

Zertifikat Lehrkompetenz: Praxisarbeit zum Thema
**Interaktion, Synchronisation und Netzwerke
als Zielsetzungen in Forschung und Lehre**

Rückblick, Reflexion und Ausblick

Mechthild Thalhammer, Institut für Mathematik

Februar 2023

Inhaltsverzeichnis

1 Zielsetzung	2
2 Eckdaten	2
3 Studium und Berufsfelder	3
3.1 Kenntnisse und Kompetenzen	4
3.2 Perspektivenwechsel und Herausforderungen	4
4 Leitgedanken	5
4.1 Reflexionen zur Umsetzung	6
5 Fachbegriffe und Sinnbilder	8
5.1 Interaktion	9
5.2 Synchronisation	13
5.3 Netzwerke	15
5.4 Schlussfolgerung	19
6 Quellen und Dank	20
7 Anhang – Unterlagen zum Zertifikat	21
8 Anhang – Ergänzende Quellen	36
9 Anhang – Lehrunterlagen	57

1 Zielsetzung

Zentrale Aspekte In dieser Praxisarbeit soll reflektiert werden, wie sich die zentralen Aspekte

- Förderung von individuellen Interessen und Fähigkeiten,
- Anleitung zu eigenständiger und selbstkontrollierter Arbeitsweise sowie
- Heranbildung zukunftsweisender algorithmischer und digitaler Kompetenzen

im Rahmen von grundlegenden Lehrveranstaltungen im Bereich Mathematik konkret umsetzen lassen.

Sinnbilder Im Sinne einer forschungsgeleiteten Lehre werden in diesem Zusammenhang die mathematischen Fachbegriffe

- Interaktion,
- Synchronisation,
- Netzwerke,

welche für aktuelle wissenschaftliche Tätigkeiten bedeutsam sind, mit sinnbildlichen Darstellungen verknüpft und in den Kontext der universitären Lehre übertragen.

Ergänzungen Ergänzungen, die aus subjektiver Sicht wesentlich für die Überlegungen im Hauptteil der Arbeit waren, wurden in der natürlichen Reihenfolge belassen und farblich gekennzeichnet.

2 Eckdaten

Zeitraumen Das Zertifikat Lehrkompetenz wurde im Herbst 2021 begonnen. Die kollegialen Hospitationen fanden im Jänner 2022 statt. Die Praxisarbeit wurde im Feber 2023 fertiggestellt.

Fokus Im Fokus dieser Praxisarbeit steht die Lehrveranstaltung

- NUMERISCHE MATHEMATIK 1 (VO 3 & PS 2, 4.5 & 3 ECTS)
Wintersemester 2022/23
Bachelorstudium Mathematik
Pflichtmodul 11 (3. Semester)

Entsprechende Überlegungen gelten für die Fortführung

- NUMERISCHE MATHEMATIK 2 (VO 3 & PS 2, 4.5 & 3 ECTS)
Sommersemester 2023
Bachelorstudium Mathematik
Pflichtmodul 14 (4. Semester)

Detaillierte Informationen und Lehrunterlagen, die im Sommersemester 2023 ergänzt werden, sind im Anhang angegeben.

Voraussetzungen Da die Lehrveranstaltung *Numerische Mathematik 1* wesentlich auf Kenntnissen aufbaut, die sich Studierende in grundlegenden Lehrveranstaltungen des ersten Studienjahres angeeignet haben, werden entsprechend dem Leitspruch

Lehrkonzepte benötigen inhaltliche Konzepte

auch die folgenden, selbst abgehaltenen Proseminare

- ANALYSIS 1 (PS 2, 2.5 ECTS)
Wintersemester 2021/22
Bachelorstudium Atmosphärenwissenschaften
Bachelorstudium Physik
(1. Semester)
- ANALYSIS 2 (PS 2, 4 ECTS-AP)
Sommersemester 2022
Bachelorstudium Atmosphärenwissenschaften
Bachelorstudium Physik
(2. Semester)
- LINEARE ALGEBRA (PS 2, 3 ECTS)
Wintersemester 2022/23
Bachelorstudium Informatik
(1. Semester)

in die Überlegungen miteinbezogen.

Inhaltliche Richtlinien Die Inhalte der zuvor erwähnten Lehrveranstaltungen sind international standardisiert und umfassen insbesondere

- Grundlagen der Linearen Algebra
(Lineare Gleichungssysteme, Eigenwertprobleme)
- Grundlagen der Analysis
(Differential- und Integralrechnung in mehreren Veränderlichen)
- Neue Sichtweisen im Rahmen der Numerik
(Methoden zur näherungsweise Lösung, Algorithmen, Implementierung)

Als Richtlinien werden die geltenden

- Studienpläne

herangezogen.

3 Studium und Berufsfelder

Im Folgenden wird reflektiert, welche der in einem Mathematikstudium erworbenen Kenntnisse und Kompetenzen wesentlich in Hinblick auf naheliegende Berufsfelder sind. Im Anschluss wird dargestellt, dass Studierende des Faches Mathematik oder verwandter Fächer, die mathematische Grundkenntnisse in ihren Curricula verankert haben, in den ersten Semestern fortwährend mit Perspektivenwechseln und neuen Herausforderungen konfrontiert sind.

3.1 Kenntnisse und Kompetenzen

Zielsetzung Eine wesentliche Zielsetzung eines universitären Studiums ist es, junge Menschen in umfassender Art und Weise auszubilden und sie damit auf ihr späteres Berufsleben vorzubereiten. Im Bereich der Mathematik geht es dabei einerseits um die Vermittlung konkreter Inhalte und andererseits um ein tiefgehendes Verständnis für theoretische Grundprinzipien und praktische Anwendungsmöglichkeiten.

Exakte Wissenschaft Im vielfältigen Spektrum wissenschaftlicher Disziplinen zeichnet sich die Mathematik als eine exakte Wissenschaft aus. Ausgehend von Axiomen und mittels logischer Schlussfolgerungen kommt man auf allgemeingültige mathematische Aussagen. Derartige Ausprägungen der Abstraktion, Strukturierung und kritischen Analyse sind in unterschiedlichen beruflichen Kontexten bedeutsam. Speziell im Zusammenhang mit komplexen Problemstellungen zeigt sich, dass systematische Vorgehensweisen erforderlich und zielführend sind.

Hilfswissenschaft Mathematische Formulierungen, Herleitungen und Resultate dienen anderen Wissenschaftszweigen als nützliche Hilfsmittel, um quantitative und qualitative Aussagen treffen zu können. Fundierte mathematische Theorien und konstruktive Methoden wurden und werden in Hinblick auf relevante Anwendungen entwickelt bzw. weiterentwickelt. Da im Rahmen eines Grundstudiums nur ausgewählte Teilgebiete der Mathematik und einzelne anwendungsorientierte Aspekte ansatzweise behandelt werden können, zählen der flexible Einsatz und Transfer von erworbenem Wissen und die Fähigkeit zur raschen Aneignung neuer Kenntnisse zu wichtigen Kompetenzen in Hinblick auf berufliche Tätigkeiten.

Moderne Technologien Angesichts der fortschreitenden Digitalisierung und der dadurch eröffneten Möglichkeiten und Herausforderungen ist es notwendig und sinnvoll, mathematisches Wissen mit Kenntnissen in den Bereichen Numerische Mathematik, Scientific Computing und Data Science, speziell zu Algorithmen, Datenstrukturen und Programmierung, zu verbinden. Im späteren Berufsleben setzt man Mathematiker:innen häufig zur Entwicklung von Lösungsstrategien und Kontrollmechanismen sowie zum Erkennen von Optimierungsmöglichkeiten ein. Unter anderem soll die Zuverlässigkeit und Aussagekraft von mittels Computern berechneten Resultaten korrekt eingeschätzt werden und das Potential von alternativen Zugängen analysiert werden. Im Zusammenhang mit komplexen Problemstellungen und Computersimulationen geht es außerdem um das Ausloten vorhandener Kapazitäten und einen reflektierten Einsatz von Ressourcen.

3.2 Perspektivenwechsel und Herausforderungen

Fokus im Schulunterricht Im Rahmen des Schulunterrichtes wird Mathematik meist mit konkreten Zahlenbeispielen und Textaufgaben in Verbindung gebracht. Bestenfalls bedingt die bestandene Matura nicht nur das erfolgreiche Automatisieren von Schemen, sondern auch ein solides Grundverständnis für zentrale Themen sowie rechnerische und sprachliche Fähigkeiten.

Fokus im ersten Semester Im ersten Semester liegt der Schwerpunkt auf Mathematik als exakter Wissenschaft. Ausgehend von Symbolen der Aussagenlogik und dem Mengenbegriff werden in erster Linie Definitionen sowie Resultate und deren Beweise behandelt. Einerseits werden bis dorthin als vertraut betrachtete Kenntnisse in Frage gestellt, jeder Schritt einer Herleitung ist präzise zu überlegen und begründen. Andererseits wird suggeriert, dass gewisse Fragestellungen allgemeingültig beantwortet werden können und dafür entwickelte systematische Methoden ohne

Einschränkungen auf die gesuchten Lösungen führen.¹

Fokus im zweiten Semester Im zweiten Semester verlässt man den aus dem Schulunterricht noch weitgehend bekannten inhaltlichen Rahmen. Man befasst sich mit anspruchsvollen Themen wie mehrdimensionaler Differentiation und Integration, welche zudem mit abstrakten Begriffen der Differentialgeometrie und Maßtheorie verknüpft werden. Die betrachteten konkreten rechnerischen Beispiele sind jedoch nach wie vor so speziell gewählt, dass sie ohne Hilfsmittel wie Taschenrechner oder Computer gelöst werden können.

Fokus im zweiten Studienjahr (Numerik) Im zweiten Studienjahr werden in den Lehrveranstaltungen *Numerische Mathematik 1–2* erstmals Computer zur näherungsweise Lösung komplexerer Probleme in realistischeren Situationen eingesetzt. Es wird verlangt, dass sich Studierende algorithmische Denkweisen aneignen und neue Programmiersprachen sowie Computeralgebra-Systeme beherrschen. Aufgrund der beschränkten Genauigkeit von Maschinen sind die postulierte Axiomatik und die entwickelten Konzepte für Unendlichkeit in diesem Kontext nicht mehr gültig. Das als selbstverständlich angesehene Assoziativgesetz zum Summieren von Zahlen ist nicht anwendbar, und als Folge einer Subtraktion zweier annähernd gleich großer Zahlen können Auslöschungsphänomene auf signifikant verfälschte und damit wertlose Ergebnisse führen. Im ersten Studienjahr besprochene Problemstellungen und systematische Methoden stellen sich als schlecht konditioniert und instabil heraus. Da deren Anwendbarkeit somit substantiell beeinträchtigt ist, müssen alternative Zugänge entwickelt werden.²

4 Leitgedanken

Begabungen Um ausgezeichnete Leistungen im Spitzensport oder als Berufsmusiker:in erbringen zu können, wird es als selbstverständlich angesehen, Talent mit intensivem Training zu verbinden. Auch als Hobbysportler:in wird man bestrebt sein, sich vor der Bewältigung einer Bergtour zunächst eine gewisse Grundkondition anzueignen oder sich als Musik-Amateur:in über Wochen hinweg im Selbststudium und gemeinsamen Proben auf eine öffentlichen Aufführung von Chor- und Orchesterwerken vorbereiten. Im Gegensatz dazu gilt Mathematik als ein Schulfach, wo wenigen Genies durch spontane Eingebungen alles zufällt, die Mehrheit hingegen beim Finden der richtigen Antwort chancenlos ist und aufgeben muss.

Aktives Tun Ein gewisses Interesse an mathematische Fragestellungen, eine rasche Auffassungsgabe und ein gutes Merkvermögen sind sicherlich hilfreich, um etwa die im Schulunterricht vermittelten Inhalte nachvollziehen zu können. Die Frustrationsgrenze bis ein tatsächliches Verständnis geschaffen wird, ist in Mathematik wohl höher einzuschätzen als in anderen Fächern. Mathematik wird vermutlich ab jenem Zeitpunkt als Angstfach wahrgenommen, ab dem man verständliche und unverständliche Vorgehensweisen nicht mehr differenzieren kann. Wenn die persönliche Motivation zum nochmaligen Anlauf für ein tiefergehendes Verstehen von Vorgehensweisen

¹Ergänzende Bemerkungen: Induktionsbeweise und indirekte Beweise sind nicht konstruktiv. Analysis: Die Konstruktion des Körpers der reellen Zahlen ist kompliziert. Der Schwerpunkt liegt auf der Existenz und Eindeutigkeit von Grenzwerten, weniger auf konkreten Berechnungen (Ableitungen, Integrale). Lineare Algebra im ersten Studienjahr versus Numerik im zweiten Studienjahr: Substantiell andere Formulierung und Analyse des Gauß-Verfahrens für Systeme linearer Gleichungen.

²Ergänzende Bemerkung: Lineare Gleichungssysteme und Eigenwertprobleme (Numerik: Pivotsuche und Vektoriteration zur Stabilisierung). Mathematische Definitionen führen auf elementare numerische Verfahren zur Differentiation und Integration.

schwindet und man sich mit oberflächlicherem Automatisieren zufrieden gibt, kann es sein, dass ab einem gewissen Anforderungsgrad der Anschluss verpasst wird.

Fähigkeiten In der modernen Mathematik trifft die Stereotype vom universellen Mathe-Genie nicht zu. Die Gesamtheit mathematischer Theorien und deren Anwendungsbereiche sind so vielfältig und umfangreich, dass sie nicht von einzelnen Personen beherrscht werden können. Es ist unumgänglich, sich auf Teilgebiete zu fokussieren und durch die kontinuierliche und wiederholte Auseinandersetzung mit Themen sein Fachwissen und seine Fähigkeiten zu erweitern und vertiefen. Auf neue wissenschaftliche Erkenntnisse führt oft ein Zusammenspiel aus Inspiration, Neugier, Einfallsreichtum, Erfahrung und Durchhaltevermögen. Speziell im Bereich der Numerischen Mathematik geht es darum, unterschiedliche Lösungsansätze zu entwickeln, gegenüberzustellen und jenen Zugang auszuwählen, der in einer bestimmten Situation unter gewissen Gesichtspunkten als optimal betrachtet wird.

Mut zu Fehlern Die erfolgreiche Beschäftigung mit komplexeren mathematischen Fragestellungen und die Implementierung von ausgeklügelten Algorithmen erfordert im Allgemeinen ein hohes Maß an Konzentration und Zeit. Obwohl unzutreffende Schlussfolgerungen oder falsche Simulationsresultate im Grunde wertlos sind, ist es zielführend, sich im ersten Anlauf nicht in Details zu verstricken. Sobald ein grober Rahmen geschaffen wurde, sollten sämtliche Zwischenschritte und Komponenten eines Codes geprüft und getestet werden. Da bei einer derartigen lösungsorientierten Vorgehensweise kleinere Irrtümer und auch gravierendere Fehler passieren können, ist es wesentlich, geeignete Kontrollmechanismen zu entwickeln, um Fehlerquellen zu lokalisieren und Programmierfehler sukzessive zu reduzieren.

Fortschritte Knüpft man nochmals an die Vergleiche mit Spitzensportler:innen und Berufsmusiker:innen an, so ist es einleuchtend, dass ein wichtiger Faktor für Erfolg das Ausmerzen von Fehlern ist. Wurde beim Probesprung auf der Bergisel-Schanze der richtige Zeitpunkt für den Absprung verpasst oder hat man in der Generalprobe einen Teil des auswendig gelernten Musikstückes nicht mehr vollständig in Erinnerung, wird man sich gleich damit befassen, um für den Wettbewerb und das Konzert eine Verbesserung zu erreichen. Entsprechend ist ein sinnvoller Modus von Proseminaren im Bereich der Mathematik die Vorbereitung von vorgegebenen Aufgaben, die gemeinsame Besprechung und insbesondere die eigenständige schriftliche Nachbereitung.

Bilder und Buntheit Aus subjektiver Sicht werden mathematische Strukturen und Formeln als faszinierend und schön wahrgenommen. Farben, zusätzliche Symbolbilder und Graphiken dienen als gedankliche Anknüpfungspunkte, zur Darstellung von Daten und zur Illustration gewisser Teilaspekte der berechneten Ergebnisse. Abhängig von der jeweiligen Persönlichkeit wird ein unterschiedliches Maß an Veranschaulichungen benötigt, um mathematische Konzepte zu erfassen und sie sich einzuprägen. Es ist auch zu beachten, dass es detaillierte Erklärungen benötigt, um den Informationsgehalt der Darstellung einer komplexeren Situation vollständig zu erfassen.

4.1 Reflexionen zur Umsetzung

Übungsaufgaben und Nachbereitungen Im Proseminar zur *Numerischen Mathematik 1* wurden neun Übungsblätter mit Aufgaben unterschiedlichen Schwierigkeitsgrades und Zeitaufwandes vorgegeben.³ Neben sehr einfachen Aufgaben zu Binärdarstellung, Horner Schema, Generieren von Matrizen, Folgen, Reihen, Rekursionen und Funktionsgraphen, deren Realisierungen wenige

³Siehe Anhang zu Lehrunterlagen.

Zeilen Code umfassen, waren insbesondere folgende numerische Methoden zu implementieren und anhand von Testbeispielen zu validieren.

- (1) Newton-Verfahren (Näherungsweise Lösung komplexer polynomialer Gleichungen, Newton-Fraktale)
- (2) Bairstow-Verfahren (Näherungsweise Berechnung der Nullstellen von Polynomen mit reellen Koeffizienten)
- (3) Gauß-Algorithmus (Näherungsweise Lösung von Systemen linearer Gleichungen mit komplexen Koeffizienten)
- (4) Cholesky-Algorithmus (Näherungsweise Lösung von Systemen linearer Gleichungen bei Symmetrie und Positivität, Lineare Regression)
- (5) Differenzenverfahren (Näherungsweise Berechnung erster und zweiter Ableitungen)
- (6) Quadratur-Formeln (Näherungsweise Berechnung bestimmter Integrale)
- (7) Dividierte Differenzen (Approximation von Funktionen durch Polynominterpolation)

Das letzte Übungsblatt griff die Methoden (3)–(7), welche im Rahmen der Vorlesung im Detail behandelt und unter verschiedenen Aspekten analysiert werden, nochmals auf. Um die erwartbare Schwierigkeiten beim erstmaligen Implementieren von hochentwickelten numerischen Verfahren in einer neuen Programmiersprache abzumildern, wurden im Rahmen von Vorlesung und Proseminar Hinweise zur Vorgehensweise gegeben und MATLAB-Scripts vorgestellt. Aufgrund des prüfungsintensiven Semesterendes wurde vereinbart, sämtliche Nachbereitungen von Aufgaben bis 15. Feber 2023 unter OLAT hochzuladen.

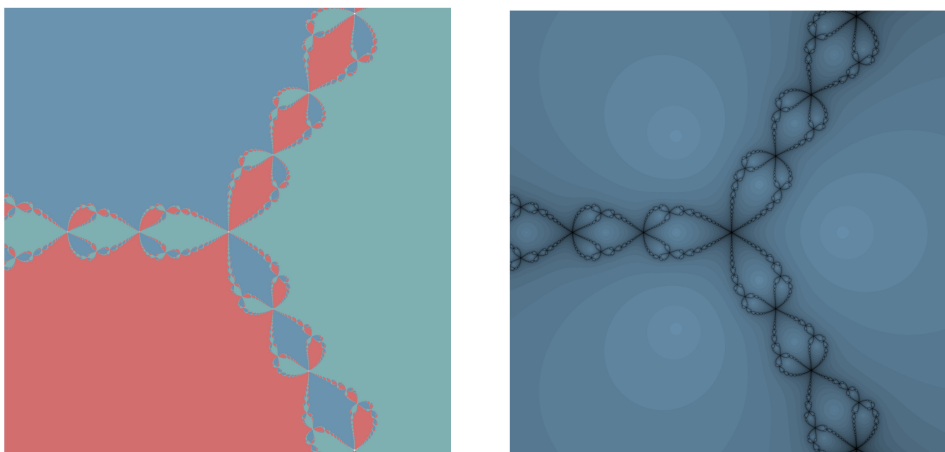


Illustration des Newton-Fraktales für die Gleichung $z^3 = 1$.

Skriptum, Tafelvortrag Zugunsten einer kompakten und konzisen Darstellung wurde im selbst erstellten Skriptum zur *Numerischen Mathematik 1–2* auf Darstellungen verzichtet.⁴ Stattdessen wurde die Strukturierung der Inhalte durch die farbliche Kennzeichnung von Begriffen, Resultaten und Beweisen hervorgehoben. In der Vorlesung wurden unterschiedliche Inhalte mittels Tafelzeichnungen veranschaulicht. Beispiele zu Problemstellungen und numerischen Methoden wurden in Form von Codes vorgeführt und verfügbar gemacht. Dies soll Studierende dazu veranlassen,

⁴Siehe Anhang zu Lehrunterlagen.

Implementierungen selbst auszuführen, damit zu experimentieren und eigenständig Graphiken und Videos zu generieren.

Umgang mit Reibungsverlusten In Hinblick auf die Lehrveranstaltung *Numerischen Mathematik* erscheint folgender Vergleich angebracht. Nach den ersten beiden Semester sind die Studierenden mit einem großen, schwer beladenen Rucksack unterwegs. Sie sollen auf einem breiten Forstweg eine Bergstation erreichen, sind kleidungs- und gerätemäßig für eine Gletschertour ausgestattet, aber haben weder Getränke noch Essen dabei. Als Lehrende gilt es, sich dessen bewusst zu sein und den Studierenden eine Karte mit auf den Weg zu geben, wo Brunnen und Einkehrmöglichkeiten eingezeichnet sind. Aus persönlicher Sicht ist es außerdem sinnvoll, auf Zeitdruck zu verzichten und gegebenenfalls Nachfristen zu gewähren.⁵

5 Fachbegriffe und Sinnbilder

Neue Sichtweisen Wenn man sich als Forschende mathematischen Theorien und Problemstellungen widmet, Zugänge entwickelt und analysiert, so erweitert und vertieft dies persönliche Interessen und Kenntnisse. Gleichzeitig führt die intensive Befassung mit verschiedenen Themenbereichen auch zu neuen Sichtweisen und Gewichtungen grundlegender Lehrinhalte. Kürzlich erschienene Publikationen und aktuelle Forschungstätigkeiten mit Kollegen aus Deutschland, Großbritannien und Spanien dienen als

- persönliche Inspiration und Motivation

sowie zur fortwährenden Reflexion

- der entstehenden Berufsfelder für Absolvent:innen,
- der Relevanz von Inhalten im internationalen Kontext,
- der verwendeten mathematischen Methoden und digitalen Mittel.

Wissenschaftlicher Kontext Die Fachbegriffe Interaktion, Synchronisation und Netzwerke wurden in den letzten Jahren im wissenschaftlichen Kontext untersucht, etwa in

T. BÖHLE, CH. KÜHN, M. THALHAMMER

On the reliable and efficient numerical integration of the Kuramoto model and related dynamical systems on graphs

doi.org/10.1080/00207160.2021.1952997

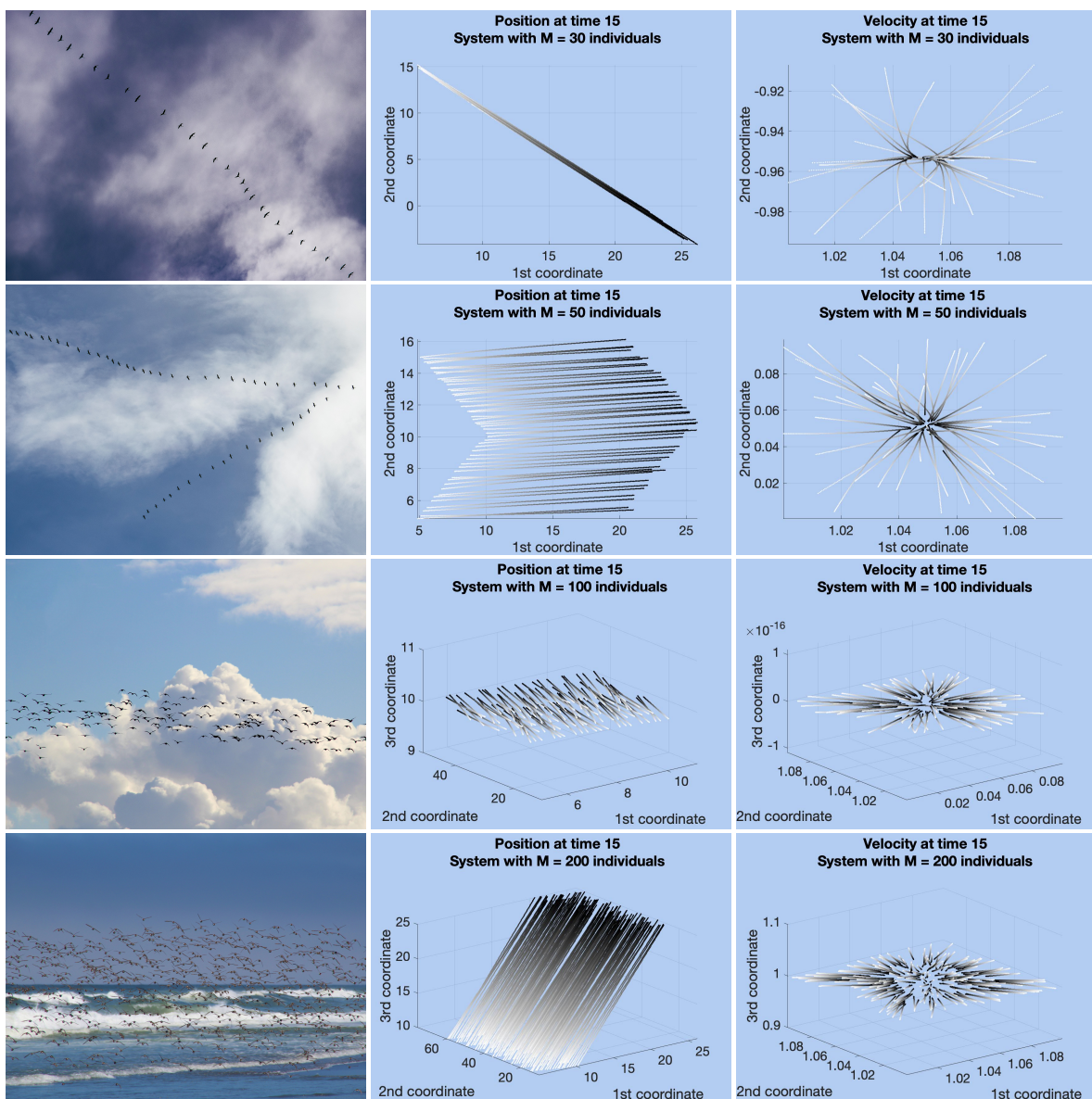
Community integration algorithms (CIAs) for dynamical systems on networks

www.sciencedirect.com/science/article/pii/S0021999122005861?via%3Dihub

⁵Im Studienplan ist für die Lehrveranstaltung *Numerische Mathematik* die Programmiersprache MATLAB vorgeschrieben. Im Gegensatz zu anderen Optionen ist sie leichter handzuhaben, so ist beispielsweise die dynamische Definition von Eingabegrößen möglich, sie bietet eine ansprechende Benutzer-Oberfläche und erleichtert die Erstellung von Graphiken oder Videos. Sie wurde vom Mathematiker CLEVE MOLER entwickelt, weshalb die meisten Programmierbefehle üblichen mathematischen Bezeichnungen entsprechen. In technischen Berufsfeldern, insbesondere im Ingenieurwesen, wird MATLAB aufgrund weiterer Vorzüge als Standard verwendet. Auch wenn MATLAB den Nachteil hat, nicht allgemein frei verfügbar zu sein, so gibt es jedoch an Universitäten Studierenden-Lizenzen und ansonsten kostenfreie Alternativen. Aus den genannten Gründen ist es wünschenswert, MATLAB bereits ab dem ersten Studienjahr zu verwenden und Grundlagen zu Algorithmen und Programmierung zu vermitteln. Der aktuell notwendige Wechsel zwischen zwei in einigen Aspekten unterschiedlichen Programmiersprachen führt (im Proseminar wöchentlich) zu erheblichen Reibungsverlusten.

5.1 Interaktion

Fachliche Aspekte Mathematische Modelle, die Interaktionen einer Gruppe von Individuen bzw. eines aus verschiedenen Komponenten zusammengesetzten komplexen Systemes beschreiben, haben vielfältige Anwendungen in verschiedenen Lebensbereichen. Aus persönlichem Interesse wurden zusätzliche Simulationen für Schwärme von Vögeln durchgeführt (siehe unten). Man stellt dabei fest, dass trotz zunehmender Komplexität der Gruppenstruktur die Bewegungen der Vögel in allen Fällen koordiniert sind. Die Positionen und Geschwindigkeiten werden so abgestimmt, dass sich momentane Flugbahnen nicht kreuzen. Zieht man hingegen den Vergleich zu sportlichen Wettbewerben wie Radrennen mit dichten Menschengruppen, so kommt es hier immer wieder zu Kollisionen.



First column: Visualisations of flocks of birds (Kranich17 and Minka2507 at pixabay.com). Second and third columns: Numerical simulations of Cucker–Smale systems modelling the collective behaviour of groups of individuals with pairwise interactions. The numbers of individuals and the positions at the initial time are inspired by real world situations. Varying colors from white to black represent the evolution of the positions and the associated velocities till the final time.

Sinnbild und Umsetzung im Lehrkontext Die angegebenen Graphiken von Vogelschärmen eignen sich als Sinnbilder für gruppendynamische Prozesse und das Erreichen von Lernzielen.

- Aspekte der Interaktion und Kommunikation in Proseminaren wurden im Protokoll zu den kollegialen Hospitationen durch Robert Hafner und Fabian Ochs reflektiert.
- Überlegungen in Bezug auf die erstmals im aktuellen Wintersemester abgehaltene Lehrveranstaltung *Numerische Mathematik 1* sind im Folgenden angegeben. Diese gelten entsprechend für die im Sommersemester 2023 stattfindende Lehrveranstaltung *Numerische Mathematik 2*. Angesichts der Diversität der für die Vorlesung angemeldeten Studierenden, liegt es nahe, die letzte Abbildung eines dichten Vogelschwarmes zu betrachten.
 - Die Pflicht-Lehrveranstaltung ist für das zweite Studienjahr vorgesehen. Die im Studienplan vorgegebenen Inhalte setzen einerseits gute Kenntnisse sowie ein gutes Verständnis der in den ersten beiden Semestern behandelten mathematischen Grundlagen und andererseits gewisse Fertigkeiten in Bezug auf Programmiersprachen und Implementierungen voraus. Ob die Voraussetzungen tatsächlich erfüllt sind, wird jedoch nicht überprüft.
 - In der unten angegebenen Aufstellung sind die Teilnehmer:innen nach Semester aufgeschlüsselt. Von den insgesamt 46 Studierenden belegt die überwiegende Mehrheit das Bachelorstudium Mathematik (Fach), vier die Fächer Atmosphärenwissenschaften oder Physik, 9 Studierende haben die STEOP nicht abgeschlossen (zwei ohne Angabe), für zwei Studierende ist es der zweite Antritt.

Pflichtmodul *Numerische Mathematik 1*, vorgesehen für 3. Semester
 46 zur Vorlesung angemeldete Studierende
 41 Studierende sind für das Bachelorstudium Mathematik inskribiert
 35 Studierende haben die STEOP abgeschlossen

1. Semester	1 Studierender
2. Semester	3 Studierende
3. Semester	5 Studierende
4. Semester	10 Studierende
5. Semester	6 Studierende
6. Semester	5 Studierende
7. Semester	8 Studierende
8. Semester	2 Studierende
9. Semester	2 Studierende
12. Semester	2 Studierende
14. Semester	1 Studierender
34. Semester	1 Studierender

Die Daten wurden vis:online entnommen (14. Februar 2023).

- Unterhalb findet sich eine Aufstellung der bisherigen Beurteilungen in Bezug auf Vorlesung und Proseminar (Stand 17. Feber 2023). Die angewandten Beurteilungsmaßstäbe werden im Folgenden unter verschiedenen Gesichtspunkten reflektiert.

Proseminar zur *Numerischen Mathematik 1*

32 angemeldete Studierende

Keine eindeutige Gruppenzuordnung (Wechsel während des Semesters)

Positive Beurteilung (21): Sehr gut 13, Gut 6, Befriedigend 1, Genügend 1

(Semester 2 – Semester 12)

Offen (1): Genügend 1

Negative Beurteilung / Abbruch (10): Nicht Genügend 1, Stornierung der Anmeldung 9

(Abbruch teils krankheitsbedingt kurz vor Semesterende)

Gegenüberstellung der Vorlesung zur *Numerischen Mathematik 1* (Aufstellung, s.o.)

Positive Beurteilung (14): Sehr gut 6, Gut 4, Befriedigend 1, Genügend 3

(Semester 2 – Semester 12)

Negative Beurteilung / Wiederholung / Nicht-Erscheinen:

Nicht Genügend 1, Wiederholung 4, Ausfall 1

(Gewährung einer *zweiten Chance* bei ausreichend vorhandenen

Grundkenntnissen und deutlichen Lernlücken)

Beurteilung Proseminar	Beurteilung Vorlesung
PS Genügend	VO Wiederholung (17. März)
—	VO Wiederholung (17. März)
—	VO Nicht Genügend
PS Gesamtleistung Gen., Quant. Krit. offen	VO (17. März)
PS Gut	VO Genügend
PS Sehr gut	VO Sehr gut
PS Gut	VO Prüfung am 17. Feber nicht angetreten
PS Sehr gut	VO Sehr gut
—	VO Genügend
PS Sehr gut	VO Offen
PS Sehr gut	VO Sehr gut
PS Sehr gut	VO Gut
PS Gut	VO Befriedigend
PS Gut	VO Wiederholung (17. März)
PS Sehr gut	VO Gut
PS Sehr gut	VO (17. März)
PS Befriedigend	VO Offen
PS Sehr gut	VO Gut
PS Nicht Genügend	VO (17. März)
PS Gut	VO Wiederholung (17. März)
PS Gut	VO Genügend
PS Sehr gut	VO Gut
PS Sehr gut	VO Sehr gut
PS Sehr gut	VO Offen
PS Sehr gut	VO Sehr gut
PS Sehr gut	VO Sehr gut

- Zieht man in der letzten Abbildung des dichten Vogelschwarmes eine Höhenlinie, so symbolisiert dies die minimalen Prüfungsanforderungen für die Lehrveranstaltung *Numerische Mathematik 1*. In der Vorlesung ist dies der klar definierte inhaltliche Rahmen (Auswahl von Themengebieten, Präzisierung der Inhalte anhand von einzelnen Kapi-

teln im selbst erstellten Skriptum). Im Proseminar betrifft es die Anzahl und Qualität der vorbereiteten, präsentierten und nachbereiteten Aufgaben sowie die Ausarbeitung und Vorstellung eines Projektes (Befassung mit theoretischen Aspekten und Erstellung einer Implementierung).⁶

- Es bleibt jedoch in horizontaler Richtung Raum, um diese Anforderungen auf individuelle Art und Weise zu erfüllen. Da die Gesamtbelastung der Studierenden zu Semesterende deutlich zunimmt, wird beispielsweise der zeitliche Horizont flexibel gehandhabt. Für mündliche Vorlesungsprüfungen werden verschiedene Termine angeboten und Kleingruppen gebildet (fixiert sind bisher 6./13./14./17. Feber und 17. März 2023). Bei den Projektpräsentationen gelang es durch vier Einheiten an zwei Tagen, Überschneidungen mit anderen Lehrveranstaltungen und angesetzten Prüfungen zu vermeiden (27./31. Jänner 2023). Das Angebot gegebenenfalls die Proseminar-Gruppe zu wechseln, nahmen mehrere Studierende wahr. Zu beachten ist, dass eine flexible Terminvereinbarung von mündlichen Vorlesungsprüfungen eine Überlastung gegen Ende des Semesters vermeidet, aber mehr Eigeninitiative der Studierenden erfordert.
- Für die Vorlesung wurde großteils ein herkömmlichen Format gewählt. Die im Skriptum zusammengefassten Inhalte wurden mittels Beamer präsentiert und durch zusätzliche Erklärungen an der Tafel ergänzt. Implementierungen und illustrierende Beispiele sowie bildliche Darstellungen wurden vorgezeigt und der entsprechende Code (MAPLE, MATLAB) verfügbar gemacht. Da am Ende des Semesters mehr Fragen zu besprechen waren, wurde mehr Zeit dafür eingeplant. Entgegen dem Usus, inhaltliche Höhepunkte am Ende des Semesters zu setzen, wurden die wichtigen Methoden und Resultate bis Mitte Jänner besprochen und anschließend weniger (prüfungs-)relevante Ergänzungen zu den Begriffen Kondition und Stabilität behandelt. Hin und wieder wurden wissenschaftliche Inputs eingestreut, so wurde beispielsweise als Abschluss der Lehrveranstaltung eine aktuelle 20-stündige Simulation und daraus zu ziehende Schlussfolgerungen interaktiv diskutiert (Kurzvortrag, rund 15 Minuten).
- Bei der Beurteilung von Prüfungsleistungen im Bereich Mathematik ergibt sich naturgemäß ein nicht auflösbarer Widerspruch.⁷ Die schriftlichen Angaben von Definitionen, Resultaten mit Beweisen und Rechnungen sind entweder vollständig richtig oder falsch. Sobald es um die Bewertung von *ungefähr richtigen* Angaben geht, ist der Interpretationsspielraum groß.⁸ Bei mündlichen Prüfungen besteht hingegen die Möglichkeit der Nachfrage und Präzisierung von Antworten.⁹
- Ein mündliches Prüfungsformat für die Vorlesung *Numerische Mathematik 1* bedeutet einen organisatorischen Mehraufwand, bietet jedoch auch die Möglichkeit, auf jede Studierende und jeden Studierenden individuell einzugehen. Durch Überblicksfragen, offene Fragen zu einzelnen Themen und detaillierte Nachfragen kann man rasch be-

⁶Das Skriptum und ergänzende Lehrunterlagen sind im Anhang aufgenommen.

⁷Multiple-Choice-Tests, die zwar rasch zu korrigieren sind, aber keine Begründungen zum *Warum ist diese Antwort richtig? Warum ist diese Antwort falsch?* umfassen und damit auch keine validen Schlussfolgerungen in Bezug auf das erworbene Verständnis zulassen, werden als Prüfungsformat von den Überlegungen ausgenommen.

⁸Vergleiche auch Kriterien zur Leistungsfeststellung: *Es kam sogar vor, dass die Beurteilungen für ein und dieselbe Mathematikschularbeit über alle fünf Beurteilungsstufen streuten.* Zitat aus www.aau.at/wp-content/uploads/2017/10/3-LEISTUNGSFESTSTELLUNG.pdf

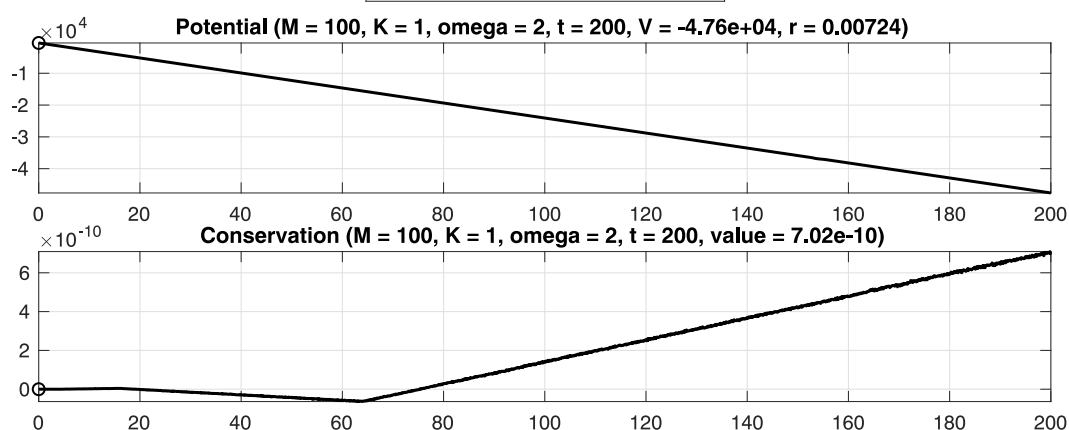
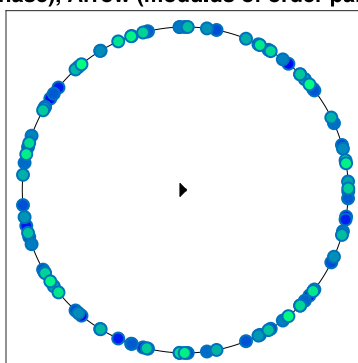
⁹Mündliche Prüfungen sind außerdem ein probates Mittel, um selbständig erbrachte Leistungen zu beurteilen. In Hinblick auf die fortwährende Weiterentwicklung digitaler Hilfsmittel und KI-Quellen wird dieser Aspekt vermutlich an Bedeutung gewinnen.

urteilen, ob ein Grundverständnis mathematischer Denkweisen und Methoden gegeben ist und wie sorgfältig sich die jeweilige Person mit dem Prüfungsstoff auseinandergesetzt hat. Auf Fehler, die möglicherweise durch Nervosität passieren, kann man aufmerksam machen und beurteilen, wie dabei reagiert wird. Hinzu kommt, dass sich Flüchtigkeitsfehler und gravierende Fehler bei mündlichen Prüfungen im Allgemeinen leichter unterscheiden lassen. Bei schriftlichen Prüfungen im Bereich Mathematik scheitern Studierende unter anderem an einzelnen Zwischenschritten, vollständigen Argumentationen und korrekten Formulierungen. Auch hier kann man im mündlichen Format Nachfragen stellen und die Fähigkeit zur Selbstkorrektur bewerten. Falls die erbrachte Leistung nicht ausreicht, so kann man direkte Rückmeldung geben, was bei der Vorbereitung auf die Wiederholung zu beachten ist.

5.2 Synchronisation

Fachliche Aspekte Die Evolution von Systemen mit geringer, partieller und signifikanter Synchronisation ist unter den im Folgenden angegebenen Links zu beobachten. Die signifikante Veränderung eines einzigen Parameters hat zur Folge, dass sich die zunächst am Kreisrand stark variierenden Komponenten des Systemes synchronisieren.

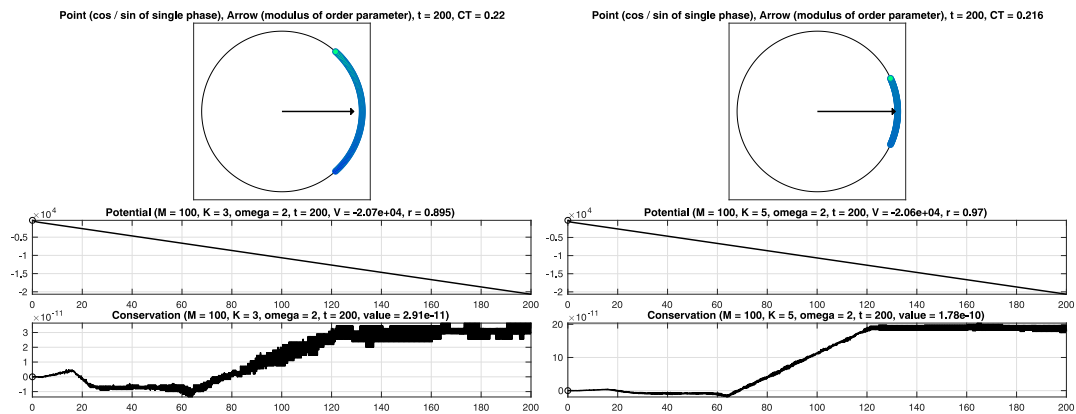
Point (cos / sin of single phase), Arrow (modulus of order parameter), $t = 200$, $CT = 0.283$



techmath.uibk.ac.at/mecht/MyHomepage/Research/MovieKuramotoClassicalK1.m4v

techmath.uibk.ac.at/mecht/MyHomepage/Research/MovieKuramotoClassicalK3.m4v

techmath.uibk.ac.at/mecht/MyHomepage/Research/MovieKuramotoClassicalK5.m4v



Sinnbild und Umsetzung im Lehrkontext Die angegebenen Illustrationen werden als sinnbildliche Darstellungen für die Lehrveranstaltung *Numerische Mathematik 1* reflektiert und entsprechend für die zukünftige Lehrveranstaltung *Numerische Mathematik 2* adaptiert. Angesichts der sehr divers zusammengesetzten Studierendengruppen, die für die Vorlesung zur *Numerischen Mathematik 1* angemeldet ist, stellt sich die Frage, welcher Grad der Synchronisation erreicht werden soll.

- Es wurde entschieden, einheitliche Regeln und quantitative sowie qualitative Beurteilungskriterien vorzugeben. Beispielsweise wurden für die Bewertung von Projektarbeiten als direkte Vergleichskriterien Korrektheit, Sorgfalt und Komplexität herangezogen. Die Gruppe wird also in dieser Hinsicht synchronisiert.
- Die Studierenden angewandter Fächer, die sowohl an der Vorlesung als auch am Proseminar teilnahmen, kamen mit der Vorgabe, dass auch von ihnen mathematische Formalismen und Beweismethoden verlangt wurden, sehr gut zurecht.
- Bei der Themenwahl der Projektarbeiten, bei der Gestaltung der Folien und Unterlagen sowie bei der verwendeten Programmiersprache gab es keine Einschränkungen. Auf naheliegende Themenbereiche wurde in den Monaten Oktober und November gelegentlich hingewiesen, vermehrt im Dezember.
- Die Abstimmung der Projektthemen erfolgte in der Lehrveranstaltung oder durch Kontaktaufnahme per Email. Einzelne Studierende entschieden sich frühzeitig, die Mehrheit vor bzw. in der Weihnachtspause, wenige erst kurz vor Ende des Semesters. Es war zulässig, Projektarbeiten in Zweiertteams zu bearbeiten und dann einzeln zu präsentieren. Interessanterweise zogen die Teilnehmer:innen die Option einer Teamarbeit nicht in Betracht, sondern präferierten alternative Themen, die sich deutlich von jenen ihrer Kolleg:innen unterschieden.
- Die Projektpräsentationen (Vortrag zu je 15 Minuten) fanden in einem Format statt, das einem wissenschaftlichen Workshop ähnelte. Die zeitliche Einteilung der vier Einheiten (Freitag und Dienstag zu Semesterende, jeweils von 8 bis 10 Uhr und von 16 bis 18 Uhr) erfolgte unerwartet reibungslos durch die Eintragung in eine im Proseminar ausgegebene Liste.
- Die Themen waren breit gefächert und behandelten sowohl theoretische als auch anwendungsorientierte numerische Methoden. Die Implementierungen basierten auf unter-

schiedlichen Programmiersprachen (C++, MATLAB, PYTHON). Zu den präsentierten Unterlagen gab es keine Vorgaben, sie reichten von professioneller entworfenen Folien bis zu eingescannten, ebenfalls schön gestalteten handschriftlichen Ausführungen.

- Die Qualität der Präsentationen war solide, genau geplant oder spontan, mehr oder weniger durchdacht, verständlicher oder weniger verständlich formuliert. Obwohl größtenteils (prüfungs-)relevante Inhalte vertieft und fortgeführt wurden, variierte die Aufmerksamkeit bei den Vorträgen der Studienkolleg:innen und das Interesse an den behandelten Problemstellungen stark. Einzelne Studierende meldeten sich mit Fragen, und eine kleine Gruppe von auch ansonsten aktiveren Studierenden setzte die begonnene Diskussion im Anschluss an eine der Einheiten fort.
- Die vielfältigen und unterschiedlich ausgerichteten Beiträge wurden von der Lehrenden als sehr interessant, abwechslungsreich und inspirierend wahrgenommen. Es ist geplant, einige Themen in der Vorlesung oder im Proseminar zur *Numerischen Mathematik 2* aufzugreifen und zu vertiefen.

Themenwahl der Projektarbeit	Programmiersprache
Hermite-Interpolation	MATLAB
Bogenlängen von Kurven	PYTHON
Nullstellen von Polynomen	MATLAB
Bairstow-Verfahren und Fraktale	MATLAB
Kettenbruchentwicklungen	MATLAB
Gauß-Chebyshev-Quadratur	PYTHON
Orthogonale Legendre-Polynome	PYTHON
Schiefer Wurf ohne / mit Luftwiderstand	PYTHON
Zeitreihenanalyse Tagestemperaturen	PYTHON
Vollständige Pivotsuche	MATLAB
Primzahlenberechnungen	PYTHON
Euklidischer Algorithmus	PYTHON
Positionsbestimmungen aus Daten	C++
Explizites Eulerverfahren	PYTHON
Heron-Verfahren	PYTHON
Auswertung der Exponentialfunktion	PYTHON
Bézier-Kurven	PYTHON
Zweidimensionale Integration	MATLAB
Vektoriteration	PYTHON
Numerisches Differenzieren	PYTHON
Gram-Schmidt-Orthogonalisierung	MATLAB
Monte-Carlo-Methoden	PYTHON
Jacobi-Iteration	MATLAB

5.3 Netzwerke

Fachliche Aspekte Netzwerke und Graphen werden verwendet, um sämtliche Verknüpfungen zwischen interagierenden Individuen zu beschreiben. Insbesondere im Zusammenhang mit kom-

plexen Systemen und Vorgängen geht es darum, gewisse Strukturen zu erkennen und diese bei der Entwicklung praktikabler und effizienter Methoden zu nützen.



Gerd Altmann pixabay.com

Sinnbild und Umsetzung im Lehrkontext Netzwerke eignen sich augenscheinlich als Sinnbilder für komplexe Inhalte, vielfältige thematische Verknüpfungen und diverse Sichtweisen, um auf mathematische Methoden zu kommen.¹⁰ Als Interpretationen bieten sich auch Gruppenstrukturen oder die verbale und nonverbale Kommunikation zwischen Lernenden und Lehrenden an. Netzwerke können ebenfalls herangezogen werden, um Perspektivenwechsel zu veranschaulichen. So kann man beispielsweise aus mehreren Wegen einen auswählen, um von einem Knoten zu einem entfernter liegenden Knoten zu gelangen.

- Die erste Darstellung kann als Sinnbild für die Zielsetzung gesehen werden, sich in Hinblick auf Lehrinhalte und digitale Mittel fortwährend an internationalen Standards und sich neu eröffnenden Berufsfeldern zu orientieren.¹¹
- Bei Lehrveranstaltungen im ersten Studienjahr sind zwei Traditionen fest verankert, die den Aufbau von Netzwerken in Bezug auf ein solides Grundverständnis hemmen. Einerseits neigt man dazu, mathematische Teilgebiete und Programmierkurse isoliert voneinander zu behandeln. Andererseits wird das erste Studienjahr im Bemühen alles unterzubringen, was man als Grundlage für später benötigt, mit Begrifflichkeiten überladen. Die Separierung in

¹⁰Als einfaches Beispiel seien die Normalgleichungen der linearen Ausgleichsrechnung genannt, die sich als Minimalbedingungen einerseits mittels Differentialrechnung und andererseits mittels orthogonalen Projektionen herleiten lassen. In Hinblick auf eine Erleichterung für Studierende im zweiten Studienjahr sollte beides bereits im ersten Studienjahr behandelt werden.

¹¹Grundlegende Überlegungen sind in Kapitel 3 angegeben.

Algebra und Analysis erweist sich in Hinblick auf die Lehrveranstaltung *Numerische Mathematik 1–2* als ungünstig, weil Abgrenzungen hier obsolet sind und die Kombination von unterschiedlichen Methoden notwendig ist.¹² Speziell im Bereich der Linearen Algebra wird eine Problemstellung mit verschiedenen Begriffen in Zusammenhang gebracht.¹³ Die in den begleitenden Proseminaren behandelten Aufgaben sind darauf ausgerichtet, aus den unterschiedlichen Begriffen auf die wohlbekanntere Problemstellung zu kommen. Die gewählten Zahlenbeispiele sollen dann durch händische Rechnungen gelöst werden. Aus persönlicher Sicht wäre es sinnvoll, inhaltliche Schwerpunkte zu setzen und zusätzliche Kenntnisse in Bezug auf den Einsatz von Software zu vermitteln. So kann bei den Studierenden ein Bewusstsein dafür geschaffen werden, dass systematische Vorgehensweisen bei komplexeren Problemen notwendig und um ein Vieles effizienter sind. Es wäre auch wünschenswert, wenn bereits im ersten Semester auf die praktischen Grenzen theoretischer Resultate aufmerksam gemacht wird.¹⁴

- Aus persönlicher Sicht hat es sich bewährt, die Studierenden wiederholt dazu einzuladen, im Netzwerk der gestellten Vorlesungsinhalte und Proseminaraufgaben eigenständig Zwischenknoten einzufügen. Bei Aufgaben, wo ein allgemeiner Fall behandelt werden soll, stehen diese Zwischenknoten für selbst gewählte konkrete Beispiele und einfache Spezialfälle. Bei Implementierungen sind dies Validierungen der einzelnen Komponenten, aus denen ein Code zusammengesetzt ist. Ein Vorbild beim Aufbau des eigenen Verständnisses durch gut gewählte, leichter zu erfassende und einprägsame Spezialfälle war der renommierte Mathematiker Peter Olver.¹⁵
- Aus persönlicher Sicht sind auch spielerische und experimentelle Aspekte ein wesentliches Mittel, um die eigene Intuition und das Verständnis für Neues zu erweitern und zu vertiefen. Im Sinnbild eines Netzwerkes können Experimente als Knoten, die als Anker dienen, gesehen werden.
- Speziell während der Lockdown-Phasen hatten Studienanfänger:innen die Option, freiwillige Zusatzaufgaben (Rätselaufgaben) zu lösen. Um einen Bonuspunkt zu bekommen, musste allerdings nicht nur die richtige Lösung angegeben, sondern auch die gewählte Vorgehensweise kurz beschrieben werden. Die Intention war es, durch derartige Aufgaben eine

¹²Das Verfahren von Bairstow zur Faktorisierung eines Polynomes mit reellen Koeffizienten beruht auf Division mit Rest (Algebra) und der Idee der Linearisierung (Analysis). Damit können sämtliche Nullstellen eines (interpolierenden) Polynomes näherungsweise berechnet werden.

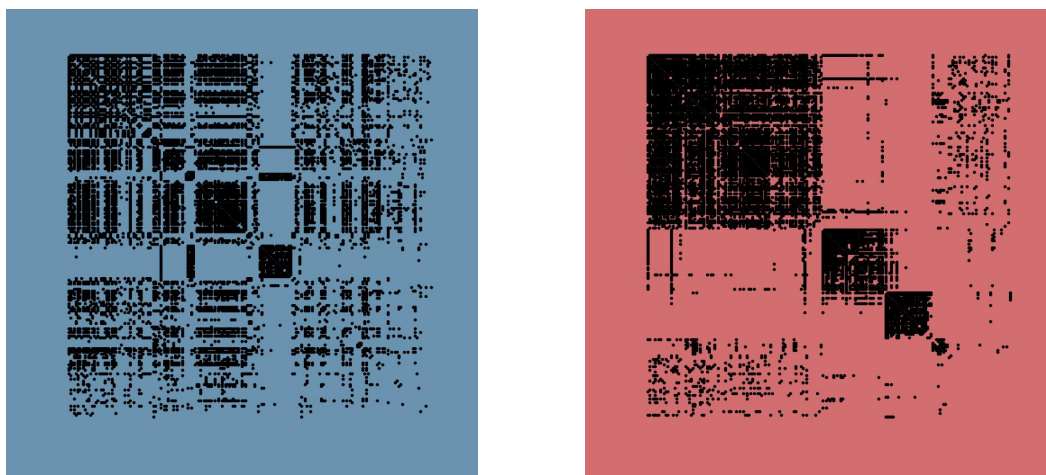
¹³Problemstellung *Lösung eines Systemes linearer Gleichungen*. Verknüpfte Begriffe in alphabetischer Reihenfolge: Affiner Unterraum, Affin-lineare Abbildung, Basis, Bild, Darstellungsmatrix, Determinante, Dimension, Eindeutigkeit, Endomorphismus, Erzeugendensystem, Existenz, Homogener Lösungsraum, Inhomogener (affiner) Lösungsraum, Injektivität, Invertierbarkeit, Kern, Lineare Abbildung, Linearkombination, Lineare Unabhängigkeit, Matrix, Rang, Spalte, Stufenform, Unterraum, Vektorraum über einem Körper, Zeile. Jeder einzelne Begriff ist relevant, die Gesamtheit aller Begriffe führt aber speziell bei Studierenden in den Bereichen Atmosphärenwissenschaften, Informatik, Physik etc. zu Irritationen. In diesem Sinne gilt *Weniger ist mehr*.

¹⁴Im Rahmen der Linearen Algebra werden Pivot-Strategien für Berechnungen mittels Computer nicht thematisiert. Für simple Matrizen der Dimension zwei führt die Anwendung des Gauß-Algorithmus ohne Spalten-Pivotsuche aufgrund der endlichen Stellenzahl auf ein signifikant falsches und damit wertloses Resultat. Der Grund dafür liegt im Phänomen der Auslöschung und anschließender Verstärkung durch Multiplikation mit einer betragsmäßig großen Zahl.

¹⁵In der Lehrveranstaltung *Numerische Mathematik 1* wurde ein wichtiges Resultat zu symmetrischen Quadraturformeln hergeleitet. Dafür wurden zunächst Spezialfälle mit wenigen Stufen betrachtet und elementare Argumente, welche dennoch längere Umformungen benötigen, verwendet. Da die Behandlung des allgemeinen Falles mit einer entsprechenden Vorgehensweise umständlich und langatmig ist, wurden zusätzliche Ideen erklärt und genutzt (Lineare Integraltransformation und Extraktion ungerader Polynome mit Integralwerten gleich Null).

Alternative zu beweislastigen Problemstellungen zu bieten. Die Rätselaufgaben sollten den Studierenden als Angelpunkte dienen, um sich auf spielerische Weise mit mathematischen Lösungsstrategien zu befassen.

- Auch für Lehrende ist es lohnenswert, bekannte Inhalte unter neuen Perspektiven zu sehen.¹⁶ Eine neue Anordnung ermöglicht es, spezielle Strukturen in (inhaltlichen) Netzwerken zu identifizieren. Als sinnbildliche Darstellungen sind zwei Graphiken eingefügt, wo links das ursprüngliche Netzwerk und rechts ein äquivalentes Netzwerk mit erkennbaren Strukturen zu sehen ist (vgl. Publikationen).

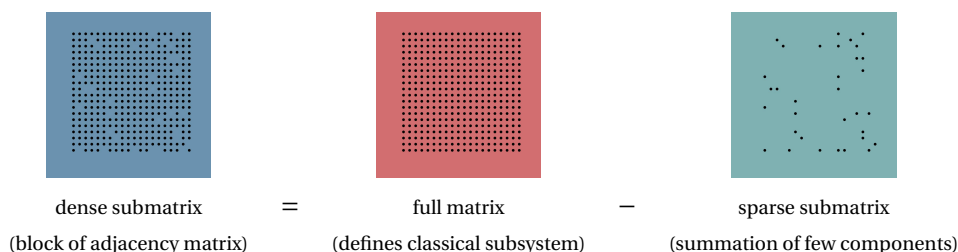


The graph of a real data network (networkrepository.com/aves-wildbird-network.php) is captured by the associated adjacency matrix comprising coefficients equal to one (blue) and zero (white). Left: Original representation, where the structure of the underlying communities and interactions is not evident. Right: An equivalent representation obtained by community detection and suitable permutation renders it possible to identify a block structure with relatively dense and sparse submatrices.

- Im aktuellen Studienjahr wurde versucht, organisatorische und inhaltliche Informationen dem Netzwerk der Studierenden mehrmals und über verschiedene Kanäle zukommen zu lassen, in mündlicher und schriftlicher Form, direkt oder indirekt (veranstaltungsübergreifend in Vorlesung und Proseminar, per Email und OLAT). Für zukünftige Lehrveranstaltungen ist geplant, den Studierenden wieder den Vorschlag zu unterbreiten, in den jeweiligen Gruppen zumindest eine Sprecherin oder einen Sprecher zu nominieren.
- Eine in der zuvor erwähnten Publikation entwickelte und analysierte Methode wird durch folgende Graphik illustriert. Diese Darstellung kann man auch im alltäglichen Kontext nachvollziehen. Falls beispielsweise eine Schilhrerin eine Gruppe mit wenigen erwachsenen Mitgliedern betreuen soll, so hat sie die Möglichkeit auf jede einzelne Person individuell einzugehen. Sie kann Abfahrten unterschiedlichen Schwierigkeitsgrades empfehlen, jene begleiten, die auf der blauen Piste unterwegs sind, und mit den anderen Treffpunkte verein-

¹⁶Eine für die Lehrveranstaltung *Numerische Mathematik 1* neuartige Sichtweise bestand darin, für allgemeine Systeme linearer Gleichungen eine äquivalente, sukzessiv lösbare Darstellung zu finden (vgl. etwa Division mit Rest, Polynominterpolation mittels dividierter Differenzen). Für die Fortsetzung *Numerische Mathematik 2* bieten sich die Begriffe Orthogonalität und Adaptivität an.

baren. Bei einer größeren Gruppe von Erwachsenen, die homogene Voraussetzungen vorweisen und ähnliche Interessen teilen, kann ebenfalls ein hoher Grad an Zufriedenheit erreicht werden. Entsprechend ist es in einer größeren, jedoch weitgehend homogenen Gruppe. Wenn nur wenige Personen andere Voraussetzungen mitbringen, kann man auf deren spezielle Bedürfnisse eingehen.



The key idea for the efficient evaluation of large-scale subsystems defined by relatively dense submatrices is to trace them back to full matrices. These correspond to classical subsystems and are solved in an efficient manner by global approximations and precomputations. Remaining contributions of relatively sparse submatrices require the summation of relatively few components.

- Signifikante Reibungsverluste sind hingegen bei größeren und inhomogenen Gruppen zu erwarten. Orientiert man sich an der Mitte, so ist es naturgemäß für die Mehrheit unpassend (Überforderung oder Unterforderung, Resignation oder Langeweile). Hier erscheint es sinnvoller, die gesamte Vielfalt und Freiraum für individuelle Zugänge zuzulassen. Wie zuvor ausgeführt, wurde dies in der Lehrveranstaltung *Numerische Mathematik 1* durch mündliche Vorlesungsprüfungen und Projektarbeiten in die Praxis umgesetzt.^{17 18}
- Die für die Lehrveranstaltung *Numerische Mathematik 1* beobachtete Zweiteilung der Gruppe mit einerseits sehr zufriedenstellenden (Sehr gut / Gut) und andererseits unzureichenden Beurteilungen oder Abbrüchen ist (im Bereich der Mathematik) öfter festzustellen. Sofern die vorausgesetzten Grundkenntnisse gegeben sind, lassen sich vorhandene Lernlücken im Allgemeinen gut schließen. Als Lehrende gilt es, (in einem angemessenen Ausmaß) für Fragen zur Verfügung zu stehen und Hinweise zu Lernquellen zu geben.
- Wie zuvor ausgeführt ist es bei einer diversen bzw. zweigeteilten Gruppe nicht sinnvoll, enge Kriterien anzulegen und Synchronisation erzwingen zu wollen. Es erscheint vielmehr wünschenswert, einen offenen Ansatz zu wählen, wo individuelle Stärken zur Geltung kommen.

5.4 Schlussfolgerung

Die vorangegangenen Beobachtungen und damit verbundenen Überlegungen im Lehrkontext können mit einem wissenschaftlichen Grundprinzip, welches sich speziell im Bereich der Nume-

¹⁷Beobachtung und persönliche Anmerkung: In Bezug auf die Lehrveranstaltung *Numerische Mathematik 1* wurde zunächst die Diversität der Studierenden unterschätzt und die Vernetzung untereinander überschätzt. Wenn mehrere Studierende bestätigten, dass die inhaltlichen Erklärungen verständlich und ausreichend waren, wurde fälschlicherweise angenommen, dass dies im Großen und Ganzen auf die gesamte Gruppe zutraf. Übungsaufgaben, die in einer der Proseminar-Gruppe sehr gut gelöst wurden, führten in der anderen Proseminar-Gruppe zu erheblichen Schwierigkeiten oder blieben un bearbeitet. In den ersten Monaten kam es also in dieser Hinsicht zu keinem Austausch zwischen den Studierenden.

¹⁸Die Diversität der Studierenden in Hinblick auf Lernstile und Wissenserwerb wurde insbesondere in den mündlichen Vorlesungsprüfungen deutlich.

rischen Mathematik immer wieder bestätigt, verknüpft werden. Es ist nicht möglich, eine im universellen Sinn *beste* (didaktische) Methode anzugeben. Es geht vielmehr darum, situationsbezogen mehrere zuverlässige Zugänge miteinander zu vergleichen und dann eine Methode hervorzuheben, die unter einem bestimmten Gesichtspunkt von Vorteil ist. Wenn allerdings eine Situation sehr vielfältige Ausprägungen zulässt, ist die Reduktion auf ein einzelnes Kriterium oder eine einzelne Kennzahl nicht sinnvoll. Hier ist es passender, verschiedene Perspektiven in Betracht zu ziehen und entsprechende Erkenntnisse umfassend darzustellen.¹⁹

6 Quellen und Dank

Quellen Zwei allgemeiner verständliche Quellen, welche sich insbesondere mit Lernprozessen und grundlegenden mathematischen Deduktionsprinzipien befassen, sind im Anhang aufgenommen. Dies sind ein Interview mit der Lernforscherin **ELSBETH STERN** (ETH Zürich) und die Niederschrift eines Vortrages des Mathematikers **GÜNTER HANISCH** (Universität Wien). Weiters sei auf Lehrunterlagen von **JOHANNES LEITNER** und seine Initiativen zur Talentförderung²⁰ sowie die Didaktikhefte der österreichischen Mathematischen Gesellschaft²¹ verwiesen. Ein Auszug eines von Kolleg:innen der Universität Klagenfurt verfassten Skriptums zu Leistungsbeurteilungen im schulischen Kontext ist ebenfalls im Anhang angegeben. In den Überlegungen werden zudem Grundprinzipien der Montessori-Pädagogik berücksichtigt.²² Auf online-Bildungsportalen wird über Studien an österreichischen Universitäten informiert und unter anderem auf die Karrierechancen von Absolvent:innen im Bereich Technische Mathematik hingewiesen.²³ Inspirierend waren außerdem Interviews mit **ANTON ZEILINGER**, der die Begeisterung seines Physik-Lehrers als einen wesentlichen Impuls für seine Studienwahl beschreibt, die Verschulung von Universitäten kritisch sieht und sich wünscht, dass Hochbegabungen und Ungewöhnliches gefördert wird.

Dank und Zukünftiges Meinen Kollegen **ROBERT HAFNER** und **FABIAN OCHS** bin ich dankbar für neue Perspektiven und ihr konstruktives Feedback bei den kollegialen Hospitationen. Ihre Praxisarbeiten zur realistischen Simulation von Konferenzsituationen und zum Umgang mit inhomogenen Studierendengruppen (MA) habe ich ansatzweise aufgegriffen und in meine Überlegungen für grundlegende Lehrveranstaltungen in den ersten beiden Studienjahren einfließen lassen. Das Lehrkonzept zum Proseminar *Analysis 1* wurde mit **JOHANNES LEITNER** im Rahmen des *Constructive Alignment für Profis in der universitären Lehre* besprochen. Ein gemeinsames Seminar zum Thema *Non scholæ sed vitæ! Wie gelingt das Erkennen und Fördern von Begabungen im eigenen Schulunterricht?* ist geplant. Sehr wertvoll waren die Anregungen der Kommunikationstrainerin **JULIA STRAUHAL**. Das Angebot von Fortbildungen soll speziell in diesem Bereich weiterhin genützt werden.

¹⁹Dies gilt ebenfalls für die Anwendung statistischer Methoden auf wenige, stark variierende Daten. Es ist allgemein bekannt, dass es kein valider statistischer Zugang ist, die in einem Datensatz vorhandene Vielfalt auf einzelne Kennwerte zu reduzieren.

²⁰Siehe www.sci-e-s.com.

²¹Verfügbar unter www.oemg.ac.at/DK/Didaktikhefte.

²²Eine kurze Darstellung und weiterführende Links finden sich unter de.wikipedia.org/wiki/Montessoripädagogik.

²³Als ein Bildungsportal sei studieren.at genannt.

7 Anhang – Unterlagen zum Zertifikat

Überblick

- Anmeldeformular
- Protokolle der Peer-Gruppen-Treffen
- Protokolle der kollegialen Hospitationen

Zertifikat Lehrkompetenz

Informationsgespräch mit Personalentwicklung geführt am 13. Juli 2021

A – Hochschuldidaktische Basisthemen (24 UE)

Constructive Alignment für Profis in der univ. Lehre	absolviert am 17. Dezember 2021	20 UE
Rechtliche Grundlagen für Ihre Lehrtätigkeit	absolviert am 15. November 2021	4 UE

B – Wahlmodul (25 UE)

Präsentationscoaching bei Verena Covi (Bestätigung der Anrechnung durch Frau Augenstein per Email am 22. Juli 2022)	absolviert am 21. September 2020	6 UE
Wie produziere ich audiovisuelle Medien für die Lehre?	absolviert am 18. Oktober 2021	5 UE
Words make a difference – Worte wirken. Wertschätzende Kommunikation im Universitätsbetrieb	absolviert am 25. April 2022	30 UE
Das psychische Immunsystem stärken - Resilienz	absolviert am 29. November 2022	4 UE

C – Praxisarbeit

Ausarbeitung des eigenen Themas	Synchronisation und Netzwerke in Forschung und Lehre
Peer-Gruppe	Robert Hafner, Fabian Ochs, Mechthild Thalhammer
6 Peer-Gruppen-Treffen	31. August 2021 7. September 2021 5. Oktober 2021 9. November 2021 7. Dezember 2021 18. Jänner 2022
3 Kollegiale Hospitationen	14. Jänner 2021, 10:15-11:45 (Fabian Ochs, PS Analysis 1, Gruppe 3) 14. Jänner 2021, 12:00-13:30 (Fabian Ochs, PS Analysis 1, Gruppe 4) 21. Jänner 2021, 10:15-11:45 (Robert Hafner, PS Analysis 1, Gruppe 3)
Fallstudiendialog	geplant für März 2023

Zertifikat Lehrkompetenz

Protokoll zum ersten Treffen der Peer-Gruppe

Anwesende: Robert Hafner, Fabian Ochs, Mechthild Thalhammer
Datum: 31. August 2021
Zeit: 8:00 – 9:00
Ort: Technikerstraße 13, Raum 514

Thematischer Schwerpunkt: Vorstellung und Kennenlernen

Die Mitglieder der Peer-Gruppe stellen ihren wissenschaftlichen Werdegang, die aktuelle thematische Fokussierung in der Forschung, die Einbettung in wissenschaftliche Communities und die derzeitigen Lehraktivitäten vor. Sie umreißen kurz, welche Schwerpunkte sie in Hinblick auf die Absolvierung des Zertifikates Lehrkompetenz setzen wollen.

Obwohl sich die jeweiligen Karrierestufen und methodischen Zugänge der Fachrichtungen deutlich unterscheiden, werden auch Gemeinsamkeiten identifiziert. Alle Mitglieder der Peer-Gruppe haben langjährige Erfahrung in der Lehre und absolvieren Modul C, einerseits als Modul im Rahmen des Zertifikats Lehrkompetenz (Robert Hafner, Mechthild Thalhammer) und andererseits als alleinstehendes Modul (Fabian Ochs). Ein für alle zentrales Thema ist die Bewältigung der beim Unterrichten von heterogenen Studierendengruppen auftretenden Herausforderungen (Knowledge Gaps).

Die Mitglieder der Peer-Gruppe tauschen sich zu den Erfahrungen im Studienjahr 2020/21 aus, wo Lehrveranstaltungen aufgrund der Covid-19-Maßnahmen fast ausschließlich in virtueller Form stattfanden. Sie stimmen überein, dass die Vorteile von Präsenzlehre überwiegen und deshalb in Hinblick auf die Absolvierung des Zertifikates Lehrkompetenz darauf der Fokus gelegt werden soll.

An diesen Meinungsaustausch schließt auch eine kurze Diskussion zu den Erfahrungen und Perspektiven zum Thema Konferenzteilnahmen und Auslandsaufenthalte unter den Gesichtspunkten Klimaschutz und Nachhaltigkeit an.

Als letzter Punkt werden die Anforderungen des Moduls besprochen und ein grober Terminplan skizziert. Um den Fortschritt bei den gewählten Schwerpunkten in der Lehre stetig verfolgen zu können, sollen monatliche Treffen stattfinden.

Als Termin für das zweite Treffen der Peer-Gruppe wird der 7. September 2021, 8 Uhr, Campus Technik fixiert.

Zertifikat Lehrkompetenz

Protokoll zum zweiten Treffen der Peer-Gruppe

Anwesende: Robert Hafner, Fabian Ochs, Mechthild Thalhammer
Datum: 7. September 2021
Zeit: 8:00 – 9:00
Ort: Technikerstraße 13, Raum 514

Thematischer Schwerpunkt: Vorstellung der Lehrkonzepte

Das zweite Treffen dient dazu, die Lehrkonzepte vorzustellen.

Robert Hafner: Aspekte der Mensch-Technik-Umwelt-Beziehung

Lehrveranstaltungsformat: Seminar im Rahmen des Masterstudiums Geographie, Globaler Wandel – regionale Nachhaltigkeit

Thematischer Schwerpunkt: Team-Teaching

Konkrete Fragestellungen: Wie lassen sich partizipative Elemente und das Simulieren von Kongress-Situationen (Vortrag, Moderation, Diskussionsbeiträge) im Rahmen eines Seminars umsetzen? Wurden die Anforderungen den Studierenden verständlich vermittelt? Wurde klar kommuniziert, dass der gewählte Modus des Seminars für Abschlussarbeiten und in Hinblick auf zukünftige Berufsfelder nützlich ist? Wird deutlich, dass es aufgrund der unterschiedlichen fachlichen Ausrichtungen der beiden Lehrenden auch unterschiedliche Zugänge zu den behandelten Themen gibt?

Fabian Ochs: Thermofluiddynamik

Lehrveranstaltungsformat: VU im Rahmen der Masterstudien Mechatronik und Umweltingenieurwissenschaften

Thematischer Schwerpunkt: Bewältigung von „Knowledge Gaps“ bei Studierendengruppen, die sowohl inhaltlich (Bauphysik, Wärmetechnik) als auch methodisch (Vorwissen betreffend Theorie und Anwendungen, Modellierung und Programmierung) heterogen sind.

Konkrete Fragestellungen: Die Rückmeldung von Studierenden zeigte, dass die Behandlung von theoretischen Aspekten und Anwendungen in einer einzigen Lehrveranstaltung als sehr anspruchsvoll wahrgenommen wird. Wie lassen sich theoretische Herangehensweisen derart mit Anwendungsbeispielen verbinden, dass Studierende einerseits die Komplexität des Themas erkennen, andererseits aber nicht überfordert werden? Wie sieht eine passende Auswahl der zu vertiefenden Themen aus? Wie kann Partizipation dem Verständnis helfen?

Mechthild Thalhammer: Analysis 1

Lehrveranstaltungsformat: Proseminar in den Bachelorstudien Mathematik, Mathematik Lehramt, Atmosphärenwissenschaften und Physik

Thematischer Schwerpunkt: Bewältigung von „Knowledge Gaps“ bei Erstsemestrigen, Fördern der Motivation durch Vermittlung der Relevanz von mathematischen Inhalten speziell für die Studienrichtungen Atmosphärenwissenschaften und Physik.

Konkrete Fragestellungen: Wie kann man Studierenden den Einstieg in ein anspruchsvolles Studium erleichtern, insbesondere bei Lehrveranstaltungen mit abstrakten mathematischen Inhalten? Wie kann man den Modus von Proseminaren gestalten, um heterogene Kenntnisse auszugleichen, und ein Monitoring gestalten, um bei einer Vielzahl von zu behandelnden Proseminar-Aufgaben zusätzlichen Erklärungsbedarf bei gewissen Themen frühzeitig zu erkennen? Durch welche Mittel kann man die Motivation der Studierenden fördern?

Als Termin für das dritte Treffen der Peer-Gruppe wird der 5. Oktober 2021, 16 Uhr, Campus Innrain fixiert.

Zertifikat Lehrkompetenz

Protokoll zum dritten Treffen der Peer-Gruppe

Anwesende: Robert Hafner, Fabian Ochs, Mechthild Thalhammer
Datum: 5. Oktober 2021
Zeit: 16:45 – 17:45
Ort: Bruno-Sander-Haus, Raum 60611

Thematischer Schwerpunkt: Terminplanung

Das dritte Treffen dient dazu, einerseits die kollegialen Hospitationen und andererseits die weiteren Treffen der Peer-Gruppe terminlich und inhaltlich zu planen.

Alle Mitglieder der Peer-Gruppe stimmen überein, dass eine Durchführung der kollegialen Hospitationen innerhalb der Gruppe ideal ist. Es werden folgende Termine bzw. Zeiträume vereinbart:

Robert Hafner:

Aspekte der Mensch-Technik-Umwelt-Beziehung, Seminar im Rahmen des Masterstudiums Geographie, Globaler Wandel – regionale Nachhaltigkeit

Kollegiale Hospitationen am 21. Oktober 2021 durch Mechthild Thalhammer, am 11. November 2021 durch Fabian Ochs, und am 13. Jänner 2022 durch Mechthild Thalhammer.

Fabian Ochs:

Thermofluidynamik, Vorlesung/Übung im Rahmen der Masterstudien Mechatronik und Umweltingenieurwissenschaften

Kollegiale Hospitationen am 25. Oktober 2021 durch Mechthild Thalhammer, am 29. November 2021 durch Robert Hafner, und zu Beginn des Sommersemesters 2022 durch Robert Hafner.

Mechthild Thalhammer:

Analysis 1, Proseminar im Rahmen der Bachelorstudien Mathematik, Mathematik Lehramt, Atmosphärenwissenschaften und Physik

Kollegiale Hospitationen am 17. Dezember 2021 bei zwei Proseminargruppen (Atmosphärenwissenschaften und Physik, 10-12 Uhr / 12-14 Uhr) durch Fabian Ochs und am 21. Jänner durch Robert Hafner.

Die wesentlichen inhaltlichen Punkte der kollegialen Hospitationen werden besprochen. Es wird vereinbart, dass allgemeine Informationen zu den Lehrveranstaltungen und detaillierte Informationen zu speziellen Aspekten, auf die bei den Hospitationen der Fokus gelegt werden soll, im Vorfeld der jeweiligen Hospitation nochmals per Email ausgetauscht werden. Nachbesprechungen zu den jeweiligen Hospitationen werden direkt im Anschluss durchgeführt.

Als Termin für das vierte Treffen der Peer-Gruppe u.a. zur gemeinsamen Besprechung der ersten kollegialen Hospitationen wird der 9. November 2021, 15 Uhr, Campus Technik fixiert. Als weitere Termine sind der 7. Dezember 2021 (Austausch zu Fortbildungen) und der 18. Jänner 2022 (Notengebung u.a. bei Gruppenarbeiten und Abschlussarbeiten) geplant.

Zertifikat Lehrkompetenz

Protokoll zum vierten Treffen der Peer-Gruppe

Anwesende: Robert Hafner, Fabian Ochs, Mechthild Thalhammer
Datum: 9. November 2021
Zeit: 15:00 – 16:00
Ort: Technikerstraße 13, Raum 514

Thematischer Schwerpunkt: Constructive Alignment und Hospitationen

Das vierte Treffen dient dazu, einerseits von Erfahrungen betreffend die Fortbildung Constructive Alignment zu berichten und sich andererseits in der Peer-Gruppe zu den bisherigen kollegialen Hospitationen auszutauschen.

Constructive Alignment

Robert Hafner hat die Fortbildung Constructive Alignment bereits absolviert. Ein Kernthema ist die umfassende Planung und Durchführung von Lehrveranstaltungen (Definition der Lernziele einer Lehrveranstaltung sowie der vermittelten inhaltlichen, methodischen und sozialen Kompetenzen, Leistungsmaßstäbe und Beurteilungskriterien, transparente Kommunikation). Dazu gibt es entsprechende Inputs, Gruppenarbeiten und Diskussionen.

Robert Hafner fasst seine Erfahrungen kurz zusammen und stellt die Eckpunkte der von ihm erstellten Abschlussarbeit vor.

Es folgt eine Diskussion zu Benotungen in Seminaren und bei Lehrformaten Vorlesung & Übung. Lassen sich theoretisch plausible Ansätze eins zu eins praktisch umsetzen? Wie groß ist der Zeitaufwand bei sehr detaillierten Leistungsfeststellungen über ein Semester hinweg? Sollen anfänglich festgesetzte Kriterien bei Bedarf adaptiert werden?

Dieses Thema soll in einer der nächsten Treffen der Peer-Gruppe weiter vertieft werden.

Hospitationen

Die überwiegend positiven Erfahrungen der kollegialen Hospitationen von Mechthild Thalhammer bei Robert Hafner und Fabian Ochs werden in der Peer-Gruppe besprochen und die bei weiteren Hospitationen zu beachtenden Punkte präzisiert.

Beim Seminar von Robert Hafner wird es in erster Linie um die Vorbereitung und Simulation von Kongress-Situationen gehen. Bei den Hospitationen von Fabian Ochs und Mechthild Thalhammer soll beobachtet werden, wie Studierende solche für sie neuartige Situationen meistern.

Bei der zweiten Hospitation von Mechthild Thalhammer bei Fabian Ochs soll es hauptsächlich um die Frage der Vereinbarkeit von Theorie und Anwendung (Implementierung in MATLAB) innerhalb einer Lehrveranstaltungseinheit gehen. Schaffen die Studierenden rasche Wechsel zwischen theoretischen und praktischen Aspekten? Wie kann man ihnen den Umstieg erleichtern?

Der Schwerpunkt der beiden Hospitationen von Fabian Ochs bei Mechthild Thalhammer wird auf den Einflussfaktoren heterogene Gruppenzusammensetzung und Gruppendynamik liegen. Wie gelingt bei

zwei unterschiedlich zusammengesetzten Proseminar-Gruppen mit Studierenden aus den Bereichen Atmosphärenwissenschaften und Physik jeweils die Durchführung der Lehrveranstaltung? Bleiben bei der späteren Proseminar-Gruppe am Freitag Nachmittag Motivation und Konzentration gleichermaßen erhalten?

Covid-19

Das Treffen endet mit einer Akkordierung der weiteren Termine in Anbetracht der aktuellen Covid-19-Situation. Man ist sich einig, dass man betreffend Hospitationen Treffen der Peer-Gruppe kurzfristig reagieren und gegebenenfalls von Präsenz auf Distanz wechseln sollte.

Zertifikat Lehrkompetenz

Protokoll zum fünften Treffen der Peer-Gruppe

Anwesende: Robert Hafner, Fabian Ochs, Mechthild Thalhammer

Datum: 7. Dezember 2021

Zeit: 15:00 – 16:00

Ort: Virtuell via BBB

Thematischer Schwerpunkt: Austausch zu Fortbildungen

Das Treffen der Peer-Gruppe wird zum Erfahrungsaustausch über von der Universität Innsbruck angebotene Fortbildungen u.a. zum Zertifikat Lehrkompetenz genützt. Man stimmt überein, dass bei derartigen Veranstaltungen das Kennenlernen von Mitgliedern anderer Institute und Fakultäten ein sehr positiver Aspekt ist.

Fortbildung zum Datenschutz

Es handelt sich um eine interne Fortbildung mit der Datenschutzkoordinatorin der Universität Innsbruck. Behandelt werden Grundfragen zu Datenschutz und Studierendendaten, speziell auch die Verwendung von Arbeitslaptops. Einerseits dürfen Daten zu Lehrveranstaltungen nicht auf Arbeitslaptops gespeichert werden, andererseits ist es zeitweise erforderlich, auf Dienstwegen oder Dienstreisen ohne stabile VPN-Verbindung zu Servern der Universität, Arbeiten zu erledigen. Eine technische Lösung liegt beispielsweise in der automatische Verschlüsselung von Speichermedien. Fabian Ochs hat an der Fortbildung teilgenommen und hebt den guten Diskussionscharakter der Veranstaltung hervor.

Fortbildung „Vom Forschen zum Führen“

Diese Fortbildung wird von zwei externen Personen geleitet. Sie behandelt Kommunikationskompetenzen und Dynamiken in der Zusammenarbeit. Konkrete Themen sind u.a. Kommunikationstypen, Verbesserung von Kommunikations- und Führungskompetenzen durch Selbstreflexion und Selbsteinordnung, Unterscheidung von Delegation, Coaching, Inspiration. Fabian Ochs empfiehlt die Fortbildung aufgrund für ihn neuer und interessanter Impulse.

Fortbildung für Führungskräfte

Robert Hafner nimmt aktuell an einem Coaching am MCB teil. Dieses bietet die Möglichkeit, auf im Arbeitsalltag auftretende konfliktreiche Situationen einzugehen. Pro Kalenderjahr werden bei 6 Stunden bis zu 75% über die Personalentwicklung und bis zu 25% über das Institut im Rahmen einer QV refundiert. Weiterführende Details finden sich unter <https://www.uibk.ac.at/personalentwicklung/fuehrungskraefte/fkcoaching/>.

Constructive Alignment

Robert Hafner und Mechthild Thalhammer haben zu unterschiedlichen Zeiten an der Fortbildung Constructive Alignment teilgenommen und berichten von ähnlichen Erfahrungen. Als hilfreich werden präsentierte neuartige didaktische Methoden bzw. Erkenntnisse und die Möglichkeit zur Selbstreflexion des erstellten Lehrkonzeptes eingeschätzt. Es werden vielfältige Themen angeschnitten, wo eine zusätzliche selbständige Vertiefung sinnvoll ist. Positiv waren auch die Interaktionen mit anderen TeilnehmerInnen.

Rechtliche Grundlagen

Robert Hafner und Mechthild Thalhammer ordnen die Fortbildung als informativ ein. Auch hier ist eine selbständige inhaltliche Vertiefung bestimmter Thematiken sinnvoll.

Terminplanung

Im Idealfall können alle weiteren kollegialen Hospitationen in Präsenz und bis Jänner 2022 abgeschlossen werden. Sollte es erforderlich sein, wird man auf virtuelle Formate wechseln oder Hospitationen auf einen späteren Zeitpunkt (März 2022) verschieben.

10.1. Kollegiale Hospitation von Robert Hafner bei Fabian Ochs (14:00-15:00)

14.1. Kollegiale Hospitation von Fabian Ochs bei Robert Hafner (8:30-9:30)

14.1. Kollegiale Hospitation von Fabian Ochs bei Mechthild Thalhammer (ab 10:15)

17.1. Kollegiale Hospitation von Mechthild Thalhammer bei Fabian Ochs (ab 14:00)

21.1. Kollegiale Hospitation Robert Hafner bei Mechthild Thalhammer (ab 10:15)

Das nächste Treffen mit dem Thema Notengebung wird für 18. Jänner 2022, 15 Uhr vereinbart, in Präsenz am Campus Technik oder gegebenenfalls in virtueller Form.

Zertifikat Lehrkompetenz

Protokoll zum sechsten Treffen der Peer-Gruppe

Anwesende: Robert Hafner, Fabian Ochs, Mechthild Thalhammer
Datum: 18.1.2021
Zeit: 15:00 – 16:00
Ort: virtuell via BBB

Thematischer Schwerpunkt: Diverses und Notengebung

Abschlussarbeiten und Fallstudiendialog

Zu Beginn des Treffens tauscht man sich zum Zertifikat Lehrkompetenz aus. Es geht insbesondere um die von den Mitgliedern der Peer-Gruppe noch zu erstellenden Arbeiten und die abschließende Präsentation, den Fallstudiendialog.

Als wertvolle Impulse für die eigenen Arbeiten werden die Treffen und die erfolgten kollegialen Hospitationen beurteilt. Auch wenn für das Zertifikat keine weiteren Treffen der Peer-Gruppe vorgegeben sind, möchte man weiterhin in Kontakt bleiben, sich gegenseitig bei der Planung und dem Schreiben der Arbeiten unterstützen (Recherchen, Literaturpools, Diskussionen, Präsentationen).

Der nächste Fallstudiendialog ist für 7. März 2022, 14:00 – 15:30 geplant und wird bis Anfang Februar im internen Fortbildungsprogramm der Universität veröffentlicht.

Robert Hafner wird als erster mit dem Verfassen der Arbeit beginnen. Er umreißt, welche Schwerpunkte er in Hinblick auf das Seminar „Aspekte der Mensch-(Technik)-Umwelt-Beziehung“ setzen möchte. Mechthild Thalhammer wird aktuelle Lehrveranstaltungen für Erstsemestrige reflektieren, durch die Fortbildung Constructive Alignment Kurs entwickelte Ideen aufnehmen und ihr Konzept für Lehrveranstaltungen im Sommersemester 2022 adaptieren.

Kollegiale Hospitationen

Aufgrund der Covid-19-Situation werden die Termine für weitere kollegiale Hospitationen nochmals abgesprochen.

21.1. Kollegiale Hospitation von Robert Hafner bei Mechthild Thalhammer (10:15-11:45)

15.3. Kollegiale Hospitation von Robert Hafner bei Fabian Ochs (8:30-10:00)

22.3. Kollegiale Hospitation von Mechthild Thalhammer bei Fabian Ochs (8:30-10:00)

Nachbesprechung der Hospitationen

Fabian Ochs bei Mechthild Thalhammer: wenig Optimierungsmöglichkeiten: Onlineformat gut adaptiert, Erklärungen gut verständlich, Studierenden gut begleitet, gutes Umfeld für Studierende, souveräne Beiträge von Studierenden; Anregung: zwischendurch Formatänderungen zugunsten von mehr Abwechslung

Fabian Ochs bei Robert Hafner: Gesamteindruck sehr positiv: souveräne Beiträge von Studierenden; Verbesserung zum letzten Hospitieren: aktive Aufforderung zum Einschalten der Studierendenkamearas

Notengebung

Als letzten Punkt des Treffens werden unterschiedliche Aspekte einer angemessenen und fairen Notengebung besprochen: detaillierte Benotungsschemata (Pros und Contras, Lehrveranstaltungstyp, quantitative und qualitative Kriterien, Zeitaufwand), Vergleichbarkeit von Arbeitsaufträgen/Projekten, Umgang mit quantitativ und qualitativ stark variierenden Studierendenbeiträgen, Gruppenarbeiten (Lerneffekt, Umgang mit dominanten/zurückhaltenden Persönlichkeiten, Konflikten vorbeugen), Tendenz zu strengen Maßstäben zu Studienbeginn und mehr Milde bei Studienende

Kollegiale Hospitationen in Proseminaren zur Analysis 1 durch Robert Hafner und Fabian Ochs

Inhalte Im Proseminar zur Analysis 1 werden grundlegende mathematische Inhalte, insbesondere zu den Themen Funktionen, Konvergenz von Folgen und Reihen, Stetigkeit, Differentiation, Integration und Taylor-Approximation besprochen.

Ablauf Im wöchentlichen Rhythmus werden im Eigenstudium schriftliche Ausarbeitungen von vorgegebenen Aufgaben erstellt (Angabe unter OLAT). Diese Ausarbeitungen werden im Proseminar vorgestellt (mittels BBB, meist eine Person pro Aufgabe, freiwillige Meldung) und daran anknüpfend Begriffe, Resultate und Methoden sowie alternative Zugänge und Fehlerquellen besprochen. Die Lehrende beantwortet auftretende Fragen, ergänzt Erklärungen zur vertiefenden oder anwendungsorientierten Aspekte und gibt Hinweise zum nächsten Übungsblatt. Falls erforderlich, revidieren die Studierenden ihre Ausarbeitungen nochmals im Eigenstudium und erstellen Nachbereitungen (Angabe unter OLAT).

Hospitationen Die kollegialen Hospitationen durch Robert Hafner und Fabian Ochs fanden am 14. und 21. Jänner 2022 statt, jeweils bei zwei Proseminar-Gruppen (10 – 12 Uhr, 12 – 14 Uhr). Insgesamt waren rund 15 Studierende pro Gruppe anwesend.

Virtuelles Format Aufgrund der geltenden Covid-19-Maßnahmen musste die Lehrveranstaltung in virtueller Form abgehalten und der Beobachtungsauftrag entsprechend angepasst werden.

Beobachtungsauftrag Ein wesentliches Lernergebnis von Proseminaren für Studienanfänger:innen liegt darin, mathematische Vorgehensweisen anhand der im Rahmen der Vorlesung behandelten Inhalte zu *verstehen*.

Absolventinnen und Absolventen dieses Moduls verstehen die Inhalte der Vorlesung und können diese wiedergeben und anwenden. Sie haben die Fertigkeit erworben, sich ähnliche Inhalte selbständig zu erarbeiten. Sie sind in der Lage, die grundlegenden Konzepte der Analysis situationsgerecht anzuwenden. Weiters haben sie ein Grundverständnis für die Denkweise der Mathematik erlangt.

Die hospitierenden Kollegen wurden deshalb gebeten, speziell auf die Interaktion der Lehrenden mit einzelnen Studierenden und der gesamten Gruppe sowie die inhaltliche und akustische Verständlichkeit ihrer mündlichen und schriftlichen Anmerkungen und zusätzlichen Erklärungen zu achten. Hat die Teilnahme am Proseminar dazu beigetragen, das Verständnis für mathematische Herangehensweisen und Inhalte zu vertiefen und Fehlkonzepte zu bereinigen, so wird in Kombination mit der nochmaligen Auseinandersetzung durch Nachbereitungen ein nachhaltiger Lernprozess unterstützt.

Beobachtungen und Bewertungen

- *Fachliche Expertise der Kollegen.* Aufgrund seines Ingenieurstudiums und seiner beruflichen Tätigkeiten ist Fabian Ochs mit mathematischen Begrifflichkeiten und Methoden wohlvertraut. Da die fachliche Expertise von Robert Hafner im Bereich Humangeographie liegt, konnte er mathematische Problemstellungen und Erklärungen nur in geringerem Ausmaß nachvollziehen. Dennoch fand er die Teilnahme an den Proseminaren sehr spannend.
- *Verständlichkeit der Lehrenden.* Robert Hafner bewertete es als positiv, dass die Lehrende mündliche organisatorische Ansagen, inhaltliche Ergänzungen und Erklärungen bei Fragen noch zusätzlich schriftlich in den *Geteilten Notizen* festhielt. Die Informationen zur Prüfung

waren für ihn klar und verständlich formuliert. Auch von Fabian Ochs wurden die Kommentare und ergänzenden Erklärungen durch die Lehrende als hilfreich und verständlich beurteilt. Dabei unter anderem die *Geteilten Notizen* zu nutzen, wurde ebenfalls als eine gute Option gesehen.

- *Tempo der Lehrveranstaltung.* Da die Mehrheit der Studierenden im ersten Semester mit mathematischen Denkweisen und Beweismethoden nicht vertraut sind und ausführliche Erklärungen benötigt werden, besteht speziell bei interaktiven Lehrveranstaltungen ein gewisser Zeitdruck. Von Fabian Ochs wurde das Tempo als passend eingeschätzt. Robert Hafner fand den Einstieg in die Lehrveranstaltung mit einem auf die Minute pünktlichen Beginn sehr schnell. Er beobachtete, dass sich Studierende sofort aktiv im Chat beteiligten. Er fand die schriftliche Zusatzerklärungen in den *Geteilten Notizen* gelungen, weil dies das anfängliche Tempo reduzierte und man zudem leichter folgen konnte.
- *Inhaltliche Beiträge der Lehrenden.* Das Proseminar wurde mit inhaltlichen Anmerkungen durch die Lehrende begonnen und mit einem Ausblick auf das nächste Mal beendet. Dies wurde von Fabian Ochs als motivierend beurteilt. Robert Hafner fand es gut, dass die Lehrende den Hinweis gab, dass sie eine ausführliche Ausarbeitung zu einer anspruchsvollen Aufgabe erstellen und verfügbar machen werde (unter OLAT).
- *Interaktion und Kommunikation.* Von Robert Hafner wurde die Interaktion zwischen der Lehrenden und den Studierenden als sehr gut beurteilt. Er stufte die Antworten und Erläuterungen als sehr direkt ein. Er fand die an die Studierenden gerichteten Feedbacks ermutigend und die Balance zwischen positiver Rückmeldung und konstruktiver Kritik gut gewählt. Er hob dabei insbesondere die Integration und das Eingehen auf im Chat gestellten Fragen von Studierenden hervor. Außerdem fiel ihm die häufige Einladung der Lehrenden an die Studierenden zu Fragen auf (*Bei Fragen bitte melden!*).
- *Wissensstand der Studierenden.* Von Fabian Ochs wurden die Antworten und Begründungen von Studierenden auf Nachfragen der Lehrenden und die Fragen von Studierenden als inhaltlich gut eingeschätzt.
- *Nutzung virtueller Möglichkeiten.* Von Robert Hafner wurde angeregt, Big Blue Button via OLAT anzulegen. Da die Studierenden dann mit vollem Namen aufscheinen, erleichtert dies der Lehrenden (zu Beginn des Semesters) die Anwesenheitskontrolle. Zur besseren Einbindung aller Studierenden, wäre eine stabilere Internetverbindung wünschenswert, damit zusätzlich alle Kameras aktiviert werden können. Von Fabian Ochs wurde erwähnt, dass im virtuellen Raum im Vergleich zur Lehre in Präsenz zwischendurch mehr Zeit für Pausen und Formatwechsel gegeben werden kann, um die Konzentration aufrecht zu erhalten.
- *Akustische Verständlichkeit.* Da die Lehrveranstaltung in virtueller Form stattfand, kam es speziell bei der kollegialen Hospitation durch Fabian Ochs trotz deaktivierter Kameras zu einer instabilen Verbindung und gewissen Störgeräuschen. Dennoch wurde die Lehrende von Fabian Ochs als akustisch gut verständlich wahrgenommen.
- *Gruppendynamik.* Von Fabian Ochs wurde zwischen den Gruppen kein großer Unterschied festgestellt. Dies ist insofern unerwartet, weil aus subjektiver Sicht bei regulären Lehrveranstaltungen ab Freitag Mittag die Erschöpfung der Studierenden deutlich zunimmt und sich auf die Gruppendynamik auswirkt. Als Lehrende versucht man dies durch erhöhte positive Aktivität und Spiegelung abzufangen.

Persönliches Fazit Aufgrund der fortwährenden Interaktion mit einzelnen Studierenden und der gesamten Gruppe sowie spontanen Entscheidungen, was zusätzliche Erklärungen und deren Ausführlichkeit betrifft, finde ich die Aufgabe, als Lehrende in Proseminaren eine ausgewogene inhaltliche Strukturierung zu erreichen, durchaus herausfordernd. Um die Nachvollziehbarkeit zu erhöhen und das Tempo zu reduzieren, erscheint es mir sinnvoll, mündliche Angaben weiterhin auch gleichzeitig schriftlich festzuhalten. Hinsichtlich einer guten Strukturierung und da dies außerdem mehr Raum für kurze Pausen zwischendurch lässt, möchte ich Folgendes ausprobieren: Fragen von Studierenden (zu einer Aufgabe bzw. einem Themenbereich), die umfassendere Erklärungen benötigen, nicht ad hoc beantworten, sondern erst einmal schriftlich sammeln und dann systematischer besprechen. Ich denke, dass dies ein weiteres praktikables Element ist, um Studierende bei Prozessen des Verstehens und Lernen zu unterstützen.

8 Anhang – Ergänzende Quellen

Überblick

- Gespräch mit der Lernforscherin Elsbeth Stern
(Kurzfassung und Transkription)
- Unterlagen eines Vortrages des Mathematikers Günter Hanisch
Österreichische Mathematische Gesellschaft, Didaktikheft 44 (2011)
- Online-Bildungsportal
(Auszug zu Studien im Bereich Technische Mathematik)
- Skriptum zur Leistungsbeurteilung
(Universität Klagenfurt, Fakultät für Kultur- und Bildungswissenschaften)

Websites

www.zeit.de
ifvll.ethz.ch/ueber-uns/personen/personen-forschung/prof-dr-elsbeth-stern.html
www.oemg.ac.at/DK/Didaktikhefte
studieren.at
www.aau.at/wp-content/uploads/2017/10/3-LEISTUNGSFESTSTELLUNG.pdf



Lernen lernen

Es gibt kein Lernen ohne Inhalt

Die Lernforscherin Elsbeth Stern im Interview (2003)

Die Lernforscherin **Elsbeth Stern** äußerte sich in einem Interview in der Wochenzeitung "Die Zeit" im Jahr 2003 über das **Verhältnis von Intelligenz und Vorwissen**. Sie warnte darin aus zwei Gründen vor einer Überschätzung der Intelligenz. Zum einen gehe es darum, dass alle Menschen etwas lernen und nicht nur die, die besonders günstige Voraussetzungen dafür mitbrächten. Und zum anderen sei auch wissenschaftlich belegt, dass Leistungsunterschiede in einem bestimmten Bereich immer weniger von der Intelligenz abhingen, je mehr man das vorhandene Vorwissen in seine Überlegungen mit einbeziehe. Auf die Frage, wie sich dies in der Schule zeige, antwortete Stern, dass es sich im Bereich schulischen Lernens ganz ähnlich verhalte. "Den größten Einfluss auf den Lernfortschritt", so betonte sie, "hatte das zu Beginn eines Schuljahres verfügbare Wissen – und zwar weitgehend unabhängig von der Intelligenz. Eine Fortführung dieser Studien für das Fach Mathematik ergab, dass die Mathematikleistung in Klasse 11 eng mit der Leistung in der Grundschule zusammenhängt, viel enger als mit dem Intelligenzquotienten der Elftklässler. Die Ergebnisse sprechen dafür, dass man sich über einen längeren Zeitraum mit mathematischen Problemen auseinander setzen muss, wenn man ein guter Mathematiker werden möchte. Zugespielt kann man sagen: Wissen schlägt Intelligenz." Und was die Vorteile der intelligenteren Kinder in der Schule anbelangt, räumte sie ein, dass diese natürlich leichter lernen, weil eben die Kombination von Wissen und Intelligenz die wirkungsvollste Voraussetzung sein. Aber zugleich betonte sie auch, dass dieser Vorteil auch wirklich genutzt werden müsse, denn auch intelligente Kinder verspielten ihren Vorsprung, wenn sie nicht lernten, und würden dann von weniger intelligenten Kindern überholt. Die Frage, ob dieser Vorsprung uneinholbar sei, wurde von Elsbeth Stern eindeutig verneint. Allerdings betonte sie in diesem Zusammenhang, dass derjenige, der sich auf seiner Intelligenz ausruhe, abgehängt werde. Denn, so begründete sie diese Auffassung, Defizite in der Intelligenz ließen sich durch Vorwissen offensichtlich wettmachen. Umgekehrt sei dies allerdings nicht der Fall. So kommt sie zum Schluss, Intelligenz komme letzten Endes nur dem wirklich zugute, der sie in Wissen umgesetzt habe. Auf die Frage, ob sie damit letzten Endes zur "guten alten Paukschule" zurückkehren wolle, antwortete die Lernforscherin: "Nein, ganz im Gegenteil, wir brauchen einen viel moderneren Unterricht. Und natürlich sind Sozialkompetenz und Lernstrategien wichtig. Aber die Vorstellung, wonach man derartige Kompetenzen losgelöst vom Inhaltswissen lernen kann, ist nicht richtig."

Wenig hält Stern davon, Lernen lernen als eine Art "Trockenübung" zu veranstalten, denn Lernstrategien fielen stets als Nebenprodukt ab, wenn Unterricht anregend sei. Denn, daran lässt sie keinen Zweifel: "Die so genannten Schlüsselqualifikationen sind lernbar, aber nicht lehrbar. Nehmen Sie das Beispiel der Sozialkompetenz. Die umfasst auch die Fähigkeit, einen schwierigen Sachverhalt anderen Menschen erklären zu können. So etwas kann nicht in einer beliebigen Situation trainiert werden. Wenn die Kinder als Übung zur Sozialkompetenz gemeinsam frühstücken, dann lernen sie dabei nicht, wie man einem anderen Kind eine Mathematikaufgabe erklärt. Der Inhalt ist eben nicht egal."

Experimente, z. B., arbeitsteilig gestalten zu lassen, und zwar von der Vorbereitung bis zur Auswertung, ist für Stern ein Weg, der als Lösung aus der Problematik hinausführen könnte, wenn es darum geht Inhalte und Lernstrategien und Kompetenzen sinnvoll miteinander zu verknüpfen. Und aus diesem Grunde spreche sie auch lieber von der "Fähigkeit zur inhaltsbezogenen Zusammenarbeit" als von Sozialkompetenz, denn schließlich habe das, worum es dabei gehe, "nichts damit zu tun, wie nett jemand ist oder wie gut er Partys schmeißen kann."

"Zum Beispiel die Schüler ein Experiment arbeitsteilig gestalten lassen, von der Vorbereitung bis zur Auswertung. Damit bereite ich sie auf Situationen vor, mit denen sie im Beruf konfrontiert werden. Ich sage statt Sozialkompetenz lieber 'Fähigkeit zur inhaltsbezogenen Zusammenarbeit'."

Auf die Aufforderung, dieses genauer zu erläutern, führte die Lernforscherin aus: "Die Auffassung, dass wir unseren Intellekt am besten an möglichst komplexen und abstrakten Problemen schulen, unabhängig vom Inhalt. Dahinter steckt häufig die Annahme, dass eine Aufgabe auf ihre abstrakte Struktur reduziert wird, welche dann auf neue Probleme übertragen werden kann. Diese Art von Wissenstransfer – das zeigen zahlreiche Studien – bleibt jedoch in der Regel aus. Es gibt kein Lernen ohne Inhalt. Wenn man möchte, dass Schüler verstehen, warum ein Auto fährt, müssen sie physikalische Gesetzmäßigkeiten verstehen. Wenn ich lernen will, Texte zu lesen und zu verstehen, dann muss ich eben anspruchsvolle Texte lesen und interpretieren. Da kann ich nicht irgendwas machen. Latein ist auch ein gutes Beispiel: Dem Lateinlernen wurde immer ein geheimnisvolles Gehirntraining nachgesagt. Unsere Studien haben aber ergeben, dass es nicht etwa das logische Denken fördert. Es fördert das, was es zum Inhalt hat, etwa das genaue Achten auf grammatische Strukturen. Wir müssen uns von der Vorstellung lösen, dass unser Gehirn mit beliebigen geistigen Aktivitäten trainiert werden kann."

ZEIT: Aber das Anhäufen von Wissen kann doch nicht die Alternative sein.

STERN: Mir geht es um intelligentes Wissen, das sich breit und flexibel einsetzen lässt, nicht einfach nur um Faktenwissen. Die Schüler sollen Konzepte verstehen, nicht isolierte Informationen sammeln."

(Interviewauszüge aus: "Wissen schlägt Intelligenz" - Der Geist kann nicht an beliebigen Themen trainiert werden. Ein Gespräch mit der Lernforscherin Elsbeth Stern, in: Die Zeit 27(2003), 26.6.2003)

Transkription eines Beitrages zum Thema *Lernen*

Wissen schlägt Intelligenz
Der Geist kann nicht an beliebigen Themen trainiert werden.
Ein Gespräch mit der Lernforscherin Elsbeth Stern

26. Juni 2003 / Quelle: (c) DIE ZEIT 26.06.2003 Nr. 27

Zeit: Frau Professor Stern, muss man als Hochschullehrer intelligent sein?

Stern: Es gibt Berufe, die eine höhere Intelligenz voraussetzen, und dazu gehört sicher auch der des Professors, insbesondere weil man als Wissenschaftler ständig Neues dazulernen muss. Ein hoher Intelligenzquotient reicht jedoch nicht. Bei einem erfolgreichen Wissenschaftler müssen Ausdauer, Fleiß, Anstrengungsbereitschaft oder Motivation hinzukommen.

Zeit: Man könnte meinen, Sie führten einen Feldzug gegen die Intelligenz.

Stern: Wie kommen Sie darauf?

Zeit: Gerade haben Sie mit Kollegen aus Österreich in einer Studie gezeigt, dass weniger intelligente Menschen ihr Fachgebiet beherrschen können, wenn sie nur üben. Und von Ihnen stammt der Satz: Wissen, nicht Intelligenz ist der Schlüssel zum Können.

Stern: Stimmt. Aber Intelligenz bleibt ein sehr sinnvolles wissenschaftliches Konstrukt, um die unterschiedliche geistige Leistungsfähigkeit von Menschen zu erklären, die die gleichen Lerngelegenheiten hatten. Intelligenz beeinflusst ohne Zweifel den Erfolg in der Schule und im Beruf.

Zeit: Weshalb warnen Sie dann vor einer Überschätzung der Intelligenz?

Stern: Aus zwei Gründen: Zum einen haben wir ja das Ziel, dass alle etwas lernen und nicht nur Menschen mit günstigeren geistigen Voraussetzungen. Da wird die Frage interessant, in welchem Ausmaß und in welchen Bereichen weniger günstige Voraussetzungen kompensiert werden können. Mit der von Ihnen angesprochenen Studie versuchen wir erstmals, diese Kompensationsmöglichkeiten anhand der Gehirntätigkeit zu erforschen. Zum anderen ist klar belegt, dass Intelligenz zur Erklärung von Leistungsunterschieden in einem bestimmten Bereich an Bedeutung verliert, sobald das Vorwissen über diesen Bereich einbezogen wird. Dies zeigen insbesondere auch Ergebnisse der Expertiseforschung.

Zeit: Zum Beispiel?

Stern: Das Schachspiel ist ein sehr gut erforschtes Gebiet. Vergleicht man nun Schachexperten und Anfänger, so genannte Novizen, dann zeigt sich, dass intelligente Novizen schlechter abschneiden als weniger intelligente Experten. Und Leistungsunterschiede zwischen Experten lassen sich nicht auf Intelligenzunterschiede zurückführen.

Zeit: Und wie sieht es in der Schule aus?

Stern: Ähnlich. Hier sind insbesondere die Arbeiten meines vor zwei Jahren verstorbenen akademischen Lehrers Franz E. Weinert vom Max-Planck-Institut für psychologische Forschung in München zu nennen. Es zeigte sich: Den größten Einfluss auf den Lernfortschritt hatte das zu Beginn eines Schuljahres verfügbare Wissen – und zwar weitgehend unabhängig von der Intelligenz. Eine Fortführung dieser Studien für das Fach Mathematik ergab, dass die Mathematikleistung in Klasse 11 eng mit der Leistung in der Grundschule zusammenhängt, viel enger als mit dem Intelligenzquotienten der Elftklässler. Die Ergebnisse sprechen dafür, dass man sich über einen längeren Zeitraum mit mathematischen Problemen auseinandersetzen muss, wenn man ein

guter Mathematiker werden möchte. Zugespißt kann man sagen: Wissen schlägt Intelligenz.

Zeit: Aber intelligente Kinder lernen leichter.

Stern: Das ist ihr Vorteil. Die Kombination aus Wissen und Intelligenz ist natürlich die beste Voraussetzung. Wenn intelligente Kinder jedoch die Gelegenheit zum Lernen nicht wahrnehmen, dann verschenken sie diesen Vorsprung. Sie werden von weniger intelligenten Kindern überholt, die sich Wissen, etwa in Mathematik, angeeignet haben.

Zeit: Uneinholbar?

Stern: Natürlich nicht. Aber wer sich auf seiner Intelligenz ausruht, der wird abgehängt. Defizite in der Intelligenz können durch Vorwissen offensichtlich wettgemacht werden. Defizite im Vorwissen hingegen nicht. Wenn man die Intelligenz nicht in Wissen umgesetzt hat, dann nützt sie einem gar nichts.

Zeit: Ihre Ansichten über die Schule und das Lernen wirken auf den ersten Blick recht altmodisch: Übung macht den Meister lassen Sie über Ihre neue Studie verlautbaren. Die Schule habe die Aufgabe, Wissen weiterzugeben und zu bewahren, schreiben Sie in einem Aufsatz. Sie polemisieren gegen das Lehren von so genannten Schlüsselqualifikationen wie Sozial- oder Lernkompetenz. Wollen Sie zurück zur guten alten Paukschule?

Stern: Nein, ganz im Gegenteil, wir brauchen einen viel moderneren Unterricht. Und natürlich sind Sozialkompetenz und Lernstrategien wichtig. Aber die Vorstellung, wonach man derartige Kompetenzen losgelöst vom Inhaltswissen lernen kann, ist nicht richtig.

Zeit: Das bedeutet?

Stern: Es ist nicht sinnvoll, vor einer Unterrichtseinheit zunächst Trockenübungen zu Lernstrategien zu machen. Lernstrategien fallen vielmehr als Nebenprodukt eines anregenden Unterrichts ab. Die so genannten Schlüsselqualifikationen sind lernbar, aber nicht lehrbar. Nehmen Sie das Beispiel der Sozialkompetenz. Die umfasst auch die Fähigkeit, einen schwierigen Sachverhalt anderen Menschen erklären zu können. So etwas kann nicht in einer beliebigen Situation trainiert werden. Wenn die Kinder als Übung zur Sozialkompetenz gemeinsam frühstücken, dann lernen sie dabei nicht, wie man einem anderen Kind eine Mathematikaufgabe erklärt. Der Inhalt ist eben nicht egal.

Zeit: Was würden Sie stattdessen machen?

Stern: Zum Beispiel die Schüler ein Experiment arbeitsteilig gestalten lassen, von der Vorbereitung bis zur Auswertung. Damit bereite ich sie auf Situationen vor, mit denen sie im Beruf konfrontiert werden. Ich sage statt Sozialkompetenz lieber Fähigkeit zur inhaltsbezogenen Zusammenarbeit. Das hat nichts damit zu tun, wie nett jemand ist oder wie gut er Partys schmeißen kann.

Zeit: Aber Fähigkeiten oder Wissen aus einem Bereich lassen sich doch auf andere Bereiche übertragen. Ist nicht gerade die berühmte Fähigkeit zur Transferleistung das Ziel der Schule?

Stern: Transfer bleibt häufig ein frommer Wunsch. In Deutschland ist die Idee der formalen Bildung noch weit verbreitet und bestimmt auch die Lehrpläne.

Zeit: Was meinen Sie damit?

Stern: Die Auffassung, dass wir unseren Intellekt am besten an möglichst komplexen und abstrakten Problemen schulen, unabhängig vom Inhalt. Dahinter steckt häufig die Annahme, dass eine Aufgabe auf ihre abstrakte Struktur reduziert wird, welche dann auf neue Probleme übertragen werden kann. Diese Art von Wissenstransfer – das zeigen zahlreiche Studien – bleibt jedoch in der

Regel aus. Es gibt kein Lernen ohne Inhalt. Wenn man möchte, dass Schüler verstehen, warum ein Auto fährt, müssen sie physikalische Gesetzmäßigkeiten verstehen. Wenn ich lernen will, Texte zu lesen und zu verstehen, dann muss ich eben anspruchsvolle Texte lesen und interpretieren. Da kann ich nicht irgendwas machen. Latein ist auch ein gutes Beispiel: Dem Lateinlernen wurde immer ein geheimnisvolles Gehirntraining nachgesagt. Unsere Studien haben aber ergeben, dass es nicht etwa das logische Denken fördert. Es fördert das, was es zum Inhalt hat, etwa das genaue Achten auf grammatische Strukturen. Wir müssen uns von der Vorstellung lösen, dass unser Gehirn mit beliebigen geistigen Aktivitäten trainiert werden kann.

Zeit: Aber das Anhäufen von Wissen kann doch nicht die Alternative sein.

Stern: Mir geht es um intelligentes Wissen, das sich breit und flexibel einsetzen lässt, nicht einfach nur um Faktenwissen. Die Schüler sollen Konzepte verstehen, nicht isolierte Informationen sammeln.

Zeit: Zum Beispiel?

Stern: In der Grundschule lernen Kinder normalerweise, dass Holz schwimmt, Eisen jedoch nicht. Erklärungen, die auf den physikalischen Konzepten Dichte und Auftrieb beruhen, werden jedoch nur selten erarbeitet. Das wäre aber genau die Form von Wissen, an die der spätere Physikunterricht anknüpfen könnte.

Zeit: Überfordert das die Schüler nicht?

Stern: Es gibt keine Untersuchung, die darauf hinweist, dass anregender Unterricht, der an das Vorwissen der Kinder anknüpft und dieses dann gezielt erweitert, schwächeren Schülern schadet.

Zeit: Gilt das auch für Klassen, in denen die Hälfte der Schüler nicht deutsch spricht?

Stern: Nein, was da in einigen deutschen Hauptschulen abläuft, ist natürlich geradezu grotesk. Guter Unterricht knüpft an das Vorwissen der Schüler an. Wenn das aufgrund von Sprachschwierigkeiten auch mit größter Anstrengung nicht kommunizierbar ist, dann müssen zunächst andere Probleme gelöst werden.

Zeit: Sie plädieren auch fürs Auswendiglernen.

Stern: Wenn es sinnvoll in verständnisförderndes Lernen eingebettet ist, ja. Wenn ich etwas auswendig kann, dann hat mein Gehirn mehr Ressourcen für andere geistige Aktivitäten verfügbar. Aber wer nur das Einmaleins runterbeten kann, ohne ein Gefühl für den Zahlenraum zu haben, der weiß natürlich nicht wirklich etwas.

Zeit: Den Grundschulunterricht kritisieren Sie als zu wenig anspruchsvoll. Bei der Iglu-Studie haben die Deutschen doch im internationalen Vergleich recht passabel abgeschnitten.

Stern: Ich sehe das nicht ganz so rosig. Beim Lesen hätten wir eigentlich besser sein müssen. Die deutsche Schriftsprache ist leichter zu erlernen als etwa die englische, denn im Deutschen erfolgt das Übertragen von Lauten auf Buchstabenfolgen viel regelmäßiger. Beim naturwissenschaftlichen Test wiederum gingen kaum Aufgaben ein, die ein Verständnis physikalischer Gesetzmäßigkeiten erfassen. Da wurde mehr nach Fakten gefragt. Das kommt dem deutschen Unterricht entgegen. Und was die mathematischen Kompetenzen angeht, ist das deutsche Ergebnis ja nicht so gut. Aber den Grundtenor in der Debatte um die Iglu-Studie teile ich: So richtig schlimm wird es nach der Klasse 4.

Zeit: Die Grundschule ist Ihnen nicht anspruchsvoll genug. An der Frühförderung von Kindern nörgeln Sie aber auch herum.

Stern: Nicht an sinnvoller Frühförderung, im Gegenteil. Wenn Vierjährige im Kindergarten singen, klatschen und reimen, dann fördert das später das Lesenlernen. Aber die fixe Idee, Kinder in den ersten Lebensjahren mit Lernstoff voll zu stopfen, weil deren Gehirn den wie ein Schwamm aufsaugt, diese Idee lässt sich mit keiner wissenschaftlichen Studie erhärten. Selbst über den Fremdsprachenerwerb in der Grundschule lässt sich trefflich streiten. Für seinen Nutzen gibt es jedenfalls keinen Beleg. Unterricht ist etwas ganz anderes als das natürliche Lernen einer zweiten Muttersprache in bilingualen Familien.

Zeit: Sie sprechen gern von nützlichem Wissen. Wo bleibt da die Bildung?

Stern: Natürlich will ich, dass die Schule gebildete Menschen, autonome Bürger entlässt. Aber gerade wenn es das Ziel ist, urteilsfähige Bürger heranzubilden, dann geht das nicht, ohne eine breite Wissensbasis zu vermitteln. Das Wissen über unsere Geschichte, das Wissen über das mächtige Werkzeug namens Mathematik oder auch das Wissen darüber, woher unser Wissen kommt – das alles ist nützlich Wissen, ohne das es keine Bildung gibt.

Das Gespräch führte Thomas Kerstan

Günter HANISCH

Warum ist die Mathematik so exakt?

0. Einleitung

Der folgende Artikel beschreibt das Wissen über die Mathematik, das nach Ansicht des Verfassers von einem/einer Maturanten/Maturantin erwartet werden sollte. Er enthält somit keine neuen mathematischen Erkenntnisse, allerdings können Teile davon 1 : 1 in den Unterricht umgesetzt werden.

Die oben gestellte Frage kann – je nachdem welches Wort man betont – ganz verschieden beantwortet werden.

1. **Warum** ist die Mathematik so exakt? Oder: *Wie* macht sie das?

Der Hauptgrund ist meines Erachtens der, dass sich ein mathematischer Beweis von einem Beweis, wie er in anderen Wissenschaften üblich ist, unterscheidet. „Ein Beweis ist in der Mathematik die als fehlerfrei anerkannte Herleitung der Richtigkeit oder auch Unrichtigkeit einer Aussage aus einer Menge von Axiomen, die als wahr vorausgesetzt werden, und anderen Aussagen, die bereits bewiesen sind.“ (Vgl.

http://de.wikipedia.org/wiki/Beweis_%28Mathematik%29)

Und daraus folgt für die mathematischen Sätze: „In diesem Sinn sind mathematische Sätze prinzipiell endgültige und allgemeingültige Wahrheiten, so dass die Mathematik als die exakte Wissenschaft betrachtet werden kann.“ (Vgl.

<http://de.wikipedia.org/wiki/Mathematik>). Mathematik unterscheidet sich eben von den anderen Wissenschaften dadurch, dass sie ihre Sätze aus sich ohne Mithilfe anderer Wissenschaften beweist, sofern man die Logik der Mathematik zurechnet.

2. Warum **ist** die Mathematik so exakt? War sie es schon immer?

Die Methode der wissenschaftlichen Mathematik verdanken wir EUKLID. Seine „Elemente“ (um 325 v. Chr. entstanden) sind das mit Abstand einflussreichste Buch der Mathematik-Geschichte. Insbesondere sind die Kapitel über die Geometrie beispielgebend für die akademische Mathematik. (Vgl. Norbert FROESE: EUKLID und die Elemente ☐ Die Entdeckung der axiomatischen Methode durch EUKLID. <http://www.antike-griechische.de/Euklid.pdf>) „Er zeigte darin die Konstruktion geometrischer Objekte, natürlicher Zahlen sowie bestimmter Größen und untersuchte deren Eigenschaften. Dazu benutzte er Definitionen, Postulate (nach ARISTOTELES Grundsätze, die akzeptiert oder abgelehnt werden können) und Axiome (nach ARISTOTELES allgemeine und unbezweifelbare Grundsätze). Viele Sätze der Elemente stammen offenbar nicht von EUKLID selbst. Seine Hauptleistung besteht vielmehr in der Sammlung und einheitlichen Darstellung des mathematischen Wissens sowie der strengen Beweisführung, die zum Vorbild für die spätere Mathematik wurde.“ (Vgl. <http://de.wikipedia.org/wiki/Euklid>)

3. Warum ist **die** Mathematik so exakt? Gibt es die *eine* Mathematik?

Das ist Definitionssache. Erinnern wir uns an EUKLID:

Eines seiner Axiome war das Parallelenaxiom: „In einer Ebene ε gibt es zu jeder Geraden g und *jedem* Punkt A außerhalb von g *genau* eine Gerade, die zu g parallel ist und durch den Punkt A geht.“

Über 2000 Jahre lang wurde versucht, dieses Axiom zu beweisen, also es aus den anderen Axiomen herzuleiten, bis Carl Friedrich GAUß (1777 - 1855) erkannte, dass dies unmöglich ist. Es müsste daher Geometrien geben, in der es mehrere parallele Geraden gibt oder in der es keine Parallele gibt. Nikolai LOBATSCHESKI (1792 - 1856) stellte als erster 1826 eine neuartige Geometrie vor – die hyperbolische Geometrie –, in der zwar die anderen Axiome der euklidischen Geometrie gelten, es aber mehr als eine Parallele durch einen Punkt zu einer gegebenen Gerade gibt. Von Bernhard RIEMANN (1826 - 1866) stammt die elliptische Geometrie, in der es keine Parallele gibt. (Ein anschauliches Modell dafür ist die Kugeloberfläche). Gibt es also nur eine Geometrie oder mehrere?

Ein ähnliches Problem ist die Kontinuumshypothese. Bei ihr geht es um die Mächtigkeit von Zahlenmengen. Für endliche Mengen ist diese die Anzahl ihrer Elemente. Zwei Mengen haben die gleiche Mächtigkeit, wenn sie sich bijektiv aufeinander abbilden lassen. So sind die Menge der natürlichen Zahlen und die Menge der ganzen Zahlen gleich mächtig, aber auch die Menge der rationalen Zahlen, was sich mit Hilfe des

CANTORSchen Diagonalverfahrens (Georg Ferdinand Ludwig Philipp CANTOR 1845 - 1918) zeigen lässt (siehe nebenstehende Abbildung, in der die bereits vorgekommenen Zahlen durchgestrichen sind).

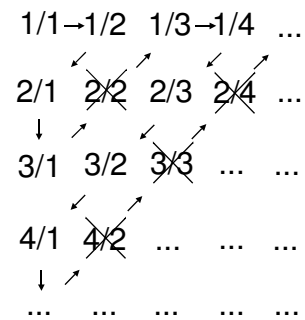
Die Menge der reellen Zahlen hingegen ist mächtiger. Die Hypothese besagt nun, dass es keine Menge gibt, deren Mächtigkeit größer als die der natürlichen Zahlen ist, aber kleiner ist als die der reellen Zahlen. Kurt Friedrich GÖDEL (1906 - 1978)

zeigte, dass diese Hypothese sich nicht widerlegen lässt und Paul Joseph COHEN (1934 - 2007) zeigte, dass sie sich auch nicht beweisen lässt. Die Hypothese ist also im Rahmen der anderen Axiome unentscheidbar. Man kann daher ein Gebäude der Mathematik errichten, in der die Kontinuumshypothese gilt, und ein solches, in der sie nicht gilt.

Dieses Dilemma kann aber einfach definitorisch gelöst werden, indem man eine Definition für Mathematik wählt, die beide Fälle einschließt:

„Die Mathematik ist die Wissenschaft, die selbst geschaffene abstrakte Muster und Strukturen auf ihre Eigenschaften und Muster untersucht.“ (Vgl.

<http://de.wikipedia.org/wiki/Mathematik>)



4. Warum ist die **Mathematik** so exakt?

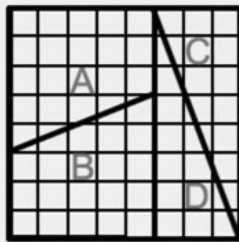
Auf diesen Punkt wurde bereits im ersten Abschnitt eingegangen. Mathematik unterscheidet sich von den anderen Wissenschaften insbesondere durch ihre Beweismethode; daneben ist auch die Art der Definition ihrer Objekte einzigartig. Daher sollten Schüler/innen im Mathematikunterricht auch das Beweisen gelehrt bekommen und selbstständig Beweise durchführen.

Dazu aber ist es notwendig, dass sie erkennen, dass und warum Beweisen notwendig ist, denn in den anderen Unterrichtsfächern geschieht dies im Allgemeinen so nicht. In Geschichte (und Sozialkunde) reicht etwa die Mitteilung einer Tatsache seitens der Lehrkraft. In Physik oder Chemie erfolgt das Beweisen im Allgemeinen durch Versuche, was aber in der Mathematik nicht als Beweis gilt. Im Mathematikunterricht sollte man aber beweisen, denn eine Mathematik ohne Beweise ergibt ein falsches Bild von Mathematik. (Aber auch eine Mathematik ohne Anwendungen täte dies.) Eine Methode der Motivation ist dabei eine kognitive Dissonanz aufzubauen, weil dadurch die Schüler/innen motiviert werden, diese aufzulösen. Dies geschieht etwa im Lehrbuch MatheFit4 wie folgt:

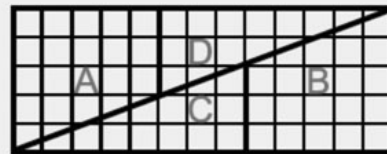
Tom hat ein Problem: „Sara, kannst du mir das erklären? Das Quadrat hat die Seitenlänge 8 und daher die Fläche $8 \times 8 = 64$, das Rechteck aber, das aus den zerschnittenen Quadratfeldern zusammengesetzt ist, eine Fläche von $13 \times 5 = 65$. Das kann doch nicht stimmen!“



Sara schaut sich auch das Ganze an und findet keinen Fehler. Das kann doch nicht sein, dass $64 = 65$ ist! Oder doch?

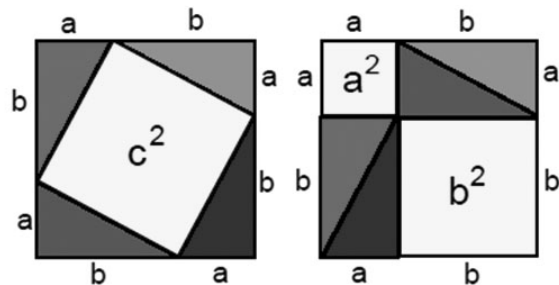


$$8 \times 8 = 13 \times 5 ?$$



„Glaubst du, dass sich die Flächenteile beim Bewegen vielleicht ändern?“, überlegt Sara. „So ein Quatsch“, entgegnet Tom. „Was gilt denn dann überhaupt noch?! Dann aber haben wir den Pythagoras auf S. 84 auch nicht bewiesen und dann gilt der vielleicht auch nicht!“

Der Pythagoreische Lehrsatz wurde – wie nebenstehende Zeichnung zeigt – bewiesen (im Original sind die Flächen farbig).



Ist nun $64 = 65$? Natürlich nicht. Die Lösung des Paradoxons ist recht einfach:

Steigung k_A der unteren schrägen Seite des Trapezes A im karierten Rechteck: $k_A = 2/5$

Steigung der Hypotenuse des Dreiecks C im karierten Rechteck: $k_C = 3/8$

Unterschied: $k_A - k_C = 1/40$

Das ist so wenig, dass es bei der gewählten Strichstärke nicht auffällt, aber doch insgesamt ein Kästchen ausmacht.

Daher gilt Zusammenlegen von Flächen nicht als Beweis!

Und wie ist es bei obigem Beweis des Lehrsatzes von Pythagoras? Hier passen die Steigungen, wie man leicht zeigen kann.

Eine weitere Möglichkeit eine kognitive Dissonanz aufzubauen, gibt die Eulersche Primzahlformel:

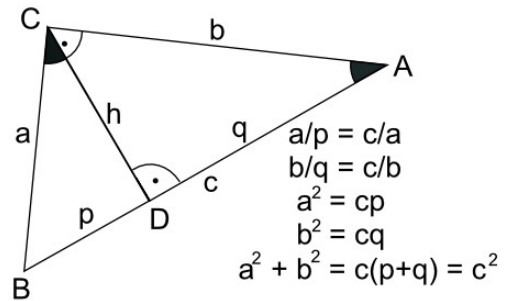
Leonhard EULER (1707 - 1783) gab folgende Formel an, mit der man Primzahlen erzeugen kann, indem man für x eine beliebige natürliche Zahl einsetzt: $x^2 + x + 41$. Untersuche, ob er Recht hatte!

Teste dabei auch 41!

Daraus erkennt man, dass Probieren in der Mathematik keine Beweismethode ist (außer es werden *sämtliche* möglichen Fälle untersucht).

Weitere einfache Aufgaben zum Beweisen wären dann (alle aus MatheFit4):

- Beweise: „In jedem Dreieck ist die Summe der Innenwinkel 180° .“
- Berechne die Winkelsumme im Viereck, im n -Eck!
- Berechne die Anzahl der Diagonalen eines n -Ecks!
- Zeige, dass in einem Rhombus die Diagonalen (1) aufeinander normal stehen und (2) einander halbieren!
- Untersuche, ob nebenstehender Beweis des pythagoreischen Lehrsatzes wirklich ein Beweis ist! Wenn nicht, ergänze, was noch fehlt!



Oder aber auch gemäß den Bildungsstandards (vgl. Praxishandbuch Bildungsstandards 8. Schulstufe, 2011, S. 54):

- Eine der binomischen Formeln lautet: $(a + b)^2 = a^2 + 2ab + b^2$. Begründe die Richtigkeit dieser Formel für positive Werte der Variablen durch einen grafischen Beweis! Schreibe dazu das Quadrieren als Multiplikation an und stelle diese Multiplikation grafisch dar! Verwende diese Darstellung dann als Grundlage für deinen Beweis in Worten und Variablen!

Oder aus Lehrbüchern der Oberstufe (vgl. Mathematik 5 – 7, 2011):

- Bd. 5, Aufg. 403: Begründe anhand geeigneter Skizzen, warum eine a) *streng monoton fallende*, b) *streng monoton steigende* reelle Funktion höchstens eine Nullstelle haben kann!
- Bd. 6, Aufg. 277a: Begründe die folgende Regel: $a > 0 \Rightarrow a^n > 0 \Rightarrow a^{2n} > 0$ und bestätige sie durch Belegung der Variablen a mit gegebenen reellen Zahlen ($n \in \mathbb{N}^*$)!
- Bd. 7, Aufg. 962a: Beweise: Die Standardabweichung ist stets kleiner oder gleich der Spannweite. Wann ist sie gleich?

5. Warum ist die Mathematik so **exakt**? Oder ist sie so genau?

Wie wir vorhin gesehen haben, ist „exakt“ und „genau“ nicht dasselbe (siehe dazu auch HANISCH, S. 203, 1990). Man kann dies leicht mit folgender Aufgabe (aus MatheFit4) den Schüler/innen verständlich machen:

- 1276** (1) Berechne den folgenden Ausdruck mit deinem Taschenrechner!
 (2) Bringe auf denselben Nenner und vereinfache soweit wie möglich!
 (3) Vergleiche die beiden Ergebnisse!

$$99 - 70\sqrt{2} - \frac{1}{99 + 70\sqrt{2}}$$

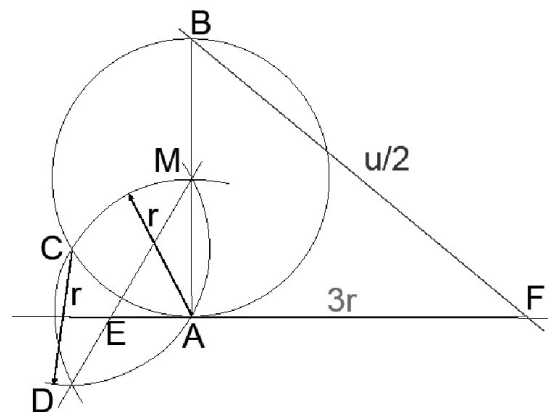
(1) Je nach der Genauigkeit deines Taschenrechners wirst du einen Wert von etwa $-2,0051043420239672797310047765089e-36$ erhalten. (Beachte, dass diese Zahl in Gleitkommadarstellung – siehe Mathe-Fit3, S. 92 – angegeben ist und sowohl der Wert als auch die Anzeige bei deinem Taschenrechner anders sein können!)

$$\begin{aligned} (2) \quad & 99 - 70\sqrt{2} - \frac{1}{99 + 70\sqrt{2}} = \\ & = \frac{(99 - 70\sqrt{2})(99 + 70\sqrt{2})}{99 + 70\sqrt{2}} - \frac{1}{99 + 70\sqrt{2}} = \\ & = \frac{99^2 - (70\sqrt{2})^2}{99 + 70\sqrt{2}} - \frac{1}{99 + 70\sqrt{2}} = \frac{9801 - 9800}{99 + 70\sqrt{2}} - \frac{1}{99 + 70\sqrt{2}} = \\ & = \frac{1 - 1}{99 + 70\sqrt{2}} = 0 \end{aligned}$$

(3) Das Ergebnis mit dem Taschenrechner ist mit einer Genauigkeit von 36 Stellen 0, das der Bruchrechnung exakt 0.

Sehr gut erkennt man diesen Unterschied auch durch die Näherungskonstruktion von Adam Adamandy KOCHAŃSKI (1631? - 1700), bei der es darum geht, den Umfang eines Kreises durch eine Strecke darzustellen. Wie man leicht sieht, ist π dadurch sehr genau, aber nicht exakt bestimmt:

$$\pi \approx \sqrt{\frac{40}{3}} - 2\sqrt{3} = 3,141533\dots$$



6. Zusammenfassung

Dieses Vorgehen macht die Mathematik im Gegensatz zu anderen Wissenschaften unanfechtbar, da alle mathematischen Aussagen durch reine Gedankenoperationen auseinander hervorgebracht oder aufeinander zurückgeführt werden. Daher muss für mathematische Erkenntnisse ein streng logischer Beweis gefunden werden, bevor sie als mathematische Sätze anerkannt werden. In diesem Sinn sind mathematische Sätze prinzipiell endgültige und allgemeingültige Wahrheiten, so dass die Mathematik als *die* exakte Wissenschaft betrachtet werden kann.

Gerade diese Exaktheit ist für viele Menschen das Faszinierende an der Mathematik. (Vgl.

<http://de.wikipedia.org/wiki/Mathematik>)

Aber sicher nicht für alle (siehe Zeichnung).



7. Literatur

1. FROESE Norbert: Euklid und die Elemente ☐ Die Entdeckung der axiomatischen Methode durch Euklid. <http://www.antike-griechische.de/Euklid.pdf>
2. GÖTZ Stefan, Reichel Hans Christian, Müller Robert, Hanisch Günter: Mathematik 5 – 7, Wien 2011.
3. HANISCH Günter: Beweisen im Mathematikunterricht. Der Unterschied zwischen Logik und Psychologik. Aus: Beiträge zum Mathematikunterricht 1985. Vorträge auf der 19. Bundestagung für Didaktik der Mathematik in Giessen, S. 138-141. Bad Salzdetfurt 1985.
4. HANISCH Günter: Problematik der Leistungsfeststellungen durch schriftliche Arbeiten am Beispiel der Mathematik. Habilitationsschrift, Wien 1990.
5. HANISCH Günter, BENISCHEK Isabella, HAUER-TYPPELT Petra, SATTLBERGER Eva.: MatheFit4. Besseres Buch Wien 2011.
6. http://de.wikipedia.org/wiki/Beweis_%28Mathematik%29
7. <http://de.wikipedia.org/wiki/Euklid>
8. <http://de.wikipedia.org/wiki/Mathematik>
9. Neureiter Hans Christian u.a.: Praxishandbuch für „Mathematik“ 8. Schulstufe. BIFIE (Hrsg.), Graz 2011.

8. Anschrift des Verfassers

Günter HANISCH
Fakultät für Mathematik der Universität Wien
Nordbergstraße 15, Zi. 201, 1090 Wien
guenter.hanisch@univie.ac.at

Online-Bildungsportal studieren.at
Studien im Bereich Technische Mathematik – Karriere
www.studieren.at/studien/technische-mathematik/#karriere
Transkription der Angaben vom 28. Jänner 2023

Wo werden Technische Mathematiker gebraucht? Ganz klar: Überall dort, wo Technik und Berechnungen eine Rolle spielen. Und das ist vor allem in Industrie, Logistik, Telekommunikation und Medizin sowie bei Energie- und Softwareunternehmen der Fall, aber natürlich werden Mathematiker auch bei Banken, Versicherungen, Unternehmensberatungen, in der Öffentlichen Verwaltung sowie in der Forschung gebraucht. Die Chance auf eine schnelle Anstellung nach dem Studium und der Aufstieg in eine Leitungsposition steht sehr gut.

Was machen Technische Mathematiker dort genau? Nun, aufgrund ihrer Fähigkeit komplexe Zusammenhänge analysieren und darauf aufbauend Prognosen erstellen zu können, nehmen Technische Mathematiker oft Schlüsselpositionen ein, wenn es um die Entwicklung eines Produkts oder eines ganzen Unternehmens geht.

Hier ein paar Beispiele:

Technische Mathematiker können mit Hilfe von Differentialgleichungen Modelle entwickeln, die Menschen mit neurologischen Schäden helfen können, zum Beispiel bei der Entwicklung eines Hörgeräts. Sie können außerdem errechnen, wie schnell sich eine Infektion ausbreiten wird und damit zur Vorsorge beitragen.

Ihre Fähigkeit komplexe Datenstrukturen zu analysieren, befähigt sie dazu, Algorithmen sozialer Netzwerke zu verstehen sowie Handlungsanweisungen zu geben, um die Online-Performance von Unternehmen zu verbessern. Sie können überdies mit Co-dierungen die Zahlungsabwicklung im Internet sichern und Daten verschlüsseln.

Technische Mathematiker sind in der Lage über Simulation und Berechnungen Erdölvorkommen und anderen Rohstoffe zu lokalisieren.

Skriptum zur Leistungsbeurteilung

Quellen:

- FISCHER, R. / MALLE, G.: Mensch und Mathematik. Eine Einführung in didaktisches Denken und Handeln. 2004 München Wien.
- SACHER, W.: Prüfen – Beurteilen – Benoten. Bad Heilbronn 1996.
- Verordnung des Bundesministers für Unterricht und Kunst vom 24. Juni 1974, BGBl. Nr. 371 über die Leistungsbeurteilung in Pflichtschulen sowie mittleren und höheren Schulen (Leistungsbeurteilungsverordnung)

LEISTUNGSFESTSTELLUNG und LEISTUNGSBEURTEILUNG – SCHULNOTEN

(1) Rechtliche Grundlagen:

Auf Schulleistungen bezogen unterscheidet der Gesetzgeber ausdrücklich zwischen Leistungsfeststellung einerseits und Leistungsbeurteilung andererseits (vgl. dazu die **Leistungsbeurteilungsverordnung (VOLB)**):

- Die **Leistungsfeststellung** ist die *Erhebungen eines Sachverhaltes*, die einer Leistungsbeurteilung vorangehen muss. Dabei geht es also um die Feststellung („Messung“), wie weit eine bestimmte Leistung vom vorgegebenen Lernziel abweicht. Ziel der Leistungsfeststellung ist die möglichst „objektive“ Erhebung des aktuellen Leistungsstandes, orientiert an vorgegebenen Zielen.
- Die **Leistungsbeurteilung** ist ein *Sachverständigengutachten* zur Beurteilung des im Zuge einer Leistungsfeststellung erhobenen Sachverhalts, wobei einer bestimmten Leistung eine von mehreren Beurteilungsstufen („Noten“) zugeordnet wird. In diese Beurteilung des zuvor erhobenen Sachverhalts durch den Lehrer fließen üblicherweise auch pädagogische und andere Überlegungen ein (vgl. (2) – „Funktionen schulischer Leistungsbeurteilung“).

Es werden verschiedene **Formen der Leistungsfeststellung** angeführt:

- Feststellung der *Mitarbeit*
- *Mündliche* Leistungsfeststellungen
- *Schriftliche* Leistungsfeststellungen
- *Praktische* Leistungsfeststellungen
- *Graphische* Leistungsfeststellungen.

Diese Formen der **Leistungsfeststellung** sind durch die Leistungsbeurteilungsverordnung genau geregelt, wobei folgende **allgemeine Grundsätze** gelten:

- Grundlage für die Leistungsfeststellung sind nur die im Lehrplan festgelegten Bildungs- und Lehraufgaben und jene Lehrstoffe, die bis zum Zeitpunkt der Leistungsfeststellung in der betreffenden Klasse behandelt worden sind.
- Leistungsfeststellungen sind möglichst gleichmäßig über den Beurteilungszeitraum zu verteilen.

- Die jeweils gewählte Form der Leistungsfeststellung ist dem Alter und Bildungsstand der Schüler, den Erfordernissen des Unterrichtsgegenstandes, den Anforderungen des Lehrplanes und dem jeweiligen Stand des Unterrichtes anzupassen.
- Leistungsfeststellungen einzelner Schüler haben während des Unterrichts zu erfolgen (Ausnahmen: Wiederholungs- und Nachtragsprüfungen sowie von einzelnen Schülern nachzuholende Schularbeiten) und sind so in den Unterricht einzubauen, dass auch die übrigen Schüler daraus Nutzen ziehen können.
- Alle Leistungsfeststellungen sind als gleichwertig zu betrachten, jedoch sind Anzahl, Stoffumfang und Schwierigkeitsgrad zu berücksichtigen. Allein auf Grund schriftlicher Leistungsfeststellungen darf keine Semester- oder Jahresbeurteilung erfolgen. Zuletzt erbrachten Leistungen ist ein höheres Gewicht beizumessen.
- An den letzten drei Unterrichtstagen vor einer Beurteilungskonferenz ist die Durchführung einer Leistungsfeststellung nur mit Zustimmung des Schulleiters zulässig. Dieser darf die Zustimmung nur bei Vorliegen wichtiger Gründe erteilen.
- Leistungsfeststellungen sind verboten, wenn der Schüler wegen einer körperlichen Behinderung eine entsprechende Leistung nicht erbringen kann oder wenn er durch die Leistungsfeststellung gesundheitlich gefährdet ist.

Nach der Leistungsfeststellung erfolgt die **Leistungsbeurteilung** durch den Lehrer, der dabei als Sachverständiger auftritt. Er hat dabei unter anderem folgende **allgemeine Grundsätze** zu beachten:

- Grundlage für die Beurteilung der Leistungen sind die oben angeführten Formen der Leistungsfeststellung.
- Die Leistungen der Schüler sind sachlich und gerecht zu beurteilen; es ist größtmögliche Objektivierung anzustreben.
- Bei schriftlichen Leistungsfeststellungen ist dem Schüler die Beurteilung spätestens bei der Rückgabe der Arbeit, bei mündlichen Leistungsfeststellungen spätestens am Ende der betreffenden Unterrichtsstunde und bei praktischen Leistungsfeststellungen spätestens am nächsten Unterrichtstag, an dem der betreffende Unterrichtsgegenstand wieder unterrichtet wird, bekannt zu geben. Die für die Beurteilung maßgeblichen Vorzüge und Mängel seiner Leistung sind dem Schüler mit der Beurteilung bekannt zu geben, ohne ihn dabei zu entmutigen oder seine Selbstachtung zu beeinträchtigen.
- Eine Information über den Leistungsstand des Schülers hat auf Wunsch des Schülers oder seiner Erziehungsberechtigten zu erfolgen.
- Vorgetäuschte Leistungen sind nicht zu beurteilen. Schularbeiten, die wegen Vortäuschung einer Leistung nicht beurteilt werden, sind wie versäumte Schularbeiten zu behandeln. Unerlaubte Hilfsmittel, deren sich der Schüler bedienen könnte, sind ihm abzunehmen und nach durchgeführter Leistungsfeststellung zurückzugeben.
- Das Verhalten des Schülers in Schule und Öffentlichkeit darf nicht in die Leistungsbeurteilung einbezogen werden.

- Die äußere Form der Arbeit darf nur in den speziell geregelten Fällen (§ 12 VOLB) bei der Leistungsbeurteilung berücksichtigt werden.
- Sachlich vertretbare Meinungsäußerungen des Schülers haben die Beurteilung auch dann nicht zu beeinflussen, wenn sie von der Meinung des Lehrers abweichen.

Für die Beurteilung von Schulleistungen bestehen folgende **Beurteilungsstufen** („**Noten**“):

- Sehr gut (1)
- Gut (2)
- Befriedigend (3)
- Genügend (4)
- Nicht genügend (5).

Die **Anforderungen** in den einzelnen Beurteilungsstufen werden nach drei Kategorien differenziert:

	(a) <u>Erfassung und Anwendung des Lehrstoffes</u> (b) <u>Durchführung der Aufgaben</u>	(c) <u>Eigenständigkeit</u>	(d) <u>Selbstständige Anwendung des Wissens und Könnens</u>
Sehr gut	Anforderungen in weit über das Wesentliche hinausgehendem Ausmaß erfüllt	Muss deutlich vorliegen (wo dies möglich ist)	Muss vorliegen (wo dies möglich ist)
Gut	Anforderungen in über das Wesentliche hinausgehendem Ausmaß erfüllt	Merkliche Ansätze (wo dies möglich ist)	Bei entsprechender Anleitung (wo dies möglich ist)
Befriedigend	Anforderungen in wesentliche Bereichen zur Gänze erfüllt	Mängel bei (b) werden durch merkliche Ansätze zur Eigenständigkeit ausgeglichen	
Genügend	Anforderungen in wesentlichen Bereichen überwiegend erfüllt		
Nicht genügend	Anforderungen nicht einmal in den wesentlichen Bereichen überwiegend erfüllt		

Besondere Bestimmungen gelten für die **Leistungsbeurteilung bei schriftlichen Arbeiten**:

- Bei schriftlichen Leistungsfeststellungen (Schularbeiten, Tests) müssen die Noten ausgeschrieben werden. Zwischennoten oder Zusätze zu Noten sind nicht zulässig. Erlaubt und zu empfehlen sind Kommentare des Lehrers.
- Identische Fehler, die in einer Arbeit mehrfach auftreten, sind nur einmal zu werten (ausgenommen in Mathematik und Darstellende Geometrie). Folgefehler sind überhaupt nicht zu werten.

- Bearbeitet ein Schüler bei einer schriftlichen Arbeit statt der gestellten Aufgabe eine andere, so ist zu prüfen, ob nicht doch noch von einer Leistung im betreffenden Arbeitsgebiet gesprochen werden kann (Beispiel: Themenverfehlung).

(2) Funktionen schulischer Leistungsbeurteilung (vgl. SACHER, S. 12 - 21):

Unsere Gesellschaft versteht sich als „Leistungsgesellschaft“, in der angeblich individuelle Leistungsfähigkeit und Leistungsbereitschaft (und eben nicht Privilegien der Geburt oder des Standes) über die jeweilige Position in der Gesellschaft entscheiden und für die Zuteilung von Lebenschancen ausschlaggebend sind. Dieses Selbstbild der Gesellschaft kann man zwar mit guten Gründen kritisch hinterfragen, es drückt sich aber jedenfalls auch in der Praxis der Leistungsbeurteilung einer „Leistungsschule“ aus. Im Bewusstsein der Öffentlichkeit wird Schule oft auf ihre Prüfungs- und Beurteilungsfunktion reduziert und da wieder auf negativ besetzte Aspekte wie Selektion (also Auslese) und Stigmatisierung (also Kennzeichnung als „Schulversager“). Benotung gilt somit immer noch primär als Auslese der in irgendeinem Sinn „Geeigneten“, wobei die diagnostischen und motivierenden Aspekte schulischer Leistungsbeurteilung oft übersehen werden.

Schulische Leistungsbeurteilung hat bestimmte „klassische“ Funktionen bzw. Aufgaben, die üblicherweise als wesentliche Kennzeichen von Schule angesehen und damit kaum kritisch reflektiert werden. Einige dieser Aufgaben schulischer Benotung seien hier angeführt:

- **Selektionsfunktion:** Leistungsbeurteilung dient der Auslese befähigter Anwärter auf höhere Bildungsabschlüsse und angesehene berufliche und gesellschaftliche Positionen.
- **Prognosefunktion:** Zeugnisnoten werden oft dazu verwendet, künftige Leistungen vorher zu sagen. Besonders an den Schnittstellen verschiedener Schulformen, also beim Übertritt in neue Bildungsgänge, werden wichtige Entscheidungen auf der Grundlage von Schulnoten getroffen.
- **Informations- und Rückmeldungsfunktion:** Schulnoten geben SchülerInnen eine Rückmeldung über den von ihnen erreichten Leistungsstand und informieren auch Erziehungsberechtigte sowie LehrerInnen weiterführender Schulen oder künftige ArbeitgeberInnen über Ergebnisse schulischer Lernprozesse.
- **Disziplinierungsfunktion:** Schulische Leistungsbeurteilung wird auch eingesetzt, um auf „problematisches“ Verhalten von SchülerInnen Einfluss zu nehmen. Darüber hinaus wird von Schulnoten auch manchmal auf Einstellungen, Grundhaltungen oder Charaktereigenschaften von SchülerInnen geschlossen.
- **Lehr- und Lerndiagnosefunktion:** Leistungserhebungen werden von LehrerInnen auch zur Einschätzung des Lernstandes verwendet und bilden damit die Grundlage für die Planung weiterer Lehr- und Lernprozesse. Sie sind aber auch eine Rückmeldung an die LehrerInnen selbst, das heißt, sie dienen der Diagnose des Lehrerfolgs.
- **Motivationsfunktion:** SchülerInnen lernen nicht selten, um gute Noten zu erhalten oder schlechte Beurteilungen zu vermeiden.

- **Sozialisationsfunktion:** Schulische Leistungsbeurteilung soll junge Menschen auf das Leben in einer Gesellschaft vorbereiten, in der Zeugnisse und Zertifikate eine wichtige Rolle spielen.
- **Lernerziehungsfunktion:** Leistungsbeurteilung als Teil einer Erziehung zur eigenverantwortlichen Gestaltung von Lernprozessen – SchülerInnen sollen befähigt werden, ihr eigenes (auch nachschulisches) Lernen selbstständig zu planen und zu steuern.

Generell lassen sich dabei zwei Kategorien unterscheiden (vgl. FISCHER / MALLE, S. 304):

- Funktionen, die **nach außen** wirken, also eine Information für nicht direkt am schulischen Geschehen beteiligte Personen (etwa zukünftige Arbeitgeber) darstellen (wie z.B. Prognose, Selektion)
- Funktionen, die **nach innen** wirken, also zur Steuerung des Lernprozesses dienen (wie z.B. Diagnose, Motivation).

Der Prozess der schulischen Leistungsbeurteilung besteht also darin, dass die in der Leistungsfeststellung erhobene Abweichung einer bestimmten Leistung vom vorgegebenen Lernziel durch den Lehrer bewertet („beurteilt“) wird: Wie gut / schlecht ist diese Leistung? Genügt sie oder genügt sie nicht? Man kann dabei verschiedene **Grundorientierungen** zugrunde legen (vgl. FISCHER / MALLE, S. 306 - 307):

- **Zielorientierung:** Wie bei der Leistungsfeststellung geht es dabei nur um die Erreichung oder Verfehlung bestimmter vorgegebener Lernziele.
- **Durchschnittsorientierung** (bzw. **Normorientierung**): Bezugsrahmen für die Beurteilung von Leistungen ist der Klassendurchschnitt, wobei oft eine bestimmte Verteilung der Noten als Ideal angestrebt wird – mittlere Noten sind am häufigsten, extrem gute oder extrem schlechte Noten sind selten („Normalverteilung“ der Noten).
- **Orientierung am Leistungszuwachs:** Entscheidend ist dabei die Veränderung des Leistungsstandes – große Fortschritte drücken sich in einer guten Note aus. Im Extremfall könnten von zwei Schülern, die dieselbe Schularbeit vorlegen, der eine die Note „Sehr gut“, der andere aber „Nicht genügend“ bekommen, wenn der eine vorher sehr schlecht und der andere sehr gut war.

(3) Die Messqualität von Schulnoten (vgl. SACHER, S. 27 – 40 und FISCHER / MALLE, S. 308 - 317):

Mit schulischer Leistungsbeurteilung ist (auch) der Anspruch von **Leistungsmessung** verbunden (→ Leistungsfeststellung). In diesem Zusammenhang ist zu fragen, wie es um Objektivität, Validität und Reliabilität von Schulnoten steht.

Zunächst ist festzuhalten, dass Schulnoten auf **Ordinalskalaniveau** „messen“ – sie legen also nur eine Rangordnung der erfassten Leistungen fest. Da eine Ordinalskala keinen Abstand zwischen den einzelnen Rangplätzen definiert, darf in eine Analyse der Messwerte (Noten) nur die Ordnungsbeziehung (größer – kleiner) der Noten untereinander eingehen.

Entgegen einer weit verbreiteten Praxis dürfen genau genommen also mit Schulnoten keine Rechenoperationen (wie etwa Bildung des Arithmetischen Mittels) ausgeführt werden – der „richtige“ Mittelwert für Schulnoten ist der Median.

(a) Die Objektivität der Leistungsbeurteilung:

Objektivität meint hier die Unabhängigkeit des Beurteilungsergebnisses von der Person des Beurteilers – verschiedene Beurteiler einer bestimmten Leistung sollten zu übereinstimmenden Ergebnissen kommen. In zahlreichen Untersuchungen konnte nachgewiesen werden, dass die Objektivität schulischer Leistungsbeurteilung nicht sehr hoch ist – Urteile verschiedener Prüfer korrelieren zwischen 0,35 und 0,85. Es kam sogar vor, dass die Beurteilungen für ein und dieselbe Mathematikschularbeit über alle fünf Beurteilungsstufen streuten.

Verantwortlich für diese großen Beurteilungsunterschiede sind vor allem Unterschiede hinsichtlich der Fehlerregistrierung, Divergenzen bezüglich der festgesetzten Höchstpunktezahlen, Unterschiede bei den angewandten Notenschlüsseln und Abweichungen bei der Punktezuordnung zu bestimmten Leistungsteilen oder Leistungsaspekten.

(b) Die Reliabilität (Zuverlässigkeit) der Leistungsbeurteilung:

Reliabilität (der Leistungsbeurteilung) bezeichnet hier den Grad der Genauigkeit, mit der Schulnoten „messen“, unabhängig von der Frage, ob sie wirklich das erfassen, was gemessen werden soll (nämlich schulische Leistungen). Ein Instrument misst zuverlässig, wenn eine Wiederholung des Messvorgangs unter gleichen Bedingungen übereinstimmende Ergebnisse bringt. Die Reliabilität schulischer Leistungsüberprüfung ist niedrig – die Korrelationskoeffizienten liegen für standardisierte Tests (die im Rahmen schulischer Leistungsbeurteilung praktisch nicht vorkommen) zwischen 0,80 und 0,95, für schriftliche Prüfungen zwischen 0,50 und 0,80 und für mündliche Prüfungen unter 0,50.

Speziell um die sogenannte Retestreliabilität (= Wiederholung der Messung nach einiger Zeit mit demselben Test) steht es schlecht. Für die wiederholte Beurteilung von Aufsätzen durch denselben Beurteiler wurden Reliabilitätskoeffizienten zwischen 0,25 und 0,51 gefunden, während die Korrelationen für Mathematikarbeiten bei 0,46 lagen.

(c) Die Validität (Gültigkeit) der Leistungsbeurteilung:

Validität eines Messverfahrens ist gegeben, wenn tatsächlich das gemessen wird, was gemessen werden soll. Im schulischen Bereich hat dieser Begriff mindestens vier Aspekte:

- **Inhaltsvalidität** bezieht sich auf die Frage, ob Prüfungen tatsächlich (nur) Kompetenzen erfassen, die zuvor im Unterricht vermittelt wurden. Ein spezieller Aspekt ist dabei die *curriculare Validität*, d.h. die Übereinstimmung der Leistungsüberprüfung mit den Lehrplananforderungen.

- **Prognosevalidität** meint die Möglichkeit, auf Grundlage bereits vorliegender Messergebnisse die Resultate zukünftiger Messungen vorherzusagen. Dieser Aspekt spielt im schulischen Leben eine nicht unbedeutende Rolle (etwa im Zusammenhang mit Aufnahme- oder Eignungsprüfungen).
- **Übereinstimmungsvalidität** ist die Übereinstimmung der mit verschiedenen Messinstrumenten gewonnenen Ergebnisse. Sie ist dann hoch, wenn die Beurteilungen schriftlicher, mündlicher und eventuell auch praktischer Leistungen eines Schülers in einem bestimmten Gegenstand sehr ähnlich sind.
- **Konstruktvalidität** ist die Übereinstimmung der gemessenen Kompetenzen mit einem theoretischen Modell.

Zahlreiche sach- und fachfremde Einflüsse wie etwa regionale, schulartspezifische oder schichtspezifische Effekte beeinträchtigen die Validität von Schulnoten. Besonders problematisch ist in diesem Zusammenhang die mangelhafte Prognosevalidität des in Notenform ausgedrückten Lehrerurteils – die entsprechenden Korrelationen liegen im Allgemeinen unter 0,50. Für Zeugnisnoten wurde eine Prognosevalidität von nur 0,25 bis 0,30 für den Erfolg in weiterführenden Schulen gefunden. Angesichts derartiger Einsichten ist die weit verbreitete Praxis zu hinterfragen, wichtige Schullaufbahnentscheidungen primär von Schulnoten abhängig zu machen.

Aufgaben

(A) Reproduktion

- (1) Erkläre den Unterschied zwischen Leistungsfeststellung und Leistungsbeurteilung.
- (2) Nenne die Formen der Leistungsbeurteilung.
- (3) Nenne einige allgemeine Grundsätze der Leistungsfeststellung.
- (4) Nenne einige allgemeine Grundsätze der Leistungsbeurteilung.
- (5) Welche grundlegende Kategorien werden bei der Definition der Beurteilungsstufen verwendet?
- (6) Nenne und erläutere wichtige Funktionen schulischer Leistungsbeurteilung.
- (7) Nenne und erläutere verschiedene Grundorientierungen für schulische Leistungsbeurteilung.
- (8) Auf welchem Skalenniveau „messen“ Schulnoten? Welche Konsequenzen ergeben sich daraus?
- (9) Nenne und erläutere die drei Gütekriterien für Messung allgemein. Wie weit erfüllen Schulnoten diese Anforderungen?

(B) Anwendung

- (1) Warum unterscheidet der Gesetzgeber im Zusammenhang mit schulischer Leistungsbeurteilung zwischen Leistungsfeststellung und Leistungsbeurteilung?

- (2) Nimm Stellung zu der Behauptung „Schulnoten sind auch ein Spiegel der Leistungsgesellschaft.“
- (3) Sind die Beurteilungsstufen („Noten“) genau (genug) definiert? Kann bzw. soll man sie genauer fassen?
- (4) Formuliere und diskutiere Argumente für bzw. gegen jede der drei Grundorientierungen schulischer Leistungsbeurteilung (Zielorientierung, Normorientierung, Orientierung am Leistungszuwachs). Bewerte die drei Alternativen.
- (5) Bewerte die einzelnen Funktionen schulischer Leistungsbeurteilung (a) in ihrer praktischen Bedeutung – wie wichtig sind sie jeweils tatsächlich in der schulischen Beurteilungspraxis und (b) in ihrer grundsätzlichen Wichtigkeit? Ordne diese Funktionen nach ihrer „Wichtigkeit“. Welche Funktionen könnte man noch nennen?
- (6) Die Messqualität von Schulnoten ist nicht hoch. Welche Konsequenzen ergeben sich daraus?

9 Anhang – Lehrunterlagen

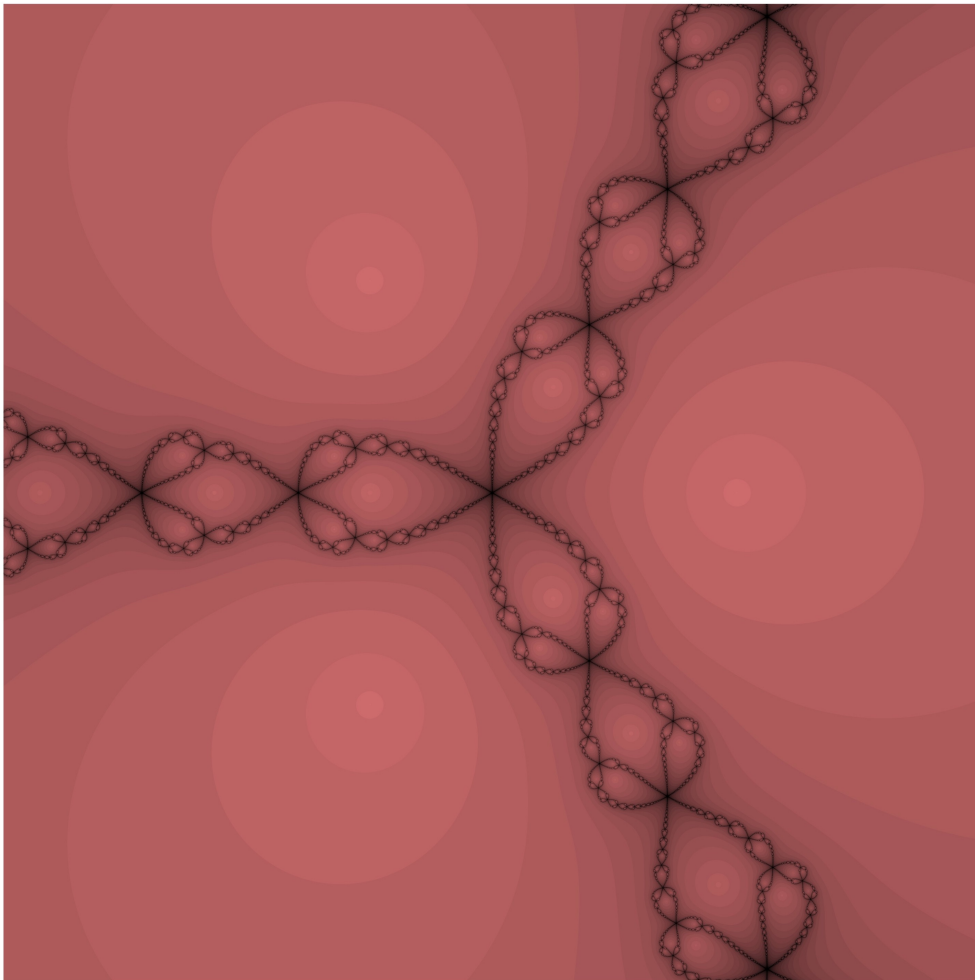
Überblick

- Skriptum zur Vorlesung *Numerische Mathematik*
(Adaptionen im Sommersemester 2023)
- Zusätzliche Unterlagen zu Vorlesung und begleitendem Proseminar
(aktuelles Wintersemester, Ergänzungen im Sommersemester 2023)

Kompendium zur Lehrveranstaltung

Numerische Mathematik

Mechthild Thalhammer



Leopold–Franzens-Universität Innsbruck

Studienjahr 2022/23

*Die Mathematik ist mehr ein Tun als eine Lehre.
Mathematik zu lernen heißt, sie immer wieder neu zu erfinden.
Frustration und Euphorie liegen in der Mathematik oft knapp nebeneinander.
Mathematik ist schön, aber für viele sperrig. Sie ist einfach nicht sozial:
In ihr kann nur der Einzelne zur Einsicht gelangen.
Wenn das die Lösung ist, will ich mein Problem zurück.
Es geht einfach oder es geht einfach nicht.*

(L. Brouwer, D. O'Shea, R. Höfer, R. Taschner, Unbekannt)

Illustration. Auf der Titelseite ist das Konvergenzverhalten des Newtonverfahrens zur näherungsweise Lösung der nichtlinearen komplexen Gleichung $z^3 = 1$ illustriert.

Das Gebiet der Numerik ist ein Teilgebiet der Mathematik, das sich insbesondere der Konstruktion und Analyse von Algorithmen widmet.

Das vorliegende Kompendium faßt die im Rahmen der Lehrveranstaltung **Numerische Mathematik** (VO 3 & PS 2) im Studienjahr 2022/23 an der Universität Innsbruck behandelten Themen zusammen. Ohne Anspruch auf Allgemeinheit und Vollständigkeit werden Methoden zur näherungsweise Lösung von grundlegenden Problemen der Analysis und Linearen Algebra angegeben. Als Illustrationen werden einfache Modellprobleme und Implementierungen betrachtet.

Das Kompendium basiert vorwiegend auf den Vorlesungsskripten

MATHIAS RICHTER

Numerische Mathematik I & II (2010/2011)

und Unterlagen, welche im darauffolgenden Studienjahr 2011/12 an der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München erstellt wurden. Die damals gewählte Strukturierung und entsprechende Verweise wurden beibehalten.

Als ergänzende Literaturquellen werden die Vorlesungsskripten

HERMANN SCHICHL

Numerik 1 & 2 (2000/2001/2011)

www.mat.univie.ac.at/~herman/skripten

ERNST HAIRER, GERHARD WANNER

Introduction à l'Analyse Numérique (2005)

www.unige.ch/~hairer/polycop

sowie dort erwähnte Standardwerke empfohlen. Was eine umfassende Einführung in die Software MATLAB betrifft, wird zudem auf

PETER ARBENZ

Einführung in MATLAB (2007/2008)

people.inf.ethz.ch/arbENZ/MatlabKurs/matlabintro.pdf

verwiesen.

Im Internet zu findende Unterlagen bieten die Vorteile kompakter Darstellungen und freier Verfügbarkeit, sollten jedoch mit einem kritischen Blick auf inhaltliche Richtigkeit und mögliche Druckfehler verwendet werden.

Themenüberblick I

1. Einführung

2. Grundbegriffe der Numerik

2.1. Maschinenzahlen und Rundung

2.2. Gleitpunktoperationen

2.3. Rundungsfehleranalyse

2.4. Kondition

2.5. Stabilität

3. Vektoren und Matrizen

3.1. Rechnen mit Vektoren und Matrizen

3.2. Elementare Matrizenmultiplikationen

3.3. Skalarprodukt und Orthogonalität

3.4. Orthogonalisierungsverfahren nach Gram–Schmidt

3.5. Normen für Vektoren und Matrizen

4. Direkte Verfahren für lineare Gleichungssysteme

4.1. Kondition linearer Gleichungssysteme

4.2. Lösung über die QR-Zerlegung

4.3. Stabilität der Lösungsmethode über die QR-Zerlegung

4.4. Gauß-Elimination und Dreieckszerlegung

4.5. Rundungsfehler-Analyse der Gauß-Elimination

4.6. Pivotwahl bei der Gauß-Elimination

5. Lineare Ausgleichsrechnung

5.1. Ein Beispiel

5.2. Normalengleichungen

5.3. Cholesky-Zerlegung

5.4. Lösung über Orthogonaltransformationen

6. Eigenwerte und SVD (Überblick)

6.1. Theoretischer Hintergrund

6.2. Singulärwertzerlegung

6.3. Algorithmen zum Eigenwertproblem (Überblick)

6.4. Vektoriteration und inverse Vektoriteration

6.5. Die Grundidee des QR-Algorithmus

7. Nichtlineare Gleichungssysteme

7.1. Grundbegriffe

7.2. Verfahren für den eindimensionalen Fall

7.3. Der mehrdimensionale Fall

1. Einführung

- **Numerische Mathematik:** Konkrete Lösung mathematischer Probleme, d.h. konstruktive Beschaffung von Lösungen mittels Zahlenrechnungen.

Theoretische Resultate und Formelmanipulationen sind von Nutzen.

- **Numerische Verfahren der Linearen Algebra:**

– Verfahren zur Lösung linearer Gleichungssysteme.

Inbesondere diese Grundaufgabe der Numerik ist nach wie vor Gegenstand aktueller Forschung.

– Verfahren zur Berechnung von Eigenwerten und Eigenvektoren einer Matrix.

Anwendungen:

– Verfahren zur Lösung nichtlinearer Gleichungssysteme

– Verfahren zur Lösung von Optimierungsproblemen

– Verfahren zur Lösung gewöhnlicher Differentialgleichungen

– Verfahren zur Lösung partieller Differentialgleichungen

- **Numerisches Problem** (Definition 2.7): Funktion, die zulässigen Eingabedaten ein Ergebnis zuordnet

$$p : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m : x \mapsto y = p(x).$$

Numerisches Verfahren bzw. **Algorithmus** zur Lösung eines Problems $p : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$: Endliche Folge von Teilproblemen (d.h. von elementaren Operationen), deren Reihenfolge beim Ablauf eindeutig festliegt

$$(p^{(1)}, \dots, p^{(k)}).$$

Die schrittweise Anwendung der Teilprobleme $p^{(i)}$ für $i = 1, 2, \dots, k$ führt auf Zwischenergebnisse bzw. das Endergebnis

$$y^{(i)} = (p^{(i)} \circ \dots \circ p^{(1)})(x), \quad 1 \leq i \leq k.$$

- **Direktes Verfahren:** Berechnung der exakten Lösung eines Problems in endlich vielen Rechenschritten (im Prinzip möglich), d.h. es ist

$$p^{(k)} \circ \dots \circ p^{(1)} = p.$$

Beispiele:

- * Formel von Vieta zur Lösung einer quadratischen Gleichung
- * Gaußsches Eliminationsverfahren zur Lösung eines linearen Gleichungssystems

- **Näherungsweise Verfahren** bzw. **Approximationsverfahren**: Berechnung einer Näherungslösung eines Problems, d.h. es ist

$$p^{(k)} \circ \dots \circ p^{(1)} \approx p.$$

Beispiele:

- * Approximation unendlicher Reihen durch endliche Reihen
- * Approximation bestimmter Integrale durch Riemann-Summen
- * Approximation von Ableitungen durch Differenzenquotienten
- * Approximation mittels Iterationsverfahren

Verfahrensfehler bzw. **Approximationsfehler**: Fehler der Näherungslösung, d.h. Differenz zwischen näherungsweise und exakter Lösung

$$y^{(k)} - y = (p^{(k)} \circ \dots \circ p^{(1)})(x) - p(x).$$

- **Beispiele** (Probleme, Algorithmen):

- **Elementare arithmetische Operationen** $*$ $\in \{+, -, \times, /\}$

$$p : D \subset \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} : (x, y) \mapsto x * y.$$

- **Auswerten einer Polynomfunktion**

$$p : \mathbb{R}^{n+1} \times \mathbb{R} \rightarrow \mathbb{R} : (c, x) \mapsto \sum_{i=0}^n c_i x^i.$$

Algorithmus basierend auf dem Horner-Schema

$$\sum_{i=0}^n c_i x^i = \left(\dots \left((c_n x + c_{n-1}) x + c_{n-2} \right) x + \dots \right) x + c_0.$$

- **Berechnung der Nullstellen eines quadratischen Polynoms**, z.B. Berechnung der Lösungen $x_{1,2}$ der quadratischen Gleichung $x^2 + 2ax - b = 0$ (unter der Annahme $a, b > 0$ folgt $x_{1,2} \in \mathbb{R}$ mit $x_1 \neq x_2$)

$$p : \mathbb{R}^2 \rightarrow \mathbb{R}^2 : (a, b) \mapsto x.$$

Algorithmus basierend auf der Formel von Vieta, Algorithmus basierend auf einer Umformulierung

$$x_1 = -a + \sqrt{a^2 + b} \quad = \quad \frac{b}{a + \sqrt{a^2 + b}}, \quad x_2 = -a - \sqrt{a^2 + b}.$$

Erweitern mit $a + \sqrt{a^2 + b}$

- **Berechnung der Nullstellen eines Polynoms** bzw. Lösung einer polynomialen Gleichung (Annahme $n \in \mathbb{N}$ mit $n \geq 1$ und $c_n \neq 0$)

$$p: \mathbb{C}^{n+1} \rightarrow \mathbb{C}^n : c \mapsto z \quad \text{mit } f(z_i) = \sum_{\ell=0}^n c_\ell z_i^\ell = 0.$$

Bemerkungen: Identifikation von \mathbb{C} mit \mathbb{R}^2 . Im allgemeinen ist keine explizite Lösungsformel bekannt.

- **Berechnung der Eigenwertzerlegung einer Matrix**, d.h. Berechnung der Eigenwerte $\lambda_1, \dots, \lambda_n$ einer (diagonalisierbaren) Matrix $A \in \mathbb{C}^{n \times n}$ und zugehöriger Eigenvektoren $v = (v_1 | \dots | v_n)$

$$p: \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^n \times \mathbb{C}^{n \times n} : A \mapsto (\lambda, v).$$

Algorithmus basierend auf der Berechnung der Nullstellen des charakteristischen Polynoms und der Lösung linearer Gleichungssysteme

$$\chi(\lambda_i) = \det(\lambda_i I - A) = 0, \quad (\lambda_i I - A)v_i = 0, \quad v_i \neq 0, \quad 1 \leq i \leq n.$$

Bemerkung: Insbesondere für $n \gg 1$ ist die Entwicklung eines alternativen Algorithmus notwendig, vgl. **Illustration 1**.

- **Lösung eines linearen Gleichungssystems** $Ax = b$ (unter Annahme $A \in \mathbb{R}^{n \times n}$ invertierbar)

$$p: \mathbb{R}^{n \times n} \times \mathbb{R}^n \rightarrow \mathbb{R}^n : (A, b) \mapsto A^{-1}b.$$

Algorithmus basierend auf dem Eliminationsverfahren nach Gauß.

- Aufgrund der **Verwendung digitaler Rechner** zur Berechnung der Lösungen von (komplexen) Problemen sind gewisse Einschränkungen unvermeidbar.
 - Einschränkung auf elementare Operationen wie arithmetische Operationen, Wurzelziehen und arithmetische Vergleiche

$$* \in \{+, -, \times, /, \sqrt{}, <, \leq, =, \geq, >, \neq\}.$$

- Einschränkung auf endlich viele darstellbare Zahlen (Maschinenzahlen, im Allgemeinen normalisierte Gleitpunktzahlen zur Basis 2 mit fester Stellenzahl und beschränktem Exponenten).
- Auftreten von Rundungsfehlern bei der Eingabe von Daten und der Berechnung von Zwischen- bzw. Endergebnissen.

Auftreten von Datenfehlern, wenn die Eingabedaten beispielsweise das Ergebnis von Messungen mit eingeschränkter Genauigkeit sind.

- **Aufgaben der Numerik:**

- Konstruktion *guter* Algorithmen beispielsweise hinsichtlich Genauigkeit (z.B. Rate der verbesserten Genauigkeit der Näherungslösung bei höherem technischem Aufwand), Effizienz (z.B. Anzahl der Rechenoperationen bzw. Rechenzeit oder benötigter Speicherplatz) und Stabilität (z.B. Auswirkung von kleinen Änderungen der Eingabedaten auf das Endergebnis).
- Analyse von Verfahrensfehlern und Herleitung von Abschätzungen für Verfahrensfehler.
- Analyse der Fortpflanzungen von Fehlern (Rundungsfehler, Datenfehler) in Algorithmen und Auswirkungen auf Endergebnisse.

- **Illustration** (Eigenwertberechnung):

- Problem: Berechnung aller Eigenwerte einer reellen und symmetrischen Matrix.
- Theoretisches Resultat sichert, daß alle Eigenwerte reell sind.
- Algorithmus basierend auf der Berechnung der Nullstellen des charakteristischen Polynoms.
- Kleine Änderungen der Koeffizienten des charakteristischen Polynoms (vergleichbar mit der Eingabe der Koeffizienten in einfacher Genauigkeit und dabei auftretenden Rundungsfehlern) und Berechnung der zugehörigen Nullstellen.
- Vergleich der Ergebnisse (Nullstellen, Graphen der Polynome).

- Vgl. **Illustration1**.

Vgl. **Illustration1_Modifikation**: Falls die Dimension n der Matrix hinreichend groß ist, ist das Ergebnis auch für kleine relative Änderungen ε der Koeffizienten nicht zufriedenstellend.

Ergänzungen: Binärdarstellung, Horner-Schema, Verfahren von Bairstow (Faktorisierung von Polynomen)

- **Schlußfolgerungen:**

- * Problemstellung *Berechnung der Nullstellen eines Polynoms* (höherer Ordnung) bei praktischen Anwendungen nicht sinnvoll. Aufgrund unvermeidbarer (kleiner) Änderungen der Koeffizienten sind (exakt) berechnete Nullstellen wertlos.
- * Notwendigkeit der Entwicklung eines alternativen Algorithmus zur Eigenwertberechnung.

- **Vorsicht!** Algorithmen, die für die Theorie sinnvoll sind, können jedoch für die Numerik wertlos sein.

2. Grundbegriffe der Numerik

- **Fragestellungen:**

- Welche Zahlen sind am Rechner darstellbar und wie werden elementare Rechenoperationen ausgeführt?
Maschinenzahlen, Rundung, Gleitpunktoperationen, Rundungsfehleranalyse
- Welche numerischen Probleme kann man zufriedenstellend lösen?
Kondition eines Problems
- Wie kann man die Güte eines numerischen Verfahrens hinsichtlich der erreichbaren Genauigkeit beurteilen?
Stabilität eines Algorithmus

Konkrete Anwendung der eingeführten Konzepte in nachfolgenden Kapiteln.

- Vertauschung von Abschnitt 2.4 und Abschnitt 2.3. Verbindung der Abschnitte 2.3 und 2.5.

2.1. Maschinenzahlen und Rundung

- **Körper der reellen Zahlen** \mathbb{R} : Gebräuchlichstes Zahlensystem in der Analysis, übliche Rechenregeln für Addition und Multiplikation, vollständige und archimedisch geordnete Menge, Veranschaulichung als unendlich lange lückenlose Linie (Zahlengerade).

Menge der Maschinenzahlen: Auf einem Rechner exakt darstellbare Zahlen (im Allgemeinen Elemente eines endlichen Systems normalisierter Gleitpunktzahlen, siehe Definition 2.2), endliche und insbesondere beschränkte Menge.

- **Normalisierte Gleitpunktzahl, Basis, Stellenzahl, Signifikand, Exponent** (Definition 2.1): Normalisierte t -stellige Gleitpunktzahl zur Basis B (wobei $B \in \mathbb{N}_{\geq 2}$, $t \in \mathbb{N}_{\geq 1}$)

$$g = 0 \quad \text{oder} \quad g = \pm |S| \cdot B^E \in \mathbb{G} = \mathbb{G}_{B,t}, \quad |S| \in \mathbb{N}_{\geq 1}, \quad B^{t-1} \leq |S| < B^t, \quad E \in \mathbb{Z}.$$

Bemerkungen:

- Die Menge \mathbb{G} ist unendlich.
- Eindeutigkeit der Darstellung (Signifikand, Exponent) aufgrund der Normalisierung des Signifikanden.

Beispiel: Darstellung der Zahl $0.012345 = 0.12345 \cdot 10^{-1} = 1.2345 \cdot 10^{-2}$ (etc.) als $12345 \cdot 10^{-6} \in \mathbb{G}_{10,5}$.

- **Auflösung** in \mathbb{G} :

$$\rho = \max \left\{ \left| \frac{\tilde{g}-g}{g} \right| : g, \tilde{g} \text{ benachbarte Zahlen in } \mathbb{G} \setminus \{0\} \right\} = B^{1-t}.$$

Denn: Für eine Zahl $g = \pm |S| \cdot B^E \in \mathbb{G}$ ist die benachbarte Zahl gegeben durch $\tilde{g} = (\pm |S| + 1) \cdot B^E$ bzw. $\tilde{g} = (\pm |S| - 1) \cdot B^E$ und daher $\left| \frac{\tilde{g}-g}{g} \right| = \frac{1}{|S|} \leq B^{1-t}$, sofern $g, \tilde{g} \neq 0$. Speziell für $\tilde{g} = 0$ folgt jedoch $\left| \frac{\tilde{g}-g}{g} \right| = 1$. \diamond

Bemerkung: Betrachtung relativer Größen anstelle absoluter Größen.

Maschinenzahl (Definition 2.2): Normalisierte t -stellige Gleitpunktzahl zur Basis B mit beschränktem Exponenten, d.h. $g = 0$ oder (wobei $B \in \mathbb{N}_{\geq 2}$, $t \in \mathbb{N}_{\geq 1}$ und $\alpha, \beta \in \mathbb{Z}$, $\alpha \leq \beta$)

$$g = \pm |S| \cdot B^E \in \mathbb{M} = \mathbb{M}_{B,t,\alpha,\beta} \subset \mathbb{G}_{B,t}, \quad |S| \in \mathbb{N}_{\geq 1}, \quad B^{t-1} \leq |S| < B^t, \quad E \in \mathbb{Z}, \quad \alpha \leq E \leq \beta.$$

Bemerkungen:

- Die Menge \mathbb{M} ist endlich.
- Maschinenzahlen sind (ebenso wie Gleitpunktzahlen) nicht äquidistant verteilt. Die Auflösung in \mathbb{M} stimmt mit der Auflösung in \mathbb{G} überein, d.h. $\rho = B^{1-t}$.

- **Kleinste positive Maschinenzahl** $\sigma = B^{t-1+\alpha}$.

Größte Maschinenzahl $\lambda = (B^t - 1) B^\beta \doteq B^{t+\beta}$.

- Übliche Genauigkeitsstufen sind **single precision** bzw. **double precision** bzw. **extended precision**, festgelegt durch den Standard ANSI/IEEE-Std-754-1985 für Gleitpunktarithmetik.

Vgl. **Illustration2_Maschinenzahlen**.

- **Bereichsüberschreitungen** treten auf, wenn Rechenergebnisse außerhalb des definierten Bereichs $[-\lambda, -\sigma] \cup \{0\} \cup [\sigma, \lambda]$ liegen. In mathematischen Software-Paketen werden daher auch zusätzliche **Sonderoperanden** verwendet.

Vgl. **Illustration2_Sonderoperanden** (Sonderoperanden $\pm\infty$ und quiet NAN not-a-number in MATLAB) und auch **Illustration2_Rundung**.

- **Vereinbarung:** Unter der Annahme, daß Bereichsüberschreitungen vermieden werden können, wird von nun an die Menge der normalisierten Gleitpunktzahlen \mathbb{G} anstelle der Menge der Maschinenzahlen \mathbb{M} betrachtet.
- Eingangsdaten und auch die Ergebnisse von elementaren arithmetischen Operationen liegen im Allgemeinen nicht im Bereich der darstellbaren Zahlen, d.h.

$$x, y \in \mathbb{G} \not\stackrel{\text{i.A.}}{\Rightarrow} x * y \in \mathbb{G}.$$

Deshalb besteht die Notwendigkeit der **Rundung** $\text{rd} : \mathbb{R} \rightarrow \mathbb{G}$, d.h. der Zuordnung einer reellen Zahl $x \in \mathbb{R}$ dem linken bzw. rechten Nachbarn in \mathbb{G}

$$g_L = \max \{g \in \mathbb{G} : g \leq x\} \leq x \leq g_R = \min \{g \in \mathbb{G} : g \geq x\}.$$

Insbesondere gilt $g_L = x = g_R$ für $x \in \mathbb{G}$.

Korrektes Runden, gerichtetes Runden (Definition 2.3): Folgende Arten der Rundung sind gebräuchlich

$$\begin{aligned} \text{Korrektes Runden :} \quad \text{rd}_* : \mathbb{R} \rightarrow \mathbb{G} : x \mapsto & \begin{cases} g_L & \text{falls } x < \frac{g_L + g_R}{2}, \\ g_L \text{ oder } g_R & \text{falls } x = \frac{g_L + g_R}{2}, \\ g_R & \text{falls } x > \frac{g_L + g_R}{2}, \end{cases} \\ \\ \text{Gerichtetes Runden :} \quad & \begin{cases} \text{Abrunden :} \quad \text{rd}_- : \mathbb{R} \rightarrow \mathbb{G} : x \mapsto g_L, \\ \text{Aufrunden :} \quad \text{rd}_+ : \mathbb{R} \rightarrow \mathbb{G} : x \mapsto g_R, \\ \text{Abhacken :} \quad \text{rd}_0 : \mathbb{R} \rightarrow \mathbb{G} : x \mapsto \begin{cases} g_L & \text{falls } x \geq 0, \\ g_R & \text{falls } x < 0. \end{cases} \end{cases} \end{aligned}$$

- Man unterscheidet den absoluten Rundungsfehler $\text{rd}(x) - x$ für $x \in \mathbb{R}$ und den **relativen Rundungsfehler**

$$\varepsilon_x = \frac{\text{rd}(x) - x}{x}, \quad 0 \neq x \in \mathbb{R}.$$

Für $x = 0$ ist $\text{rd}(x) = x$ und man setzt speziell $\varepsilon_x = 0$.

Maschinengenauigkeit (Satz 2.5): (Relative) Maschinengenauigkeit (wobei $\varrho = B^{1-t}$ Auflösung in \mathbb{G} bzw. \mathbb{M})

$$\varepsilon_{\text{mach}} = \begin{cases} \frac{1}{2} \varrho & \text{für korrektes Runden,} \\ \varrho & \text{für gerichtetes Runden.} \end{cases}$$

Abschätzung des relativen Rundungsfehlers (Satz 2.4): Für $0 \neq x \in \mathbb{R}$ gilt $\varepsilon_x = \frac{\text{rd}(x)-x}{x}$ mit $|\varepsilon_x| \leq \varepsilon_{\text{mach}}$ und damit

$$\text{rd}(x) = x(1 + \varepsilon_x), \quad |\varepsilon_x| \leq \varepsilon_{\text{mach}}.$$

Denn: Bei korrekter Rundung folgt für $g_L \leq x < \frac{g_L + g_R}{2}$ und damit $\text{rd}(x) = g_L$ die Abschätzung $|\varepsilon_x| = \left| \frac{\text{rd}(x)-x}{x} \right| \leq \frac{1}{2} \left| \frac{g_R - g_L}{g_L} \right| \leq \frac{1}{2} \varrho$. Ähnliche Überlegungen gelten für $\frac{g_L + g_R}{2} \leq x \leq g_R$ sowie gerichtetes Runden. \diamond

- **Vorbemerkung:** Summenformel für **geometrische Reihe**

$$\sum_{i=n_0}^{n_1} q^i = \frac{q^{n_0} - q^{n_1+1}}{1 - q}, \quad \sum_{i=n_0}^{\infty} q^i = \frac{q^{n_0}}{1 - q}, \quad |q| < 1.$$

Insbesondere gilt (wegen $B \geq 2$)

$$\sum_{i=n_0}^{\infty} B^{-i} = \frac{B^{-n_0}}{1 - B^{-1}} = \frac{B^{1-n_0}}{B-1}.$$

Bemerkung: Korrekte Rundung einer Zahl $x \in \mathbb{R}$ auf t Stellen bei Kenntnis von $t+1$ Stellen. Mittels der Darstellung (mit Ziffern $0 \leq s_j \leq B-1$, wobei $s_\ell < B-1$ für ein $\ell \leq -2$)

$$x = \pm \sum_{j=-\infty}^{t-1} s_j \cdot B^{j+E} = \pm s \cdot B^E,$$

$$s = \sum_{j=-\infty}^{t-1} s_j \cdot B^j = \sum_{j=0}^{t-1} s_j \cdot B^j + s_{-1} \cdot B^{-1} + r, \quad 0 \leq r = \sum_{j=-\infty}^{-2} s_j \cdot B^j < (B-1) \sum_{i=2}^{\infty} B^{-i} = B^{-1},$$

ergibt sich für die mathematisch korrekte Rundung

$$\text{rd}(x) = \pm B^E \begin{cases} \sum_{j=0}^{t-1} s_j \cdot B^j, & \text{falls } s_{-1} < \frac{1}{2} B, \\ \sum_{j=0}^{t-1} s_j \cdot B^j + 1, & \text{falls } s_{-1} \geq \frac{1}{2} B. \end{cases}$$

Beachte, daß die Zifferndarstellung durch die obige Forderung eindeutig ist (beispielsweise identifiziert man im Dezimalsystem $4.999\dots = 5$). Falls $s_{-1} \geq \frac{1}{2} B$ und $s_0 = B-1$ kommt es zu Überlauf.

Vgl. **Illustration2_Rundung**.

2.2. Gleitpunktoperationen

- Die Ergebnisse der elementaren arithmetischen Operationen $* \in \{+, -, \times, / \}$ liegen im Allgemeinen nicht mehr im Bereich der darstellbaren Zahlen, d.h. für $a, b \in \mathbb{G}$ wird statt des exakten Ergebnisses

$$a * b \underset{\text{i.A.}}{\notin} \mathbb{G}$$

eine Näherungslösung

$$a \overset{\circ}{*} b \in \mathbb{G}$$

berechnet.

Ideale Arithmetik: Die berechnete Näherungslösung $a \overset{\circ}{*} b$ ergibt sich durch Rundung des exakten Ergebnisses, d.h. es gilt

$$a \overset{\circ}{*} b = \text{rd}(a * b) \in \mathbb{G}.$$

Vereinbarung: Entsprechend dem Standard ANSI/IEEE-Std-754-1985 für Gleitpunktarithmetik wird von nun an angenommen, daß die ideale Arithmetik in allen Genauigkeitsstufen für alle elementaren arithmetischen Operationen sowie die Berechnung der Wurzel und alle Rundungsarten gilt.

- Rundungsfehlerschranken** (Satz 2.5): In idealer Arithmetik gilt

$$\begin{aligned} a \overset{\circ}{+} b &= \text{rd}(a + b) = (a + b)(1 + \varepsilon_1), & a \overset{\circ}{-} b &= \text{rd}(a - b) = (a - b)(1 + \varepsilon_2), \\ a \overset{\circ}{\times} b &= \text{rd}(ab) = ab(1 + \varepsilon_3), & a \overset{\circ}{/} b &= \text{rd}\left(\frac{a}{b}\right) = \frac{a}{b}(1 + \varepsilon_4), \end{aligned}$$

mit relativen Rundungsfehlern $\varepsilon_i = \varepsilon_i(a, b)$ und $|\varepsilon_i| \leq \varepsilon_{\text{mach}}$.

Beispiel: Vgl. **Illustration2_Rundung** ($B = 10, t = 4, \varepsilon_{\text{mach}} = \frac{1}{2} \cdot 10^{1-t}$ bei korrekter Rundung).

- Beispiel** (Rundungsfehler bei Addition dreier Zahlen, Vorwärtsanalyse):

– Aufgabe: Berechnung von

$$x = a + b + c = (a + b) + c = a + (b + c)$$

unter dem Einfluß von Rundungsfehlern mittels Satz 2.5.

- Bestimmung von $(a + b) + c$ unter dem Einfluß von Rundungsfehlern (wobei $|\varepsilon_1|, |\varepsilon_2| \leq \varepsilon_{\text{mach}}$)

$$\begin{aligned} \tilde{x} &= \text{rd}(\text{rd}(a + b) + c) \\ &= \text{rd}((a + b)(1 + \varepsilon_1) + c) \\ &= ((a + b)(1 + \varepsilon_1) + c)(1 + \varepsilon_2) \\ &= (a + b + c + (a + b)\varepsilon_1)(1 + \varepsilon_2) \\ &= x \left(1 + \frac{a+b}{a+b+c} \varepsilon_1(1 + \varepsilon_2) + \varepsilon_2\right). \end{aligned}$$

Als relativer Fehler ergibt sich

$$\varepsilon_{\tilde{x}} = \frac{\tilde{x}-x}{x} = \frac{a+b}{a+b+c} \varepsilon_1(1 + \varepsilon_2) + \varepsilon_2.$$

- Bestimmung von $a + (b + c)$ unter dem Einfluß von Rundungsfehlern (wobei $|\varepsilon_3|, |\varepsilon_4| \leq \varepsilon_{\text{mach}}$)

$$\begin{aligned}\hat{x} &= \text{rd}(a + \text{rd}(b + c)) \\ &= \text{rd}(a + (b + c)(1 + \varepsilon_3)) \\ &= (a + (b + c)(1 + \varepsilon_3))(1 + \varepsilon_4) \\ &= (a + b + c + (b + c)\varepsilon_3)(1 + \varepsilon_4) \\ &= x \left(1 + \frac{b+c}{a+b+c} \varepsilon_3(1 + \varepsilon_4) + \varepsilon_4\right).\end{aligned}$$

Als relativer Fehler ergibt sich

$$\varepsilon_{\hat{x}} = \frac{\hat{x}-x}{x} = \frac{b+c}{a+b+c} \varepsilon_3(1 + \varepsilon_4) + \varepsilon_4.$$

- **Schlußfolgerung:** Der relative Fehler des Ergebnisses bei der Addition von drei Zahlen hängt i.A. von der gewählten Reihenfolge der Operationen ab, d.h. Assoziativgesetz und Distributivgesetz gelten im Allgemeinen nicht mehr. Im Gegensatz zur Addition von zwei Zahlen kann der Fall eintreten, daß der relative Fehler nicht beschränkt ist.
- **Zahlenbeispiel,** vgl. **Illustration2_Rundung.**

Vgl. **Illustration2_Rundung:** Günstige Wahl der Reihenfolge bei der Berechnung von (wobei $n \gg 1$)

$$\sum_{i=1}^n \frac{1}{i}, \quad \sum_{i=1}^n \frac{(-1)^i}{i}.$$

2.4. Kondition

- **Situation:** Lösung eines numerischen Problems $p : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$, d.h. Berechnung von

$$y = p(x).$$

Annahme p regulär (hinreichend oft differenzierbar).

- **Fragestellung:** Beurteilung der Sinnhaftigkeit der numerischen Lösung eines Problems, d.h. der Berechenbarkeit des Ergebnisses. Untersuchung der Auswirkungen kleiner Änderungen der Eingabedaten auf die Ergebnisse bei Anwendung von $p : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$\begin{array}{llll} \text{Eingabe:} & x + \xi & \text{statt } x & \text{mit relativem Fehler } \frac{\xi}{x}, \\ \text{Ergebnis:} & p(x + \xi) = y + \eta & \text{statt } y = p(x) & \text{mit relativem Fehler } \frac{\eta}{y}, \end{array}$$

d.h. für $\xi \in \mathbb{R}^n$ (*klein* und jedenfalls so gewählt, daß $x + \xi \in D$) bestimme $\eta \in \mathbb{R}^m$ mit

$$\eta = p(x + \xi) - y = p(x + \xi) - p(x).$$

Beispielsweise aufgrund der Darstellung der Eingabedaten als Maschinenzahlen sind solche kleinen Änderungen unvermeidbar.

- **Vorbemerkungen:**
 - Betrachtung von kleinen Änderungen (komponentenweise Abschätzung mit Inkrement $\Delta x \in \mathbb{R}^n$, wobei die Relation \leq komponentenweise zu verstehen ist, oder Abschätzung bzgl. einer Norm mit $\Delta x \in \mathbb{R}$)

$$|\xi| \leq \Delta x \quad \text{oder} \quad \|\xi\| \leq \Delta x.$$

- Mittels Taylorreihenentwicklung erhält man

$$\eta = p(x + \xi) - p(x) = p'(x)\xi + \mathcal{O}(\|\xi\|^2).$$

In Komponenten

$$\begin{pmatrix} \eta_1 \\ \vdots \\ \eta_m \end{pmatrix} = \begin{pmatrix} \partial_{x_1} p_1(x) & \dots & \partial_{x_n} p_1(x) \\ \vdots & & \vdots \\ \partial_{x_1} p_m(x) & \dots & \partial_{x_n} p_m(x) \end{pmatrix} \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix} + \mathcal{O}(\|\xi\|^2).$$

Für ξ *klein genug* folgt

$$\eta_i \approx (p'(x)\xi)_i = \sum_{j=1}^n \partial_{x_j} p_i(x) \xi_j, \quad 1 \leq i \leq m,$$

und damit für den relativen Fehler

$$\frac{\eta_i}{y_i} \approx \sum_{j=1}^n \partial_{x_j} p_i(x) \frac{\xi_j}{y_i} = \sum_{j=1}^n \frac{x_j}{y_i} \partial_{x_j} p_i(x) \frac{\xi_j}{x_j}.$$

- **Kondition, Konditionszahlen** (Definition 2.8): Unter der Kondition eines numerischen Problems versteht man die Sensitivität des Ergebnisses $y + \eta = p(x + \xi)$ gegenüber *kleinen* Änderungen der Eingangsdaten. Falls *kleine* Änderungen ξ *kleine* Änderungen η im Ergebnis bewirken, spricht man von einem *gut* konditionierten Problem, falls hingegen *kleine* Änderungen ξ *große* Änderungen η im Ergebnis bewirken, spricht man von einem *schlecht* konditionierten Problem. Die Größen (unabhängig von ξ, η)

$$|\partial_{x_j} p_i(x)| \quad \text{bzw.} \quad \left| \frac{x_j}{y_i} \partial_{x_j} p_i(x) \right|,$$

bezeichnet man als absolute bzw. relative Konditionszahlen.

Bemerkung: Anstelle der absoluten Konditionszahlen wird auch oft eine Operatornorm $\|p'(x)\|$ betrachtet.

- **Beispiele** (Kondition):

- **Relative Konditionszahlen elementarer Operationen:** Einsetzen in obige Relation speziell für

$$p: \mathbb{R}^2 \rightarrow \mathbb{R}: (a, b) \mapsto y = p(a, b)$$

führt auf (wobei $\xi = (\xi_a, \xi_b)^T$)

$$\frac{\eta}{y} = \frac{a}{y} \partial_a p(a, b) \frac{\xi_a}{a} + \frac{b}{y} \partial_b p(a, b) \frac{\xi_b}{b} + \mathcal{O}(\xi_a^2 + \xi_b^2).$$

- * Addition:

$$y = p(a, b) = a + b, \quad \partial_a p(a, b) = \partial_b p(a, b) = 1, \quad \frac{\eta}{y} = \frac{a}{a+b} \frac{\xi_a}{a} + \frac{b}{a+b} \frac{\xi_b}{b}.$$

Schlecht konditioniertes Probleme für $a + b \approx 0$ bzw. $b \approx -a$ (Annahme $|a|, |b| > 0$), da dann die relativen Konditionszahlen sehr groß sind, d.h. Verstärkung relativer Eingabefehler.

Phänomen der Auslöschung signifikanter Stellen, vgl. **Illustration2_Kondition**.

Bemerkung: Gleichheit gilt aufgrund der Linearität der Funktion.

- * Subtraktion:

$$y = p(a, b) = a - b, \quad \partial_a p(a, b) = 1, \quad \partial_b p(a, b) = -1, \quad \frac{\eta}{y} = \frac{a}{a-b} \frac{\xi_a}{a} + \frac{b}{a-b} \frac{\xi_b}{b}.$$

Bemerkung: Zurückführen auf Addition.

- * Multiplikation, Division:

$$y = p(a, b) = a b, \quad \partial_a p(a, b) = b, \quad \partial_b p(a, b) = a, \quad \frac{\eta}{y} \approx \frac{\xi_a}{a} + \frac{\xi_b}{b},$$

$$y = p(a, b) = \frac{a}{b}, \quad \partial_a p(a, b) = \frac{1}{b}, \quad \partial_b p(a, b) = -\frac{a}{b^2}, \quad \frac{\eta}{y} \approx \frac{\xi_a}{a} - \frac{\xi_b}{b}.$$

Gut konditionierte Probleme, da die relativen Konditionszahlen durch 1 beschränkt sind, d.h. keine Verstärkung relativer Eingabefehler (für die lineare Näherung).

* Wurzelziehen:

$$y = p(a) = \sqrt{a}, \quad \partial_a p(a) = \frac{1}{2\sqrt{a}}, \quad \frac{\eta}{y} \approx \frac{1}{2} \frac{\xi_a}{a}.$$

Gut konditioniertes Problem, da die relativen Konditionszahlen durch $\frac{1}{2}$ beschränkt sind, d.h. Verkleinerung relativer Eingabefehler (für lineare Näherung).

- **Relative Konditionszahlen der Formel von Vieta:** Bestimmung einer der beiden Lösungen $x_{1,2} = -a \pm \sqrt{a^2 + b}$ der quadratischen Gleichung $x^2 + 2ax - b = 0$ (Annahme $|a|, |b| > 0$, Relationen $x_1 + x_2 = -2a$ und $x_1 x_2 = -b$)

$$y = p(a, b) = -a + \sqrt{a^2 + b},$$

$$\partial_a p(a, b) = -1 + \frac{a}{\sqrt{a^2 + b}} = -\frac{y}{\sqrt{a^2 + b}}, \quad \partial_b p(a, b) = \frac{1}{2\sqrt{a^2 + b}},$$

$$\frac{\eta}{y} \approx -\frac{a}{\sqrt{a^2 + b}} \frac{\xi_a}{a} + \frac{b}{2y\sqrt{a^2 + b}} \frac{\xi_b}{b} = -\frac{a}{\sqrt{a^2 + b}} \frac{\xi_a}{a} + \frac{a + \sqrt{a^2 + b}}{2\sqrt{a^2 + b}} \frac{\xi_b}{b}.$$

Gut konditioniertes Problem beispielsweise für $a > 0$ und $b > 0$, da dann die relativen Konditionszahlen durch 1 beschränkt sind. Falls jedoch $a^2 + b \approx 0$ bzw. $b \approx -a^2$ ($x_1 \approx x_2$, sogenannter *schleifender Schnitt*) ist das Problem schlecht konditioniert. Daran ändert auch die Umformulierung des Problems

$$y = p(a, b) = \frac{b}{a + \sqrt{a^2 + b}},$$

nichts (Kettenregel: partielle Ableitungen und damit Konditionszahlen gleich, vgl. **Illustration2_Kondition**).

Aber! (Vorbemerkung zur Stabilität von Algorithmen) Für $a^2 + b \approx a^2$ bzw. $b \approx 0$ ist das Problem gut konditioniert (z.B. $a \gg b > 0$ und $b \approx 0$, relative Konditionszahlen durch 1 beschränkt)

$$(a, b) \xrightarrow{p} -a + \sqrt{a^2 + b}.$$

$$\left| -\frac{a}{\sqrt{a^2 + b}} \right| \leq 1, \quad \left| \frac{a + \sqrt{a^2 + b}}{2\sqrt{a^2 + b}} \right| \leq 1$$

Berechnung von $y = -a + \sqrt{a^2 + b}$ mit dem kanonischen Algorithmus

$$(a, b) \xrightarrow[1,1]{p^{(1)}} a^2 \xrightarrow{\left| \frac{a^2}{a^2 + b} \right| \leq 1, \left| \frac{b}{a^2 + b} \right| \leq 1}{p^{(2)}} a^2 + b \xrightarrow[\frac{1}{2}]{p^{(3)}} \sqrt{a^2 + b} \xrightarrow{p^{(4)}} -a + \sqrt{a^2 + b}.$$

$$\frac{-a}{-a + \sqrt{a^2 + b}}, \frac{\sqrt{a^2 + b}}{-a + \sqrt{a^2 + b}}$$

Die Teilprobleme $p^{(1)}, p^{(2)}, p^{(3)}$ sind in der vorliegenden Situation gut konditioniert, beim letzten Teilproblem $p^{(4)}$ kommt es jedoch zur Auslöschung signifikanter Stellen, d.h. der gewählte Algorithmus führt für ein gut konditioniertes

Problem auf ein unzufriedenstellendes Ergebnis (instabiler Algorithmus). Daher sollte man jedenfalls den (stabilen) Algorithmus

$$\begin{array}{c}
 (a, b) \xrightarrow[1,1]{p^{(1)}} a^2 \xrightarrow{\left| \frac{a^2}{a^2+b} \right| \leq 1, \left| \frac{b}{a^2+b} \right| \leq 1} p^{(2)} a^2 + b \xrightarrow[\frac{1}{2}]{p^{(3)}} \sqrt{a^2 + b} \\
 \xrightarrow{\left| \frac{-a}{a+\sqrt{a^2+b}} \right| \leq 1, \left| \frac{\sqrt{a^2+b}}{a+\sqrt{a^2+b}} \right| \leq 1} p^{(4)} a + \sqrt{a^2 + b} \xrightarrow[1,1]{p^{(5)}} \frac{b}{a+\sqrt{a^2+b}}
 \end{array}$$

mit in der vorliegenden Situation gut konditionierten Teilproblemen verwenden. Vgl. **Illustation2_Kondition**.

Bemerkung: Eine etwas andere Situation liegt für den Fall $b \approx -a^2$ ($a, b \gg 0$) vor. In dieser Situation ist das Problem schlecht konditioniert und damit dessen numerische Lösung nur eingeschränkt sinnvoll. Außerdem beinhalten beide Algorithmen das schlecht konditionierte Teilproblem $p^{(2)}$.

2.3. Rundungsfehleranalyse

- **Situation:** Anwendung eines Algorithmus $(p^{(1)}, \dots, p^{(k)})$ zur Lösung eines numerischen Problems $p: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$. Betrachtung eines direkten Verfahren, d.h. es gelte

$$p^{(k)} \circ \dots \circ p^{(1)} = p.$$

- **Fragestellung:** Beurteilung der Güte eines Algorithmus. Untersuchung der Auswirkungen von kleinen Änderungen bzw. Rundungen der Eingabedaten sowie von bei der Lösung der Teilprobleme auftretenden Rundungen auf das Endergebnis.
- **Rundungsfehleranalyse:** Untersuchung der Fortpflanzung von Rundungsfehlern auf das Endergebnis.

- **Vorwärtsanalyse:** Vergleich des unter dem Einfluß von Rundungsfehlern erhaltenen Endergebnisses mit dem exakten Ergebnis und Herleitung einer Relation bzw. Abschätzung für den relativen Fehler.

Nachteil der Vorwärtsanalyse: Analyse bei komplexeren Algorithmen kompliziert.

- **Rückwärtsanalyse:** Interpretation des unter dem Einfluß von Rundungsfehlern erhaltenen Endergebnisses als exaktes Ergebnis zu veränderten Eingabedaten. Keine Aussage über die tatsächliche Größe des relativen Rundungsfehlers im Ergebnis.

Vorteil der Rückwärtsanalyse: Analyse auch bei komplexeren Algorithmen leichter durchführbar als die Vorwärtsanalyse.

- **Beispiele** (Rückwärtsanalyse):

- **Elementare arithmetische Operationen:** Mit Hilfe von Satz 2.5 und dem Ansatz $\text{rd}(a * b) = \tilde{a} * \tilde{b}$ ergibt sich

$$\text{rd}(a * b) = (a * b)(1 + \varepsilon) = \tilde{a} * \tilde{b}$$

mit $|\varepsilon| \leq \varepsilon_{\text{mach}}$ (Annahme $1 - \varepsilon_{\text{mach}} > 0$). Wähle beispielsweise für die Addition (Subtraktion analog)

$$\begin{aligned} \text{rd}(a + b) &= (a + b)(1 + \varepsilon) = \tilde{a} + \tilde{b}, \\ \tilde{a} &= a(1 + \varepsilon), \quad \tilde{b} = b(1 + \varepsilon), \quad \left| \frac{\tilde{a} - a}{a} \right| = \left| \frac{\tilde{b} - b}{b} \right| = |\varepsilon| \leq \varepsilon_{\text{mach}}, \end{aligned}$$

für die Multiplikation

$$\begin{aligned} \text{rd}(ab) &= ab(1 + \varepsilon) = \tilde{a}\tilde{b}, \\ \tilde{a} &= a\sqrt{1 + \varepsilon}, \quad \tilde{b} = b\sqrt{1 + \varepsilon}, \quad \left| \frac{\tilde{a} - a}{a} \right| = \left| \frac{\tilde{b} - b}{b} \right| = \left| \sqrt{1 + \varepsilon} - 1 \right| = \frac{|\varepsilon|}{\sqrt{1 + \varepsilon} + 1} \leq \varepsilon_{\text{mach}}, \end{aligned}$$

und für die Division

$$\begin{aligned} \text{rd}\left(\frac{a}{b}\right) &= \frac{a}{b} (1 + \varepsilon) = \frac{\tilde{a}}{\tilde{b}}, \\ \tilde{a} &= a\sqrt{1 + \varepsilon}, \quad \tilde{b} = \frac{b}{\sqrt{1 + \varepsilon}}, \\ \left|\frac{\tilde{a}-a}{a}\right| &\leq \varepsilon_{\text{mach}}, \quad \left|\frac{\tilde{b}-b}{b}\right| = \frac{|1-\sqrt{1+\varepsilon}|}{\sqrt{1+\varepsilon}} = \frac{|\varepsilon|}{\sqrt{1+\varepsilon}(1+\sqrt{1+\varepsilon})} \leq \frac{\varepsilon_{\text{mach}}}{\sqrt{1-\varepsilon_{\text{mach}}}}. \end{aligned}$$

Schlußfolgerung: Die bei der Anwendung der elementaren arithmetischen Operationen auftretenden Rundungsfehler verfälschen das exakte Ergebnis so wie relative Änderungen der Größenordnung $\varepsilon_{\text{mach}}$ in den Eingangsdaten.

– **Horner-Schema** zur Berechnung des Funktionswertes einer Polynomfunktion:

* **Bernoullische Ungleichung:** Für $x \in \mathbb{R}$ mit $x \geq -1$ und $n \in \mathbb{N}_{\geq 0}$ gilt

$$1 + nx \leq (1 + x)^n.$$

Denn: Mittels Induktion folgt

$$(1 + x)^{n+1} = (1 + x)(1 + x)^n \geq (1 + x)(1 + nx) = 1 + (n + 1)x + nx^2 \geq 1 + (n + 1)x$$

und damit die Abschätzung. \diamond

Abschätzung für Fehlerterme der Rundungsfehleranalyse (Lemma 2.6): Für eine Folge reeller Zahlen $(\varepsilon_i)_{1 \leq i \leq n}$ mit $|\varepsilon_i| \leq \varepsilon_{\text{mach}}$ gilt für $0 \leq k \leq n$

$$\left| \prod_{i=1}^k (1 + \varepsilon_i) \prod_{j=k+1}^n \frac{1}{1 + \varepsilon_j} - 1 \right| \leq \frac{n \varepsilon_{\text{mach}}}{1 - n \varepsilon_{\text{mach}}}.$$

Denn: Wegen $|\varepsilon_i| \leq \varepsilon_{\text{mach}}$ ist $-\varepsilon_{\text{mach}} \leq \varepsilon_i \leq \varepsilon_{\text{mach}}$ sowie $-\varepsilon_{\text{mach}} \leq -\varepsilon_i \leq \varepsilon_{\text{mach}}$ für $1 \leq i \leq n$. Zusammen mit der Relation $(1 + \varepsilon_i)(1 - \varepsilon_i) = 1 - \varepsilon_i^2 \leq 1$ folgt (Annahme $n \varepsilon_{\text{mach}} < 1$ und insbesondere $\varepsilon_{\text{mach}} < 1$)

$$1 - \varepsilon_{\text{mach}} \underset{-\varepsilon_{\text{mach}} \leq \varepsilon_i}{\leq} 1 + \varepsilon_i \underset{(1+\varepsilon_i)(1-\varepsilon_i) \leq 1}{\leq} \frac{1}{1 - \varepsilon_i} \underset{-\varepsilon_{\text{mach}} \leq -\varepsilon_i}{\leq} \frac{1}{1 - \varepsilon_{\text{mach}}}.$$

Mittels Bernoullischer Ungleichung ergibt sich weiters

$$\begin{aligned} 1 - \frac{n \varepsilon_{\text{mach}}}{1 - n \varepsilon_{\text{mach}}} &\underset{n \varepsilon_{\text{mach}} \leq \frac{n \varepsilon_{\text{mach}}}{1 - n \varepsilon_{\text{mach}}}}{\leq} 1 - n \varepsilon_{\text{mach}} \underset{\text{Bernoulli}}{\leq} (1 - \varepsilon_{\text{mach}})^n \\ &\underset{\text{Relation}}{\leq} \prod_{i=1}^k (1 + \varepsilon_i) \prod_{j=k+1}^n \frac{1}{1 + \varepsilon_j} \underset{\text{Relation}}{\leq} \frac{1}{(1 - \varepsilon_{\text{mach}})^n} \\ &\underset{\text{Bernoulli}}{\leq} \frac{1}{1 - n \varepsilon_{\text{mach}}} = 1 + \frac{n \varepsilon_{\text{mach}}}{1 - n \varepsilon_{\text{mach}}} \end{aligned}$$

und damit die Behauptung. \diamond

- * Für eine Polynomfunktion der Form

$$p(x) = \sum_{i=0}^n c_i x^i$$

beruht das Horner-Schema auf der Umformulierung

$$y = c_n x^n + c_{n-1} x^{n-1} + \dots + c_0 = \left(\dots \left((c_n x + c_{n-1}) x + c_{n-2} \right) x + \dots \right) x + c_0.$$

Mögliche Implementierung (Pseudo-Code)

```

y = c_n
for i = n-1:-1:0
    y = y x + c_i
end

```

Vgl. auch **Illustration1_Modifikation**.

- * Unter dem Einfluß von Rundungsfehlern erhält man stattdessen

```

y-tilde = c_n
for i = n-1:-1:0
    y-tilde = rd(rd(y-tilde x) + c_i) = (y-tilde x (1 + epsilon_i) + c_i) (1 + epsilon-tilde_i)
end

```

wobei $|\epsilon_i|, |\tilde{\epsilon}_i| \leq \epsilon_{\text{mach}}$. Das berechnete Ergebnis läßt sich in der Form

$$\tilde{y} = \tilde{c}_n x^n + \tilde{c}_{n-1} x^{n-1} + \dots + \tilde{c}_0$$

mit Koeffizienten (Induktion)

$$\tilde{c}_0 = c_0 (1 + \tilde{\epsilon}_0), \quad \tilde{c}_n = c_n \prod_{\ell=0}^{n-1} (1 + \epsilon_\ell) (1 + \tilde{\epsilon}_\ell),$$

$$\tilde{c}_i = c_i (1 + \tilde{\epsilon}_i) \prod_{\ell=0}^{i-1} (1 + \epsilon_\ell) (1 + \tilde{\epsilon}_\ell), \quad 1 \leq i \leq n-1,$$

darstellen. Mittels Lemma 2.6 folgt außerdem (wegen $\frac{k\epsilon_{\text{mach}}}{1-k\epsilon_{\text{mach}}} \leq \frac{n\epsilon_{\text{mach}}}{1-n\epsilon_{\text{mach}}}$ für $0 \leq k \leq n$)

$$\tilde{c}_i = c_i (1 + \delta_i), \quad |\delta_i| \leq \frac{n\epsilon_{\text{mach}}}{1-n\epsilon_{\text{mach}}}, \quad 0 \leq i \leq n.$$

- * **Schlußfolgerung:** Die bei der Anwendung des Horner-Schemas auftretenden Rundungsfehler verfälschen das exakte Ergebnis so wie relative Änderungen der Größenordnung $\frac{n\epsilon_{\text{mach}}}{1-n\epsilon_{\text{mach}}}$ in den Eingangsdaten, d.h. den Koeffizienten des Polynoms.

- **Numerische Stabilität** eines Algorithmus: Ein Algorithmus heißt numerisch stabil bzw. gutartig im Sinne der Rückwärtsanalyse, wenn die Fortpflanzung von Rundungsfehlern zu Änderungen im Endergebnis führt, die in ihrer Größenordnung mit dem unvermeidlichen Fehler aufgrund von Ungenauigkeiten in den Eingangsdaten vergleichbar sind.
- **Spezialfall:** Betrachte speziell einen aus zwei Teilproblemen bestehenden Algorithmus zur Lösung eines Problems $p: \mathbb{R}^n \rightarrow \mathbb{R}^m$ (wobei $q: \mathbb{R}^n \rightarrow \mathbb{R}^k$, $r: \mathbb{R}^k \rightarrow \mathbb{R}^m$)

$$p(x) = (r \circ q)(x) = r(q(x)).$$

Die Anwendung der Kettenregel ergibt

$$p'(x) = r'(q(x)) q'(x),$$

$$p' = \begin{pmatrix} \partial_{x_1} p_1 & \dots & \partial_{x_n} p_1 \\ \vdots & & \vdots \\ \partial_{x_1} p_m & \dots & \partial_{x_n} p_m \end{pmatrix}, \quad r' = \begin{pmatrix} \partial_{x_1} r_1 & \dots & \partial_{x_k} r_1 \\ \vdots & & \vdots \\ \partial_{x_1} r_m & \dots & \partial_{x_k} r_m \end{pmatrix}, \quad q' = \begin{pmatrix} \partial_{x_1} q_1 & \dots & \partial_{x_n} q_1 \\ \vdots & & \vdots \\ \partial_{x_1} q_k & \dots & \partial_{x_n} q_k \end{pmatrix},$$

$$\partial_{x_j} p_i(x) = \sum_{\ell=1}^k \partial_{x_\ell} r_i(q(x)) \partial_{x_j} q_\ell(x).$$

Damit ergibt sich für die relativen Konditionszahlen

$$\frac{x_j}{y_i} \partial_{x_j} p_i(x) = \sum_{\ell=1}^k \frac{z_\ell}{y_i} \partial_{x_\ell} r_i(z) \frac{x_j}{z_\ell} \partial_{x_j} q_\ell(x), \quad z = q(x).$$

Die relativen Konditionszahlen des Problems sind unabhängig von der gewählten Zerlegung, d.h. von der Wahl der Teilprobleme r und q .

Aber! Die Wahl der Teilprobleme wirkt sich auf die Güte des Algorithmus aus, insbesondere bestimmt die Größe der Konditionszahlen die Fortpflanzung von Fehlern und damit die Stabilität des Algorithmus. Untersuchung der Auswirkung kleiner Änderungen der Eingangsdaten auf das Ergebnis des Algorithmus, d.h. insbesondere Hinzunahme auftretender Änderungen (Rundungsfehler) bei der Berechnung des Zwischenergebnisses und des Endergebnisses

Eingabe :	$x + \xi$	statt x ,
Zwischenergebnis :	$q(x + \xi) + \zeta$	statt $z = q(x)$,
Endergebnis :	$r(q(x + \xi) + \zeta) + \eta$	statt $y = r(q(x))$.

Mittels Taylorreihenentwicklung von q folgt

$$q(x + \xi) = z + q'(x) \xi + \mathcal{O}(\|\xi\|^2).$$

Eine Taylorreihenentwicklung von r ergibt weiters

$$\begin{aligned} r(q(x + \xi) + \zeta) &= r\left(z + q'(x) \xi + \mathcal{O}(\|\xi\|^2) + \zeta\right) \\ &= y + \mathbf{r}'(\mathbf{z}) \mathbf{q}'(\mathbf{x}) \xi + \mathbf{r}'(\mathbf{z}) \zeta + \mathcal{O}(\|\xi\|^2) + \mathcal{O}(\|\zeta\|^2). \end{aligned}$$

Falls das Problem gut konditioniert ist, ist

$$\|p'(x)\| = \|r'(z) q'(x)\|$$

von moderater Größenordnung und damit $r'(z) q'(x) \xi$ eine kleine Änderung des exakten Endergebnisses. Ist der Algorithmus jedoch so gewählt, daß das erste Teilproblem sehr gut und das zweite sehr schlecht konditioniert sind (d.h. $\|r'(z)\| \gg \|q'(z)\|$) werden unvermeidbare Fehler ζ im Zwischenergebnis (Rundung) sehr verstärkt, d.h. der Beitrag $r'(z) \zeta$ führt auf ein unzufriedenstellendes Endergebnis und somit ist der Algorithmus instabil.

• **Beispiele:**

- **Formel von Vieta:** Siehe obige Überlegungen. Das Problem ist für $a \gg b$ gut konditioniert. Auswirkung von kleinen relativen Änderungen der Eingangsdaten und relativen Rundungsfehlern bei der Anwendung der Formel von Vieta

$$\begin{array}{l}
 (a, b), (\varepsilon_a, \varepsilon_b) \xrightarrow[\mathbf{1,1}]{p^{(1)}} a^2, (\varepsilon_a, \varepsilon_1) \\
 \xrightarrow{p^{(2)}} a^2 + b, (\varepsilon_a, \varepsilon_b, \varepsilon_1, \varepsilon_2) \\
 \left| \frac{a^2}{a^2+b} \right| \leq 1, \left| \frac{b}{a^2+b} \right| \leq 1 \\
 \xrightarrow[\mathbf{\frac{1}{2}}]{p^{(3)}} \sqrt{a^2 + b}, (\varepsilon_a, \varepsilon_b, \varepsilon_1, \varepsilon_2, \varepsilon_3) \\
 \xrightarrow{p^{(4)}} -a + \sqrt{a^2 + b}, (\varepsilon_a, \varepsilon_b, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4). \\
 \frac{-a}{-a + \sqrt{a^2+b}}, \frac{\sqrt{a^2+b}}{-a + \sqrt{a^2+b}}
 \end{array}$$

Der Algorithmus ist instabil, da der Verstärkungsfaktor der relativen Rundungsfehler im letzten Teilproblem sehr groß ist. Z.B. für $a = 1000$, $b = 0.018000000081$

$$\frac{\sqrt{a^2+b}}{-a + \sqrt{a^2+b}} \approx 10^8.$$

Vgl. **Illustration2_Kondition.**

- **Eigenwertberechnung:** Das Problem *Berechnung der Eigenwerte einer symmetrischen Matrix* ist sehr gut konditioniert. Der Algorithmus basierend auf der Nullstellenberechnung des charakteristischen Polynoms ist instabil, da das Teilproblem *Berechnung der Nullstellen eines Polynoms* höherer Ordnung sehr schlecht konditioniert ist.

Siehe **Illustration1.**

• **Schlußfolgerungen:**

- Die numerische Lösung eines schlecht konditionierten Problems ist nicht oder nur eingeschränkt sinnvoll.

- Da ein einzelnes schlecht konditioniertes Teilproblem das Endergebnis stark verfälschen kann, ist bei der numerischen Lösung eines gut konditionierten Problems darauf zu achten, daß der verwendete Algorithmus gut konditionierte Teilprobleme umfaßt.

2.5. Stabilität

- **Situation:** Anwendung eines Algorithmus $(p^{(1)}, \dots, p^{(k)})$ zur Lösung eines numerischen Problems $p : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$p^{(k)} \circ \dots \circ p^{(1)} \approx p.$$

Numerische Stabilität eines Algorithmus (im Sinne der Rückwärtsanalyse): Änderungen im Endergebnis aufgrund von Rundungsfehlern und Verfahrensfehlern vergleichbar mit *kleinen* Änderungen der Eingabedaten.

Nun: Präzisierung des Begriffes der numerische Stabilität.

- Betrachtung einer Umgebung $U = U_{x,\varepsilon}$ der exakten Eingabedaten (komponentenweise, bzgl. Norm)

$$U = \{x + \xi \in D : |\xi| \leq \varepsilon |x|\} \subset \mathbb{R}^n \quad \text{oder} \quad U = \{x + \xi \in D : \|\xi\| \leq \varepsilon \|x\|\} \subset \mathbb{R}^n.$$

Elemente der zugehörigen Umgebung des exakten Ergebnisses (Bildmenge)

$$p(U) = \{p(x + \xi) : x + \xi \in U\} \subset \mathbb{R}^m$$

(oder nahe bei $p(U)$ liegende Elemente) werden als Näherungslösungen akzeptiert.

Akzeptable Näherungslösung (Definition 2.9): Eine anstelle des exakten Ergebnisses $y = p(x)$ berechnete Näherungslösung $\tilde{y} \approx y$ heißt akzeptabel bezüglich der Eingabemenge U , wenn

$$\tilde{y} \in p(U)$$

oder die abgeschwächten Bedingungen (komponentenweise, bzgl. Norm)

$$\exists z \in p(U) \quad \text{soda\ss} \quad |\tilde{y} - z| \leq \mathcal{O}(\varepsilon_{\text{mach}}) |\tilde{y}| \quad \text{bzw.} \quad \|\tilde{y} - z\| \leq \mathcal{O}(\varepsilon_{\text{mach}}) \|\tilde{y}\|$$

erfüllt sind.

Numerische Stabilität eines Algorithmus (Definition 2.11): Ein Algorithmus zur Lösung eines Problems $p : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ ist numerisch stabil (im Sinne der Rückwärtsanalyse), wenn für alle zulässigen Eingabedaten $x \in D$ die unter dem Einfluß von Rundungsfehlern und Verfahrensfehlern berechneten Näherungslösungen $\tilde{y} \approx y = p(x)$ akzeptabel bezüglich der Eingabemenge $U = U_{x,\varepsilon_{\text{mach}}}$ sind.

Bemerkungen:

- Die elementaren arithmetischen Operationen und das Wurzelziehen sind numerisch stabil.
- Kondition eines numerischen Problems: Signifikanz der berechneten Ergebnisse. Numerische Stabilität eines Algorithmus (im Sinne der Rückwärtsanalyse): Gutartigkeit eines Verfahrens hinsichtlich der Auswirkung von Rundungsfehlern. Je schlechter die Kondition des numerischen Problems ist, desto größere Fehler im Endresultat sind akzeptabel.

• **Lösung eines linearen Gleichungssystems** $Ax = b$:

- Numerisches Problem (unter der vereinfachenden Annahme, daß das lineare Gleichungssystem eindeutig lösbar ist, d.h. A invertierbar)

$$p: D \subset \mathbb{R}^{n \times n} \times \mathbb{R}^n \rightarrow \mathbb{R}^n : (A, b) \mapsto x = A^{-1}b.$$

- Eingabemenge

$$U = U_{(A,b),\varepsilon} = \{(A + \alpha, b + \beta) \in D : |\alpha| \leq \varepsilon |A|, |\beta| \leq \varepsilon |b|\} \subset \mathbb{R}^{n \times n} \times \mathbb{R}^n$$

und zugehörige Bildmenge (Annahme $A + \alpha$ invertierbar, etwa $\|\alpha\| < \|A^{-1}\|^{-1}$, vgl. Satz 4.1)

$$p(U) = \{(A + \alpha)^{-1}(b + \beta) : (A + \alpha, b + \beta) \in U\}.$$

Akzeptable Näherungslösungen (bzgl. U , im strengen Sinn)

$$\tilde{x} = (A + \alpha)^{-1}(b + \beta) \in p(U)$$

bzw. $\tilde{x} = \tilde{A}^{-1}\tilde{b}$ mit $\tilde{A} = A + \alpha$, $|\alpha| \leq \varepsilon |A|$ und $\tilde{b} = b + \beta$, $|\beta| \leq \varepsilon |b|$ bzw.

$$\tilde{A}\tilde{x} = \tilde{b} \quad \text{mit} \quad |\tilde{A} - A| \leq \varepsilon |A| \quad \text{und} \quad |\tilde{b} - b| \leq \varepsilon |b|.$$

- Resultat über die Größe des Residuums beim Einsetzen einer akzeptablen Näherungslösung in $Ax = b$.

Satz von Prager und Oettli (Satz 2.10): Eine Näherungslösung \tilde{x} des linearen Gleichungssystems $Ax = b$ ist akzeptabel (bzgl. U , im strengen Sinn), genau dann wenn das Residuum $r(\tilde{x}) = b - A\tilde{x}$ die folgende Abschätzung erfüllt

$$|A\tilde{x} - b| \leq \varepsilon (|A||\tilde{x}| + |b|).$$

Denn: Einerseits: Falls die Näherungslösung \tilde{x} akzeptabel ist, ergibt sich die Abschätzung (Bezeichnungen \tilde{A} und \tilde{b} wie zuvor, komponentenweise Relation \leq)

$$\begin{aligned} |A\tilde{x} - b| & \underset{\mp \tilde{A}\tilde{x} = -\tilde{A}\tilde{x} + \tilde{b}}{=} |(A - \tilde{A})\tilde{x} + \tilde{b} - b| = |\alpha\tilde{x} + \beta| \leq |\alpha||\tilde{x}| + |\beta| \\ & \leq \varepsilon (|A||\tilde{x}| + |b|). \end{aligned}$$

Andererseits: Es ist zu zeigen, daß für Näherungslösungen \tilde{x} , welche die Abschätzung $|A\tilde{x} - b| \leq \varepsilon (|A||\tilde{x}| + |b|)$ erfüllen, die Relation $\tilde{A}\tilde{x} = \tilde{b}$ mit $|\tilde{A} - A| \leq \varepsilon |A|$ und $|\tilde{b} - b| \leq \varepsilon |b|$ folgt. Für eine Näherungslösung \tilde{x} definiere \tilde{A} und \tilde{b} durch

$$\begin{aligned} \tilde{A} &= (a_{ij} - \text{sign}(\tilde{x}_j) \varepsilon \vartheta_i |a_{ij}|)_{1 \leq i, j \leq n}, & \tilde{b} &= (b_i + \varepsilon \vartheta_i |b_i|)_{1 \leq i \leq n}, \\ r = A\tilde{x} - b, & \quad z = \varepsilon (|A||\tilde{x}| + |b|), & \vartheta_i &= \begin{cases} \frac{r_i}{z_i} & \text{falls } z_i \neq 0 \\ 0 & \text{falls } z_i = 0. \end{cases} \end{aligned}$$

Eine kurze Rechnung zeigt ($\text{sign}(x) x = |x|$)

$$\begin{aligned}\tilde{A}\tilde{x} - \tilde{b} &= \left(\sum_{j=1}^n (a_{ij} - \text{sign}(\tilde{x}_j) \varepsilon \vartheta_i |a_{ij}|) \tilde{x}_j - (b_i + \varepsilon \vartheta_i |b_i|) \right)_{1 \leq i \leq n} \\ &= A\tilde{x} - b - \underbrace{\left(\vartheta_i \varepsilon \sum_{j=1}^n (|a_{ij}| |\tilde{x}_j| + |b_i|) \right)_{1 \leq i \leq n}}_{= z_i} = A\tilde{x} - b - r = 0.\end{aligned}$$

Da nach Voraussetzung die Näherungslösung \tilde{x} die Abschätzung $|r| \leq z$ (komponentenweise) erfüllt, folgt außerdem $|\vartheta_i| \leq 1$ für $1 \leq i \leq n$ und damit

$$\begin{aligned}\tilde{A} - A &= \left(-\text{sign}(\tilde{x}_j) \varepsilon \vartheta_i |a_{ij}| \right)_{1 \leq i, j \leq n}, & |\tilde{A} - A| &\leq \varepsilon |A|, \\ \tilde{b} - b &= \left(\varepsilon \vartheta_i |b_i| \right)_{1 \leq i \leq n}, & |\tilde{b} - b| &\leq \varepsilon |b|.\end{aligned}$$

Damit folgt die Behauptung. \diamond

– **Bemerkungen:**

- * Für eine **genaue Näherungslösung** ist die Differenz zur exakten Lösung des linearen Gleichungssystems $|\tilde{x} - x|$ *klein*.
- * Für eine **akzeptable Näherungslösung** ist das Residuum beim Einsetzen in das lineare Gleichungssystem $|A\tilde{x} - b|$ *klein*.
- * **Zusammenfassung:** Ein Algorithmus zur Lösung eines linearen Gleichungssystems $Ax = b$ (mit $A \in \mathbb{K}^{n \times n}$ invertierbar) ist numerisch stabil (im Sinne der Rückwärtsanalyse), wenn für alle zulässigen Eingabedaten (A, b) die unter dem Einfluß von Rundungsfehlern und Verfahrensfehlern berechneten Näherungslösungen $\tilde{x} \approx x = A^{-1}b$ akzeptabel bezüglich der Eingabemenge $U_{(A,b), \varepsilon_{\text{mach}}}$ sind, d.h. es gilt (mittels Satz von Prager und Oettli)

$$\tilde{x} \in p(U_{(A,b), \varepsilon_{\text{mach}}}) \iff |A\tilde{x} - b| \leq \varepsilon_{\text{mach}} (|A| |\tilde{x}| + |b|).$$

3. Vektoren und Matrizen

- **Inhalte:**

- Grundlegende Begriffe und Resultate der Linearen Algebra, insbesondere zu Vektoren, Matrizen, Skalarprodukt, Orthogonalität, Norm.
- QR-Zerlegung einer Matrix, Orthogonalisierungsverfahren nach Gram–Schmidt und Modifikationen.

3.1. Rechnen mit Vektoren und Matrizen

- Endlich dimensionaler **Vektorraum** bzw. **linearer Raum** \mathbb{K}^n mit $\mathbb{K} = \mathbb{R}, \mathbb{C}$ (übliche Bezeichnungen für Komponenten, identifiziere $x \in \mathbb{K}^n$ mit Spalte $x \in \mathbb{K}^{n \times 1}$, Addition und Skalarmultiplikation)

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{K}^n, \quad x + y = \begin{pmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{pmatrix}, \quad \lambda x = \begin{pmatrix} \lambda x_1 \\ \vdots \\ \lambda x_n \end{pmatrix}, \quad x, y \in \mathbb{K}^n, \quad \lambda \in \mathbb{K}.$$

- Menge der komplexen bzw. reellen **Matrizen** $\mathbb{K}^{m \times n}$ (Anordnung in Schema mit m Zeilen und n Spalten, komponentenweise Addition und Skalarmultiplikation, **quadratische Matrix** für $m = n$)

$$A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \in \mathbb{K}^{m \times n}.$$

Matrizenmultiplikation (assoziativ, nicht kommutativ)

$$C = AB = \left(\sum_{k=1}^m a_{ik} b_{kj} \right)_{1 \leq i \leq \ell, 1 \leq j \leq n} \in \mathbb{K}^{\ell \times n}, \quad A \in \mathbb{K}^{\ell \times m}, \quad B \in \mathbb{K}^{m \times n}$$

$$ABC = (AB)C = A(BC), \quad AB \neq BA. \quad \text{i.A.}$$

- Für $A \in \mathbb{K}^{m \times n}$ ist die zugehörige **lineare Abbildung** (mit Eigenschaften Additivität und Homogenität) gegeben durch

$$f: \mathbb{K}^n \rightarrow \mathbb{K}^m: x \mapsto Ax.$$

Bild der linearen Abbildung bzw. der zugehörigen Matrix

$$\mathcal{B}_A = f(\mathbb{K}^n) = \{y = Ax \in \mathbb{K}^m : x \in \mathbb{K}^n\} \subset \mathbb{K}^m.$$

Nullraum der linearen Abbildung bzw. der zugehörigen Matrix

$$\mathcal{N}_A = f^{-1}(0) = \{x \in \mathbb{K}^n : Ax = 0 \in \mathbb{K}^m\} \subset \mathbb{K}^n.$$

- Lässt sich ein Vektor $w \in \mathbb{K}^n$ in der Form

$$w = \sum_{i=1}^k \lambda_i v_i$$

mit $v_1, \dots, v_k \in \mathbb{K}^n$ und $\lambda_1, \dots, \lambda_k \in \mathbb{K}$ darstellen, heißt w eine **Linearkombination** von v_1, \dots, v_k . Die Menge aller Linearkombinationen von v_1, \dots, v_k bezeichnet man als **lineare Hülle** der Vektoren v_1, \dots, v_k bzw. den von v_1, \dots, v_k **aufgespannten Raum**

$$\langle v_1, \dots, v_k \rangle = \left\{ w = \sum_{i=1}^k \lambda_i v_i \in \mathbb{K}^n : \lambda_1, \dots, \lambda_k \in \mathbb{K} \right\} \subset \mathbb{K}^n.$$

Läßt sich kein Vektor in $\{v_1, \dots, v_k\}$ als Linearkombination der restlichen Vektoren darstellen, d.h. es gilt

$$\sum_{i=1}^k \lambda_i v_i = 0 \implies \lambda_i = 0, \quad 1 \leq i \leq k,$$

heißen v_1, \dots, v_k **linear unabhängig**. Folglich ist auch die Darstellung von $w \in \langle v_1, \dots, v_k \rangle$ als Linearkombination der v_1, \dots, v_k eindeutig.

Sind n Vektoren $v_1, \dots, v_n \in \mathbb{K}^n$ linear unabhängig, so bilden sie eine **Basis** des Vektorraumes \mathbb{K}^n , d.h. jeder Vektor in \mathbb{K}^n läßt sich als Linearkombination der Basisvektoren darstellen (Erzeugendensystem) und die Darstellung ist eindeutig (lineare Unabhängigkeit).

Die **Standardbasisvektoren** bzw. **kanonischen Einheitsvektoren** $e_1, \dots, e_n \in \mathbb{K}^n$ (definiert durch $(e_i)_j = \delta_{ij}$ für $1 \leq i, j \leq n$) bilden eine Basis des Vektorraumes \mathbb{K}^n . Für $x \in \mathbb{K}^n$ folgt direkt die Darstellung als Linearkombination

$$x = \sum_{i=1}^n x_i e_i \in \mathbb{K}^n.$$

Veranschaulichung im $\mathbb{R}^2, \mathbb{R}^3$.

- **Grundlegender Zusammenhang zwischen linearen Gleichungssystemen und Darstellungen als Linearkombinationen:** Angabe und Umformulierung des Matrix-Vektor Produktes $Ax = b$ ergibt (wobei $A \in \mathbb{K}^{m \times n}$, $x \in \mathbb{K}^n$, $b \in \mathbb{K}^m$)

$$Ax = \left(\sum_{k=1}^n a_{ik} x_k \right)_{1 \leq i \leq m} = \sum_{k=1}^n x_k \begin{pmatrix} a_{1k} \\ \vdots \\ a_{ik} \\ \vdots \\ a_{mk} \end{pmatrix} = b,$$

k -te Spalte a_k von A

d.h. die Lösung des linearen Gleichungssystems $Ax = b$ entspricht der Darstellung der rechten Seite b als Linearkombination der Spalten der Matrix A .

Schlußfolgerung: Das lineare Gleichungssystem $Ax = b$ ist genau dann lösbar, wenn sich die rechte Seite b als Linearkombination der Spalten der Matrix A darstellen läßt (somit ist $b \in \mathcal{R}_A$).

- Der **Rang** einer Matrix $A \in \mathbb{K}^{m \times n}$ ist definiert als die Anzahl der linear unabhängigen Spalten (bzw. Zeilen). Es gilt

$$\text{rg}(A) = n - \dim \mathcal{N}_A.$$

Eine Matrix $A \in \mathbb{K}^{m \times n}$ hat **vollen Rang**, wenn $\text{rg}(A) = \min\{m, n\}$.

Falls für eine Matrix $A \in \mathbb{K}^{m \times n}$ die Anzahl der Zeilen größer als die Anzahl der Spalten sind, d.h. es gilt $m \geq n$, sind folgende Aussagen äquivalent:

- Die Matrix $A \in \mathbb{K}^{m \times n}$ hat vollen Rang, d.h. es ist $\text{rg}(A) = n$, d.h. alle Spalten der Matrix sind linear unabhängig.
- Die Abbildung $f: \mathbb{K}^n \rightarrow \mathbb{K}^m: x \mapsto Ax$ ist injektiv.

Denn: Falls alle Spalten der Matrix linear unabhängig sind, folgt

$$Ax = Ay \iff A(x - y) = 0 \iff \sum_{k=1}^n (x_k - y_k) a_k = 0 \iff x = y$$

und damit die Injektivität von f . \diamond

- Der Nullraum von A ist gegeben durch $\mathcal{N}_A = \{0\} \subset \mathbb{K}^n$.

Denn: Ähnlich wie zuvor folgt aus der linearen Unabhängigkeit der Spalten von A

$$Ax = 0 \iff \sum_{k=1}^n x_k a_k = 0 \iff x = 0$$

und damit die Behauptung. \diamond

Für **quadratische Matrizen** gelten insbesondere die folgenden äquivalenten Aussagen:

- Die Matrix $A \in \mathbb{K}^{n \times n}$ hat vollen Rang, d.h. es ist $\text{rg}(A) = n$, d.h. alle Spalten der Matrix sind linear unabhängig.

Die Spalten von A bilden somit eine Basis von \mathbb{K}^n , d.h. jeder Vektor in \mathbb{K}^n lässt sich als Linearkombination der Basisvektoren darstellen und die Darstellung ist eindeutig.

- Die Abbildung $f: \mathbb{K}^n \rightarrow \mathbb{K}^n: x \mapsto Ax$ ist bijektiv.

Die zugehörige inverse Abbildung $f^{-1}: \mathbb{K}^n \rightarrow \mathbb{K}^n: x \mapsto A^{-1}x$ ist durch die **inverse Matrix** (bzw. kurz **Inverse**) $A^{-1} \in \mathbb{K}^{n \times n}$ definiert, und es gilt

$$AA^{-1} = I.$$

- Der Nullraum von A ist gegeben durch $\mathcal{N}_A = \{0\} \subset \mathbb{K}^n$.

Die Inverse einer Matrix $A \in \mathbb{K}^{n \times n}$ erfüllt die Matrix-Gleichung

$$\begin{aligned} AX &= I, \\ (Ax_1 | \dots | Ax_n) &= (e_1 | \dots | e_n), \\ \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{nn} \end{pmatrix} &= \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}, \end{aligned}$$

d.h. die inverse Matrix A^{-1} ergibt sich als Lösung der n linearen Gleichungssysteme

$$Ax_i = e_i, \quad 1 \leq i \leq n, \quad A^{-1} = (x_1 | \dots | x_n).$$

Bemerkung: Für invertierbare Matrizen $A, B \in \mathbb{K}^{n \times n}$ gilt

$$(AB)^{-1} = B^{-1}A^{-1}.$$

- **Eindeutige Lösbarkeit eines linearen Gleichungssystems:** Falls die Matrix $A \in \mathbb{K}^{n \times n}$ invertierbar ist, bilden die Spalten von A eine Basis von \mathbb{K}^n . Folglich ist die Darstellung der rechten Seite b als Linearkombination der Spalten der Matrix A eindeutig und damit die Lösung des linearen Gleichungssystems $Ax = b$ eindeutig bestimmt.
- Vgl. **Illustration3_VektorenMatrizen**.

3.2. Elementare Matrix-Multiplikationen

- **Vorbemerkung:** Die Lösung eines (eindeutig lösbaren) linearen Gleichungssystems $Ax = b$ (mittels Gaußschem Eliminationsverfahren) beruht auf der Transformation der Matrix A auf Dreiecksgestalt. Für theoretische Untersuchungen ist eine Beschreibung der Transformation mittels elementarer Matrix-Multiplikationen vorteilhaft.
- **Elementare Matrix-Umformungen:** Die Multiplikation einer Matrix A von links mit einer elementaren Transformationsmatrix T (d.h. Bildung von TA) wirkt auf die Zeilen der Matrix A .

- **Skalierung** durch Multiplikation mit einer **Diagonalmatrix**

$$D = \text{diag}(d_1, \dots, d_n) = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} \in \mathbb{K}^{n \times n}, \quad d_i \neq 0, \quad 1 \leq i \leq n.$$

Es gilt $D^{-1} = \text{diag}(\frac{1}{d_1}, \dots, \frac{1}{d_n})$.

- **Vertauschung von Zeilen** (oder Spalten) durch Multiplikation mit einer **Permutationsmatrix (Standardmatrix E_{ij})** mit Eintrag 1 bei (i, j) -tem Koeffizienten und ansonsten Einträgen 0)

$$P = I - E_{ii} - E_{jj} + E_{ij} + E_{ji} \in \mathbb{K}^{n \times n}, \quad 1 \leq i, j \leq n, \quad i \neq j.$$

Es gilt $P^{-1} = P = P^T$.

- **Addition des skalaren Vielfachen einer Zeile** (oder Spalte) zu einer anderen Zeile (oder Spalte)

$$N_{ij}(\alpha) = I + \alpha E_{ij} \in \mathbb{K}^{n \times n}, \quad 1 \leq i, j \leq n, \quad i \neq j, \quad \alpha \in \mathbb{K}.$$

Es gilt $N_{ij}(\alpha)^{-1} = N_{ij}(-\alpha)$ und beispielsweise für $n = 3$ (Reihenfolge der Matrizen wesentlich!)

$$N_{21}(\alpha_{21}) N_{31}(\alpha_{31}) N_{32}(\alpha_{32}) = \begin{pmatrix} 1 & & \\ \alpha_{21} & \ddots & \\ \alpha_{31} & \alpha_{32} & 1 \end{pmatrix} \in \mathbb{K}^{3 \times 3}.$$

Vgl. **Illustration3_VektorenMatrizen**.

3.3. Skalarprodukt und Orthogonalität

- Für eine komplexe Zahl $z = \Re z + i\Im z \in \mathbb{C}$ ist die **komplex konjugierte Zahl** gegeben durch $\bar{z} = \Re z - i\Im z \in \mathbb{C}$.
- Für eine reelle oder komplexe Matrix

$$A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \in \mathbb{K}^{m \times n}$$

ist die **transponierte Matrix** (bzw. kurz **Transponierte**) gegeben durch

$$A^T = (a_{ji})_{1 \leq j \leq n, 1 \leq i \leq m} = \begin{pmatrix} a_{11} & \dots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \dots & a_{mn} \end{pmatrix} \in \mathbb{K}^{n \times m}$$

und die **adjungierte Matrix** (bzw. kurz **Adjungierte**) $A^* \in \mathbb{K}^{n \times m}$ gegeben durch

$$A^* = \overline{A}^T = (\overline{a_{ji}})_{1 \leq j \leq n, 1 \leq i \leq m} = \begin{pmatrix} \overline{a_{11}} & \dots & \overline{a_{m1}} \\ \vdots & & \vdots \\ \overline{a_{1n}} & \dots & \overline{a_{mn}} \end{pmatrix} \in \mathbb{K}^{n \times m}.$$

Offensichtlich gilt (wobei $A, B \in \mathbb{K}^{m \times n}$, $C \in \mathbb{K}^{n \times k}$, für letzte Relation verwende Annahme $A \in \mathbb{K}^{n \times n}$ invertierbar und z.B. die Relation $I = (A^{-1}A)^* = A^*(A^{-1})^*$)

$$\begin{aligned} (A+B)^T &= A^T + B^T, & (\lambda A)^T &= \lambda A^T, & (AC)^T &= C^T A^T, & (A^T)^{-1} &= (A^{-1})^T, \\ (A+B)^* &= A^* + B^*, & (\lambda A)^* &= \overline{\lambda} A^*, & (AC)^* &= C^* A^*, & (A^*)^{-1} &= (A^{-1})^*, \end{aligned}$$

und insbesondere $A^* = A^T$ für $A \in \mathbb{R}^{m \times n}$.

- Eine (quadratische) Matrix $A \in \mathbb{K}^{n \times n}$ heißt **symmetrisch**, wenn

$$A^T = A, \quad a_{ji} = a_{ij}, \quad 1 \leq i, j \leq n.$$

Eine (quadratische) Matrix $A \in \mathbb{K}^{n \times n}$ heißt **selbstadjungiert** bzw. **hermitesch**, wenn

$$A^* = A, \quad \overline{a_{ji}} = a_{ij}, \quad 1 \leq i, j \leq n.$$

Offensichtlich ist eine reelle selbstadjungierte Matrix insbesondere symmetrisch.

Eine selbstadjungierte Matrix $A \in \mathbb{K}^{n \times n}$ heißt **positiv semi-definit** oder **positiv definit**, falls für alle $x \in \mathbb{K}^n$ die Bedingung

$$x^* A x \geq 0 \quad \text{oder} \quad x^* A x > 0$$

erfüllt ist. Beachte, daß die quadratische Form

$$q: \mathbb{K}^n \rightarrow \mathbb{R}: x \mapsto x^* A x,$$

$$q(x) = x^* A x = (\overline{x_1} \quad \dots \quad \overline{x_n}) \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \overline{x_i} x_j,$$

aufgrund der geforderten Selbstadjungiertheit von A reelle Werte annimmt, denn es folgt (für $v, w \in \mathbb{K}^n$ ist $v^T w = w^T v$)

$$\overline{x^* A x} \underset{A^*=A}{=} \overline{x^* A^* x} = \overline{(Ax)^* x} \underset{\overline{(Ax)^*} = (Ax)^T}{=} (Ax)^T \overline{x} \underset{(Ax)^T \overline{x} = \overline{x^T (Ax)}}{=} x^* A x \implies x^* A x \in \mathbb{R}.$$

- Das **euklidische Skalarprodukt** ist definiert durch (komplexe Konjugation bzgl. des zweiten Argumentes!)

$$\langle \cdot | \cdot \rangle_2: \mathbb{K}^n \times \mathbb{K}^n \rightarrow \mathbb{K}: (x, y) \mapsto \langle x | y \rangle_2 = y^* x = (\overline{y_1} \quad \dots \quad \overline{y_n}) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^n \overline{y_i} x_i.$$

Die zugehörige **euklidische Norm** ist gegeben durch

$$\| \cdot \|_2: \mathbb{K}^n \rightarrow \mathbb{R}_{\geq 0}: x \mapsto \|x\|_2 = \sqrt{\langle x | x \rangle_2} = \sqrt{\sum_{i=1}^n |x_i|^2}.$$

- Allgemeiner fordert man von einem **Skalarprodukt** bzw. **inneren Produkt** folgende Eigenschaften (**positiv definite hermitesche Sesquilinearform**, insbesondere für $\mathbb{K} = \mathbb{R}$ **positiv definite symmetrische Bilinearform**)

$$\begin{aligned} \langle \cdot | \cdot \rangle: \mathbb{K}^n \times \mathbb{K}^n &\rightarrow \mathbb{K} \\ \langle x + \tilde{x} | y \rangle &= \langle x | y \rangle + \langle \tilde{x} | y \rangle, \quad \langle \lambda x | y \rangle = \lambda \langle x | y \rangle, \quad \lambda \in \mathbb{K}, \quad x, y \in \mathbb{K}^n, \\ \langle y | x \rangle &= \overline{\langle x | y \rangle}, \quad x, y \in \mathbb{K}^n, \\ \langle x | x \rangle &\geq 0, \quad \langle x | x \rangle = 0 \Leftrightarrow x = 0, \quad x \in \mathbb{K}^n. \end{aligned}$$

Die **zugehörige Norm**, definiert durch

$$\| \cdot \|: \mathbb{K}^n \rightarrow \mathbb{R}_{\geq 0}: x \mapsto \|x\| = \sqrt{\langle x | x \rangle},$$

erfüllt die **Ungleichung von Cauchy-Schwarz**

$$|\langle x | y \rangle| \leq \|x\| \|y\|, \quad x, y \in \mathbb{K}^n.$$

Denn: Insbesondere für den Spezialfall $y = 0$ ist die Behauptung klar. Unter der Annahme $y \neq 0$ und $\mathbb{K} = \mathbb{R}$ führt die Minimierung der quadratischen Funktion

$$f: \mathbb{K} \rightarrow \mathbb{R}_{\geq 0}: \lambda \mapsto \|x + \lambda y\|^2 = \langle x + \lambda y | x + \lambda y \rangle = \|x\|^2 + |\lambda|^2 \|y\|^2 + (\overline{\lambda} \langle x | y \rangle + \lambda \langle y | x \rangle)$$

wegen $f(\lambda) = \|x\|^2 + \lambda^2 \|y\|^2 + 2\lambda \langle x|y \rangle$ auf die Ungleichung

$$f'(\lambda_{\min}) = 2\lambda_{\min} \|y\|^2 + 2\langle x|y \rangle = 0, \quad \lambda_{\min} = -\frac{1}{\|y\|^2} \langle x|y \rangle, \quad f''(\lambda_{\min}) = 2\|y\|^2 > 0,$$

$$0 \leq f(\lambda_{\min}) = \|x\|^2 - \frac{1}{\|y\|^2} \langle x|y \rangle^2 \implies |\langle x|y \rangle| \leq \|x\| \|y\|.$$

Für $\mathbb{K} = \mathbb{C}$ ergibt sich die Behauptung direkt durch Einsetzen von $\lambda = -\frac{1}{\|y\|^2} \langle x|y \rangle$

$$0 \leq f\left(-\frac{1}{\|y\|^2} \langle x|y \rangle\right) = \|x\|^2 - \frac{1}{\|y\|^2} |\langle x|y \rangle|^2$$

und Wurzelziehen. \diamond

Allgemein fordert man für eine **Norm** $\|\cdot\|: \mathbb{K}^n \rightarrow \mathbb{R}_{\geq 0}$ die folgenden Eigenschaften (Positive Definitheit, Homogenität, Sub-Additivität bzw. Dreiecksungleichung)

$$\begin{aligned} \|x\| = 0 &\Leftrightarrow x = 0, & x &\in \mathbb{K}^n, \\ \|\lambda x\| &= |\lambda| \|x\|, & \lambda &\in \mathbb{K}, \quad x \in \mathbb{K}^n, \\ \|x + y\| &\leq \|x\| + \|y\|, & x, y &\in \mathbb{K}^n. \end{aligned}$$

Bemerkung: Die durch ein Skalarprodukt definierte Norm erfüllt die Normeigenschaften. **Denn:** Die Eigenschaften positive Definitheit und Homogenität sind leicht nachzuweisen. Mittels der Ungleichung von Cauchy-Schwarz folgt

$$\begin{aligned} \langle x|y \rangle + \langle y|x \rangle &\leq 2\|x\| \|y\| \\ \implies \langle x|x \rangle + \langle x|y \rangle + \langle y|x \rangle + \langle y|y \rangle &\leq \|x\|^2 + 2\|x\| \|y\| + \|y\|^2 \\ \implies \|x + y\|^2 &\leq \|x\|^2 + 2\|x\| \|y\| + \|y\|^2 \\ \implies \|x + y\| &\leq \|x\| + \|y\| \end{aligned}$$

und damit die Dreiecksungleichung. \diamond

- Insbesondere im \mathbb{R}^2 ist der von zwei Vektoren $0 \neq x, y \in \mathbb{R}^2$ eingeschlossene **Winkel** gegeben durch

$$\cos \alpha = \frac{\langle x|y \rangle_2}{\|x\|_2 \|y\|_2}.$$

Speziell für $\|x\|_2 = 1$ und mit der Bezeichnung $r = \|y\|_2$ für die Länge von y ergibt sich

$$\langle x|y \rangle_2 = r \cos \alpha,$$

d.h. $\langle x|y \rangle_2$ beschreibt die Projektion von y auf x und $\langle x|y \rangle_2 x$ die Komponente von y in Richtung x (vgl. Abbildung, Skriptum, S. 42). Falls die beiden Vektoren einen rechten Winkel einschließen, d.h. $\alpha = \frac{\pi}{2}$, folgt $\langle x|y \rangle_2 = 0$.

Zwei Vektoren $x, y \in \mathbb{K}^n$ heißen **orthogonal**, wenn die Bedingung $\langle x|y \rangle = 0$ (bzw. speziell für das euklidische Skalarprodukt $\langle x|y \rangle_2 = y^* x = 0$) erfüllt ist.

Satz von Pythagoras: Für orthonormale Vektoren $z_1, \dots, z_k \in \mathbb{K}^n$ und skalare $c_1, \dots, c_k \in \mathbb{K}$ gilt die folgende Relation

$$\left\| \sum_{i=1}^k c_i z_i \right\|^2 = \left\langle \sum_{i=1}^n c_i z_i \mid \sum_{j=1}^n c_j z_j \right\rangle = \sum_{i,j=1}^n c_i \overline{c_j} \underbrace{\langle z_i \mid z_j \rangle}_{=z_j^* z_i = \delta_{ij}} = \sum_{i=1}^n |c_i|^2.$$

Eine Teilmenge $V = \{v_1, \dots, v_k\} \subset \mathbb{K}^n \setminus \{0\}$ heißt **orthogonal**, wenn je zwei Elemente orthogonal sind, d.h. es gilt $\langle v_i \mid v_j \rangle = 0$ für $1 \leq i, j \leq k$ mit $i \neq j$. Nach **Satz 3.1** sind die Elemente einer orthogonalen Menge linear unabhängig, denn es gilt

$$\sum_{i=1}^k \lambda_i v_i = 0 \quad \Rightarrow \quad \sum_{i=1}^k \lambda_i \underbrace{\langle v_i \mid v_j \rangle}_{=\|v_j\|^2 \delta_{ij}} = 0, \quad 1 \leq j \leq k \quad \stackrel{\|v_j\| \neq 0}{\implies} \quad \lambda_j = 0, \quad 1 \leq j \leq k.$$

Sind außerdem alle Vektoren in V normiert, d.h. $\|v_i\| = 1$ für $1 \leq i \leq k$, heißt die Menge **orthonormal**.

Die Elemente einer orthonormalen Menge $V = \{v_1, \dots, v_n\} \subset \mathbb{K}^n$ bilden eine **Orthonormalbasis** von \mathbb{K}^n , vgl. **Satz 3.1**. Für einen Vektor $y \in \mathbb{K}^n$ gilt dann die folgende Darstellung bezüglich der Orthonormalbasis

$$y = \sum_{i=1}^n \langle y \mid v_i \rangle v_i.$$

Denn: Da die Vektoren in V eine Basis von \mathbb{K}^n bilden, läßt sich $y \in \mathbb{K}^n$ als Linearkombination von v_1, \dots, v_n darstellen

$$y = \sum_{i=1}^n \lambda_i v_i.$$

Durch Bilden des Skalarproduktes mit den Basisvektoren folgt

$$\langle y \mid v_j \rangle = \sum_{i=1}^n \lambda_i \underbrace{\langle v_i \mid v_j \rangle}_{=\delta_{ij}} = \lambda_j, \quad 1 \leq j \leq n \quad \implies \quad \lambda_j = \langle y \mid v_j \rangle, \quad 1 \leq j \leq n,$$

und damit die angegebene Darstellung. \diamond

Zwei Teilmengen $X, Y \subset \mathbb{K}^n$ heißen **orthogonal** zueinander, wenn für jedes Element $x \in X$ und jedes Element $y \in Y$ die Bedingung $\langle x \mid y \rangle = 0$ gilt.

- Eine quadratische Matrix $Q \in \mathbb{K}^{n \times n}$, deren Spalten orthonormal sind, heißt eine **unitäre Matrix** (oder speziell **orthonormale Matrix** für $\mathbb{K} = \mathbb{R}$).

Folglich gilt (Produkt der i -ten Zeile mit der j -ten Spalte von Q ergibt Kronecker-Delta δ_{ij} , d.h. Wert 1 falls $i = j$ und ansonsten Wert 0)

$$Q^* Q = I \quad \text{bzw.} \quad Q^{-1} = Q^*.$$

Die durch eine unitäre Matrix definierte lineare Abbildung $f : \mathbb{K}^n \rightarrow \mathbb{K}^n : x \mapsto Qx$ ist **längenerhaltend**, d.h. es gilt

$$\|Qx\|_2 = \|x\|_2$$

wegen $\|Qx\|_2^2 = x^* Q^* Qx = x^* x = \|x\|_2^2$.

- **Zusammenhang zwischen linearen Gleichungssystemen und Darstellungen als Linearkombinationen:** Die Darstellung eines Vektors $b \in \mathbb{K}^n$ als Linearkombination orthogonaler Vektoren q_1, \dots, q_k entspricht der Lösung eines linearen Gleichungssystems mit unitärer Matrix

$$Qx = b \iff b = \sum_{i=1}^n x_i q_i, \quad Q = (q_1 | \dots | q_n).$$

Die Lösung erhält man in diesem Fall auf einfache Weise durch Multiplikation mit der adjungierten Matrix

$$x = Q^* b = \begin{pmatrix} \overline{q_1} \\ \vdots \\ \overline{q_n} \end{pmatrix} b = \begin{pmatrix} \overline{q_1} b \\ \vdots \\ \overline{q_n} b \end{pmatrix}.$$

3.4. Orthogonalisierungsverfahren nach Gram–Schmidt

- **Vorbemerkungen:**

- Die Lösung eines linearen Gleichungssystems

$$Ax = b, \quad A = (a_1 | \dots | a_n) \in \mathbb{K}^{n \times n}, \quad x \in \mathbb{K}^n, \quad b \in \mathbb{K}^n,$$

entspricht der Darstellung der rechten Seite b als Linearkombination der Spaltenvektoren a_1, \dots, a_n .

- Besonders einfach ist die Lösung eines linearen Gleichungssystems

$$Qy = c, \quad Q = (q_1 | \dots | q_n) \in \mathbb{K}^{n \times n}, \quad y \in \mathbb{K}^n, \quad c \in \mathbb{K}^n,$$

mit unitärer Matrix Q , d.h. die Spaltenvektoren q_1, \dots, q_n bilden eine Orthonormalbasis des Vektorraumes \mathbb{K}^n . In diesem Fall ist das Gleichungssystem eindeutig lösbar und man erhält die Lösung durch Multiplikation mit der adjungierten Matrix

$$y = Q^* c, \quad Q^* = \begin{pmatrix} \overline{q_1} \\ \vdots \\ \overline{q_n} \end{pmatrix}.$$

- Formulierung in Hinblick auf die Lösung **überbestimmter linearer Gleichungssysteme**

$$Ax = b, \quad A = (a_1 | \dots | a_n) \in \mathbb{K}^{m \times n}, \quad x \in \mathbb{K}^n, \quad b \in \mathbb{K}^m, \quad m \geq n.$$

Zusätzliche Annahme, daß die Matrix A vollen Rang hat, d.h. die Spaltenvektoren a_1, \dots, a_n sind linear unabhängig.

- **Problemstellung:** Zu linear unabhängigen Vektoren $a_1, \dots, a_n \in \mathbb{K}^m$ finde orthonormale Vektoren $q_1, \dots, q_n \in \mathbb{K}^m$ so, daß die Vektoren a_1, \dots, a_k und q_1, \dots, q_k für $1 \leq k \leq n$ denselben Raum aufspannen, d.h. es gelte

$$\langle q_1, \dots, q_k \rangle = \langle a_1, \dots, a_k \rangle, \quad 1 \leq k \leq n.$$

Dies ist äquivalent zur Berechnung der **reduzierten QR-Zerlegung** der Matrix A

$$\begin{aligned} A &= \widehat{Q} \widehat{R}, \\ A &= (a_1 | \dots | a_n) \in \mathbb{K}^{m \times n}, \\ \widehat{Q} &= (q_1 | \dots | q_n) \in \mathbb{K}^{m \times n}, \\ \widehat{R} &= \begin{pmatrix} r_{11} & \dots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix} \in \mathbb{K}^{n \times n}, \quad r_{ii} \neq 0, \quad 1 \leq i \leq n. \end{aligned}$$

Denn: Durch Bestimmung des Produktes $\widehat{Q}\widehat{R}$ erhält man wegen $r_{kj} = 0$ für $k \geq j + 1$

$$A = \widehat{Q}\widehat{R},$$

$$a_{ij} = \sum_{k=1}^n q_{ik}r_{kj} = \sum_{k=1}^j q_{ik}r_{kj}, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n,$$

$$a_j = \sum_{k=1}^j r_{kj}q_k, \quad 1 \leq j \leq n,$$

$$a_1 = r_{11}q_1, \quad a_2 = r_{12}q_1 + r_{22}q_2, \quad \dots, \quad a_n = r_{1n}q_1 + \dots + r_{n-1,n}q_{n-1} + r_{nn}q_n.$$

Die zeigt, daß $a_1, \dots, a_k \in \langle q_1, \dots, q_k \rangle$ für $1 \leq k \leq n$. Andererseits ist wegen $r_{ii} \neq 0$ für $1 \leq i \leq n$ die Matrix \widehat{R} invertierbar und deshalb $A = \widehat{Q}\widehat{R}$ äquivalent zu $\widehat{Q} = A\widehat{R}^{-1}$ (die Inverse von R wird nicht explizit berechnet, Einsetzen der bereits bestimmten Spalten von \widehat{Q}). Sukzessives Auflösen der obigen Relationen führt auf

$$q_j = \frac{1}{r_{jj}} \left(a_j - \sum_{i=1}^{j-1} r_{ij}q_i \right), \quad 1 \leq j \leq n,$$

$$q_1 = \frac{1}{r_{11}} a_1, \quad q_2 = \frac{1}{r_{22}} (a_2 - r_{12}q_1), \quad \text{etc.}, \quad q_n = \frac{1}{r_{nn}} (a_n - r_{1n}q_1 - \dots - r_{n-1,n}q_{n-1}),$$

was zeigt, daß auch $q_1, \dots, q_k \in \langle a_1, \dots, a_k \rangle$ für $1 \leq k \leq n$. \diamond

Bemerkung: Als (volle) **QR-Zerlegung** einer Matrix $A \in \mathbb{K}^{m \times n}$ bezeichnet man die Darstellung

$$A = QR,$$

$$A = (a_1 | \dots | a_n) \in \mathbb{K}^{m \times n},$$

$$Q = (\widehat{Q} | q_{n+1} | \dots | q_m) \in \mathbb{K}^{m \times m}, \quad R = \begin{pmatrix} \widehat{R} \\ 0 \end{pmatrix} \in \mathbb{K}^{m \times n},$$

wobei die Matrix \widehat{Q} um $m - n$ orthonormale Spalten q_{n+1}, \dots, q_m zu einer unitären Matrix $Q \in \mathbb{K}^{m \times m}$ ergänzt wird und die Matrix \widehat{R} um $m - n$ Nullzeilen zur Matrix $R \in \mathbb{K}^{m \times n}$ ergänzt wird.

- Das **Orthogonalisierungsverfahren nach Gram-Schmidt** ist ein Verfahren zur Berechnung der reduzierten QR-Zerlegung einer Matrix $A = (a_1 | \dots | a_n) \in \mathbb{K}^{m \times n}$, d.h. orthonormalen Vektoren $q_1, \dots, q_n \in \mathbb{K}^m$ mit

$$q_k = \frac{1}{r_{kk}} \left(a_k - \sum_{i=1}^{k-1} r_{ik}q_i \right), \quad 1 \leq k \leq n,$$

wobei $r_{ij} \in \mathbb{K}$ für $1 \leq i, j \leq n$ mit $r_{ij} = 0$ für $i \geq j + 1$ und $r_{ii} \neq 0$ für $1 \leq i \leq n$.

Bemerkung: Wähle im Folgenden $\langle \cdot | \cdot \rangle = \langle \cdot | \cdot \rangle_2$ und $\| \cdot \| = \| \cdot \|_2$.

Herleitung des Verfahrens:

- Betrachte die erste Relation $q_1 = \frac{1}{r_{11}} a_1$ mit unbekanntem Koeffizienten r_{11} . Die Forderung $\|q_1\| = 1$ impliziert $|r_{11}| = \|a_1\|$. Setze etwa $r_{11} = \|a_1\|$.

- Betrachte die zweite Relation $q_2 = \frac{1}{r_{22}}(a_2 - r_{12}q_1)$ mit unbekanntem Koeffizienten r_{12} und r_{22} . Die Forderungen $\langle q_2 | q_1 \rangle = 0$ und $\|q_2\| = 1$ implizieren

$$0 = \langle q_2 | q_1 \rangle = \left\langle \frac{1}{r_{22}}(a_2 - r_{12}q_1) | q_1 \right\rangle = \frac{1}{r_{22}}(\langle a_2 | q_1 \rangle - r_{12}) \implies_{r_{22} \neq 0} r_{12} = \langle a_2 | q_1 \rangle,$$

$$\|q_2\| = \frac{1}{|r_{22}|} \|a_2 - r_{12}q_1\| = 1 \implies |r_{22}| = \|\tilde{q}_2\|, \quad \tilde{q}_2 = a_2 - r_{12}q_1.$$

Setze etwa $r_{22} = \|\tilde{q}_2\|$.

- Betrachte allgemein die k -te Relation (beinhaltet auch den Spezialfall $k = 1$)

$$q_k = \frac{1}{r_{kk}} \left(a_k - \sum_{i=1}^{k-1} r_{ik} q_i \right), \quad 1 \leq k \leq n,$$

mit unbekanntem Koeffizienten r_{jk} für $1 \leq j \leq k$. Die Forderungen $\langle q_k | q_j \rangle = 0$ für $1 \leq j \leq k-1$ und $\|q_k\| = 1$ implizieren

$$0 = \langle q_k | q_j \rangle = \left\langle \frac{1}{r_{kk}} \left(a_k - \sum_{i=1}^{k-1} r_{ik} q_i \right) | q_j \right\rangle = \frac{1}{r_{kk}} (\langle a_k | q_j \rangle - r_{jk}) \implies_{r_{kk} \neq 0} r_{jk} = \langle a_k | q_j \rangle,$$

$$\|q_k\| = \frac{1}{|r_{kk}|} \left\| a_k - \sum_{i=1}^{k-1} r_{ik} q_i \right\| \implies |r_{kk}| = \|\tilde{q}_k\|, \quad \tilde{q}_k = a_k - \sum_{i=1}^{k-1} r_{ik} q_i.$$

Setze etwa $r_{kk} = \|\tilde{q}_k\|$.

Die obigen Überlegungen führen auf folgendes Resultat.

Existenz und Eindeutigkeit der QR-Zerlegung einer Matrix (Satz 3.2): Für jede Matrix $A = (a_1 | \dots | a_n) \in \mathbb{K}^{m \times n}$ von vollem Rang existiert die reduzierte QR-Zerlegung

$$A = \hat{Q} \hat{R},$$

$$\hat{Q} = (q_1 | \dots | q_n) \in \mathbb{K}^{m \times n},$$

$$\hat{R} = \begin{pmatrix} r_{11} & \dots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix} \in \mathbb{K}^{n \times n}, \quad r_{ii} \neq 0, \quad 1 \leq i \leq n,$$

mit orthonormalen Vektoren $q_1, \dots, q_n \in \mathbb{K}^m$. Durch die zusätzliche Forderung $r_{ii} > 0$ für $1 \leq i \leq n$ ist die reduzierte QR-Zerlegung eindeutig bestimmt.

Das **klassische Orthogonalisierungsverfahren nach Gram-Schmidt** zur Berechnung der reduzierten QR-Zerlegung einer Matrix

$$r_{jk} = \langle a_k | q_j \rangle, \quad 1 \leq j \leq k-1,$$

$$\tilde{q}_k = a_k - \sum_{j=1}^{k-1} r_{jk} q_j, \quad r_{kk} = \|\tilde{q}_k\|, \quad q_k = \frac{1}{r_{kk}} \tilde{q}_k, \quad 1 \leq k \leq n,$$

als Pseudo-Code formuliert lautet beispielsweise folgendermaßen

```

Eingabedaten:  $a_k$  für  $1 \leq k \leq n$ 
for k=1:n
   $\tilde{q} = a_k$ 
  for j=1:k-1
     $r_{jk} = \langle a_k | q_j \rangle$ 
     $\tilde{q} = \tilde{q} - r_{jk} q_j$ 
  end
   $r_{kk} = \|\tilde{q}\|$ 
   $q_k = \frac{1}{r_{kk}} \tilde{q}$ 
end
Ergebnisse:  $q_k, r_{jk}$  für  $1 \leq j \leq k$  und  $1 \leq k \leq n$ 

```

Vgl. **Illustration3_OrthogonalisierungGramSchmidt**.

Die **geometrische Veranschaulichung** des Orthogonalisierungsverfahrens nach Gram-Schmidt ist, daß vom Basisvektor a_k schrittweise die Anteile $r_{jk} q_j = \langle a_k | q_j \rangle q_j$ (Erinnerung: $\langle a_k | q_j \rangle$ ist die Projektion von a_k auf q_j und $\langle a_k | q_j \rangle q_j$ die Komponente von a_k in Richtung von q_j) subtrahiert werden. Der resultierende Vektor \tilde{q}_k steht dann orthogonal (senkrecht) auf q_1, \dots, q_{k-1} bzw. die lineare Hülle $\langle q_1, \dots, q_{k-1} \rangle$. Durch Normierung (Einheitslänge) erhält man den orthonormalen Vektor q_k .

Vorsicht! Speziell für zwei Vektoren $a_1, a_2 \in \mathbb{K}^m$ führt das klassische Orthogonalisierungsverfahren nach Gram-Schmidt auf

$$q_1 = \frac{1}{\|a_1\|} a_1, \quad \tilde{q}_2 = a_2 - \langle a_2 | q_1 \rangle q_1 = a_2 - \frac{1}{\|a_1\|^2} \langle a_2 | a_1 \rangle a_1, \quad q_2 = \frac{1}{\|\tilde{q}_2\|} \tilde{q}_2.$$

Falls die beiden Vektoren a_1, a_2 **fast linear abhängig** sind, d.h. es ist $a_2 = c a_1 + \delta$ mit $\delta \in \mathbb{K}^m$ und $\|\delta\|$ *klein*, folgt

$$\tilde{q}_2 = \delta - \frac{1}{\|a_1\|^2} \langle \delta | a_1 \rangle a_1,$$

d.h. es ist insbesondere $\|\tilde{q}_2\|$ *klein* und damit der bei q_2 auftretende Faktor $\frac{1}{\|\tilde{q}_2\|}$ *groß*. Relative Rundungsfehler bei der Berechnung von \tilde{q}_2 werden deshalb bei der Berechnung von q_2 erheblich verstärkt und der Algorithmus ist **numerisch instabil**.

Betrachte das **Beispiel von Läuchli** mit

$$A = \begin{pmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{pmatrix} \in \mathbb{R}^{4 \times 3},$$

vgl. **Illustration3_OrthogonalisierungGramSchmidt**.

- Ein **modifiziertes Orthogonalisierungsverfahren nach Gram–Schmidt**

Eingabedaten: a_k für $1 \leq k \leq n$
 for k=1:n
 $\tilde{q}_k = a_k$
 end
 for k=1:n
 $r_{kk} = \|\tilde{q}_k\|$
 $q_k = \frac{1}{r_{kk}} \tilde{q}_k$
 for j=k+1:n
 $r_{kj} = \langle a_j | q_k \rangle$
 $\tilde{q}_j = \tilde{q}_j - r_{kj} q_k$
 end
 end
 Ergebnisse: q_k, r_{jk} für $1 \leq j \leq k$ und $1 \leq k \leq n$

beruht auf einer zeilenweisen Berechnung der Einträge der Matrix R (anstelle einer spaltenweisen Berechnung).

Eine zweite **Modifikation** lautet

Eingabedaten: a_k für $1 \leq k \leq n$
 for k=1:n
 $\tilde{q} = a_k$
 for j=1:k-1
 $r_{jk} = \langle \tilde{q} | q_j \rangle$
 $\tilde{q} = \tilde{q} - r_{jk} q_j$
 end
 $r_{kk} = \|\tilde{q}\|$
 $q_k = \frac{1}{r_{kk}} \tilde{q}$
 end
 Ergebnisse: q_k, r_{jk} für $1 \leq j \leq k$ und $1 \leq k \leq n$

Bemerkungen:

- Das Orthogonalisierungsverfahren nach Gram–Schmidt und auch beide Modifikationen sind in gewissen Situation numerisch instabil, vgl. Beispiel von Läuchli. Für die Modifikationen kann man zeigen, daß unter dem Einfluß von Rundungsfehlern Matrizen \tilde{Q} (im Allgemeinen nicht unitär!) und \tilde{R} (obere Dreiecksmatrix)

berechnet werden, die von den exakten Matrizen Q und R der QR -Zerlegung einer Matrix $A \in \mathbb{K}^{n \times n}$ folgendermaßen abweichen

$$\begin{aligned} \|\tilde{Q}\tilde{R} - A\|_2 &\leq C_1(n) \varepsilon_{\text{mach}} \|A\|_2, \\ \|\tilde{Q}^* \tilde{Q} - I\|_2 &\leq C_2(n) \varepsilon_{\text{mach}} \kappa_2(A) + \mathcal{O}(\varepsilon_{\text{mach}}^2 \kappa_2(A)^2). \end{aligned}$$

Für eine *fast* singuläre Matrix A ist die Konditionzahl $\kappa_2(A)$ *groß*. Vgl. Bemerkung im Skriptum, S. 49 sowie **Beispiel von Lächli**.

Orthogonalisierungsverfahren und Modifikationen werden aber dennoch im Zusammenhang mit iterativen Verfahren (Arnoldi, GMRES) angewendet.

- Eine numerisch stabile Modifikation des Orthogonalisierungsverfahren nach Gram–Schmidt verwendet zusätzlich die Idee der **Re-Orthogonalisierung**. Eine weitere stabile Alternative der QR -Zerlegung einer Matrix mittels Householder-Reflexionen wird in Abschnitt 4.2 besprochen.
- Im **Vergleich** mit dem Orthogonalisierungsverfahren nach Gram–Schmidt speziell für den Fall $n = 3$

$$k = 1: \quad r_{11} = \|a_1\|, \quad q_1 = \frac{1}{r_{11}} \|a_1\|$$

$$k = 2: \quad \tilde{q} = a_2,$$

$$j = 1: \quad r_{12} = \langle a_2 | q_1 \rangle, \quad \tilde{q} = a_2 - r_{12} q_1, \quad r_{22} = \|\tilde{q}\|, \quad q_2 = \frac{1}{r_{22}} \|\tilde{q}\|$$

$$k = 3: \quad \tilde{q} = a_3,$$

$$j = 1: \quad r_{13} = \langle a_3 | q_1 \rangle, \quad \tilde{q} = a_3 - r_{13} q_1,$$

$$j = 2: \quad \mathbf{r}_{23} = \langle \mathbf{a}_3 | \mathbf{q}_2 \rangle, \quad \tilde{q} = a_3 - r_{13} q_1 - r_{23} q_2, \quad r_{33} = \|\tilde{q}\|, \quad q_3 = \frac{1}{r_{33}} \|\tilde{q}\|$$

führt die erste Modifikation auf

$$k = 1: \quad r_{11} = \|a_1\|, \quad q_1 = \frac{1}{r_{11}} \|a_1\|$$

$$j = 2: \quad r_{12} = \langle a_2 | q_1 \rangle, \quad \tilde{q}_2 = a_2 - r_{12} q_1$$

$$j = 3: \quad r_{13} = \langle a_3 | q_1 \rangle, \quad \tilde{q}_3 = a_3 - r_{13} q_1$$

$$k = 2: \quad r_{22} = \|\tilde{q}_2\|, \quad q_2 = \frac{1}{r_{22}} \|\tilde{q}_2\|$$

$$j = 3: \quad r_{23} = \langle a_3 | q_2 \rangle, \quad \tilde{q}_3 = a_3 - r_{13} q_1 - r_{23} q_2$$

$$k = 3: \quad r_{33} = \|\tilde{q}_3\|, \quad q_3 = \frac{1}{r_{33}} \|\tilde{q}_3\|$$

und die zweite Modifikation auf

$$k = 1: \quad r_{11} = \|a_1\|, \quad q_1 = \frac{1}{r_{11}} \|a_1\|$$

$$k = 2: \quad \tilde{q} = a_2$$

$$j = 1: \quad r_{12} = \langle a_2 | q_1 \rangle, \quad \tilde{q} = a_2 - r_{12} q_1, \quad r_{22} = \|\tilde{q}\|, \quad q_2 = \frac{1}{r_{22}} \|\tilde{q}\|$$

$$k = 3: \quad \tilde{q} = a_3$$

$$j = 1: \quad r_{13} = \langle \tilde{q} | q_1 \rangle, \quad \tilde{q} = \tilde{q} - r_{13} q_1$$

$$j = 2: \quad \mathbf{r}_{23} = \langle \tilde{q} | \mathbf{q}_2 \rangle = \langle \mathbf{a}_3 - \mathbf{r}_{13} \mathbf{q}_1 | \mathbf{q}_2 \rangle, \quad \tilde{q} = \tilde{q} - r_{23} q_2$$

$$r_{33} = \|\tilde{q}\|, \quad q_3 = \frac{1}{r_{33}} \|\tilde{q}\|$$

3.5. Normen für Vektoren und Matrizen

- Für Vektoren $x \in \mathbb{K}^n$ und Matrizen $A \in \mathbb{K}^{m \times n}$ sind Beträge sowie die arithmetischen Vergleiche **komponentenweise** zu verstehen, d.h. man definiert

$$|x| = \begin{pmatrix} |x_1| \\ \vdots \\ |x_n| \end{pmatrix}, \quad |A| = \begin{pmatrix} |a_{11}| & \dots & |a_{1n}| \\ \vdots & & \vdots \\ |a_{m1}| & \dots & |a_{mn}| \end{pmatrix},$$

$$A \leq B \quad \text{für } A, B \in \mathbb{K}^{m \times n} \iff a_{ij} \leq b_{ij} \quad \text{für alle } 1 \leq i \leq m, \quad 1 \leq j \leq n.$$

- Erinnerung:** Eine **Norm** auf einem Vektorraum \mathbb{K}^n ist eine Funktion $\|\cdot\| : \mathbb{K}^n \rightarrow \mathbb{R}_{\geq 0}$, welche die folgenden Eigenschaften erfüllt (positive Definitheit, Homogenität, Sub-Additivität)

$$\begin{aligned} \|x\| = 0 &\iff x = 0, & x \in \mathbb{K}^n, \\ \|\lambda x\| &= |\lambda| \|x\|, & \lambda \in \mathbb{K}, \quad x \in \mathbb{K}^n, \\ \|x + y\| &\leq \|x\| + \|y\|, & x, y \in \mathbb{K}^n. \end{aligned}$$

Neben der **euklidischen Norm** $\|\cdot\|_2$ (definiert durch das euklidische Skalarprodukt) werden üblicherweise die **Betragssummennorm** $\|\cdot\|_1$ und die **Maximumsnorm** $\|\cdot\|_\infty$ verwendet

$$\|x\|_1 = \sum_{i=1}^n |x_i|, \quad \|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}, \quad \|x\|_\infty = \max\{|x_i| : 1 \leq i \leq n\}.$$

Vgl. Definition 3.3 und Veranschaulichungen zum **Abstand** zweier Vektoren im \mathbb{R}^2 und der **Normkugel** im \mathbb{R}^2 (Skriptum, S. 51)

$$U = U_{0,1} = \{x \in \mathbb{K}^n : \|x\| \leq 1\}.$$

Bemerkungen:

- Für die Summen- und Maximumsnorm sind die Normeigenschaften leicht nachzuweisen (Eigenschaften des Betrages).
- Für die euklidische Norm nützt man den Zusammenhang mit dem euklidischen Skalarprodukt (vgl. früher) und insbesondere die Ungleichung von Cauchy-Schwarz zum Beweis der Dreiecksungleichung.
- Speziell für die euklidische Norm und $\mathbb{K} = \mathbb{R}$ lautet die Ungleichung von Cauchy-Schwarz

$$\left(\sum_{i=1}^n x_i y_i\right)^2 \leq \left(\sum_{i=1}^n x_i^2\right) \left(\sum_{i=1}^n y_i^2\right), \quad x, y \in \mathbb{R}^n.$$

Ein direkter Nachweis beruht etwa auf der **arithmetischen-geometrischen Mittelungleichung**

$$\left(\prod_{i=1}^n x_i\right)^{\frac{1}{n}} \leq \frac{1}{n} \sum_{i=1}^n x_i, \quad x \in \mathbb{R}^n.$$

- Der Begriff der Norm lässt sich direkt auf allgemeine Vektorräume V übertragen (Funktion $\|\cdot\| : V \rightarrow \mathbb{R}_{\geq 0}$ mit den Eigenschaften positive Definitheit, Homogenität und Sub-Additivität).

Zwei Normen $\|\cdot\|, \|\|\cdot\|\| : V \rightarrow \mathbb{R}_{\geq 0}$ auf einem Vektorraum heißen **äquivalent**, wenn Konstanten $c_1, c_2 > 0$ existieren so, daß für alle $x \in V$ die folgende Relation gilt

$$c_1 \|x\| \leq \|\|x\|\| \leq c_2 \|x\|.$$

In endlichdimensionalen Vektorräumen und insbesondere im \mathbb{K}^n sind alle Normen äquivalent (im Gegensatz zu unendlichdimensionalen Vektorräumen).

Beispielsweise gilt

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty.$$

Denn: Der Index $1 \leq \ell \leq n$ sei so gewählt, daß $\|x\|_\infty = \max\{|x_i| : 1 \leq i \leq n\} = |x_\ell|$. Dann folgt einerseits

$$\|x\|_\infty^2 = |x_\ell|^2 \leq \sum_{i=1}^{\ell-1} |x_i|^2 + |x_\ell|^2 + \sum_{i=\ell+1}^n |x_i|^2 = \|x\|_2^2$$

und andererseits

$$\|x\|_2^2 = \sum_{i=1}^n |x_i|^2 \leq \sum_{i=1}^n |x_\ell|^2 = n \|x\|_\infty^2.$$

Durch Wurzelziehen folgt die Behauptung. \diamond

- Für $A \in \mathbb{K}^{m \times n}$ betrachte die lineare Abbildung $f : (\mathbb{K}^n, \|\cdot\|_{\mathbb{K}^n}) \rightarrow (\mathbb{K}^m, \|\cdot\|_{\mathbb{K}^m}) : x \mapsto Ax$. Die zugehörige **Operatornorm** $\|\cdot\|_{\mathbb{K}^m \leftarrow \mathbb{K}^n} : \mathbb{K}^{m \times n} \rightarrow \mathbb{R}_{\geq 0}$ ist definiert durch

$$\|A\|_{\mathbb{K}^m \leftarrow \mathbb{K}^n} = \max_{0 \neq x \in \mathbb{K}^n} \frac{\|Ax\|_{\mathbb{K}^m}}{\|x\|_{\mathbb{K}^n}} = \max_{\|x\|_{\mathbb{K}^n}=1} \|Ax\|_{\mathbb{K}^m},$$

vgl. Definition 3.4.

Bemerkungen:

- Nach Festlegung der betrachteten Normen schreibt man kurz $\|A\|$.
- Wegen der Linearität von A und der Homogenität der Norm gilt $\frac{1}{\|x\|} \|Ax\| = \|A \frac{x}{\|x\|}\|$ für $x \neq 0$ und deshalb die angegebene Identität bei der Definition der Norm.
- Für die Identität $I : \mathbb{K}^n \rightarrow \mathbb{K}^n$ ergibt sich die Operatornorm $\|I\| = 1$.

In Analogie zu den Eigenschaften einer Norm auf \mathbb{K}^n erfüllt eine Operatornorm $\|\cdot\| : \mathbb{K}^{m \times n} \rightarrow \mathbb{R}_{\geq 0}$ folgende **Eigenschaften** (positive Definitheit, Homogenität, Sub-Additivität bzw. Dreiecksungleichung, Sub-Multiplikativität, Konsistenz)

$$\begin{aligned} \|A\| = 0 &\Leftrightarrow A = 0, & A \in \mathbb{K}^{m \times n}, \\ \|\lambda A\| &= |\lambda| \|A\|, & \lambda \in \mathbb{K}, \quad A \in \mathbb{K}^{m \times n}, \\ \|A + B\| &\leq \|A\| + \|B\|, & A, B \in \mathbb{K}^{m \times n}, \\ \|AB\| &\leq \|A\| \|B\|, & A \in \mathbb{K}^{\ell \times m}, \quad B \in \mathbb{K}^{m \times n}, \\ \|Ax\| &\leq \|A\| \|x\|, & A \in \mathbb{K}^{m \times n}, \quad x \in \mathbb{K}^n. \end{aligned}$$

Denn: Die Eigenschaften positive Definitheit, Homogenität sowie Sub-Additivität folgen aus den Eigenschaften der Norm auf \mathbb{K}^m ($\|A\| = 0 \Leftrightarrow \|Ax\| = 0$ für alle Vektoren $x \in \mathbb{K}^n \Leftrightarrow A = 0$, $\|\lambda Ax\| = |\lambda| \|Ax\|$, $\|(A+B)x\| \leq \|Ax\| + \|Bx\|$). Die Eigenschaft Sub-Multiplikativität erhält man aus

$$\begin{aligned} \|AB\| &= \max_{0 \neq x \in \mathbb{K}^n} \frac{\|ABx\|}{\|x\|} = \max_{0 \neq Bx \in \mathbb{K}^m} \frac{\|ABx\|}{\|x\|} = \max_{0 \neq Bx \in \mathbb{K}^m} \frac{\|ABx\|}{\|Bx\|} \frac{\|Bx\|}{\|x\|} \\ &\leq \max_{0 \neq Bx \in \mathbb{K}^m} \frac{\|ABx\|}{\|Bx\|} \max_{0 \neq x \in \mathbb{K}^n} \frac{\|Bx\|}{\|x\|} \leq \max_{0 \neq y \in \mathbb{K}^m} \frac{\|Ay\|}{\|y\|} \max_{0 \neq x \in \mathbb{K}^n} \frac{\|Bx\|}{\|x\|} = \|A\| \|B\|. \end{aligned}$$

Für $x = 0$ ist die Eigenschaft Konsistenz offensichtlich und wegen

$$\frac{\|Ax\|}{\|x\|} \leq \max_{0 \neq x \in \mathbb{K}^n} \frac{\|Ax\|}{\|x\|} = \|A\| \quad \Rightarrow \quad \|Ax\| \leq \|A\| \|x\|$$

ergibt sich die Behauptung für $0 \neq x \in \mathbb{K}^n$. \diamond

- **Berechnung von Operatornormen** (Satz 3.6): Für die Betragssummennorm, euklidische Norm und Maximumsnorm ist die zugehörige Operatornorm einer Matrix $A \in \mathbb{K}^{m \times n}$ gegeben durch die größte Spaltenbetragssumme, die Wurzel des maximalen Eigenwertes von $A^*A \in \mathbb{K}^{n \times n}$ und die größte Zeilenbetragssumme

$$\begin{aligned} \|A\|_1 &= \max \left\{ \sum_{i=1}^m |a_{ij}| : 1 \leq j \leq n \right\}, \\ \|A\|_2 &= \max \left\{ \sqrt{\lambda} : \lambda \text{ Eigenwert von } A^*A \right\}, \\ \|A\|_\infty &= \max \left\{ \sum_{j=1}^n |a_{ij}| : 1 \leq i \leq m \right\}. \end{aligned}$$

Denn (Nachweis der Relation für $\|A\|_2$): Beachte, daß die Matrix

$$B = A^*A \in \mathbb{K}^{n \times n}$$

positiv semi-definit ist ($x^*Bx = \|Ax\|_2^2 \geq 0$ für alle $x \in \mathbb{K}^n$) und folglich alle Eigenwerte nicht-negativ sind (jeder Eigenwert $\lambda \in \mathbb{K}$ mit zugehörigem Eigenvektor $v \in \mathbb{K}^n$ erfüllt $0 \leq v^*Bv = \lambda \|v\|_2^2$, d.h. $\lambda \geq 0$). Außerdem ist B wegen $B^* = (A^*A)^* = A^*A^{**} = A^*A = B$ insbesondere eine **normale Matrix**, d.h. es gilt die Gleichheit $B^*B = BB^*$. Eine normale Matrix ist **unitär diagonalisierbar**, d.h. es gilt

$$QBQ^* = \Lambda \quad \text{bzw.} \quad B = Q^*\Lambda Q$$

mit einer unitären Matrix $Q \in \mathbb{K}^{n \times n}$ und einer Diagonalmatrix $\Lambda \in \mathbb{K}^{n \times n}$. Nun folgt mittels $\|Qx\|_2 = \|x\|_2$

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\|x\|_2=1} \sqrt{x^*Bx} = \max_{\|x\|_2=1} \sqrt{x^*Q^*\Lambda Qx} = \max_{y=Qx, \|y\|_2=1} \sqrt{y^*\Lambda y}$$

und schließlich durch Einsetzen der Standardbasisvektoren die Behauptung. \diamond

Bemerkung: Für den Spezialfall $A \in \mathbb{R}^{2 \times 2}$ ergibt sich

$$A = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}, \quad x = \begin{pmatrix} a \\ b \end{pmatrix}, \quad \|x\|_1 = |a| + |b|, \quad Ax = \begin{pmatrix} \alpha a + \beta b \\ \gamma a + \delta b \end{pmatrix}, \quad \|A\|_1 = \max_{\|x\|_1=1} \|Ax\|_1,$$

und damit die Abschätzung

$$\begin{aligned} \|Ax\|_1 &= |\alpha a + \beta b| + |\gamma a + \delta b| \leq (|\alpha| + |\gamma|)|a| + (|\beta| + |\delta|)|b| \\ &\leq \max\{|\alpha| + |\gamma|, |\beta| + |\delta|\}(|a| + |b|), \\ \|A\|_1 &\leq \max_{|a|+|b|=1} \max\{|\alpha| + |\gamma|, |\beta| + |\delta|\} \underbrace{(|a| + |b|)}_{=1} = \max\{|\alpha| + |\gamma|, |\beta| + |\delta|\}. \end{aligned}$$

Durch Einsetzen der Standardbasisvektoren folgt weiters (verwende $\|e_1\|_1 = 1 = \|e_2\|_1$)

$$\begin{cases} |\alpha| + |\gamma| = \|Ae_1\|_1 \leq \max_{\|x\|_1=1} \|Ax\|_1 = \|A\|_1, \\ |\beta| + |\delta| = \|Ae_2\|_1 \leq \|A\|_1, \end{cases} \quad \Rightarrow \quad \max\{|\alpha| + |\gamma|, |\beta| + |\delta|\} \leq \|A\|_1.$$

Damit folgt die Gleichheit der Operatornorm von A mit dem Maximum der Betragssummen der ersten bzw. zweiten Spalte von A

$$\|A\|_1 = \max\{|\alpha| + |\gamma|, |\beta| + |\delta|\}.$$

Ähnliche Überlegungen gelten für die Maximumsnorm.

• **Bemerkungen und Veranschaulichungen von Operatornormen:**

- Die Norm einer Matrix

$$\|A\| = \max_{\|x\|=1} \|Ax\|, \quad A \in \mathbb{K}^{m \times n},$$

gibt für Elemente der Normkugel in \mathbb{K}^n , d.h. Argumente $x \in \mathbb{K}^n$ mit $\|x\| = 1$, die maximale Ausdehnung der entsprechenden Bildelemente $\|Ax\|$ an.

- Speziell für invertierbare Matrizen $A \in \mathbb{K}^{n \times n}$ gilt wegen der eindeutigen Zuordnung $x \leftrightarrow y = Ax$ bzw. $y \leftrightarrow x = A^{-1}y$ für $x, y \in \mathbb{K}^n$ (verwende außerdem die Relation $\max\{x : x \in X\} = \frac{1}{\min\{x : x \in X\}}$ für eine beschränkte Menge $X \subset \mathbb{R} \setminus \{0\}$)

$$\|A^{-1}\| = \max_{y \neq 0} \frac{\|A^{-1}y\|}{\|y\|} = \max_{Ax \neq 0} \frac{\|x\|}{\|Ax\|} = \max_{x \neq 0} \frac{\|x\|}{\|Ax\|} = \frac{1}{\min_{x \neq 0} \frac{\|Ax\|}{\|x\|}} = \frac{1}{\min_{\|x\|=1} \|Ax\|}$$

und folglich

$$\frac{1}{\|A^{-1}\|} = \min_{\|x\|=1} \|Ax\| \quad \text{für } A \in \mathbb{K}^{n \times n} \text{ invertierbar.}$$

Anschaulich bedeutet das, daß die Operatornorm $\frac{1}{\|A^{-1}\|}$ für Argumente $x \in \mathbb{K}^n$ mit $\|x\| = 1$ die minimale Ausdehnung der entsprechenden Bildelemente $\|Ax\|$ angibt.

- Insbesondere im euklidischen Raum $(\mathbb{R}^2, \|\cdot\|_2)$ bilden für Elemente des Einheitskreises (d.h. $x_1 = \cos t$, $x_2 = \sin t$) die zugehörige Bildelemente unter einer linearen Abbildung $x \mapsto Ax$ mit $A \in \mathbb{R}^{2 \times 2}$ eine Ellipse, vgl. Abbildung, Skriptum, S. 53.
- Die obigen Überlegungen motivieren die Definition der Konditionzahl einer Matrix.

Die **Konditionszahl** einer quadratischen Matrix $A \in \mathbb{K}^{n \times n}$ ist definiert durch

$$\kappa(A) = \begin{cases} \|A\| \|A^{-1}\|, & \text{falls } A \in \mathbb{K}^{n \times n} \text{ invertierbar,} \\ \infty, & \text{falls } A \in \mathbb{K}^{n \times n} \text{ singulär,} \end{cases}$$

vgl. Definition 3.5. Allgemein definiert man für $A \in \mathbb{K}^{m \times n}$

$$\kappa(A) = \frac{\max_{\|x\|=1} \|Ax\|}{\min_{\|x\|=1} \|Ax\|}.$$

Bemerkungen:

- Für die Konditionszahl einer Matrix gilt $\kappa(A) \geq 1$ (vgl. Veranschaulichung im \mathbb{R}^2).
- Für unitäre Matrizen $Q \in \mathbb{K}^{n \times n}$ gilt (wegen $Q^*Q = I = QQ^*$ ist $\|Qx\|_2 = \|x\|_2$ und $\|Q^{-1}x\|_2 = \|Q^*x\|_2 = \|x\|_2$)

$$\|Q\|_2 = 1, \quad \|Q^{-1}\|_2 = 1.$$

Somit gilt auch

$$\kappa(Q) = 1.$$

Weiters bleibt bei Transformation einer Matrix $A \in \mathbb{K}^{m \times n}$ mittels einer unitären Matrix $Q \in \mathbb{K}^{m \times m}$ wegen $\|QAx\|_2 = \|Ax\|_2$ für $x \in \mathbb{K}^n$ die Operatornorm und folglich auch die Konditionzahl erhalten

$$\|QA\|_2 = \|A\|_2, \quad \kappa(QA) = \kappa(A).$$

- Für $m \times n$ Matrizen ist die **Frobenius-Norm** $\|\cdot\|_F : \mathbb{K}^{m \times n} \rightarrow \mathbb{R}_{\geq 0}$ definiert durch

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}, \quad A \in \mathbb{K}^{m \times n}.$$

Dies entspricht der euklidischen Norm des entsprechenden Vektors $a \in \mathbb{K}^{m \cdot n}$ mit den Komponenten a_{ij} für $1 \leq i \leq m$ und $1 \leq j \leq n$. Die Frobenius-Norm erfüllt die Eigenschaften positive Definitheit, Homogenität, Sub-Additivität, Sub-Multiplikativität und Konsistenz, ist jedoch keine Operatornorm, da $\|I\|_F = \sqrt{n} \neq 1$ für $I \in \mathbb{K}^{n \times n}$ mit $n \geq 2$.

4. Direkte Verfahren für lineare Gleichungssysteme

- Die näherungsweise **Lösung linearer Gleichungssysteme** ist die wichtigste Grundaufgabe der Numerischen Linearen Algebra und findet beispielsweise Anwendung bei
 - Verfahren zur Lösung nichtlinearer Gleichungssysteme,
 - Verfahren zur Lösung von Optimierungsproblemen,
 - Verfahren zur Lösung gewöhnlicher Differentialgleichungen,
 - Verfahren zur Lösung partieller Differentialgleichungen.
- **Inhalte:**
 - Kondition des Problems.
 - QR-Zerlegung einer Matrix mittels Householder-Reflexionen zur direkten Lösung linearer Gleichungssysteme.
 - Gaußsches Eliminationsverfahren bzw. LR-Zerlegung einer Matrix zur direkten Lösung linearer Gleichungssysteme.
 - Numerische Stabilität der Verfahren.

4.1. Kondition linearer Gleichungssysteme

- **Fragestellung:** Bestimmung der Kondition des numerischen Problems der Lösung eines linearen Gleichungssystems $Ax = b$, d.h.

$$p: \mathbb{K}^{m \times n} \times \mathbb{K}^m \rightarrow \mathbb{K}^n: (A, b) \mapsto x.$$

Zu untersuchen ist also die Sensitivität der Lösung $x + \xi = p(A + \alpha, b + \beta)$ gegenüber kleinen Änderungen α und β der Eingangsdaten, d.h. das Ziel ist es, die Lösung x des linearen Gleichungssystems $Ax = b$ ist mit der Lösung $x + \xi$ des linearen Gleichungssystems $(A + \alpha)(x + \xi) = b + \beta$ in Relation zu setzen und eine Abschätzung für den relativen Fehler $\frac{\|\xi\|}{\|x\|}$ herzuleiten.

- **Annahme:** Es sei $A \in \mathbb{K}^{n \times n}$ invertierbar und die Änderung $\alpha \in \mathbb{K}^{n \times n}$ so klein, daß $A + \alpha$ ebenfalls invertierbar ist.

Resultat zur Invertierbarkeit der Matrix $A + \alpha$ (Satz 4.1):

$$\|\alpha\| < \frac{1}{\|A^{-1}\|} \implies A + \alpha \text{ invertierbar.}$$

Denn: Die Matrix $A + \alpha$ ist genau dann invertierbar, wenn das lineare Gleichungssystem $(A + \alpha)x = 0$ nur die triviale Lösung $x = 0$ besitzt. Mit Hilfe der geforderten Invertierbarkeit von A sowie der Bedingung $\|\alpha\| < \frac{1}{\|A^{-1}\|} \Leftrightarrow 1 - \|A^{-1}\|\|\alpha\| > 0$ folgt

$$\begin{aligned} (A + \alpha)x = 0 &\implies Ax = -\alpha x \implies x = -A^{-1}\alpha x \\ &\implies \|x\| \leq \|A^{-1}\|\|\alpha\|\|x\| \implies (1 - \|A^{-1}\|\|\alpha\|)\|x\| \leq 0 \implies \|x\| = 0 \end{aligned}$$

und damit $x = 0$. \diamond

- **Annahme:** Es sei $A \in \mathbb{K}^{n \times n}$ invertierbar und es gelte $\|\alpha\| < \frac{1}{\|A^{-1}\|}$ für $\alpha \in \mathbb{K}^{n \times n}$, d.h. die Matrix $A + \alpha$ ist ebenfalls invertierbar. Weiters sei $0 \neq b \in \mathbb{K}^n$ sowie $0 \neq x \in \mathbb{K}^n$.

Abschätzung: Unter der Voraussetzung, daß die relativen Fehler der Eingangsdaten den Schranken (wobei $\varepsilon > 0$ hinreichend klein, sodaß $\varepsilon \kappa(A) = \varepsilon \|A\|\|A^{-1}\| < 1$)

$$\frac{\|\alpha\|}{\|A\|} \leq \varepsilon, \quad \frac{\|\beta\|}{\|b\|} \leq \varepsilon,$$

genügen, ergibt sich folgende Abschätzung für den relativen Fehler des Ergebnisses (beachte $\frac{\|b\|}{\|A\|\|x\|} \leq 1$ wegen $\|b\| = \|Ax\| \leq \|A\|\|x\|$)

$$\frac{\|\xi\|}{\|x\|} \leq \frac{\varepsilon \kappa(A)}{1 - \varepsilon \kappa(A)} \left(1 + \frac{\|b\|}{\|A\|\|x\|} \right).$$

Bemerkung: Die Konditionszahl der Matrix A ist ausschlaggebend für die Kondition des Problems. Falls $\varepsilon \kappa(A) \ll 1$ gilt, ist das Problem gut konditioniert. Falls hingegen $\varepsilon \kappa(A) \approx 1$ gilt, ist die Lösung des linearen Gleichungssystems ein schlecht konditioniertes Problem.

Denn: Es gilt (verwende $1 - \|A^{-1}\|\|\alpha\| > 0$)

$$\begin{aligned}
 (A + \alpha)(x + \xi) = b + \beta &\implies Ax + A\xi + \alpha x + \alpha\xi = b + \beta \\
 &\implies A\xi = \beta - \alpha x - \alpha\xi \\
 &\stackrel{Ax=b}{\implies} \xi = A^{-1}(\beta - \alpha x - \alpha\xi) \\
 &\implies \|\xi\| \leq \|A^{-1}\|(\|\beta\| + \|\alpha\|\|x\| + \|\alpha\|\|\xi\|) \\
 &\implies (1 - \|A^{-1}\|\|\alpha\|)\|\xi\| \leq \|A^{-1}\|(\|\beta\| + \|\alpha\|\|x\|) \\
 &\implies \|\xi\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\alpha\|} (\|\beta\| + \|\alpha\|\|x\|).
 \end{aligned}$$

Einsetzen der vorausgesetzten Schranken $\|\alpha\| \leq \varepsilon\|A\|$ und $\|\beta\| \leq \varepsilon\|b\|$ sowie der Konditionszahl $\kappa(A) = \|A\|\|A^{-1}\|$ unter Verwendung der Relation

$$\begin{aligned}
 0 \leq q = \|A^{-1}\|\|\alpha\| < 1, \quad q \leq \tilde{q} = \varepsilon\kappa(A) < 1, \\
 \frac{1}{1 - \|A^{-1}\|\|\alpha\|} = \frac{1}{1 - q} = \sum_{k=0}^{\infty} q^k \leq \sum_{k=0}^{\infty} \tilde{q}^k = \frac{1}{1 - \tilde{q}} = \frac{1}{1 - \varepsilon\kappa(A)},
 \end{aligned}$$

ergibt die Abschätzung

$$\|\xi\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\alpha\|} (\|\beta\| + \|\alpha\|\|x\|) \leq \frac{\varepsilon\kappa(A)}{1 - \varepsilon\kappa(A)} \left(\frac{\|b\|}{\|A\|} + \|x\| \right),$$

und damit folgt die Behauptung. \diamond

• **Bedeutung des Residuums:**

- Für eine Näherungslösung \tilde{x} an die Lösung x des linearen Gleichungssystems $Ax = b$ ergibt sich durch Einsetzen das **Residuum**

$$r(\tilde{x}) = A\tilde{x} - b.$$

Klarerweise gilt $r(x) = Ax - b = 0$.

- **Aber!** Aus der Größe des Residuums kann man im Allgemeinen nichts über die Güte der Näherungslösung, d.h. die Größe des Fehlers $\|\tilde{x} - x\|$, ableiten.
- Es gilt die Abschätzung

$$\|\tilde{x} - x\| \leq \kappa(A) \frac{\|r(\tilde{x})\|}{\|A\|},$$

die aus der Überlegung

$$\begin{aligned}
 r(\tilde{x}) = A\tilde{x} - b = A(\tilde{x} - x) &\implies \tilde{x} - x = A^{-1}r(\tilde{x}) \\
 &\implies \|\tilde{x} - x\| \leq \|A^{-1}\|\|r(\tilde{x})\| = \kappa(A) \frac{\|r(\tilde{x})\|}{\|A\|}
 \end{aligned}$$

folgt.

- Aus der Relation

$$A\tilde{x} = b + r(\tilde{x})$$

folgt man, daß bei einem *kleinen* Residuum die Näherungslösung \tilde{x} der exakten Lösung eines linearen Gleichungssystems mit exakter Matrix A und leicht abgeänderter rechter Seite $b + r(\tilde{x})$ entspricht und damit eine akzeptable Näherungslösung ist.

- Für zwei Näherungslösungen \tilde{x} und \hat{x} kann aber folgende Situation eintreten

$$\|\tilde{x} - x\| \approx \|\hat{x} - x\| \quad \text{und} \quad \|r(\tilde{x})\| \ll \|r(\hat{x})\|$$

oder sogar

$$\|\tilde{x} - x\| \gg \|\hat{x} - x\| \quad \text{und} \quad \|r(\tilde{x})\| \ll \|r(\hat{x})\|.$$

- **Beispiel von Kahan**, vgl. **Illustration4_Kahan** und Abbildung, Skriptum, S. 59.

4.2. Lösung über die QR-Zerlegung

- **Situation:** Es sei $A \in \mathbb{K}^{n \times n}$ invertierbar und es bezeichne

$$A = QR,$$

$$(a_1 | \cdots | a_n) = (q_1 | \cdots | q_n) \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix},$$

die (eindeutig bestimmte reduzierte bzw. volle) QR-Zerlegung von A mit unitärer Matrix $Q \in \mathbb{K}^{n \times n}$ (d.h. $Q^*Q = I$) und oberer Dreiecksmatrix $R \in \mathbb{K}^{n \times n}$ mit positiven Diagonaleinträgen $r_{ii} > 0$ für $1 \leq i \leq n$.

Erinnerung: Die Berechnung der QR-Zerlegung von A mittels des Orthogonalisierungsverfahrens nach Gram–Schmidt basiert auf der Idee, die Spaltenvektoren von A zu orthogonalisieren. Die Forderung $\langle a_1, \dots, a_k \rangle = \langle q_1, \dots, q_k \rangle$ für $1 \leq k \leq n$ führt auf eine Dreiecksmatrix R . Die numerische Instabilität des Verfahrens in gewissen Situationen erfordert die Konstruktion eines geeigneten alternativen Verfahrens.

Alternativer Zugang: Ein alternatives Verfahren zur Berechnung der QR-Zerlegung einer (invertierbaren) Matrix $A \in \mathbb{K}^{n \times n}$ verwendet die Idee der **Triangulierung** von A durch Multiplikation mit geeignet gewählten unitären Matrizen Q_n^*, \dots, Q_1^* (d.h. $Q_i^* = Q_i$ für $1 \leq i \leq n$, wegen $(Q_{i+1}Q_i)^*(Q_{i+1}Q_i) = Q_i^*Q_{i+1}^*Q_{i+1}Q_i = I$ ist das Produkt unitärer Matrizen und damit Q^* unitär)

$$\underbrace{Q_1^* \cdots Q_n^*}_{=Q^*} A = R \iff A = \underbrace{Q_n \cdots Q_1}_{=Q} R.$$

Transformationen zur Triangulierung von A basieren auf der Verwendung von **Householder-Reflexionen** oder Givens-Rotationen.

- Eine **Householder-Reflexion** ist eine Matrix der Form

$$T = I - 2vv^* \in \mathbb{K}^{n \times n}, \quad v \in \mathbb{K}^n, \quad \|v\|_2 = 1.$$

Bemerkungen zu Householder-Reflexionen:

- Im allgemeinen ist eine Householder-Reflexion eine *volle* Matrix.

Der Rang einer Householder-Reflexion ist 1.

Zur **Berechnung des Produktes** Tx mit $x \in \mathbb{K}^n$ verwendet man **ausschließlich** die folgende Relation

$$Tx = x - 2v \underbrace{v^*x}_{=c \in \mathbb{K}} = x - 2cv.$$

Zur **Berechnung der transformierten Matrix** TA verwendet man

$$TA = (Ta_1 | \cdots | Ta_n), \quad Ta_j = a_j - 2(v^*a_j)v, \quad 1 \leq j \leq n.$$

- Eine Householder-Reflexion ist eine selbstadjungierte und unitäre Matrix

$$T^* = (I - 2vv^*)^* = I^* - 2(vv^*)^* = I - 2v^{**}v^* = I - 2vv^* = T,$$

$$T^*T = T^2 = (I - 2vv^*)(I - 2vv^*) = I - 4vv^* + 4v \underbrace{v^*v}_{=\|v\|_2^2=1} v^* = I.$$

- Die Bezeichnung Householder-Reflexion erklärt sich durch folgende Eigenschaften: Es sei v, w_1, \dots, w_{n-1} eine Orthonormalbasis des \mathbb{K}^n . Einerseits wird v bei Anwendung von T auf $-v$ abgebildet

$$Tv = v - 2v \underbrace{v^*v}_{=1} = v - 2v = -v.$$

Andererseits wird jeder auf v orthogonale Vektor $w \in \mathbb{K}^n$, d.h. $w \in \langle w_1, \dots, w_{n-1} \rangle$ (**Hyperebene** im \mathbb{K}^n), auf sich selbst abgebildet

$$Tw = w - 2v \underbrace{v^*w}_{=0} = w.$$

Für einen beliebigen Vektor $x \in \mathbb{K}^n$ folgt mittels eindeutiger Darstellung als Linearkombination der Basisvektoren

$$x = \lambda v + w, \quad \lambda = v^*x, \quad w \in \langle w_1, \dots, w_{n-1} \rangle,$$

$$Tx = T(\lambda v + w) = \lambda Tv + Tw = -\lambda v + w,$$

d.h. die Multiplikation mit der Matrix T bewirkt eine Spiegelung an der zum Vektor v orthogonalen Hyperebene, vgl. Abbildung, Skriptum, S. 62.

- In Hinblick auf die Triangulierung einer Matrix möchte man beispielsweise erreichen, daß durch Multiplikation mit einer Householder-Reflexion der erste Spaltenvektor auf ein Vielfaches des ersten Standardbasisvektors transformiert wird. Die entsprechende Householder-Reflexion wird nun abgeleitet.

Vorbemerkung: Aufgrund der Selbstadjungiertheit von T (d.h. es gilt $T^* = T$) folgt speziell für $y = Tx$

$$x^*y \underset{y=Tx}{=} x^*Tx \underset{T^*=T}{=} x^*T^*x = (Tx)^*x \underset{y=Tx}{=} y^*x \underset{\bar{y}^T x = x^T \bar{y} = \overline{x^T y}}{=} \overline{x^*y} \implies x^*y \in \mathbb{R}.$$

Allgemeiner Fall: Eine Householder-Reflexion T soll so bestimmt werden, daß zwei vorgegebene Vektoren gleicher Norm ineinander übergehen

$$y = Tx, \quad x, y \in \mathbb{K}^n, \quad x \neq y, \quad \|x\|_2 = \|y\|_2, \quad x^*y = y^*x \in \mathbb{R}.$$

Die obigen Überlegungen ergeben

$$x = \lambda v + w, \quad y = -\lambda v + w \implies x - y = 2\lambda v$$

und führen damit auf die Wahl (wegen $v = c(x - y)$ und $\|v\|_2 = 1$)

$$v = \frac{1}{\|x-y\|_2} (x - y).$$

Man setzt somit

$$T = I - 2vv^*, \quad v = \frac{1}{\|x-y\|_2} (x - y).$$

Spezialfall: Für ein vorgegebenes Argument $x \in \mathbb{K}^n$ und ein (zu bestimmendes) Vielfaches des ersten Standardbasisvektors als zugehöriges Bildelement $y \in \mathbb{K}^n$ führen die obigen Überlegungen auf (Definition von α in Übereinstimmung mit obigen Bedingungen, Wahl des Vorzeichens von α zur Vermeidung von Auslöschung bei der Berechnung von $x - y = (x_1 + \alpha, x_2, \dots, x_n)$, Norm von $x - y$ vereinfacht sich zu $\|x - y\|_2^2 = \|x\|_2^2 + |\alpha|^2 + 2\Re(\alpha \bar{x}_1) = 2\|x\|_2^2 + 2|x_1| \|x\|_2$)

$$Tx = y = -\alpha e_1, \quad \alpha = \begin{cases} \|x\|_2 \frac{x_1}{|x_1|}, & \text{falls } x_1 \neq 0, \\ \|x\|_2, & \text{falls } x_1 = 0, \end{cases}$$

$$T = I - 2vv^*, \quad v = \frac{1}{\|x-y\|_2} (x - y).$$

- **Berechnung der QR-Zerlegung** einer Matrix mittels Householder-Reflexionen: Die schrittweise Anwendung der obigen Überlegungen führt auf die Triangulierung von

$$A = (a_1 | \dots | a_n) \in \mathbb{K}^{n \times n}.$$

- 1. Schritt: Elimination in der ersten Spalte

$$T_1 = I - 2v_1v_1^*, \quad v_1 = \frac{a_1 + \alpha_1 e_1}{\|a_1 + \alpha_1 e_1\|_2}, \quad \alpha_1 = \begin{cases} \|a_1\|_2 \frac{a_{11}}{|a_{11}|}, & a_{11} \neq 0, \\ \|a_1\|_2, & a_{11} = 0, \end{cases}$$

$$T_1 A = (T_1 a_1 | \dots | T_1 a_n), \quad T_1 a_1 = -\alpha_1 e_1, \quad T_1 a_j = a_j - 2(v_1^* a_j) v_1, \quad 2 \leq j \leq n,$$

$$T_1 A = \begin{pmatrix} -\alpha_1 & * & \dots & * \\ & * & \dots & * \\ & \vdots & & \vdots \\ & * & \dots & * \end{pmatrix} = \begin{pmatrix} -\alpha_1 & * \\ & A_1 \end{pmatrix} \in \mathbb{K}^{n \times n}, \quad A_1 \in \mathbb{K}^{(n-1) \times (n-1)}.$$

- 2. Schritt: Elimination in der zweiten Spalte

$$T_2 = \begin{pmatrix} 1 & \\ & \tilde{T}_2 \end{pmatrix} \in \mathbb{K}^{n \times n},$$

$$\tilde{T}_2 = \begin{pmatrix} -\alpha_2 & * \\ & A_2 \end{pmatrix} \in \mathbb{K}^{(n-1) \times (n-1)}, \quad A_2 \in \mathbb{K}^{(n-2) \times (n-2)},$$

$$T_2 T_1 A = \begin{pmatrix} -\alpha_1 & * & * & \dots & * \\ & -\alpha_2 & * & \dots & * \\ & & \vdots & & \vdots \\ & & * & \dots & * \end{pmatrix}.$$

Insgesamt ergibt sich nach $n - 1$ Schritten die QR-Zerlegung von A

$$\underbrace{T_{n-1} \cdots T_1}_{=Q^*} A = R \iff A = \underbrace{T_1 \cdots T_{n-1}}_{=Q} R.$$

Als Pseudo-Code formuliert lautet die Berechnung der QR-Zerlegung einer Matrix mittels Householder-Reflexionen folgendermaßen

Eingabedaten: a_{ij} für $1 \leq i, j \leq n$

for $k = 1:n-1$

$x = A(k : \text{end}, k)$

$u = x, \quad u(1) = u(1) + \|x\|_2 \frac{x_1}{|x_1|}$

$v_k = \frac{1}{\|u\|_2} u$

$A(k : \text{end}, k : \text{end}) = A(k : \text{end}, k : \text{end}) - 2 v_k (v_k^* A(k : \text{end}, k : \text{end}))$

end

Ergebnisse: a_{ij} für $1 \leq i \leq j \leq n$ und v_{ik} für $1 \leq i \leq n + 1 - k$ und $1 \leq k \leq n - 1$

Bemerkungen:

- Beachte, daß $v_k \in \mathbb{K}^{n+1-k}$ für $1 \leq k \leq n - 1$.
- Die Koeffizienten von A können sukzessive durch die neu berechneten Einträge der Dreiecksmatrix R und die Komponenten von v_k überschrieben werden. Zusätzlich werden die Diagonalelemente von R abgespeichert.
- Die Anzahl der zur Berechnung der QR-Zerlegung einer Matrix $A \in \mathbb{K}^{n \times n}$ benötigten Operationen (Addition oder Multiplikation in \mathbb{K}) ist

$$\mathcal{O}\left(\frac{4}{3} n^3\right),$$

vgl. Skriptum, S. 65.

- Die Erweiterung des Algorithmus auf Matrizen $A \in \mathbb{K}^{m \times n}$ von vollem Rang verwendet die Hinzunahme von T_n (d.h. im Pseudo-Algorithmus ersetzt man die Schleife for $k = 1:n-1 \dots \text{end}$ durch for $k = 1:n \dots \text{end}$), vgl. Skriptum, S. 66.
- Die Berechnung des Produktes

$$Q^* b = T_{n-1} \cdots T_1 b$$

für $b \in \mathbb{K}^n$ benötigt die Kenntnis der Vektoren $v_k \in \mathbb{K}^{n+1-k}$ für $1 \leq k \leq n - 1$. Als Pseudo-Code ergibt sich

Eingabedaten: b_i für $1 \leq i \leq n$ und v_{ik} für $1 \leq i \leq n + 1 - k$ und $1 \leq k \leq n - 1$

for $k = 1:n-1$

$b(k : \text{end}) = b(k : \text{end}) - 2 v_k (v_k^* b(k : \text{end}))$

end

Ergebnisse: b_i für $1 \leq i \leq n$

- **Lösung eines linearen Gleichungssystems mittels QR-Zerlegung:** Falls die QR-Zerlegung der Matrix $A \in \mathbb{K}^{n \times n}$ bekannt ist, ist die Lösung des linearen Gleichungssystems $Ax = b$ (wobei $x \in \mathbb{K}^n$ und $b \in \mathbb{K}^n$) einfach durchzuführen. Einsetzen der QR-Zerlegung von A und Multiplikation des Gleichungssystems mit der adjungierten Matrix Q^* (invertierbar) führt auf ein äquivalentes **gestaffeltes lineares Gleichungssystem**

$$Ax = b \quad \begin{array}{c} \iff \\ A=QR \end{array} \quad QRx = b \quad \begin{array}{c} \iff \\ Q^*Q=I \end{array} \quad Rx = \underbrace{Q^*b}_{=c}$$

dessen Lösung direkt mittels **Rückwärtssubstitution** berechnet werden kann (verwende $r_{ij} = 0$ für $1 \leq j < i \leq n$ und $r_{ii} > 0$ für $1 \leq i \leq n$)

$$Rx = c,$$

$$\begin{pmatrix} r_{11} & \dots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix},$$

$$\sum_{j=i}^n r_{ij} x_j = c_i, \quad x_i = \frac{1}{r_{ii}} \left(c_i - \sum_{j=i+1}^n r_{ij} x_j \right), \quad 1 \leq i \leq n,$$

$$x_n = \frac{1}{r_{nn}} c_n, \quad \dots, \quad x_1 = \frac{1}{r_{11}} \left(c_1 - \sum_{j=2}^n r_{1j} x_j \right).$$

Als Pseudo-Code lautet die Rückwärtssubstitution

```

Eingabedaten:  $c_i, r_{ij}$  für  $1 \leq i \leq j \leq n$ 
for i = n:-1:1
     $x_i = c_i$ 
    for j = i+1:n
         $x_i = x_i - r_{ij} x_j$ 
    end
     $x_i = \frac{1}{r_{ii}} x_i$ 
end
Ergebnisse:  $x_i$  für  $1 \leq i \leq n$ 

```

4.3. Stabilität der Lösungsmethode über die QR-Zerlegung

- **Erinnerung:** Das in Abschnitt 4.2 besprochene Verfahren zur Lösung eines linearen Gleichungssystems $Ax = b$ mit $A \in \mathbb{K}^{n \times n}$ invertierbar und $b \in \mathbb{K}^n$ umfaßt die folgenden Schritte.
 - QR-Zerlegung von $A = QR$ bzw. Triangulierung von A mittels Householder-Reflexionen,
 - Berechnung des Produktes $c = Q^* b$,
 - Rückwärtssubstitution zur Berechnung von x aus $Rx = c$.

Stabilität des Verfahrens (Satz 4.2): Die unter dem Einfluß von Rundungsfehlern berechnete Näherungslösung \tilde{x} ist die exakte Lösung eines linearen Gleichungssystems

$$\tilde{A}\tilde{x} = b \quad \text{mit} \quad \|A - \tilde{A}\|_2 \leq \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2,$$

d.h. das Verfahren ist im strengen Sinn numerisch stabil.

Bemerkungen:

- Die Größe $\mathcal{O}(\varepsilon_{\text{mach}})$ hängt von der Dimension n der Matrix A ab.
 - Begründung der Stabilitätsaussage für die Rückwärtssubstitution. Zusätzliche Überlegungen zur Ableitung der Abschätzung.
 - Zusätzliche Berechnung des Residuums $A\tilde{x} - b$ sichert eine akzeptable Näherungslösung bei kleinem Residuum, vgl. Bemerkung, Skriptum, S. 69.
- **Rundungsfehleranalyse der Rückwärtssubstitution** (im Sinne der Rückwärtsanalyse): Die Rückwärtssubstitution für das lineare Gleichungssystem $Rx = c$ mit invertierbarer oberer Dreiecksmatrix $R \in \mathbb{K}^{n \times n}$ beruht auf der Berechnung der unbekanntenen Komponenten x_n, x_{n-1}, \dots, x_1 mittels der Relation

$$x_i = \frac{1}{r_{ii}} \left(c_i - \sum_{j=i+1}^n r_{ij} x_j \right), \quad 1 \leq i \leq n.$$

Unter dem Einfluß von Rundungsfehlern ergibt sich folgende Näherungslösung (verwende $\text{rd}(a * b) = (a * b)(1 + \varepsilon)$ mit $|\varepsilon| \leq \varepsilon_{\text{mach}}$)

$$\begin{aligned} \longrightarrow & \text{rd}(c_i - r_{i,i+1} \tilde{x}_{i+1} (1 + \tilde{\varepsilon}_{i1})) = c_i (1 + \varepsilon_{i1}) - r_{i,i+1} \tilde{x}_{i+1} (1 + \varepsilon_{i1}) (1 + \tilde{\varepsilon}_{i1}) \\ \longrightarrow & \text{rd}\left(c_i (1 + \varepsilon_{i1}) - r_{i,i+1} \tilde{x}_{i+1} (1 + \varepsilon_{i1}) (1 + \tilde{\varepsilon}_{i1}) - r_{i,i+2} \tilde{x}_{i+2} (1 + \tilde{\varepsilon}_{i2})\right) \\ & = c_i (1 + \varepsilon_{i1}) (1 + \varepsilon_{i2}) - r_{i,i+1} \tilde{x}_{i+1} (1 + \varepsilon_{i1}) (1 + \varepsilon_{i2}) (1 + \tilde{\varepsilon}_{i1}) \\ & \quad - r_{i,i+2} \tilde{x}_{i+2} (1 + \varepsilon_{i2}) (1 + \tilde{\varepsilon}_{i2}) \\ \longrightarrow & \text{rd}\left(c_i (1 + \varepsilon_{i1}) (1 + \varepsilon_{i2}) - r_{i,i+1} \tilde{x}_{i+1} (1 + \varepsilon_{i1}) (1 + \varepsilon_{i2}) (1 + \tilde{\varepsilon}_{i1})\right. \\ & \quad \left. - r_{i,i+2} \tilde{x}_{i+2} (1 + \varepsilon_{i2}) (1 + \tilde{\varepsilon}_{i2})\right) \\ & = c_i (1 + \varepsilon_{i1}) (1 + \varepsilon_{i2}) (1 + \varepsilon_{i3}) - r_{i,i+1} \tilde{x}_{i+1} (1 + \varepsilon_{i1}) (1 + \varepsilon_{i2}) (1 + \varepsilon_{i3}) (1 + \tilde{\varepsilon}_{i1}) \\ & \quad - r_{i,i+2} \tilde{x}_{i+2} (1 + \varepsilon_{i2}) (1 + \varepsilon_{i3}) (1 + \tilde{\varepsilon}_{i2}) \end{aligned}$$

und insgesamt (zur Vereinfachung werden die Größen $\tilde{\varepsilon}_{ij}$ nicht unterschieden)

$$\begin{aligned} \dots &\longrightarrow c_i \prod_{\ell=1}^{n-i} (1 + \varepsilon_{i\ell}) - \sum_{j=i+1}^n r_{ij} \tilde{x}_j \prod_{\ell=j-i}^{n-i} (1 + \varepsilon_{i\ell}) (1 + \tilde{\varepsilon}) \\ &\longrightarrow \tilde{x}_i = \frac{1+\tilde{\varepsilon}}{r_{ii}} \left(c_i \prod_{\ell=1}^{n-i} (1 + \varepsilon_{i\ell}) - \sum_{j=i+1}^n r_{ij} \tilde{x}_j \prod_{\ell=j-i}^{n-i} (1 + \varepsilon_{i\ell}) (1 + \tilde{\varepsilon}) \right) \end{aligned}$$

mit $|\varepsilon_{i\ell}| \leq \varepsilon_{\text{mach}}$ für $1 \leq \ell \leq n-i$ und $|\tilde{\varepsilon}| \leq \varepsilon_{\text{mach}}$. Weiters erhält man

$$\begin{aligned} \tilde{x}_i &= \frac{1+\tilde{\varepsilon}}{r_{ii}} \left(c_i \prod_{\ell=1}^{n-i} (1 + \varepsilon_{i\ell}) - \sum_{j=i+1}^n r_{ij} \tilde{x}_j \prod_{\ell=j-i}^{n-i} (1 + \varepsilon_{i\ell}) (1 + \tilde{\varepsilon}) \right) \\ \iff &\frac{r_{ii}}{1+\tilde{\varepsilon}} \tilde{x}_i + \sum_{j=i+1}^n r_{ij} \tilde{x}_j \prod_{\ell=j-i}^{n-i} (1 + \varepsilon_{i\ell}) (1 + \tilde{\varepsilon}) = c_i \prod_{\ell=1}^{n-i} (1 + \varepsilon_{i\ell}) \\ \iff &c_i = r_{ii} \tilde{x}_i \prod_{\ell=1}^{n-i} \frac{1}{1+\varepsilon_{i\ell}} \frac{1}{1+\tilde{\varepsilon}} + \sum_{j=i+1}^n r_{ij} \tilde{x}_j \prod_{\ell=1}^{n-i} \frac{1}{1+\varepsilon_{i\ell}} \prod_{\ell=j-i}^{n-i} (1 + \varepsilon_{i\ell}) (1 + \tilde{\varepsilon}) \\ \iff &c_i = r_{ii} \tilde{x}_i \prod_{\ell=1}^{n-i} \frac{1}{1+\varepsilon_{i\ell}} \frac{1}{1+\tilde{\varepsilon}} + \sum_{j=i+1}^n r_{ij} \tilde{x}_j \prod_{\ell=1}^{j-i-1} \frac{1}{1+\varepsilon_{i\ell}} (1 + \tilde{\varepsilon}). \end{aligned}$$

Mittels Lemma 2.6. folgt damit die Relation (beachte die Abhängigkeit von n)

$$c_i = \sum_{j=i}^n r_{ij} (1 + \delta_{ij}) \tilde{x}_j, \quad |\delta_{ij}| \leq \frac{n\varepsilon_{\text{mach}}}{1-n\varepsilon_{\text{mach}}} = \mathcal{O}(\varepsilon_{\text{mach}}), \quad 1 \leq i \leq j \leq n.$$

Dies zeigt, daß die Näherungslösung \tilde{x} die exakte Lösung eines linearen Gleichungssystems ist, dessen Matrix die folgende komponentenweise Abschätzung bzw. Normabschätzung erfüllt (verwende $R = Q^* A$ und folglich $\|R\|_2 = \|A\|_2$)

$$\tilde{R} \tilde{x} = c, \quad |\tilde{R} - R| \leq \mathcal{O}(\varepsilon_{\text{mach}}) |R|, \quad \|\tilde{R} - R\|_2 \leq \mathcal{O}(\varepsilon_{\text{mach}}) \|R\|_2 = \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2.$$

- **Zusätzliche Überlegungen:** Es seien $\tilde{v}_1, \dots, \tilde{v}_{n-1}$ die bei der Triangulierung einer invertierbaren Matrix $A \in \mathbb{K}^{n \times n}$ mittels Householder-Reflexionen berechneten Vektoren und es bezeichnen $\tilde{T}_k = I - 2 \tilde{v}_k \tilde{v}_k^*$ für $1 \leq k \leq n-1$ und $\tilde{Q} = \tilde{T}_1 \cdots \tilde{T}_{n-1}$ die daraus entstehenden unitären Matrizen sowie \tilde{R} die berechnete obere Dreiecksmatrix. Dann gilt im Vergleich mit der reduzierten QR-Zerlegung von A die folgende Abschätzung (ohne Begründung)

$$\|\tilde{Q} \tilde{R} - A\|_2 \leq \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2.$$

Weiters erfüllt die bei der Berechnung von $\tilde{Q}^* b$ resultierende Näherungslösung \tilde{c} die Relation (ohne Begründung)

$$\Delta Q \tilde{c} = b - \tilde{Q} \tilde{c}, \quad \|\Delta Q\|_2 \leq \mathcal{O}(\varepsilon_{\text{mach}}).$$

Folglich ergibt sich mit ΔR definiert durch $\Delta R \tilde{x} = \tilde{c} - \tilde{R} \tilde{x}$, d.h. $\tilde{c} = (\tilde{R} + \Delta R) \tilde{x}$, die Relation

$$b = (\tilde{Q} + \Delta Q) \tilde{c} = (\tilde{Q} + \Delta Q) (\tilde{R} + \Delta R) \tilde{x} = \tilde{A} \tilde{x}.$$

Wegen $\tilde{A} - A = (\tilde{Q} + \Delta Q) (\tilde{R} + \Delta R) - A = \tilde{Q} \tilde{R} - A + \Delta Q \tilde{R} + \tilde{Q} \Delta R + \Delta Q \Delta R$ ergibt sich weiters (verwende die Abschätzungen $\|\tilde{R}\|_2 = \|\tilde{Q} \tilde{R}\|_2 \leq (1 + \mathcal{O}(\varepsilon_{\text{mach}})) \|A\|_2 \leq \mathcal{O}(1) \|A\|_2$ und $\|\tilde{Q} \Delta R\|_2 = \|\tilde{R} - R\|_2 \leq \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2$)

$$\begin{aligned} \|\tilde{A} - A\|_2 &= \|\tilde{Q} \tilde{R} - A + \Delta Q \tilde{R} + \tilde{Q} \Delta R + \Delta Q \Delta R\|_2 \\ &\leq \underbrace{\|\tilde{Q} \tilde{R} - A\|_2}_{\leq \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2} + \underbrace{\|\Delta Q\|_2}_{\leq \mathcal{O}(\varepsilon_{\text{mach}})} \underbrace{\|\tilde{R}\|_2}_{\leq \mathcal{O}(1) \|A\|_2} + \underbrace{\|\tilde{Q} \Delta R\|_2}_{\leq \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2} + \underbrace{\|\Delta Q \Delta R\|_2}_{\leq \mathcal{O}(\varepsilon_{\text{mach}}^2) \|A\|_2} \\ &\leq \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2. \end{aligned}$$

Dies führt auf die Abschätzung von Satz 4.2

$$\tilde{A} \tilde{x} = b, \quad \|\tilde{A} - A\|_2 \leq \mathcal{O}(\varepsilon_{\text{mach}}) \|A\|_2.$$

4.4. Gauß-Elimination und Dreieckszerlegung

- **Situation:** Lösung eines linearen Gleichungssystems

$$Ax = b$$

mit $A \in \mathbb{K}^{n \times n}$ invertierbar und $b \in \mathbb{K}^n$.

Das **Gaußsche Eliminationsverfahren** zur Lösung des linearen Gleichungssystems basiert auf folgender Idee:

- Eine der n Gleichungen wird explizit nach einer der Unbekannten aufgelöst (z.B. nach x_1). Die resultierende Bedingung (z.B. $x_1 = f_1(x_2, \dots, x_n)$) wird später zur Bestimmung dieser Unbekannten verwendet und in die restlichen $n - 1$ Gleichungen eingesetzt (Elimination z.B. von x_1 führt auf $n - 1$ Gleichungen $g_i(x_2, \dots, x_n)$ für $1 \leq i \leq n - 1$ in den $n - 1$ Unbekannten x_2, \dots, x_n). Damit ergibt sich ein Gleichungssystem in $n - 1$ Unbekannten, welches $n - 1$ Gleichungen umfaßt.
 - Durch induktive Fortführung dieser Idee der Elimination ergibt sich schließlich eine einzige Gleichung in einer Unbekannten, die direkt aufgelöst werden kann.
 - Das sukzessive Einsetzen der bereits bestimmten Unbekannten in die expliziten Gleichungen führt auf die Lösung des linearen Gleichungssystems (Rückwärts-substitution).
- **Beschreibung des Gaußschen Eliminationsverfahrens** unter der Annahme, daß bei der schrittweisen Elimination der Unbekannten die **natürliche Reihenfolge** x_1, x_2, \dots, x_n gewählt werden kann.

- Lineares Gleichungssystem

$$A^{(1)}x = b^{(1)}, \quad A^{(1)} = A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \in \mathbb{K}^{n \times n}, \quad b^{(1)} = b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{K}^n.$$

- 1. Schritt: Wahl der ersten Zeile von A als **Pivotzeile**. Elimination der Unbekannten x_1 in der i -ten Gleichung durch Subtraktion des $\frac{a_{i1}}{a_{11}}$ -Vielfachen der ersten Zeile von der i -ten Zeile für $2 \leq i \leq n$ führt auf das äquivalente Gleichungssystem

$$A^{(2)}x = b^{(2)}, \quad A^{(2)} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & \tilde{a}_{22} & \dots & \tilde{a}_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & \tilde{a}_{n2} & \dots & \tilde{a}_{nn} \end{pmatrix} \in \mathbb{K}^{n \times n}, \quad b^{(2)} = \begin{pmatrix} b_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{pmatrix} \in \mathbb{K}^n,$$

$$\ell_{i1} = \frac{a_{i1}}{a_{11}}, \quad \tilde{a}_{ij} = a_{ij} - \ell_{i1} a_{1j}, \quad \tilde{b}_i = b_i - \ell_{i1} b_1, \quad 2 \leq i \leq n, \quad 1 \leq j \leq n.$$

Theoretische Beschreibung des ersten Eliminationsschritts durch Multiplikation von A mit den elementaren Matrizen $N_{21}(-\ell_{21}), \dots, N_{n1}(-\ell_{n1})$, d.h. es ist

$$A^{(2)} = N_{n1}(-\ell_{n1}) \cdots N_{21}(-\ell_{21}) A^{(1)}.$$

- Analoge Elimination der Variablen x_2, \dots, x_{n-1} .

Theoretische Beschreibung der weiteren Eliminationsschritte durch Multiplikation mit elementaren Matrizen, beispielsweise

$$A^{(3)} = N_{n2}(-\ell_{n2}) \cdots N_{32}(-\ell_{32}) A^{(2)}.$$

- Nach $n - 1$ Eliminationsschritten ergibt sich ein zu $Ax = b$ äquivalentes lineares Gleichungssystem der Form

$$Rx = c$$

mit oberer Dreiecksmatrix $R \in \mathbb{K}^{n \times n}$, dessen Lösung mittels Rückwärtssubstitution bestimmt wird. Da die Matrix A nach Voraussetzung invertierbar ist, ist R ebenfalls invertierbar, d.h. die Diagonalelemente erfüllen $r_{ii} \neq 0$ für $1 \leq i \leq n$.

Theoretische Beschreibung der Eliminationsschritte durch Multiplikation mit elementaren Matrizen

$$R = \underbrace{N_{n,n-1}(-\ell_{n2}) \cdots N_{n2}(-\ell_{n2}) \cdots N_{32}(-\ell_{32}) N_{n1}(-\ell_{n1}) \cdots N_{21}(-\ell_{21})}_{=L^{-1}} A.$$

Dies ist äquivalent zur **LR-Zerlegung** bzw. **Dreieckszerlegung** bzw. **LU-Zerlegung** der Matrix A (verwende die Relation für die Inverse von $N_{ij}(\alpha)$ und erhalte damit $L = N_{21}(\ell_{21}) \cdots N_{n1}(\ell_{n1}) N_{32}(\ell_{32}) \cdots N_{n2}(\ell_{n2}) \cdots N_{n,n-1}(\ell_{n2})$)

$$A = LR,$$

$$L = \begin{pmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n1} & \ell_{n2} & \dots & \ell_{n,n-1} & 1 \end{pmatrix}, \quad R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix},$$

vgl. Skriptum, S. 74/75. Die Lösung des linearen Gleichungssystems $Ax = b$ mittels Gaußschem Eliminationsverfahren in natürlicher Reihenfolge entspricht damit der Berechnung der LR-Zerlegung von A

$$Ax = LRx = b \iff Ly = b, \quad Rx = y$$

und der Lösung des gestaffelten Gleichungssystems $Ly = b$ mittels **Vorwärtssubstitution**

$$Ly = b, \quad y_i = b_i - \sum_{j=1}^{i-1} \ell_{ij} y_j, \quad 1 \leq i \leq n,$$

sowie der Lösung des gestaffelten Gleichungssystems $Rx = y$ mittels **Rückwärts-substitution**

$$Rx = y, \quad x_i = \frac{1}{r_{ii}} \left(y_i - \sum_{j=i+1}^n r_{ij} x_j \right), \quad 1 \leq i \leq n.$$

Als Pseudo-Code lautet das Gaußsche Eliminationsverfahren beispielsweise (Überschreiben der Koeffizienten von A und b mit den berechneten Koeffizienten von L, R und c)

```

Eingabedaten :  $A, b$ 
for  $k = 1:n-1$ 
  for  $i = k+1:n$ 
     $A(i, k) = \frac{A(i, k)}{A(k, k)}$ 
    for  $j = k+1:n$ 
       $A(i, j) = A(i, j) - A(i, k) A(k, j)$ 
    end
     $b(i) = b(i) - A(i, k) b(k)$ 
  end
end
Zwischenergebnis :  $A, b$ 
for  $i = n:-1:1$ 
   $x(i) = b(i)$ 
  for  $j = i+1:n$ 
     $x(i) = x(i) - A(i, j) x(j)$ 
  end
   $x(i) = \frac{x(i)}{A(i, i)}$ 
end

```

Vgl. **Illustration4_GaussElimination**.

Bemerkungen:

- Üblicherweise wird zuerst die LR -Zerlegung von A berechnet und anschließend die Lösung des linearen Gleichungssystems mittels Vorwärtssubstitution und Rückwärtssubstitution berechnet.
- Die Anzahl der zur Berechnung der LR -Zerlegung von $A \in \mathbb{K}^{n \times n}$ benötigten Operationen (Addition oder Multiplikation in \mathbb{K}) ist

$$\mathcal{O}\left(\frac{2}{3} n^3\right).$$

- Günstige Wahl des Pivotelementes, vgl. Abschnitt 4.6.

4.5. Rundungsfehler-Analyse der Gauß-Elimination

- **Vorüberlegungen:** Das in Abschnitt 4.4 besprochene Verfahren zur Lösung eines linearen Gleichungssystems $Ax = b$ mit $A \in \mathbb{K}^{n \times n}$ invertierbar und $b \in \mathbb{K}^n$ umfaßt die folgenden Schritte.
 - LR-Zerlegung von $A = LR$ bzw. Triangulierung von A mittels elementarer Umformungen (beschrieben durch Multiplikation mit geeignet gewählten Matrizen $N_{ij}(\alpha)$).
 - Vorwärtssubstitution zur Berechnung von y aus $Ly = b$.
 - Rückwärtssubstitution zur Berechnung von x aus $Rx = y$.

Dieses Verfahren ist äquivalent zum Gaußschen Eliminationsverfahren in natürlicher Reihenfolge. Hinsichtlich einer Rundungsfehleranalyse wird verwendet, daß die Koeffizienten der unteren Dreiecksmatrix L und der oberen Dreiecksmatrix R durch folgende Relationen gegeben sind (zeilenweise Berechnung der Koeffizienten z.B. von L , es ist $\ell_{ik} = 0$ für $k \geq i + 1$ und $\ell_{ii} = 1$ für $1 \leq i \leq n$)

$$A = LR,$$

$$L = \begin{pmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n1} & \ell_{n2} & \dots & \ell_{n,n-1} & 1 \end{pmatrix}, \quad R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix},$$

$$\begin{cases} \ell_{ik} = \frac{1}{r_{kk}} \left(a_{ik} - \sum_{j=1}^{k-1} \ell_{ij} r_{jk} \right), & 1 \leq k \leq i-1, \\ r_{ik} = a_{ik} - \sum_{j=1}^{i-1} \ell_{ij} r_{jk}, & i \leq k \leq n, \end{cases} \quad 1 \leq i \leq n.$$

Vgl. **Illustration4_LRZerlegung**.

- **Rundungsfehleranalyse:** Ähnliche Überlegungen wie in Abschnitt 4.3 im Zusammenhang mit der Stabilität der Rückwärtssubstitution zeigen, daß sich unter dem Einfluß von Rundungsfehlern folgende Näherungslösungen beispielsweise für die Koeffizienten der Matrix L ergeben (verwende wieder $\text{rd}(a * b) = (a * b) (1 + \varepsilon)$ mit $|\varepsilon| \leq \varepsilon_{\text{mach}}$)

$$\ell_{ik} = \frac{1}{r_{kk}} \left(a_{ik} - \sum_{j=1}^{k-1} \ell_{ij} r_{jk} \right), \quad 1 \leq k \leq i-1, \quad 1 \leq i \leq n,$$

$$\tilde{\ell}_{ik} = \frac{1+\tilde{\varepsilon}}{\tilde{r}_{kk}} \left(a_{ik} \prod_{\ell=1}^{k-1} (1 + \varepsilon_{i\ell}) - \sum_{j=1}^{k-1} \tilde{\ell}_{ij} \tilde{r}_{jk} \prod_{\ell=j}^{k-1} (1 + \varepsilon_{i\ell}) (1 + \tilde{\varepsilon}) \right), \quad 1 \leq k \leq i-1, \quad 1 \leq i \leq n,$$

mit $|\varepsilon_{i\ell}| \leq \varepsilon_{\text{mach}}$ für $1 \leq \ell \leq k-1$ und $|\tilde{\varepsilon}| \leq \varepsilon_{\text{mach}}$. Mittels Lemma 2.6. folgt damit die Relation (beachte die Abhängigkeit von n)

$$a_{ik} = \sum_{j=1}^k \tilde{\ell}_{ij} \tilde{r}_{ik} (1 + \delta_{ijk}), \quad |\delta_{ijk}| \leq \frac{n\varepsilon_{\text{mach}}}{1-n\varepsilon_{\text{mach}}} = \mathcal{O}(\varepsilon_{\text{mach}}), \quad 1 \leq i, k \leq n.$$

Ähnliche Überlegungen führen auf folgende Abschätzungen für die berechneten Größen (zusätzliche Abhängigkeit von Dimension n)

$$\begin{aligned} |A - \tilde{L}\tilde{R}| &\leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{L}||\tilde{R}|, \\ \Delta L \tilde{y} &= b - \tilde{L} \tilde{y}, \quad |\Delta L| \leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{L}|, \\ \Delta R \tilde{x} &= \tilde{y} - \tilde{R} \tilde{x}, \quad |\Delta R| \leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{R}|. \end{aligned}$$

Daraus folgt weiters

$$\begin{aligned} b &= (\tilde{L} + \Delta L) \tilde{y} = (\tilde{L} + \Delta L) (\tilde{R} + \Delta R) \tilde{x} = \tilde{A} \tilde{x} \\ \tilde{A} - A &= (\tilde{L} + \Delta L) (\tilde{R} + \Delta R) - A = \tilde{L}\tilde{R} - A + \Delta L\tilde{R} + \tilde{L}\Delta R + \Delta L\Delta R, \\ |\tilde{A} - A| &\leq \underbrace{|\tilde{L}\tilde{R} - A|}_{\leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{L}||\tilde{R}|} + \underbrace{|\Delta L|}_{\leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{L}|} |\tilde{R}| + |\tilde{L}| \underbrace{|\Delta R|}_{\leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{R}|} + \underbrace{|\Delta L\Delta R|}_{\leq \mathcal{O}(\varepsilon_{\text{mach}}^2)|\tilde{L}||\tilde{R}|} \leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{L}||\tilde{R}|, \\ r &= b - A\tilde{x} = (\tilde{A} - A)\tilde{x}. \end{aligned}$$

Schlußfolgerungen:

- Die obigen Überlegungen zeigen, daß die mittels LR-Zerlegung und Substitutionen berechnete Näherungslösung \tilde{x} die exakte Lösung eines linearen Gleichungssystems ist

$$\tilde{A}\tilde{x} = b, \quad |\tilde{A} - A| \leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{L}||\tilde{R}|.$$

- **Erinnerung:** Der **Satz von Prager und Oettli** besagt, daß eine Näherungslösung \tilde{x} des linearen Gleichungssystems $Ax = b$ genau dann akzeptabel ist (bzgl U_ε , im strengen Sinn), wenn das Residuum $A\tilde{x} - b$ die folgende Abschätzung erfüllt

$$|A\tilde{x} - b| \leq \varepsilon (|A||\tilde{x}| + |b|).$$

- Nach dem Satz von Prager und Oettli ist die berechnete Näherungslösung \tilde{x} akzeptabel (bzgl. $U_{\varepsilon_{\text{mach}}}$), wenn das Residuum die Abschätzung

$$|r| = |\tilde{A} - A||\tilde{x}| \leq \varepsilon_{\text{mach}} (|A||\tilde{x}| + |b|)$$

erfüllt. Insbesondere ist das Verfahren numerisch stabil, wenn

$$|r| = |\tilde{A} - A||\tilde{x}| \leq \mathcal{O}(\varepsilon_{\text{mach}})|\tilde{L}||\tilde{R}||\tilde{x}| \leq \varepsilon_{\text{mach}} (|A||\tilde{x}| + |b|),$$

d.h. $|\tilde{L}||\tilde{R}| \approx |A|$.

Allerdings gibt es Situation, in denen

$$|\tilde{L}||\tilde{R}| \gg |\tilde{L}\tilde{R}| \approx |LR| = |A|$$

gilt, und somit das Verfahren numerisch instabil ist. Um dies (teilweise) zu vermeiden, verwendet man eine **Spalten-Pivotsuche** oder **vollständige Pivotsuche**, vgl. Abschnitt 4.6.

4.6. Pivotwahl bei der Gauß-Elimination

- **Situation:** Lösung eines linearen Gleichungssystems

$$Ax = b$$

mit $A \in \mathbb{K}^{n \times n}$ invertierbar und $b \in \mathbb{K}^n$.

Vorbemerkung: Das **Gaußsche Eliminationsverfahren in natürlicher Reihenfolge** benötigt, daß im k -ten Eliminationsschritt das **natürliche Pivotelement**, d.h. das k -te Diagonalelement der Matrix $A^{(k)}$ ungleich Null ist (notwendige Bedingung für Elimination und Rückwärtssubstitution). Wenn diese Bedingung verletzt ist und auch zur Verbesserung der Stabilität des Verfahrens führt man eine **Pivotsuche** durch.

Spalten-Pivotsuche mit Maximums-Strategie: Falls das k -te natürliche Pivotelement gleich Null ist, wählt man unter den Koeffizienten a_{ik} für $k+1 \leq i \leq n$ ein Pivotelement $a_{\ell k} \neq 0$. Bei Berechnungen mit endlicher Genauigkeit beeinflusst die Wahl des Pivotelements das Ergebnis. Bei der Spalten-Pivotsuche mit Maximums-Strategie wählt man im k -ten Eliminationsschritt den betragsmäßig größten Koeffizienten in der k -ten Spalte der aktuellen Matrix als Pivotelement, d.h. jenes Element $a_{\ell k}$ mit

$$|a_{\ell k}| = \max \{|a_{ik}| : k \leq i \leq n\},$$

und vertauscht die entsprechenden Zeilen. Die Spalten-Pivotsuche mit Maximums-Strategie sichert, daß die komponentenweise Abschätzung $|L| \leq 1$ erfüllt ist.

Bemerkung: Sollte bei der Durchführung des Gaußschen Eliminationsverfahrens mit Spaltenpivotsuche im k -ten Eliminationsschritt der Fall $a_{ik} = 0$ für alle $k \leq i \leq n$ eintreten, ist die Voraussetzung $A \in \mathbb{K}^{n \times n}$ invertierbar verletzt und man bricht das Gaußsche Eliminationsverfahren ab.

Vollständige Pivotsuche mit Maximums-Strategie: Bei der vollständigen Pivotsuche mit Maximums-Strategie wählt man im k -ten Eliminationsschritt das betragsmäßig größte Element der Koeffizienten $a_{i\ell}$ für $k \leq i, \ell \leq n$ und vertauscht die entsprechenden Zeilen und Spalten (entspricht einer Umnummerierung der Unbekannten). Da die vollständige Pivotsuche vergleichsweise aufwendig ist, wird sie selten angewendet.

- **Bemerkung:** Überlegungen in Abschnitt 4.5 haben gezeigt, daß das Gaußsche Eliminationsverfahren bzw. die Lösung des linearen Gleichungssystems mittels LR-Zerlegung numerisch stabil ist, wenn

$$|r| = |\tilde{A} - A| |\tilde{x}| \leq \mathcal{O}(\varepsilon_{\text{mach}}) |\tilde{L}| |\tilde{R}| |\tilde{x}| \leq \varepsilon_{\text{mach}} (|A| |\tilde{x}| + |b|),$$

wobei \tilde{L} und \tilde{R} die berechneten Matrizen bei Durchführung der LR-Zerlegung bezeichnen.

Schlußfolgerung: Bei günstiger Wahl der Pivotelemente sind die Koeffizienten der berechneten Matrix $|\tilde{L}| |\tilde{R}| \approx |L| |R|$ minimal.

Aber! Es gibt Situationen in denen die Wahl des betragsmäßig größten Elementes nicht ideal ist.

Beispiel:

- Für den Spezialfall $A \in \mathbb{R}^{2 \times 2}$ führt das Gaußsche Eliminationsverfahren in natürlicher Reihenfolge (unter Annahme $a \neq 0$) auf die LR-Zerlegung

$$A = LR,$$

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 \\ \frac{c}{a} & 1 \end{pmatrix}, \quad R = \begin{pmatrix} a & b \\ 0 & \frac{ad-bc}{a} \end{pmatrix},$$

$$|A| = |LR| = \begin{pmatrix} |a| & |b| \\ |c| & |d| \end{pmatrix}, \quad |L||R| = \begin{pmatrix} |a| & |b| \\ |c| & \frac{|b||c|}{|a|} + \frac{|ad-bc|}{|a|} \end{pmatrix}.$$

Eine kurze Rechnung ergibt (Fallunterscheidung $|a||b| = ab$ oder $|a||b| = -ab$)

$$d = 0: \quad \alpha = \frac{|b||c|}{|a|} + \frac{|ad-bc|}{|a|} = 2 \frac{|b||c|}{|a|},$$

$$d \neq 0: \quad \alpha = \frac{|b||c|}{|a|} + \frac{|ad-bc|}{|a|} = |d| \left(\frac{|b||c|}{|a||d|} + \frac{|ad-bc|}{|a||d|} \right) = |d| \left(\frac{|b||c|}{|a||d|} + \left| \frac{ad}{|a||d|} - \frac{bc}{|a||d|} \right| \right)$$

$$= |d| \left(\frac{|b||c|}{|a||d|} + \left| 1 - \frac{bc}{ad} \right| \right) = |d| (|x| + |1-x|), \quad x = \frac{bc}{ad}.$$

Die Funktion $f: \mathbb{R} \rightarrow \mathbb{R}: x \mapsto |1-x| + |x|$ hat für $0 \leq x \leq 1$ den konstanten Wert 1 und wächst ansonsten an. Falls $0 \leq \frac{bc}{ad} \leq 1$ oder $0 \leq \left| \frac{bc}{ad} \right| \leq 1$, folgt $\alpha \leq 3$ und somit

$$|bc| \leq |ad|: \quad |L||R| \approx |LR|,$$

d.h. das Verfahren ist numerisch stabil. Falls hingegen

$$|bc| \gg |ad|: \quad |L||R| \gg |LR|$$

ist das Verfahren numerisch instabil.

- Beachte, daß das Stabilitätskriterium **skalierungsinvariant** ist, d.h. Skalierungen der Zeilen oder Spalten von A bewirken keine Änderung der Größe $\frac{bc}{ad}$, denn

$$DA = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} d_1 a & d_1 b \\ d_2 c & d_2 d \end{pmatrix}, \quad \frac{(d_1 b)(d_2 c)}{(d_1 a)(d_2 d)} = \frac{bc}{ad},$$

$$AD = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} = \begin{pmatrix} d_1 a & d_2 b \\ d_1 c & d_2 d \end{pmatrix}, \quad \frac{(d_2 b)(d_1 c)}{(d_1 a)(d_2 d)} = \frac{bc}{ad}.$$

- Obige Überlegungen zeigen, daß bei einer günstigen Pivotalwahl die natürliche Reihenfolge der Zeilen belassen wird, falls $|bc| \leq |ad|$, und ansonsten die beiden Zeilen der Matrix vertauscht werden. Es gibt jedoch Situationen, in denen eine Spaltenpivotsuche (nicht skalierungsinvariant) zu einer ungünstigen Pivotalwahl führt.

- **Beispiel von Dahlquist und Björck**, vgl. **Illustration4_Pivotwahl**.
- *Bis heute ist keine Methode bekannt, wie die in numerischer Hinsicht beste Pivotwahl getroffen werden kann — bzw. keine praktikable Skalierungsmethode bekannt, bei der die Maximums-Strategie immer gut wäre.*

Übliche Strategie: Äquilibrierung der Matrix A , d.h. Skalierung von A

$$A \rightarrow \hat{A} = DA = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} d_1 a_{11} & \dots & d_1 a_{1n} \\ \vdots & & \vdots \\ d_n a_{n1} & \dots & d_n a_{nn} \end{pmatrix}$$

derart, daß die Zeilenbetragssumme konstant ist

$$\sum_{j=1}^n |\hat{a}_{ij}| = 1, \quad 1 \leq i \leq n \iff d_i = \left(\sum_{j=1}^n |a_{ij}| \right)^{-1}, \quad 1 \leq i \leq n$$

und dann Anwendung der Spalten-Pivotsuche mit Maximums-Strategie. Dies entspricht der folgenden Pivotwahl im k -ten Eliminationsschritt

$$|a_{\ell k}^{(k)}| = \max_{1 \leq i \leq n} \frac{d_i}{d_\ell} |a_{ik}^{(k)}|.$$

5. Lineare Ausgleichsrechnung

- **Problemstellung:** Näherungsweise Lösung eines **überbestimmten linearen Gleichungssystems** (mehr Gleichungen als Unbekannte, im Allgemeinen existiert keine Lösung)

$$Ax = b, \quad A \in \mathbb{K}^{m \times n}, \quad x \in \mathbb{K}^n, \quad b \in \mathbb{K}^m, \quad m > n.$$

Bestimme eine **Lösung des linearen Ausgleichsproblems**, d.h. eine Lösung im Sinn der **kleinsten Fehlerquadrate** (Gauß)

$$\|Ax - b\|_2 \xrightarrow{!} \min.$$

Falls $\|Ax - b\|_2 = 0 \Leftrightarrow Ax = b$ erfüllt ist, ist x tatsächlich eine Lösung des linearen Gleichungssystems.

5.1. Ein Beispiel

- **Polynominterpolation:** Bestimme das eindeutig bestimmte Polynom vom Grad $\leq m-1$

$$p: \mathbb{K} \rightarrow \mathbb{K}: x \mapsto \sum_{i=0}^{m-1} c_{i+1} x^i$$

durch m vorgegebene Datenpunkte $(x_j, y_j) \in \mathbb{K} \times \mathbb{K}$ für $1 \leq j \leq m$, d.h. bestimme die Koeffizienten $c \in \mathbb{K}^m$ des Polynoms durch Lösung des linearen Gleichungssystems

$$p(x_j) = c_1 + c_2 x_j + \dots + c_m x_j^{m-1} = y_j, \quad 1 \leq j \leq m,$$

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^{m-1} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}.$$

Vgl. **Illustration5_PolynomInterpolation** (Auswerten mittels Horner-Schema).

Vgl. Abbildung, Skriptum, S. 86. Für $m > 10$ und äquidistante Stützstellen sind Oszillationen des Interpolationspolynoms insbesondere an den Intervallenden typisch.

Bemerkung: Im Allgemeinen sind Konditionszahlen von **Vandermonde-Matrizen** groß. Eine bessere Alternative zur Berechnung der Koeffizienten des Interpolationspolynoms verwendet die Darstellung des Interpolationspolynoms nach Newton, vgl. Numerische Mathematik II.

- **Approximation mittels Ausgleichspolynom:** Bestimme ein Polynom

$$p: \mathbb{K} \rightarrow \mathbb{K}: x \mapsto \sum_{i=0}^{n-1} c_{i+1} x^i, \quad n < m,$$

durch m vorgegebene Datenpunkte $(x_j, y_j) \in \mathbb{K} \times \mathbb{K}$ für $1 \leq j \leq m$, d.h. bestimme die Koeffizienten $c \in \mathbb{K}^n$ des Polynoms durch Lösung des linearen Ausgleichsproblems

$$\|Ac - y\|_2 = \sum_{j=1}^m |p(x_j) - y_j|^2 \xrightarrow{!} \min,$$

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^{n-1} \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}.$$

Vgl. Abbildung, Skriptum, S. 87. Glatterer Verlauf des approximierenden Polynoms.

5.2. Normalgleichungen

- **Situation:** Näherungsweise Lösung des linearen Ausgleichsproblems

$$\|Ax - b\|_2 \xrightarrow{!} \min, \quad A \in \mathbb{K}^{m \times n}, \quad x \in \mathbb{K}^n, \quad b \in \mathbb{K}^m, \quad m > n.$$

- **Mittels Differentialrechnung:**

- Für $\mathbb{K} = \mathbb{C}$ betrachte das äquivalente reelle lineare Gleichungssystem der Dimension $2m \times 2n$.
- Verwende die äquivalente Formulierung

$$\|Ax - b\|_2^2 \xrightarrow{!} \min.$$

Bestimme das Minimum mittels Differenzieren (betrachte z.B. den Fall $m = n = 2$, für eine symmetrische Matrix $B \in \mathbb{R}^{2 \times 2}$ bestimme die erste und zweite Ableitung von $x^T B x = b_{11}x_1^2 + 2b_{12}x_1x_2 + b_{22}x_2^2$, beachte $f(x) = f(x_1, \dots, x_n) \in \mathbb{R}$ und folglich $f'(x) = (\partial_{x_1}f(x), \dots, \partial_{x_n}f(x)) \in \mathbb{R}^{1 \times n}$)

$$\begin{aligned} f(x) &= \|Ax - b\|_2^2 = (Ax - b)^T (Ax - b) = (x^T A^T - b^T)(Ax - b) \\ &= x^T A^T Ax - b^T Ax - x^T A^T b + b^T b \stackrel{x^T A^T b = b^T Ax \in \mathbb{R}}{=} x^T A^T Ax - 2b^T Ax + b^T b, \end{aligned}$$

$$f'(x) = 2x^T A^T A - 2b^T A \stackrel{!}{=} 0 \iff \underset{\text{Transponieren}}{A^T A x} = A^T b,$$

$$f''(x) = A^T A \quad \text{positiv semi-definit.}$$

Folglich erfüllt die Lösung des linearen Ausgleichsproblems die **Normalgleichungen**

$$A^T A x = A^T b$$

mit symmetrischer Matrix $A^T A$.

- Wenn alle Spalten von A linear unabhängig sind, d.h. die Matrix A vollen Rang hat, ist die Lösung der Normalgleichungen eindeutig bestimmt und die Hessematrix $f''(x) = A^T A$ positiv definit.

Denn: Verwende $x^T f''(x) x = \|Ax\|_2^2 = 0 \iff Ax = 0 \iff x = 0$ bzw. $x^T f''(x) x > 0$ für alle $x \neq 0$ und $A^T A x = 0 \iff A^T y = 0, y = Ax \iff 0 = y = Ax \iff x = 0$. \diamond

- **Mittels Ergebnissen der Linearen Algebra:**

- Es bezeichnen a_1, \dots, a_n die Spalten der Matrix

$$A = (a_1 | \dots | a_n) \in \mathbb{K}^{m \times n}.$$

Früheren Überlegungen zeigten

$$\text{Gleichungssystem } Ax = b \text{ lösbar} \iff b \in \mathcal{R}_A = \langle a_1, \dots, a_n \rangle \quad \text{bzw.}$$

$$\text{Gleichungssystem } Ax = b \text{ nicht lösbar} \iff b \notin \mathcal{R}_A = \langle a_1, \dots, a_n \rangle.$$

- Die Zerlegung von \mathbb{K}^m in den Unterraum \mathcal{R}_A und den dazu orthogonalen Unterraum U führt auf (wäre $w = b - A\hat{x}$ nicht orthogonal auf \mathcal{R}_A , wäre $\|A\hat{x} - b\|_2$ wegen der Dreiecksungleichung nicht minimal)

$$\min_{x \in \mathbb{K}^n} \|Ax - b\|_2 = \hat{x} \iff b = v + w, \quad v = A\hat{x} \in \mathcal{R}_A, \quad w \in U, \text{ d.h. } A^*w = 0.$$

Multiplikation von $A\hat{x} + w = b$ mit A^* führt auf die **Normalgleichungen**

$$\min_{x \in \mathbb{K}^n} \|Ax - b\|_2 = \hat{x} \implies A^*A\hat{x} = A^*b.$$

- Es bleibt zu zeigen, daß für eine Lösung $z \in \mathbb{K}^n$ der Normalgleichungen, d.h. $A^*Az = A^*b$, und einen beliebigen Vektor $x \in \mathbb{K}^n$ die Abschätzung

$$\|Az - b\|_2 \leq \|Ax - b\|_2 \iff \|Az - b\|_2^2 \leq \|Ax - b\|_2^2$$

gilt. Unter der Annahme $\|Ax - b\|_2^2 \leq \|Az - b\|_2^2$ für ein $x \in \mathbb{K}^n$ folgt

$$\begin{aligned} x^*A^*Ax - b^*Ax - x^*A^*b + b^*b &= (x^*A^* - b^*)(Ax - b) = \|Ax - b\|_2^2 \\ &\leq \|Az - b\|_2^2 = z^*A^*Az - b^*Az - z^*A^*b + b^*b \\ \iff x^*A^*Ax - \underbrace{b^*A}_{=z^*A^*A}x - x^*\underbrace{A^*b}_{=A^*Az} &\leq z^*A^*Az - \underbrace{b^*A}_{=z^*A^*A}z - z^*\underbrace{A^*b}_{=A^*Az} \\ \iff x^*A^*Ax - z^*A^*Ax - x^*A^*Az + z^*A^*Az &\leq 0 \\ \iff \|A(x - z)\|_2^2 = (x - z)^*A^*A(x - z) &\leq 0 \\ \iff Ax = Az, \end{aligned}$$

d.h. für Lösungen der Normalgleichungen und des linearen Ausgleichsproblems gilt die Identität $Az = Ax$. Insbesondere ist also das Residuum $Ax - b$ von Lösungen zum linearen Ausgleichsproblem eindeutig bestimmt.

- Falls die Matrix A vollen Rang hat, ist die Lösung des linearen Ausgleichsproblems (wegen $Az = Ax \Leftrightarrow x = y$) eindeutig bestimmt.

- **Resultat zur Lösung des linearen Ausgleichsproblems** (Satz 5.1): Das lineare Ausgleichsproblem ist äquivalent zu den Normalgleichungen

$$\min_{x \in \mathbb{K}^n} \|Ax - b\|_2 = \hat{x} \iff A^*A\hat{x} = A^*b.$$

Das Residuum ist eindeutig bestimmt, d.h. für zwei Lösungen $x_1, x_2 \in \mathbb{K}^n$ des linearen Ausgleichsproblems gilt $b - Ax_1 = b - Ax_2$. Falls die Matrix A vollen Rang hat, ist die Lösung des linearen Ausgleichsproblems eindeutig bestimmt.

- **Kondition des linearen Ausgleichsproblems:** Es bezeichne \hat{x} die Lösung des (eindeutig lösbaren) linearen Ausgleichsproblems und \hat{r} das entsprechende Residuum

$$\hat{x} = \min_{x \in \mathbb{K}^n} \|Ax - b\|_2, \quad \hat{r} = A\hat{x} - b.$$

Bei Änderungen α und β der Eingabedaten A und B gilt für den Fehler der zugehörigen Lösung $\hat{x} + \xi$ und des entsprechenden Residuums $\hat{r} + \varrho = (A + \alpha)(\hat{x} + \xi) - (b + \beta)$ folgende Abschätzung (ohne Begründung)

$$\|\xi\|_2 \leq \frac{\kappa_1}{\|A\|_2} F, \quad \|\varrho\|_2 \leq F,$$

$$F = \|\alpha\|_2 \|\hat{x}\|_2 + \|\beta\|_2 + \kappa_1 \frac{\|\alpha\|_2 \|\hat{r}\|_2}{\|A\|_2}, \quad \kappa_1 = \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\alpha\|_2}{\|A\|_2}},$$

Es ist zu beachten, daß im Vergleich mit einem linearen Gleichungssystem beim linearen Ausgleichsproblem das Quadrat der Kondition der Matrix A (jedoch in Kombination mit dem Residuum) auftritt.

5.3. Cholesky-Zerlegung

- **Vorbemerkung:** Falls die Matrix $A \in \mathbb{K}^{m \times n}$ mit $m > n$ vollen Rang hat, d.h. die Spalten von A linear unabhängig sind, ist die Lösung der Normalgleichungen

$$A^* A x = A^* b$$

eindeutig bestimmt. Beachte, daß die quadratische Matrix $B = A^* A \in \mathbb{K}^{n \times n}$ selbstadjungiert und positiv definit ist

$$B^* = (A^* A)^* = A^* A^{**} = A^* A = B, \quad x^* B x = x^* A^* A x = \|Ax\|_2^2 > 0 \quad \text{für } 0 \neq x \in \mathbb{K}^n.$$

Zur Lösung der Normalgleichungen werden üblicherweise

- das Cholesky-Verfahren (vgl. Abschnitt 5.3) oder
- Orthogonaltransformationen mittels Householder-Reflexionen (vgl. Abschnitt 5.4)

verwendet.

- **Situation:** Die Durchführung des Gaußschen Eliminationsverfahrens (Implementierung) bzw. der LR-Zerlegung (theoretische Überlegungen) unter den Voraussetzungen
 - $A \in \mathbb{K}^{n \times n}$ selbstadjungiert, d.h. $A^* = A$,
 - A positiv definit, d.h. $x^* A x > 0$ für alle $0 \neq x \in \mathbb{K}^n$,

führt auf die **Cholesky-Zerlegung** von A . Die effiziente Implementierung der Cholesky-Zerlegung einer Matrix beruht auf der direkten Berechnung der LR-Zerlegung der Matrix (s.u.).

Resultat: Unter den obigen Voraussetzungen an die Matrix A kann das Gaußsche Eliminationsverfahren in natürlicher Reihenfolge durchgeführt werden. Die Eigenschaften der Selbstadjungiertheit und positiven Definitheit bleiben in den entstehenden Teilmatrizen erhalten.

Denn:

- Im ersten Eliminationsschritt ist die Pivotwahl a_{11} möglich (Anwendung der positiven Definitheit von A mit $x = e_1$)

$$a_{11} = e_1^* A e_1 > 0.$$

Elimination in der ersten Spalte von (wegen $A^* = A$ hat A die angegebene Form)

$$A = A^{(1)} = \begin{pmatrix} a_{11} & \alpha^* \\ \alpha & \beta \end{pmatrix}, \quad \alpha \in \mathbb{K}^{n-1}, \quad \beta^* = \beta \in \mathbb{K}^{(n-1) \times (n-1)},$$

führt auf (z.B. Transformation 2. Zeile \rightarrow 2. Zeile $-\frac{\alpha_1}{a_{11}} \times$ 1. Zeile)

$$A^{(2)} = \begin{pmatrix} a_{11} & \alpha^* \\ 0 & B^{(2)} \end{pmatrix}, \quad B^{(2)} = \beta - \frac{1}{a_{11}} \alpha \alpha^* \in \mathbb{K}^{(n-1) \times (n-1)}.$$

Die entstehende Teilmatrix $B^{(2)}$ ist offensichtlich selbstadjungiert ($a_{11} \in \mathbb{R}$)

$$(B^{(2)})^* = (\beta - \frac{1}{a_{11}} \alpha \alpha^*)^* = \beta^* - \frac{1}{a_{11}} \alpha^{**} \alpha^* = \beta - \frac{1}{a_{11}} \alpha \alpha^* = B^{(2)}.$$

Es bleibt zu zeigen, daß $B^{(2)}$ positiv definit ist, d.h. für alle $0 \neq \xi \in \mathbb{K}^{n-1}$ gilt

$$\xi^* B^{(2)} \xi = \xi^* (\beta - \frac{1}{a_{11}} \alpha \alpha^*) \xi = \xi^* \beta \xi - \frac{1}{a_{11}} \xi^* \alpha \alpha^* \xi > 0.$$

Aus der positiven Definitheit von A folgt

$$\begin{aligned} 0 < x^* A x &= (x_1 \quad \xi^*) \begin{pmatrix} a_{11} & \alpha^* \\ \alpha & \beta \end{pmatrix} \begin{pmatrix} x_1 \\ \xi \end{pmatrix} = (x_1 \quad \xi^*) \begin{pmatrix} a_{11} x_1 + \alpha^* \xi \\ x_1 \alpha + \beta \xi \end{pmatrix} \\ &= a_{11} x_1^2 + x_1 (\alpha^* \xi + \xi^* \alpha) + \xi^* \beta \xi, \quad 0 \neq x = \begin{pmatrix} x_1 \\ \xi \end{pmatrix} \in \mathbb{K}^n, \end{aligned}$$

und speziell mit $x_1 = -\frac{1}{a_{11}} \alpha^* \xi$ ergibt sich die Behauptung

$$\begin{aligned} 0 < a_{11} x_1^2 + x_1 (\alpha^* \xi + \xi^* \alpha) + \xi^* \beta \xi &= \frac{1}{a_{11}} (\alpha^* \xi)^2 - \frac{1}{a_{11}} \alpha^* \xi (\alpha^* \xi + \xi^* \alpha) + \xi^* \beta \xi \\ &= -\frac{1}{a_{11}} \alpha^* \xi \xi^* \alpha + \xi^* \beta \xi = \xi^* B^{(2)} \xi. \end{aligned}$$

Für den Fall $\mathbb{K} = \mathbb{R}$ ist dies motiviert durch (wobei $z = \alpha^* \xi$, $(\Re z)^2 = |z|^2$ für $z \in \mathbb{R}$)

$$\begin{aligned} a_{11} x_1^2 + 2 \Re z x_1 = -\frac{1}{a_{11}} |z|^2 &\iff x_1^2 + \frac{2}{a_{11}} \Re z x_1 + \frac{1}{a_{11}^2} |z|^2 = 0 \\ &\iff x_1 = -\frac{1}{a_{11}} \Re z \pm \frac{1}{a_{11}} \sqrt{(\Re z)^2 - |z|^2} = -\frac{1}{a_{11}} \alpha^* \xi. \end{aligned}$$

– Die obigen Überlegungen für A lassen sich direkt auf $B^{(2)}$ anwenden.

Mittels Induktion ergibt sich die Behauptung des Resultates. \diamond

Bemerkungen:

- Für selbstadjungierte und positiv definite Matrizen ist das Gaußsche Eliminationsverfahren in natürlicher Reihenfolge ein numerisch stabiler Algorithmus (ohne Begründung).
- Da sämtliche entstehende Teilmatrizen selbstadjungiert sind, ist es ausreichend, die Koeffizienten in und oberhalb der Diagonale abzuspeichern, z.B. für $n = 3$ gilt

$$\begin{aligned} A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = A^* = \begin{pmatrix} \overline{a_{11}} & \overline{a_{21}} & \overline{a_{31}} \\ \overline{a_{12}} & \overline{a_{22}} & \overline{a_{32}} \\ \overline{a_{13}} & \overline{a_{23}} & \overline{a_{33}} \end{pmatrix} \\ \iff a_{11}, a_{22}, a_{33} \in \mathbb{R}, \quad a_{21} = \overline{a_{12}}, \quad a_{31} = \overline{a_{13}}, \quad a_{32} = \overline{a_{23}}. \end{aligned}$$

Die Anzahl der zur Durchführung des Gaußschen Eliminationsverfahren bzw. äquivalent dazu die Anzahl der zur Berechnung der LR-Zerlegung von $A \in \mathbb{K}^{n \times n}$ benötigten Operationen (Addition oder Multiplikation in \mathbb{K}) ist somit (vergleiche dies mit dem Aufwand für die LR-Zerlegung bzw. QR-Zerlegung einer allgemeinen quadratischen Matrix)

$$\mathcal{O}\left(\frac{1}{3}n^3\right).$$

Rationale Cholesky-Zerlegung und Cholesky-Zerlegung:

* Die geforderte Selbstadjungiertheit von $A \in \mathbb{K}^{n \times n}$ und der Ansatz

$$R = DS, \quad D = \begin{pmatrix} R_{11} & & \\ & \ddots & \\ & & R_{nn} \end{pmatrix}, \quad S = \begin{pmatrix} 1 & S_{12} & \dots & S_{1n} \\ & \ddots & & \vdots \\ & & \ddots & S_{n-1,n} \\ & & & 1 \end{pmatrix},$$

zeigt die Relation $S = L^*$. Somit führt die LR-Zerlegung von A auf die **rationale Cholesky-Zerlegung** von A

$$A = LDL^*,$$

$$D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad d_i > 0, \quad 1 \leq i \leq n,$$

$$L = \begin{pmatrix} 1 & & & \\ L_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ L_{n1} & \dots & L_{n,n-1} & 1 \end{pmatrix} \in \mathbb{K}^{n \times n}.$$

* Setzt man weiters

$$\hat{L} = LD^{\frac{1}{2}}, \quad D^{\frac{1}{2}} = \begin{pmatrix} \sqrt{d_1} & & \\ & \ddots & \\ & & \sqrt{d_n} \end{pmatrix} \in \mathbb{R}^{n \times n},$$

ergibt sich die **Cholesky-Zerlegung** von A

$$A = \hat{L}\hat{L}^*,$$

$$\hat{L} = \begin{pmatrix} \hat{L}_{11} & & \\ \vdots & \ddots & \\ \hat{L}_{n1} & \dots & \hat{L}_{nn} \end{pmatrix} \in \mathbb{K}^{n \times n}.$$

Die Cholesky-Zerlegung berechnet man direkt durch **sukzessive Bestimmung der Koeffizienten** von \hat{L} .

Vgl. **Illustration5_Cholesky**.

- **Anwendung der Cholesky-Zerlegung** zur Lösung der Normalgleichungen

$$A^* Ax = \widehat{L}\widehat{L}^* x = A^* b.$$

Die Berechnung der Cholesky-Zerlegung von $B = A^* A$ erfolgt gleichzeitig mit der Berechnung von $A^* A$. Ebenso wird die Berechnung von $A^* b$ gleichzeitig mit der Vorwärtssubstitution $\widehat{L}y = A^* b$ ausgeführt.

Bemerkung: Ein Nachteil des Cholesky-Verfahrens zur Lösung der Normalgleichungen liegt darin, daß die Konditionszahl der Matrix $B = A^* A$, d.h. das Quadrat der Konditionszahl der Matrix A auftritt. Falls die Gleichungen nahezu konsistent sind, ist das Residuum klein. In diesem Fall dürften sich Rundungsfehler wie $\kappa(A)$ (aber nicht wie $\kappa(A)^2$) auswirken, d.h. in diesem Fall ist der Algorithmus numerisch instabil. Eine stabile Alternative wird in Abschnitt 5.4 angegeben.

5.4. Lösung über Orthogonaltransformationen

- **Situation:** Es sei $A \in \mathbb{K}^{m \times n}$ mit $m > n$ eine Matrix von vollem Rang, d.h. die Spalten von A seien linear unabhängig.

Überlegungen: Die Triangulierung der Matrix $A \in \mathbb{K}^{m \times n}$ mittels Householder-Reflexionen $T_1, \dots, T_n \in \mathbb{K}^{m \times m}$ führt auf ein äquivalentes lineares Gleichungssystem

$$Ax = b \iff Rx = \underbrace{T_n \cdots T_1}_{=Q^*} Ax = T_n \cdots T_1 b = c,$$

vgl. Abschnitt 4.2. Diese Transformation entspricht der Bestimmung der (vollen) QR-Zerlegung von A mit unitärer Matrix $Q = (T_n \cdots T_1)^* = T_1^* \cdots T_n^* = T_1 \cdots T_n \in \mathbb{K}^{m \times m}$ (beachte die Selbstadjungiertheit von T_i für $1 \leq i \leq n$) und oberer Dreiecksmatrix $R \in \mathbb{K}^{m \times n}$

$$A = QR,$$

$$Q = T_1 \cdots T_n \in \mathbb{K}^{m \times m}, \quad Q^* Q = I, \quad R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} \in \mathbb{K}^{m \times n}.$$

Da die Transformation mittels einer unitären Matrix die euklidische Norm erhält

$$\|Ax - b\|_2 = \|Q^* Ax - Q^* b\|_2 = \|Rx - Q^* b\|_2,$$

gilt für die Lösung des linearen Ausgleichsproblems

$$\|Ax - b\|_2 \stackrel{!}{\rightarrow} \min \iff \|Rx - Q^* b\|_2 \stackrel{!}{\rightarrow} \min.$$

Dies entspricht (wobei $R \in \mathbb{K}^{m \times n}$, $\hat{R} \in \mathbb{K}^{n \times n}$, $x \in \mathbb{K}^n$, $c = Q^* b \in \mathbb{K}^m$, $\hat{c} \in \mathbb{K}^n$, $\tilde{c} \in \mathbb{K}^{m-n}$)

$$Rx - c = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} x - \begin{pmatrix} \hat{c} \\ \tilde{c} \end{pmatrix} = \begin{pmatrix} \hat{R}x - \hat{c} \\ -\tilde{c} \end{pmatrix},$$

$$\|Rx - Q^* b\|_2 = \|\hat{R}x - \hat{c}\|_2 + \|\tilde{c}\|_2 \stackrel{!}{\rightarrow} \min \iff \|\hat{R}x - \hat{c}\|_2 \stackrel{!}{\rightarrow} \min.$$

Aufgrund der geforderten Rangbedingung an A ist die quadratische Teilmatrix $\hat{R} \in \mathbb{K}^{n \times n}$ invertierbar. Deshalb entspricht die Lösung des linearen Ausgleichsproblems gerade der Lösung eines linearen Gleichungssystems ($\|\hat{R}x - \hat{c}\|_2 = 0 \Leftrightarrow \hat{R}x = \hat{c}$)

$$\|Ax - b\|_2 \stackrel{!}{\rightarrow} \min \iff \hat{R}x = \hat{c}.$$

6. Eigenwerte und SVD (Überblick)

- Die näherungsweise **Lösung des Eigenwertproblems** ist eine wichtige Grundaufgabe der Numerischen Linearen Algebra und findet beispielsweise Anwendung bei
 - Verfahren zur Lösung gewöhnlicher Differentialgleichungen, beispielsweise bei Stabilitätsuntersuchungen für dynamische Systeme,
 - Verfahren zur Lösung partieller Differentialgleichungen, beispielsweise bei der Berechnung von *Basislösungen* (Separationsansatz, Fourieranalyse, Stationäre Lösungen).
- **Gekoppelte Schwingungen**, vgl. Skriptum, S. 93.
- **Quantenmechanische Vielteilchenzustände**, vgl. Berechnungen von A. Läuchli am Supercomputer MACH der Universitäten Innsbruck und Linz.

6.1. Theoretischer Hintergrund

- **Eigenwerte und Eigenvektoren** (Definition 6.1): Für eine quadratische Matrix $A \in \mathbb{K}^{n \times n}$ heißt $\lambda \in \mathbb{C}$ ein **Eigenwert** von A und $0 \neq v \in \mathbb{C}^n$ ein zugehöriger **Eigenvektor**, wenn folgende Relation gilt

$$Av = \lambda v.$$

Der zum Eigenwert λ zugehörige **Eigenraum** ist gegeben durch (Unterraum von \mathbb{C}^n durch Hinzunahme von $v = 0$)

$$\mathcal{N}_{A-\lambda I} = \{v \in \mathbb{C}^n : (A - \lambda I)v = 0\}.$$

- **Bemerkungen:**

- Für theoretische Überlegungen wird verwendet, daß die Eigenwerte von A Nullstellen des **charakteristischen Polynoms** $\chi: \mathbb{K} \rightarrow \mathbb{K}$ sind (wegen $Av = \lambda v, v \neq 0 \Leftrightarrow (A - \lambda I)v = 0, v \neq 0 \Leftrightarrow \det(A - \lambda I) = 0$)

$$\lambda \text{ Eigenwert von } A \iff \chi(\lambda) = 0, \quad \chi(\lambda) = \det(A - \lambda I) = \sum_{i=0}^n c_i \lambda^i, \quad c_n = (-1)^n.$$

Der Fundamentalsatz der Algebra besagt, daß ein Polynom vom Grad n mit Koeffizienten in \mathbb{K} genau n komplexe Nullstellen besitzt (mit Vielfachheit gezählt) und folglich besitzt eine Matrix $A \in \mathbb{K}^{n \times n}$ genau n komplexe Eigenwerte (mit Vielfachheit gezählt).

Die Vielfachheit eines Eigenwertes bezeichnet man als **algebraische Multiplizität** $\mu(\lambda)$ des Eigenwertes und die Dimension des zugehörigen Eigenraumes als seine **geometrische Multiplizität** $\nu(\lambda)$. Es gilt $\nu(\lambda) \leq \mu(\lambda)$ (s.u.).

Beispiel: Die Matrix

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{2 \times 2}$$

besitzt den Eigenwert $\lambda = 1 \in \mathbb{R}$ mit algebraischer Multiplizität $\mu(1) = 2$. Wegen

$$(A - \lambda I)v = 0, \quad v \neq 0 \iff \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \iff v = v_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad v_1 \neq 0$$

sind alle Vielfachen des ersten Standardbasisvektors zugehörige Eigenvektoren. Folglich ist der zugehörige Eigenraum gegeben durch

$$\mathcal{N}_{A-\lambda I} = \langle e_1 \rangle \subset \mathbb{R}^2,$$

d.h. die geometrische Multiplizität des Eigenwertes ist

$$\nu(1) = 1 \leq \mu(1) = 2.$$

- Die Menge aller Eigenwerte einer Matrix $A \in \mathbb{K}^{n \times n}$ heißt **Spektrum** von A und der betragsmäßig größte Eigenwert gibt den **Spektralradius** der Matrix an

$$\sigma = \{\lambda \in \mathbb{C} : \lambda \text{ Eigenwert von } A\}, \quad \rho = \sup \{|\lambda| : \lambda \in \sigma\}.$$

- Falls für alle Eigenwerte der Matrix $A \in \mathbb{K}^{n \times n}$ algebraische und geometrische Multiplizität übereinstimmen

$$\mu(\lambda) = \nu(\lambda),$$

gibt es eine Basis v_1, \dots, v_n des \mathbb{C}^n , die von zugehörigen Eigenvektoren gebildet wird. Wegen

$$AV = V\Lambda,$$

$$A \underbrace{(v_1 | \dots | v_n)}_{=V \in \mathbb{C}^{n \times n}} = (Av_1 | \dots | Av_n) = (\lambda_1 v_1 | \dots | \lambda_n v_n) = \underbrace{(v_1 | \dots | v_n)}_{=V \in \mathbb{C}^{n \times n}} \underbrace{\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}}_{=\Lambda \in \mathbb{C}^{n \times n}},$$

ist die Matrix **diagonalisierbar** mit zugehöriger **Diagonalisierung** $\Lambda \in \mathbb{C}^{n \times n}$ (aufgrund der linearen Unabhängigkeit der Basisvektoren ist V invertierbar)

$$AV = V\Lambda \iff A = V\Lambda V^{-1} \iff \Lambda = V^{-1}AV.$$

Falls die Matrix $A \in \mathbb{K}^{n \times n}$ **unitär diagonalisierbar** ist, d.h. eine Orthonormalbasis aus Eigenvektoren existiert, folgt insbesondere (wegen $V^{-1} = V^*$)

$$AV = V\Lambda \iff A = V\Lambda V^* \iff \Lambda = V^*AV.$$

- Die Transformation

$$F_T : \mathbb{K}^{n \times n} \rightarrow \mathbb{K}^{n \times n} : A \mapsto T^{-1}AT$$

mit invertierbarer Matrix $T \in \mathbb{K}^{n \times n}$ heißt eine **Ähnlichkeitstransformation**. Dies entspricht einem Basiswechsel im Definitionsbereich und Bildbereich (verwende $y = Ax$, $\tilde{x} = Tx$, $\tilde{y} = Ty = TAx = TAT^{-1}\tilde{x}$).

Die Matrix A und die transformierte Matrix $T^{-1}AT$ besitzen dieselben Eigenwerte und mittels F_T transformierte Eigenräume

$$\begin{aligned} Av = \lambda v, \quad 0 \neq v \in \mathcal{N}_{A-\lambda I} &= \{v \in \mathbb{K}^n : (A - \lambda I)v = 0\} \\ \iff ATw = \lambda Tw, \quad 0 \neq v \in \mathcal{N}_{A-\lambda I}, \quad 0 \neq w = T^{-1}v \\ \iff T^{-1}ATw = \lambda w, \quad 0 \neq w \in \mathcal{N}_{T^{-1}AT-\lambda I} &= \{w \in \mathbb{K}^n : (T^{-1}AT - \lambda I)w = 0\}. \end{aligned}$$

- Nicht jede (quadratische) Matrix ist diagonalisierbar, jedoch kann jede (quadratische) Matrix mittels einer **unitären Ähnlichkeitstransformation auf Dreiecksgestalt** transformiert werden.

Lemma von Schur (Satz 6.2): Für jede Matrix $A \in \mathbb{K}^{n \times n}$ mit (beliebig angeordneten) Eigenwerten $\lambda_1, \dots, \lambda_n \in \mathbb{C}$ gibt es eine unitäre Matrix $T \in \mathbb{C}^{n \times n}$ (d.h. $T^*T = I$) derart, daß

$$T^*AT = S, \quad S = \begin{pmatrix} \lambda_1 & S_{12} & \dots & S_{1n} \\ & \ddots & & \vdots \\ & & \ddots & S_{n-1,n} \\ & & & \lambda_n \end{pmatrix}.$$

Denn: Es sei $\lambda_1 \in \mathbb{C}$ ein Eigenwert der Matrix $A \in \mathbb{K}^{n \times n}$ und $0 \neq v_1 \in \mathbb{C}^n$ ein zugehöriger Eigenvektor

$$Av_1 = \lambda_1 v_1,$$

wobei zusätzlich $\|v_1\|_2 = 1$ und $v_{11} \geq 0$ (v_{11} bezeichne die erste Komponente von v_1) angenommen wird. Frühere Überlegungen zeigten, wie eine Householder-Reflexion $T_1 \in \mathbb{C}^{n \times n}$ konstruiert werden kann, sodaß (Elimination in der ersten Spalte, verwende die Relation $T_1^{-1} = T_1^* = T_1$, explizite Angabe von T_1 wird nicht benötigt)

$$T_1 v_1 = e_1, \quad T_1 e_1 = T_1^{-1} e_1 = v_1.$$

Mit Hilfe der Relation

$$e_i^T B e_j = \sum_{k,\ell=1}^n \delta_{ik} B_{k\ell} \delta_{j\ell} = b_{ij}, \quad 1 \leq i, j \leq n,$$

folgt somit

$$(T_1 A T_1^{-1})_{i1} = e_i^T T_1 A T_1^{-1} e_1 = e_i^T T_1 A \underbrace{T_1 e_1}_{=v_1} = e_i^T T_1 \underbrace{A v_1}_{=\lambda_1 v_1} = \lambda_1 e_i^T \underbrace{T_1 v_1}_{=e_1} = \lambda_1 \delta_{i1},$$

$$T_1 A T_1^{-1} = \begin{pmatrix} \lambda_1 & * & \dots & * \\ 0 & & & \\ \vdots & & A_2 & \\ 0 & & & \end{pmatrix}, \quad A_2 \in \mathbb{C}^{(n-1) \times (n-1)}.$$

Da A und $T_1 A T_1^{-1}$ ähnliche Matrizen sind, besitzen sie dieselben Eigenwerte. Weiters gilt für das charakteristische Polynom (Determinantenentwicklungssatz)

$$\chi(\lambda) = \det(A - \lambda I) = \det(T_1 A T_1^{-1} - \lambda I) = (\lambda_1 - \lambda) \det(A_2 - \lambda I),$$

d.h. die restlichen Eigenwerte $\lambda_2, \dots, \lambda_n$ von A sind Eigenwerte von A_2 . Induktiv folgt mittels der Transformationsmatrizen

$$T_2 = \begin{pmatrix} 1 & \\ & \tilde{T}_2 \end{pmatrix}, \quad T_3 = \begin{pmatrix} 1 & & \\ & 1 & \\ & & \tilde{T}_3 \end{pmatrix}, \quad \text{etc.,}$$

die Relation

$$\underbrace{T_{n-1} \cdots T_1}_{=T^*} A \underbrace{T_1^{-1} \cdots T_{n-1}^{-1}}_{=T_1 \cdots T_{n-1}=T} = \begin{pmatrix} \lambda_1 & * & \cdots & * \\ & \ddots & & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{pmatrix}$$

und damit die Behauptung. \diamond

Bemerkung: Das Lemma von Schur wird vorwiegend für theoretische Überlegungen verwendet (s.u.). Die Berechnung der Matrizen S, T (entsprechend der Herleitung) benötigt die Kenntnis der Eigenwerte von A und zugehöriger Eigenvektoren.

Folgerung: Es sei $\lambda \in \mathbb{C}$ ein Eigenwert von $A \in \mathbb{K}^{n \times n}$ mit algebraischer Multiplizität $\mu(\lambda)$ und geometrischer Multiplizität $\nu(\lambda)$. Es gilt

$$\nu(\lambda) \leq \mu(\lambda).$$

Denn: Nach dem Lemma von Schur reicht es aus, die unitär transformierte Matrix

$$T^* A T = S, \quad S = \begin{pmatrix} \lambda_1 & S_{12} & \cdots & S_{1n} \\ & \ddots & & \vdots \\ & & \ddots & S_{n-1,n} \\ & & & \lambda_n \end{pmatrix},$$

zu betrachten, wobei die Eigenwerte $\lambda_1, \dots, \lambda_n$ von A derart angeordnet sein sollen, daß $\lambda_1 = \lambda, \dots, \lambda_{\mu(\lambda)} = \lambda$. Betrachte beispielsweise den Fall

$$\mu(\lambda) = n = 3$$

und bestimme den zugehörigen Eigenraum durch Lösung des linearen Gleichungssystems

$$(T^* A T - \lambda I)v = \begin{pmatrix} 0 & S_{12} & S_{13} \\ & 0 & S_{23} \\ & & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} S_{12}v_2 + S_{13}v_3 \\ S_{23}v_3 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Die Unterscheidung der folgenden Fälle

$$\begin{aligned} S_{12} = 0, \quad S_{13} = 0, \quad S_{23} = 0 &\implies v_1, v_2, v_3 \text{ beliebig} \implies \nu(\lambda) = 3, \\ S_{12} \neq 0, \quad S_{13} = 0, \quad S_{23} = 0 &\implies v_2 = 0, v_1, v_3 \text{ beliebig} \implies \nu(\lambda) = 2, \\ S_{12} = 0, \quad S_{13} \neq 0, \quad S_{23} = 0 &\implies v_3 = 0, v_1, v_2 \text{ beliebig} \implies \nu(\lambda) = 2, \\ S_{12} = 0, \quad S_{13} = 0, \quad S_{23} \neq 0 &\implies v_3 = 0, v_1, v_2 \text{ beliebig} \implies \nu(\lambda) = 2, \\ S_{12} \neq 0, \quad S_{13} \neq 0, \quad S_{23} = 0 &\implies v_2 = -\frac{S_{13}}{S_{12}}v_3, v_1, v_3 \text{ beliebig} \implies \nu(\lambda) = 2, \\ S_{12} \neq 0, \quad S_{13} = 0, \quad S_{23} \neq 0 &\implies v_2 = v_3 = 0, v_1 \text{ beliebig} \implies \nu(\lambda) = 1, \\ S_{12} = 0, \quad S_{13} \neq 0, \quad S_{23} \neq 0 &\implies v_3 = 0, v_1, v_2 \text{ beliebig} \implies \nu(\lambda) = 2, \\ S_{12} \neq 0, \quad S_{13} \neq 0, \quad S_{23} \neq 0 &\implies v_2 = 0, v_3 = 0, v_1 \text{ beliebig} \implies \nu(\lambda) = 1, \end{aligned}$$

zeigt $\nu(\lambda) \leq \mu(\lambda)$. Ähnlich Überlegungen zeigen die Behauptung des Resultates im allgemeinen Fall. \diamond

- Eine quadratische Matrix $A \in \mathbb{K}^{n \times n}$ heißt **normal**, wenn

$$A^*A = AA^*.$$

Spezialfälle:

- Reelle symmetrische Matrizen

$$A^* = A^T = A \implies A^*A = A^2 = AA^*.$$

- Selbstadjungierte Matrizen

$$A^* = A \implies A^*A = A^2 = AA^*.$$

- Reelle **antisymmetrische Matrizen** (d.h. $A^T = -A$)

$$A^* = A^T = -A \implies A^*A = -A^2 = AA^*.$$

- Reelle orthogonale Matrizen

$$A^* = A^T = A^{-1} \implies A^*A = I = AA^*.$$

- Unitäre Matrizen

$$A^* = A^{-1} \implies A^*A = I = AA^*.$$

Resultat zur unitären Diagonalisierbarkeit (Korollar 6.3): Eine Matrix $A \in \mathbb{K}^{n \times n}$ ist genau dann normal, wenn es eine unitäre Matrix $T \in \mathbb{C}^{n \times n}$ gibt, derart daß

$$T^*AT = \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}.$$

Folglich besitzt eine normale Matrix A ein vollständiges System orthonormaler Eigenvektoren und insbesondere gilt $\mu(\lambda) = \nu(\lambda)$ für alle Eigenwerte λ von A .

Denn: Mittels Lemma von Schur folgt die Darstellung

$$T^*AT = S$$

mit unitärer Matrix T und oberer Dreiecksmatrix S und folglich (wegen $T^{-1} = T^*$)

$$A = T^{*-1}ST^{-1} = TST^*,$$

$$A^* = (TST^*)^* = T^{**}S^*T^* = TS^*T^*,$$

$$TS^*T^* = TS^*T^*TST^* = A^*A = AA^* = TST^*TS^*T^* = TSS^*T^* \Leftrightarrow S^*S = SS^*.$$

Die schrittweise Betrachtung der Koeffizienten der Differenz

$$S^*S - SS^* = 0$$

impliziert $S_{ij} = 0$ für $1 \leq i \leq i+1 \leq j \leq n$ und damit $S = \Lambda$ mit Diagonalmatrix Λ . Ist andererseits $S = \Lambda$ und $T^*AT = \Lambda$ so folgt $A^*A = T\Lambda^2T^* = AA^*$. \diamond

Folgerung für selbstadjungierte Matrizen:

- Eine selbstadjungierte Matrix $A \in \mathbb{K}^{n \times n}$ ist unitär diagonalisierbar und alle Eigenwerte sind reell.
- Eine selbstadjungierte Matrix $A \in \mathbb{K}^{n \times n}$ ist genau dann positiv definit, wenn alle Eigenwerte positiv sind.

Denn: Eine selbstadjungierte Matrix (d.h. $A^* = A$) ist insbesondere normal und damit unitär diagonalisierbar

$$T^*AT = \Lambda, \quad A = T\Lambda T^*.$$

Damit folgt

$$\begin{aligned} T\Lambda^*T^* &= T^{**}\Lambda^*T^* = (T\Lambda T^*)^* = A^* = A = T\Lambda T^* \\ \iff \Lambda^* &= \Lambda \iff \overline{\lambda_i} = \lambda_i, \quad 1 \leq i \leq n \iff \lambda_i \in \mathbb{R}, \quad 1 \leq i \leq n. \end{aligned}$$

Die Wahl $0 \neq x = v_i = T_{-i}$ für $1 \leq j \leq n$ zeigt, daß die Matrix positiv definit ist, wenn

$$0 < x^*Ax = v_i^* \underbrace{Av_i}_{=\lambda_i v_i} = \lambda_i \|v_i\|_2^2 \iff \lambda_i > 0, \quad 1 \leq i \leq n.$$

Gilt andererseits $\lambda_i > 0$ für alle $1 \leq i \leq n$, so folgt für jedes Element $0 \neq x \in \mathbb{K}^n$ mittels der Darstellung bezüglich der Orthonormalbasis v_1, \dots, v_n

$$0 \neq x = \sum_{i=1}^n c_i v_i, \quad v_i = T_{-i} \quad 1 \leq i \leq n,$$

die Relation

$$x^*Ax = \sum_{i=1}^n c_i x^*Av_i = \sum_{i=1}^n c_i \lambda_i x^*v_i = \sum_{i,j=1}^n c_i \overline{c_j} \lambda_i \underbrace{v_j^*v_i}_{=\delta_{ij}} = \sum_{i=1}^n |c_i|^2 \lambda_i > 0.$$

Dies zeigt die Behauptung. \diamond

- **Resultat zur Kondition der Eigenwertberechnung** einer normalen Matrix (Satz 6.4):
Es sei $A \in \mathbb{K}^{n \times n}$ eine normale Matrix (d.h. $A^*A = AA^*$) mit Eigenwerten $\lambda_1, \dots, \lambda_n \in \mathbb{C}$. Dann gilt für die entsprechenden Eigenwerte $\tilde{\lambda}_1, \dots, \tilde{\lambda}_n$ der Matrix $A + \alpha$ die folgende Abschätzung (ohne Begründung)

$$|\tilde{\lambda}_j - \lambda_j| \leq \|\alpha\|_2.$$

Somit ist die Eigenwertberechnung einer normalen Matrix sehr gut konditioniert.

6.2. Singulärwertzerlegung

- **Vorüberlegungen:**

- Betrachte eine Matrix $A \in \mathbb{K}^{m \times n}$ mit $m \geq n$ und $\text{rg}(A) = k$ mit $1 \leq k \leq n$. Die Matrix $B = A^* A \in \mathbb{K}^{n \times n}$ ist selbstadjungiert und positiv semi-definit

$$B^* = (A^* A)^* = A^* A^{**} = A^* A = B, \quad x^* B x = x^* A^* A x = \|Ax\|_2^2 \geq 0, \quad x \in \mathbb{K}^n.$$

Insbesondere ist die Matrix $B = A^* A$ unitär diagonalisierbar mit nicht-negativen (reellen) Eigenwerten, d.h. es existiert eine unitäre Matrix $T \in \mathbb{C}^{n \times n}$ derart, daß

$$T^* A^* A T = \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}, \quad T^* T = I, \quad \lambda_i = \sigma_i^2 \geq 0, \quad 1 \leq i \leq n.$$

- Wegen $\text{rg}(B) = \text{rg}(A^* A) = \text{rg}(A) = k$ (verwende die Relation $\text{rg}(A) = n - \dim \mathcal{N}_A$ sowie $\text{rg}(A^* A) = n - \dim \mathcal{N}_{A^* A} = n - \dim \mathcal{N}_A = \text{rg}(A)$) folgt nach eventueller Umordnung der Eigenwerte (und entsprechender Anpassung von T)

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k > 0, \quad \lambda_{k+1} = \dots = \lambda_n = 0.$$

- Die Relation

$$T^* A^* A T = \Lambda \quad \underset{S=AT}{\iff} \quad S^* S = \Lambda = \begin{pmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_k & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix}$$

zeigt, daß die Spalten der Matrix $AT = S = (s_1 | \dots | s_n) \in \mathbb{C}^{m \times n}$ orthogonal aufeinander stehen, d.h. $s_j^* s_i = 0$ für $1 \leq i, j \leq n$ mit $i \neq j$ (die Spalten sind nicht notwendigerweise normiert, es gilt $s_i^* s_i = \lambda_i \geq 0$). Zusätzliche Normierung der ersten k Spalten von S

$$u_i = \frac{1}{\sqrt{\lambda_i}} s_i = \frac{1}{\sigma_i} s_i \in \mathbb{C}^m, \quad 1 \leq i \leq k,$$

und Ergänzung mit orthonormalen Vektoren $u_{k+1}, \dots, u_n \in \mathbb{C}^m$ führt auf die Relation

$$AT = S = (\sigma_1 u_1 | \dots | \sigma_k u_k | \sigma_{k+1} u_{k+1} | \dots | \sigma_n u_n) = (u_1 | \dots | u_n) \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & & \sigma_n \end{pmatrix}$$

und weiters auf die **reduzierte Singulärwertzerlegung** von A

$$A = \widehat{U} \widehat{\Sigma} T^*,$$

$$A = (a_1 | \cdots | a_n) \in \mathbb{K}^{m \times n}, \quad T \in \mathbb{C}^{n \times n}, \quad T^* T = I,$$

$$\widehat{U} = (u_1 | \cdots | u_n) \in \mathbb{C}^{m \times n}, \quad \widehat{U}^* \widehat{U} = I, \quad \widehat{\Sigma} = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} \in \mathbb{R}^{n \times n}$$

Die (volle) **Singulärwertzerlegung** von A ergibt sich bei Ergänzung von \widehat{U} zu einer Orthonormalbasis des \mathbb{C}^m und entsprechender Ergänzung von $\widehat{\Sigma}$ um Nullzeilen

$$A = U \Sigma T^*,$$

$$A = (a_1 | \cdots | a_n) \in \mathbb{K}^{m \times n}, \quad T \in \mathbb{C}^{n \times n}, \quad T^* T = I,$$

$$U = (u_1 | \cdots | u_m) \in \mathbb{C}^{m \times m}, \quad U^* U = I, \quad \Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{m \times n},$$

vgl. Skriptum, S. 100.

Bemerkungen:

- Die **Erweiterung** auf den Fall $m < n$ ergibt sich durch Betrachtung der Singulärwertzerlegung der adjungierten Matrix $A^* = U \Sigma V^* \in \mathbb{K}^{n \times m}$ mit $n \geq m$ und Adjunktion $A = (U \Sigma V^*)^* = V \Sigma^T U^* \in \mathbb{K}^{m \times n}$.
- Wie üblich werden von nun an die Bezeichnungen $A = U \Sigma V^*$ verwendet.

Resultat zur Singulärwertzerlegung (Satz 6.5): Jede Matrix $A \in \mathbb{K}^{m \times n}$ mit Rang $\text{rg}(A) = k$, wobei $1 \leq k \leq \min\{m, n\}$, besitzt die Darstellung

$$A = U \Sigma V^*,$$

$$U \in \mathbb{C}^{m \times m}, \quad U^* U = I, \quad V \in \mathbb{C}^{n \times n}, \quad V^* V = I,$$

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{\min\{m, n\}}) \in \mathbb{R}^{m \times n}, \quad \sigma_1 \geq \dots \geq \sigma_k > 0, \quad \sigma_{k+1} = \dots = \sigma_{\min\{m, n\}} = 0,$$

mit unitären Matrizen U, V und Matrix Σ definiert durch die **Singulärwerte** von A .

Mögliche **Anwendung** der Singulärwertzerlegung zur Lösung eines linearen Gleichungssystems (entspricht einer Entkopplung der Gleichungen durch geeignete Basiswechsel)

$$Ax = b \quad \underset{A=U\Sigma V^*}{\iff} \quad U \Sigma V^* x = b \quad \iff \quad \Sigma y = c, \quad y = V^* x, \quad c = U^* b.$$

Falls die Matrix A unitär diagonalisierbar ist, ergibt sich insbesondere

$$Ax = b \quad \underset{AT=TA}{\iff} \quad T \Lambda T^* x = b \quad \iff \quad \Lambda y = c, \quad y = T^* x, \quad c = T^* b.$$

- **Eigenschaften der Singulärwertzerlegung** (Korollar 6.6): Für die Singulärwertzerlegung $A = U\Sigma V^*$ einer Matrix gelten folgende Eigenschaften:
 - Die Quadrate der Singulärwerte $\{\sigma_1^2, \dots, \sigma_n^2\}$ stimmen mit den Eigenwerten von A^*A und AA^* überein.
 - Es gilt $\mathcal{R}_A = \langle u_1, \dots, u_k \rangle$ und $\mathcal{N}_A = \langle v_{k+1}, \dots, v_n \rangle$.
 - Es gilt (wobei $\kappa(A) = \infty$ für $\sigma_n = 0$)

$$\|A\|_2 = \sigma_1, \quad \kappa(A) = \frac{\sigma_1}{\sigma_n}.$$

- **Vorüberlegung:** Mittels der Singulärwertzerlegung einer Matrix $A \in \mathbb{K}^{m \times n}$

$$A = U\Sigma V^*,$$

$$V = (v_1 | \dots | v_n) \in \mathbb{C}^{n \times n}, \quad V^*V = I, \quad U = (u_1 | \dots | u_m) \in \mathbb{C}^{m \times m}, \quad U^*U = I,$$

$$\Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ 0 & \dots & 0 & \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad m \geq n, \quad \Sigma = \begin{pmatrix} \sigma_1 & & & 0 \\ & \ddots & & \vdots \\ & & \sigma_n & 0 \\ & & & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad m \leq n,$$

ergibt sich wegen

$$Ax = U \underbrace{\Sigma V^* x}_{=y} = Uy = \sum_{i=1}^m y_i u_i = \sum_{i=1}^k \sigma_i v_i^* x u_i = \sum_{i=1}^k \sigma_i u_i v_i^* x,$$

$$y = \Sigma \begin{pmatrix} v_1^* x \\ \vdots \\ v_n^* x \end{pmatrix} = \begin{pmatrix} \sigma_1 v_1^* x \\ \vdots \\ \sigma_k v_k^* x \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

die Darstellung von A als Summe der Rang 1 Matrizen $u_i v_i^* \in \mathbb{C}^{m \times n}$ für $1 \leq i \leq k$

$$A = \sum_{i=1}^k \sigma_i u_i v_i^*.$$

Diese Darstellung wird zur **Approximation** der Matrix A verwendet.

Bemerkung: Für orthonormale Vektoren $z_1, \dots, z_k \in \mathbb{K}^m$ und Skalare $c_1, \dots, c_k \in \mathbb{K}$ gilt die folgende Relation (**Satz von Pythagoras**)

$$\left\| \sum_{i=1}^k c_i z_i \right\|_2^2 = \left\langle \sum_{i=1}^n c_i z_i \mid \sum_{j=1}^n c_j z_j \right\rangle_2 = \sum_{i,j=1}^n c_i \bar{c}_j \underbrace{\langle z_i \mid z_j \rangle_2}_{=z_j^* z_i = \delta_{ij}} = \sum_{i=1}^n |c_i|^2.$$

Resultat zur Approximation einer Matrix durch Matrizen niedrigeren Ranges (Satz 6.7): Für die Approximation der Matrix A durch eine Matrix A_j vom Rang $1 \leq j \leq k-1$

$$A_j = \sum_{i=1}^j \sigma_i u_i v_i^* \approx A = \sum_{i=1}^k \sigma_i u_i v_i^*$$

gilt die Relation

$$\|A_j - A\|_2 = \inf \{ \|A - B\|_2 : B \in \mathbb{C}^{m \times n}, \text{rg}(B) \leq j \} = \sigma_{j+1}.$$

Denn: (i) Zeige zunächst die Relation

$$\|A_j - A\|_2 = \sigma_{j+1}.$$

Einerseits gilt für $1 \leq j \leq k-1$ (Anwendung des Satzes von Pythagoras auf die Orthonormalbasis u_1, \dots, u_m)

$$A - A_j = \sum_{i=1}^k \sigma_i u_i v_i^* - \sum_{i=1}^j \sigma_i u_i v_i^* = \sum_{i=j+1}^k \sigma_i u_i v_i^*,$$

$$\|A - A_j\|_2 = \max_{\|x\|_2=1} \|(A - A_j)x\|_2 = \max_{\|x\|_2=1} \left\| \sum_{i=j+1}^k \sigma_i v_i^* x u_i \right\| = \max_{\|x\|_2=1} \sqrt{\sum_{i=j+1}^k \sigma_i^2 |v_i^* x|^2}.$$

Mit Hilfe der Darstellung bezüglich der Basis $v_1, \dots, v_n \in \mathbb{C}^n$ folgt weiters für jedes Element $x \in \mathbb{C}^n$ mit $\|x\|_2 = 1$ (wegen $\sigma_i \leq \sigma_{j+1}$ für $j+1 \leq i \leq k$)

$$x = \sum_{i=1}^n \xi_i v_i, \quad 1 = \|x\|_2 = \left\| \sum_{i=1}^n \xi_i v_i \right\|_2 = \sqrt{\sum_{i=1}^n |\xi_i|^2}, \quad v_j^* x = \sum_{i=1}^n \xi_i \underbrace{v_j^* v_i}_{=\delta_{ij}} = \xi_j,$$

$$\|A - A_j\|_2 = \max_{\|x\|_2=1} \sqrt{\sum_{i=j+1}^k \sigma_i^2 |v_i^* x|^2} \leq \max_{\|x\|_2=1} \sigma_{j+1} \underbrace{\sqrt{\sum_{i=j+1}^k |\xi_i|^2}}_{\leq 1} \leq \sigma_{j+1}.$$

Mit der Wahl $x = v_{j+1}$ gilt (beachte, daß $\|v_{j+1}\|_2 = 1$)

$$\sigma_{j+1} = \sqrt{\sum_{i=j+1}^k \sigma_i^2 |v_i^* v_{j+1}|^2} \leq \|A - A_j\|_2 \leq \sigma_{j+1} \implies \|A - A_j\|_2 = \sigma_{j+1}.$$

(ii) Zeige die Relation

$$\|A_j - A\|_2 = \inf \{ \|A - B\|_2 : B \in \mathbb{C}^{m \times n}, \text{rg}(B) \leq j \}.$$

Unter der Annahme für $B \in \mathbb{C}^{m \times n}$ gilt

$$\text{rg}(B) \leq j \Leftrightarrow \dim \mathcal{N}_B \geq n - j, \quad \|A - B\|_2 < \sigma_{j+1}.$$

Wählt man einen $(n - j)$ -dimensionalen Unterraum $U_1 \subset \mathcal{N}_B \subset \mathbb{C}^n$, so folgt

$$\begin{aligned} 0 \neq u \in U_1 : \quad (A - B)u &= Au - Bu = Au \\ \implies \quad \|Au\|_2 &= \|(A - B)u\|_2 \leq \|A - B\|_2 \|u\|_2 < \sigma_{j+1} \|u\|_2. \end{aligned}$$

Betrachtet man den $j + 1$ dimensionalen Unterraum $U_2 = \langle v_1, \dots, v_{j+1} \rangle \subset \mathbb{C}^n$, so gilt (beachte, daß $j + 1 \leq k$)

$$\begin{aligned} u &= \sum_{i=1}^{j+1} c_i v_i \in U_2, & A &= \sum_{\ell=1}^k \sigma_\ell u_\ell v_\ell^*, \\ Au &= \sum_{i=1}^{j+1} c_i A v_i = \sum_{i=1}^{j+1} \sum_{\ell=1}^k c_i \sigma_\ell u_\ell \underbrace{v_\ell^* v_i}_{=\delta_{i\ell}} = \sum_{i=1}^{j+1} c_i \sigma_i u_i, \\ \|u\|_2 &= \sqrt{\sum_{i=1}^{j+1} |c_i|^2}, & \|Au\|_2 &= \sqrt{\sum_{i=1}^{j+1} \sigma_i^2 |c_i|^2} \geq \sigma_{j+1} \sqrt{\sum_{i=1}^{j+1} |c_i|^2} = \sigma_{j+1} \|u\|_2. \end{aligned}$$

Wegen $1 \leq j \leq k - 1 < k \leq n$ folgt $\dim U_1 \geq n - j \geq 1$ und $\dim U_2 = j + 1 \geq 2$ und damit $\dim(U_1 \cap U_2) \geq 1$, d.h. es existiert ein Element $0 \neq u \in U_1 \cap U_2$ mit einerseits $\|Au\|_2 < \sigma_{j+1} \|u\|_2$ und andererseits $\|Au\|_2 \geq \sigma_{j+1} \|u\|_2$. Widerspruch! Somit folgt die Behauptung. \diamond

- **Bemerkung:** Niedrigrangapproximationen von Matrizen haben Anwendungen in verschiedenen Bereichen der Numerischen Mathematik (Theorie der Inversen Probleme, Bildkompression).

Beispiel, vgl. Skriptum S. 104: Ein Bild bestehend aus $m \times n$ Pixel entspricht einer Matrix $A \in \mathbb{R}^{m \times n}$ (Koeffizient a_{ij} entspricht dem Wert des Pixel an der Position (i, j) , d.h. einer Farbstufe). Eine Kompression des Bildes entspricht der Approximation der Matrix A durch eine Matrix A_j vom Rang $1 \leq j \leq k - 1$

$$A_j = \sum_{i=1}^j \sigma_i u_i v_i^* \approx A = \sum_{i=1}^k \sigma_i u_i v_i^*.$$

Ergebnis für $m = 576, n = 768, j = 30$.

- **Bemerkungen:**
 - Zur Berechnung der Singulärwertzerlegung $A = U\Sigma V^*$ einer Matrix könnte man im Prinzip die Eigenwertzerlegung von $B = A^*A$ verwenden (zunächst die Relation $V^*BV = \Sigma^2$ zur Berechnung von Σ und V sowie die Relation $AV = U\Sigma$ zur Berechnung von U). Diese Vorgehensweise kann allerdings zu einem numerisch instabilen Algorithmus führen, weil sich kleine Änderungen der Koeffizienten von A stärker auf die Eigenwerte von B auswirken als auf die Singulärwerte von A .

- Zur Berechnung der Singulärwertzerlegung ist es vorteilhafter, die Matrix

$$\mathbf{A} = \begin{pmatrix} & A^* \\ A & \end{pmatrix}$$

und die zugehörige Eigenwertrelation (ohne explizite Berechnung von \mathbf{A} , verwende $A = U\Sigma V^* \Leftrightarrow AV = U\Sigma$ bzw. $A^* = V\Sigma U^* \Leftrightarrow A^*U = V\Sigma$)

$$\mathbf{AV} = \mathbf{V}\Sigma, \quad \Sigma = \begin{pmatrix} \Sigma & \\ & -\Sigma \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} V & V \\ U & -U \end{pmatrix},$$

$$\begin{pmatrix} A^*U & -A^*U \\ AV & AV \end{pmatrix} = \begin{pmatrix} & A^* \\ A & \end{pmatrix} \begin{pmatrix} V & V \\ U & -U \end{pmatrix} = \begin{pmatrix} V & V \\ U & -U \end{pmatrix} \begin{pmatrix} \Sigma & \\ & -\Sigma \end{pmatrix} = \begin{pmatrix} V\Sigma & -V\Sigma \\ U\Sigma & -U\Sigma \end{pmatrix},$$

zu verwenden.

6.3. Algorithmen zum Eigenwertproblem (Überblick)

- **Vorbemerkungen:**

- Da es keine explizite Formel für die Nullstellen eines allgemeinen Polynoms höheren Grades gibt, kann es keine explizite Formel für die Eigenwerte einer allgemeinen Matrix höherer Dimension geben. Ein Polynom vom Grad $m \geq 1$ mit Koeffizienten $c_i \in \mathbb{K}$ für $0 \leq i \leq m$ (Leitkoeffizient $c_m = 1$)

$$p: \mathbb{K} \rightarrow \mathbb{K} : z \mapsto p(z) = \sum_{i=0}^m c_i z^i$$

entspricht nämlich dem charakteristischen Polynom der folgenden Matrix (zusätzliches Vorzeichen $(-1)^n$, geeignete Gauß-Elimination zur Berechnung von $\det(A - \lambda I)$ zeigt den Zusammenhang)

$$A = \begin{pmatrix} & & -c_0 \\ 1 & & -c_1 \\ & \ddots & \vdots \\ & & 1 & -c_{m-1} \end{pmatrix} \in \mathbb{K}^{n \times n}, \quad \chi(\lambda) = \det(A - \lambda I).$$

- Die obigen Überlegungen zeigen, daß ein numerisches Verfahren zur Berechnung eines (bzw. mehrerer) Eigenwertes λ einer allgemeinen Matrix $A \in \mathbb{K}^{n \times n}$ notwendigerweise ein iteratives Verfahren sein muß, d.h. ein Verfahren der Form

$$\lambda^{(k+1)} = \Psi(\lambda^{(k)}), \quad k \geq 0,$$

welches eine Folge von Näherungswerten an λ ergibt. Sofern das Verfahren konvergiert, d.h. es ist

$$\lim_{k \rightarrow \infty} \lambda^{(k)} = \lambda,$$

bricht man die Iteration ab, sobald eine ausreichende Genauigkeit erreicht werden konnte (beispielsweise verwendet man das Abbruchkriterium $|\lambda^{(k+1)} - \lambda^{(k)}| \leq \text{TOL}$)

$$|\lambda^{(k)} - \lambda| \leq \text{TOL}.$$

- Das meistverwendete numerische Verfahren zur Berechnung sämtlicher Eigenwerte einer Matrix, der QR-Algorithmus (mit Shift), basiert auf dem Lemma von Schur. Dieses besagt, daß jede Matrix $A \in \mathbb{K}^{n \times n}$ mittels einer unitären Matrix $T \in \mathbb{C}^{n \times n}$ (d.h. $T^* T = I$) auf Dreiecksform transformiert werden kann

$$A \longrightarrow S = T^* A T = \begin{pmatrix} \lambda_1 & S_{12} & \dots & S_{1n} \\ & \ddots & & \vdots \\ & & \ddots & S_{n-1,n} \\ & & & \lambda_n \end{pmatrix}.$$

Die Diagonalelemente von S entsprechen den Eigenwerten $\lambda_1, \dots, \lambda_n \in \mathbb{C}$ von A (die Eigenwerte von A bleiben unter Ähnlichkeitstransformationen erhalten). Die grundlegende Idee ist es, eine Folge von unitären Matrizen $Q_k \in \mathbb{C}^{n \times n}$ für $k \geq 1$ zu konstruieren, derart daß die transformierten Matrizen gegen S konvergieren

$$A_1 = A, \quad A_{k+1} = Q_k^* A_k Q_k, \quad k = 1, 2, 3, \dots$$

$$A_1 = A \longrightarrow A_2 = Q_1^* A Q_1 \longrightarrow A_3 = Q_2^* Q_1^* A Q_1 Q_2 \longrightarrow \dots$$

$$\lim_{k \rightarrow \infty} A_k = S.$$

Dazu verwendet man die Transformation von $A \in \mathbb{K}^{n \times n}$ auf **obere Hessenberg-Form** sowie die Idee der (inversen) Vektoriteration.

- Beispielsweise mittels Householder-Reflexionen läßt sich eine allgemeinen Matrix $A \in \mathbb{K}^{n \times n}$ auf **obere Hessenberg-Form** transformieren

$$A \longrightarrow H = T^* A T = \begin{pmatrix} H_{11} & \dots & \dots & H_{1n} \\ H_{21} & \ddots & & \vdots \\ & \ddots & \ddots & \vdots \\ & & H_{n,n-1} & H_{nn} \end{pmatrix}.$$

- * 1. Schritt: Konstruktion einer Householder-Reflexion $\tilde{Q}_1 \in \mathbb{K}^{(n-1) \times (n-1)}$ derart, daß die Spaltenelemente $(a_{21}, \dots, a_{n1})^T \in \mathbb{K}^{n-1}$ auf ein Vielfaches des Standardbasisvektors $e_1 \in \mathbb{R}^{n-1}$ abgebildet werden. Mit

$$Q_1 = \begin{pmatrix} 1 & \\ & \tilde{Q}_1 \end{pmatrix}$$

folgt somit (beachte, daß die entstehenden Matrizen AQ_1 und $Q_1^* A Q_1$ die gewünschte Form haben)

$$A \longrightarrow Q_1^* A Q_1 = \begin{pmatrix} * & * & \dots & * \\ * & * & \dots & * \\ & \vdots & & \vdots \\ & & * & \dots & * \end{pmatrix}.$$

- * Die Anwendung analoger Ideen auf die entstehenden Teilmatrizen führt nach $n - 2$ Schritten auf eine Matrix in oberer Hessenberg-Form. Die Anzahl der benötigten Operationen (Addition oder Multiplikation in \mathbb{K}) ist

$$\mathcal{O}\left(\frac{4}{3} n^3\right).$$

- * Man beachte, daß im Spezialfall einer selbstadjungierten Matrix $A \in \mathbb{K}^{n \times n}$ (d.h. es ist $A^* = A$) die entstehenden Matrizen ebenfalls selbstadjungiert sind.

Somit führt die obige Vorgehensweise auf eine selbstadjungierte **Tridiagonalmatrix**

$$A \longrightarrow H = T^* A T = \begin{pmatrix} H_{11} & H_{12} & & \\ H_{21} & \ddots & \ddots & \\ & \ddots & \ddots & H_{n-1,n} \\ & & H_{n,n-1} & H_{nn} \end{pmatrix},$$

$$H_{ii} \in \mathbb{R}, \quad H_{i+1,i} = \overline{H_{i,i+1}} \in \mathbb{R}, \quad 1 \leq i \leq i+1 \leq n.$$

- Im Folgenden werden grundlegende Ideen behandelt, die zur Konstruktion von Algorithmen für Eigenwertberechnungen führen. Für die praktische Berechnung von Eigenwerten und Eigenvektoren sollte man ausgeklügelte Software-Pakete verwenden.

6.4. Vektoriteration und inverse Vektoriteration

- **Situation:** Es sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische und reelle (quadratische) Matrix (d.h. es ist $A^T = A$). Folglich sind alle Eigenwerte $\lambda_1, \dots, \lambda_n$ von A reell und es gibt ein vollständiges System von orthonormalen Eigenvektoren $v_1, \dots, v_n \in \mathbb{R}^n$, d.h. es gilt

$$AV = V\Lambda, \quad V = (v_1 | \dots | v_n) \in \mathbb{R}^{n \times n}, \quad V^{-1} = V^T, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n) \in \mathbb{R}^{n \times n},$$

vgl. früheres Resultat zur unitären Diagonalisierbarkeit einer normalen Matrix und Folgerung für selbstadjungierte Matrizen.

Bemerkung: Symmetrische reelle Matrizen sind bedeutsam in Hinblick auf praktische Anwendungen und erlauben gewisse Vereinfachungen. Beispielsweise führen Ähnlichkeitstransformationen mittels Householder-Reflexionen auf symmetrische Tridiagonalmatrizen.

- Für $A \in \mathbb{R}^{n \times n}$ und $x \in \mathbb{R}^n$ ist der **Rayleigh-Quotient** gegeben durch

$$r: \mathbb{R}^n \setminus \{0\} \rightarrow \mathbb{R}: x \mapsto r(x) = \frac{x^T A x}{x^T x}, \quad x^T A x = \sum_{i,j=1}^n x_i a_{ij} x_j, \quad x^T x = \sum_{i=1}^n x_i^2.$$

Für den Gradienten ergibt sich folgende Relation (verwende Symmetrie $a_{ji} = a_{ij}$ für $1 \leq i, j \leq n$ und beachte $r'(x) = (\partial_{x_1} r(x), \dots, \partial_{x_n} r(x)) \in \mathbb{R}^{1 \times n}$)

$$\begin{aligned} r'(x) &= 2 \frac{1}{x^T x} \left((Ax)^T - r(x) x^T \right), \\ \partial_{x_\ell} r(x) &= \frac{\sum_{i,j=1}^n (\delta_{i\ell} a_{ij} x_j + x_i a_{ij} \delta_{j\ell})}{x^T x} - 2 r(x) \frac{x_\ell}{x^T x} \\ &= \frac{\sum_{j=1}^n a_{\ell j} x_j + \sum_{i=1}^n x_i a_{i\ell}}{x^T x} - 2 r(x) \frac{x_\ell}{x^T x} \\ &= 2 \frac{(Ax)_{\ell 1}}{x^T x} - 2 r(x) \frac{x_\ell}{x^T x} \\ &= 2 \frac{1}{x^T x} \left((Ax)_{\ell 1} - r(x) x_\ell \right). \end{aligned}$$

Ist $0 \neq v \in \mathbb{R}^n$ ein Eigenvektor zum Eigenwert $\lambda \in \mathbb{R}$, folgt insbesondere

$$r(v) = \lambda, \quad r'(v) = 2 \frac{1}{v^T v} \left(\lambda v^T - \underbrace{r(v) v^T}_{=\lambda} \right) = 0.$$

Taylorreihenentwicklung von r um v zeigt somit

$$r(x) = r(v) + \mathcal{O}(\|x - v\|_2^2).$$

Dies zeigt, daß der Rayleigh-Quotient für eine gute Näherung x an einen Eigenvektor v eine *gute* Approximation an den zugehörigen Eigenwert darstellt

$$r(x) \approx r(v) = \lambda, \quad r(x) - r(v) = \mathcal{O}(\|x - v\|_2^2).$$

- Die obigen Überlegungen motivieren die (direkte) **Vektoriteration**

$$x^{(k)} = Ax^{(k-1)} = A^k x^{(0)}, \quad \lambda^{(k)} = r(x^{(k)}) \approx \lambda,$$

mit einem *geeignet* gewählten Startvektor $x^{(0)} \in \mathbb{R}^n$ mit $\|x^{(0)}\|_2 = 1$ (mit Ergänzung eines Abbruchkriteriums)

```

x = x(0)
for k = 1, 2, ...
    x = Ax
    λ = r(x)
end

```

Mittels der Darstellung des Startvektors bezüglich der Orthonormalbasis von Eigenvektoren ergibt sich (unter der Annahme $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n| \geq 0$ und $\xi_1 = v_1^T x^{(0)} \neq 0$)

$$\begin{aligned}
 x^{(0)} &= \sum_{i=1}^n \xi_i v_i, \\
 x^{(k)} = A^k x^{(0)} &= \sum_{i=1}^n \xi_i A^k v_i = \sum_{i=1}^n \xi_i \lambda_i^k v_i = \xi_1 \lambda_1^k \left(v_1 + \sum_{i=2}^n \frac{\xi_i}{\xi_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k v_i \right), \\
 \frac{1}{\xi_1 \lambda_1^k} x^{(k)} &= v_1 + \sum_{i=2}^n \frac{\xi_i}{\xi_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k v_i \xrightarrow[k \rightarrow \infty]{|\frac{\lambda_i}{\lambda_1}| = \rho_i < 1} v_1.
 \end{aligned}$$

Für den Rayleigh-Quotienten folgt weiters

$$\begin{aligned}
 \|x^{(k)}\|_2^2 &= \sum_{i=1}^n \xi_i^2 \lambda_i^{2k} = \xi_1^2 \lambda_1^{2k} + \sum_{i=2}^n \xi_i^2 \lambda_i^{2k} = \xi_1^2 \lambda_1^{2k} \left(1 + \sum_{i=2}^n \frac{\xi_i^2}{\xi_1^2} \left(\frac{\lambda_i}{\lambda_1} \right)^{2k} \right), \\
 (x^{(k)})^T A x^{(k)} &= \left(\sum_{i=1}^n \xi_i \lambda_i^k v_i \right)^T \sum_{i=1}^n \xi_i \lambda_i^{k+1} v_i = \sum_{i,j=1}^n \xi_i \xi_j \lambda_i^{k+1} \lambda_j^k \underbrace{v_j^T v_i}_{=\delta_{ij}} = \sum_{i=1}^n \xi_i^2 \lambda_i^{2k+1}, \\
 r(x^{(k)}) &= \frac{1}{\|x^{(k)}\|_2^2} (x^{(k)})^T A x^{(k)} = \frac{1}{\|x^{(k)}\|_2^2} \sum_{i=1}^n \xi_i^2 \lambda_i^{2k+1} = \frac{\xi_1^2 \lambda_1^{2k+1} \left(1 + \sum_{i=2}^n \frac{\xi_i^2}{\xi_1^2} \left(\frac{\lambda_i}{\lambda_1} \right)^{2k+1} \right)}{\xi_1^2 \lambda_1^{2k} \left(1 + \sum_{i=2}^n \frac{\xi_i^2}{\xi_1^2} \left(\frac{\lambda_i}{\lambda_1} \right)^{2k} \right)} \\
 &= \lambda_1 \frac{1 + \sum_{i=2}^n \frac{\xi_i^2}{\xi_1^2} \left(\frac{\lambda_i}{\lambda_1} \right)^{2k+1}}{1 + \sum_{i=2}^n \frac{\xi_i^2}{\xi_1^2} \left(\frac{\lambda_i}{\lambda_1} \right)^{2k}} \xrightarrow{k \rightarrow \infty} \lambda_1.
 \end{aligned}$$

Hier wurden die folgenden Relationen verwendet (geometrische Reihe, für ρ hinreichend klein)

$$\frac{1}{1-q} = \sum_{\ell=0}^{\infty} q^{\ell}, \quad |q| < 1, \quad \frac{1+\mathcal{O}(\rho)}{1+\mathcal{O}(\rho)} = 1 + \mathcal{O}(\rho).$$

Dies zeigt, daß die mittels der Vektoriteration berechneten Näherungswerte gegen den dominanten (d.h. betragsmäßig größten) Eigenwert und einen zugehörigen Eigenvektor konvergieren (unter der Annahme $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n| \geq 0$ und der Forderung $v_1^T x^{(0)} = \xi_1 \neq 0, c = \pm 1$)

$$\frac{1}{\|x^{(k)}\|_2} x^{(k)} \xrightarrow{k \rightarrow \infty} c v_1, \quad r(x^{(k)}) \xrightarrow{k \rightarrow \infty} \lambda_1.$$

Bemerkungen:

- Beachte, daß der Rayleigh-Quotient skalierungsinvariant ist

$$r(cx) = \frac{(cx)^T A(cx)}{(cx)^T (cx)} = \frac{cx^T A x}{x^T x} = r(x), \quad c \neq 0.$$

Eine Normierung der Approximationen $x^{(k)}$ dient dazu, rasch auftretenden Overflow (falls $|\lambda_1| > 1$) bzw. Underflow (falls $|\lambda_1| < 1$) zu verhindern.

- Die Konvergenzgeschwindigkeit der Vektoriteration wird durch die Größe

$$|\rho_2| = \left| \frac{\lambda_2}{\lambda_1} \right|$$

bestimmt. Falls $|\rho| \approx 1$ ist die Konvergenzrate sehr dürftig.

- Analoge Überlegungen gelten für eine diagonalisierbare Matrix $A \in \mathbb{K}^{n \times n}$.

- **Vorbemerkungen:** Wesentliche Nachteile der direkten Vektoriteration sind, daß
 - nur der betragsmäßig größte Eigenwerte und ein zugehöriger Eigenwert berechnet werden können und
 - die Konvergenzrate unzufriedenstellend ist, falls $|\rho_2| = \left| \frac{\lambda_2}{\lambda_1} \right| \approx 1$.

Die **inverse Vektoriteration** basiert auf der Umformulierung der Eigenrelation (wobei $\mu \in \mathbb{R}$ mit $\mu \neq \lambda_i$ für $1 \leq i \leq n$)

$$\begin{aligned} Av_i = \lambda_i v_i, \quad 1 \leq i \leq n &\iff (A - \mu I) v_i = (\lambda_i - \mu) v_i, \quad 1 \leq i \leq n \\ &\iff (A - \mu I)^{-1} v_i = \frac{1}{\lambda_i - \mu} v_i, \quad 1 \leq i \leq n \\ &\iff \tilde{A}_\mu v_i = \tilde{\lambda} v_i, \quad B = (A - \mu I)^{-1}, \quad \tilde{\lambda} = \frac{1}{\lambda_i - \mu}, \quad 1 \leq i \leq n. \end{aligned}$$

Unter der Annahme, daß eine Näherung $\mu \approx \lambda_i$ an den gesuchten Eigenwert der Matrix A bekannt ist, führt man die direkte Vektoriteration für die Matrix \tilde{A}_μ durch

$$x^{(k)} = \tilde{A}_\mu x^{(k-1)} = (A - \mu I)^{-1} x^{(k-1)}, \quad \tilde{\lambda}^{(k)} = r(x^{(k)}) \approx \tilde{\lambda} = \frac{1}{\lambda_i - \mu}, \quad \lambda_i \approx \frac{1}{\tilde{\lambda}^{(k)}} + \mu.$$

Dies erfordert in jedem Iterationsschritt die Lösung eines linearen Gleichungssystems (effiziente Umsetzung durch Berechnung z.B. einer LR-Zerlegung der Matrix, pro Iterationsschritt sind dann eine Vorwärtssubstitution und eine Rückwärtssubstitution nötig)

$$(A - \mu I) x^{(k)} = x^{(k-1)}.$$

Sofern eine gute Näherung $\mu \approx \lambda_i$ bekannt ist und die restlichen Eigenwerte von A deutlich verschieden von λ_i sind, ist die Konvergenzrate der inversen Vektoriteration wegen

$$|\lambda_i - \mu| \ll |\lambda_j - \mu|, \quad 1 \leq i, j \leq n, \quad j \neq i \iff \frac{|\lambda_i - \mu|}{|\lambda_j - \mu|} \ll 1, \quad 1 \leq i, j \leq n, \quad j \neq i$$

ausgezeichnet.

Bemerkung: Es ist zu beachten, daß bei der inversen Vektoriteration im allgemeinen Matrizen mit großer Konditionszahl auftreten (falls $|\lambda_i - \mu| \approx 0$ ist $(A - \mu I)^{-1}$ nahezu singulär). Dennoch ist die numerische Anwendung sinnvoll. Offene Fragen sind außerdem die Konstruktion geeigneter Startvektoren und optimale Abbruchkriterien.

6.5. Die Grundidee des QR-Algorithmus

- **Vorbemerkung:** Der QR-Algorithmus (genauer der QR-Algorithmus mit Shift) ist ein effizientes numerisch stabiles Verfahren zur Berechnung aller Eigenwerte einer Matrix $A \in \mathbb{K}^{n \times n}$.
- Ausgehend von der bereits auf Hessenberg-Form transformierten Matrix $A^{(0)} = A$, basiert der **QR-Algorithmus** auf einer QR-Zerlegung der Matrix und anschließender Matrizenmultiplikation

$$\left\{ \begin{array}{l} \text{QR-Zerlegung: } Q^{(k)} R^{(k)} = A^{(k)}, \\ \text{Rekombination: } A^{(k+1)} = R^{(k)} Q^{(k)}, \end{array} \right. \quad k \geq 0.$$

Ohne Einschränkung der Allgemeinheit kann angenommen werden, daß alle Diagonalelemente von $R^{(k)}$ positiv sind (ansonsten erfolgt eine Reduktion des Problems, vgl. Skriptum, S. 110). Beachte, daß der Übergang von A zu $A^{(k)}$ einer unitären Transformation entspricht

$$A^{(k+1)} = \underbrace{R^{(k)}}_{R^{(k)} = (Q^{(k)})^* A^{(k)} Q^{(k)}} \quad Q^{(k)} = (Q^{(k)})^* A^{(k)} Q^{(k)},$$

$$A^{(k)} = (X^{(k)})^* A X^{(k)}, \quad X^{(k)} = Q^{(0)} \dots Q^{(k-1)}.$$

Weiters erhält der QR-Algorithmus Eigenschaften wie Selbstadjungiertheit

$$(A^{(k)})^* = A^{(k)} \implies (A^{(k+1)})^* = \left((Q^{(k)})^* A^{(k)} Q^{(k)} \right)^* = (Q^{(k)})^* \underbrace{(A^{(k)})^*}_{=A^{(k)}} Q^{(k)} = A^{(k+1)}$$

und auch die Hessenberg-Form einer Matrix.

Konvergenz des QR-Algorithmus: Unter der Annahme $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > 0$, konvergieren die beim QR-Algorithmus entstehenden Matrizen gegen eine obere Dreiecksmatrix mit den Eigenwerten der Matrix A als Diagonalelemente

$$\lim_{k \rightarrow \infty} A^{(k)} = \begin{pmatrix} \lambda_1 & * & \dots & \dots & * \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & * \\ & & & \ddots & * \\ & & & & \lambda_n \end{pmatrix}.$$

Denn: Es reicht aus, die Konvergenz der ersten Spalte und der letzten Zeile von $A^{(k)}$ nachzuweisen

$$\lim_{k \rightarrow \infty} A^{(k)} = \begin{pmatrix} \lambda_1 & * & \dots & \dots & * \\ & \vdots & & & \vdots \\ & \vdots & & & \vdots \\ & * & \dots & \dots & * \\ & & & & \lambda_n \end{pmatrix},$$

weil sich dann die Überlegungen auf die entstehende Teilmatrix der Dimension $n - 2$ anwenden lassen.

- Zunächst stellt man mittels $X^{(k)} = Q^{(0)} \dots Q^{(k-1)}$, d.h. es ist $X^{(k+1)} = X^{(k)} Q^{(k)}$, folgenden Zusammenhang her

$$A^{(k)} = (X^{(k)})^* A X^{(k)} \iff A X^{(k)} = X^{(k)} \underbrace{A^{(k)}}_{=Q^{(k)} R^{(k)}} = X^{(k+1)} R^{(k)}.$$

- Speziell für die erste Spalte der entstehenden Matrix ergibt sich aufgrund der oberen Dreiecksform von $R^{(k)}$

$$\begin{aligned} A X^{(k)} &= X^{(k+1)} R^{(k)}, \\ A X_{-,1}^{(k)} &= X^{(k+1)} R_{-,1}^{(k)} = R_{11}^{(k)} X^{(k+1)} e_1 = R_{11}^{(k)} X_{-,1}^{(k+1)}, \end{aligned}$$

d.h. die Iteration für die erste Spalte entspricht einer direkten Vektoriteration

$$y^{(k+1)} = c^{(k)} A y^{(k)}, \quad c^{(k)} = \frac{1}{R_{11}^{(k)}} > 0, \quad y^{(k)} = X_{-,1}^{(k)},$$

mit Konvergenz gegen einen Eigenvektor von A zum betragsmäßig größten Eigenwert λ_1 .

- Adjunktion und Inversion der obigen Relation ergibt (Transformationsmatrizen sind unitär, d.h. es ist $T^{-1} = T^*$ und $(T^{-1})^* = T$)

$$\begin{aligned} A X^{(k)} &= X^{(k+1)} R^{(k)}, \\ (X^{(k)})^* A^* &= (A X^{(k)})^* = (X^{(k+1)} R^{(k)})^* = (R^{(k)})^* (X^{(k+1)})^*, \\ (A^{-1})^* X^{(k)} &= X^{(k+1)} \left((R^{(k)})^* \right)^{-1}. \end{aligned}$$

Speziell für die letzte Spalte der entstehenden Matrix ergibt sich aufgrund der unteren Dreiecksform von $(R^{(k)})^*$

$$\begin{aligned} (A^{-1})^* X_{-,n}^{(k)} &= X^{(k+1)} \left((R^{(k)})^* \right)_{-,n}^{-1} = \left((R^{(k)})^* \right)_{n,n}^{-1} X^{(k+1)} e_n = \frac{1}{R_{nn}^{(k)}} X_{-,n}^{(k+1)}, \\ z^{(k+1)} &= d^{(k)} (A^{-1})^* z^{(k)}, \quad d^{(k)} = R_{nn}^{(k)} > 0, \quad z^{(k)} = X_{-,n}^{(k)}, \end{aligned}$$

d.h. die Iteration für die letzte Spalte entspricht einer inversen Vektoriteration mit Konvergenz gegen einen Eigenvektor z von A^* zum betragsmäßig kleinsten Eigenwert μ . Wegen (Eigenrelation für die adjungierte Matrix und Zusammenhang mit den zugehörigen Eigenwerten und linksseitigen Eigenvektoren)

$$A^* z = \mu z \Leftrightarrow z^* A = \mu z^*$$

stimmt μ mit λ_n überein.

- Insgesamt ergibt sich somit für die erste Spalte bzw. letzte Zeile von $A^{(k)}$ (vgl. Abschnitt 6.4)

$$\begin{aligned}
 A^{(k)} &= (X^{(k)})^* A X^{(k)}, \\
 A_{-,1}^{(k)} &= A^{(k)} e_1 = (X^{(k)})^* A X^{(k)} e_1 = (X^{(k)})^* \underbrace{A X_{-,1}^{(k)}}_{\rightarrow \pm \lambda_1 v_1} \xrightarrow{k \rightarrow \infty} \lambda_1 e_1, \\
 A_{n,-}^{(k)} &= e_n^T A^{(k)} = (A^* X^{(k)} e_n^T)^* X^{(k)} = \underbrace{(A^* X_{-,n}^{(k)})^*}_{\rightarrow \lambda_n z} X^{(k)} \xrightarrow{k \rightarrow \infty} \lambda_n e_n^T.
 \end{aligned}$$

Dies ergibt die Behauptung. \diamond

- **QR-Algorithmus mit Shift:** Im Allgemeinen wird eine Modifikation des QR-Algorithmus verwendet mit einem zusätzlichen **Shift**, ähnlich der Idee der inversen Vektoriteration. Damit verbessert man die Konvergenzrate der letzten Zeile der entstehenden Matrix $A^{(k)}$.

Beispiel: Speziell für die folgende Matrix

$$\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

ist die Konvergenzrate des QR-Algorithmus sehr dürftig. Ein zufriedenstellendes Ergebnis ergibt sich hingegen mittels des QR-Algorithmus mit Shift.

- **Bemerkung:** Nicht behandelt werden alternative Verfahren zu Eigenwertberechnungen. Für symmetrische und reelle Tridiagonalmatrizen verwendet ein numerisch stabiles Verfahren die **Bisektionsmethode nach Givens** oder **Sturmsche Ketten**. Ein weiteres Verfahren ist das **Verfahren von Arnoldi**.

7. Nichtlineare Gleichungssysteme

- **Problemstellung:** Betrachtet wird eine nichtlineare Funktion (wie üblich sei $\mathbb{R}^n = \mathbb{R}^{n \times 1}$)

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^n : x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \mapsto f(x) = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{pmatrix},$$

die als hinreichend oft differenzierbar angenommen wird. Gesucht ist eine Näherungslösung an eine (zumindest lokal eindeutige) Lösung $\bar{x} \in \mathbb{R}^n$ des **nichtlinearen Gleichungssystems** (n Gleichungen, n Unbekannte)

$$f(x) = 0.$$

Anwendungen:

- Verfahren zur Lösung nichtlinearer Optimierungsprobleme
- Verfahren zur Lösung nichtlinearer gewöhnlicher Differentialgleichungen
- Verfahren zur Lösung nichtlinearer partieller Differentialgleichungen

Vgl. auch den Zusammenhang mit technischen Regelkreisen, Skriptum, S. 116.

Bemerkungen:

- Im Gegensatz zu linearen Gleichungssystemen sind Resultate zur **Existenz und (lokalen) Eindeutigkeit** von Lösungen nichtlinearer Gleichungssysteme nicht allgemein sondern nur in speziellen Situationen gültig. Deshalb wird im Folgenden die (lokale) Existenz und Eindeutigkeit der Lösung des betrachteten nichtlinearen Gleichungssystems angenommen.
- Hinsichtlich praktischer Anwendungen (komplexe Funktionsvorschrift, zusätzliche Berechnungen zur Funktionsauswertung erforderlich) ist es wesentlich, die Anzahl der Funktionsauswertungen von f möglichst gering zu halten.
- Für den Spezialfall einer skalaren nichtlinearen Gleichung

$$f(x) = 0, \quad f: \mathbb{R} \rightarrow \mathbb{R},$$

gibt es iterative Verfahren mit ausgezeichneten Konvergenzeigenschaften (sofern die Existenz und lokale Eindeutigkeit einer Lösung gesichert ist und ein einschließendes Intervall bekannt ist).

In mehreren Dimensionen ist die Lösung eines nichtlinearen Gleichungssystems hingegen ein **schwieriges Problem**, und es gibt kein allgemein anwendbares und mit Sicherheit erfolgreiches numerisches Verfahren.

Die Konvergenzeigenschaften der verwendeten Iterationsverfahren hängen wesentlich vom gewählten **Startwert** ab. Bisher gibt es für die Berechnung eines geeigneten Startwertes kein allgemein anwendbares Verfahren.

– Beachte, daß

$$\begin{aligned}
 f &: \mathbb{R}^n \rightarrow \mathbb{R}^k : x \mapsto f(x), \\
 f'(x) &: \mathbb{R}^n \rightarrow \mathbb{R}^k : y \mapsto f'(x)y, \quad x \in \mathbb{R}^n, \\
 f''(x) &: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^k : (y, z) \mapsto f''(x)(y, z), \quad x \in \mathbb{R}^n, \\
 f(x) &= \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_k(x_1, \dots, x_n) \end{pmatrix} \in \mathbb{R}^k, \quad x \in \mathbb{R}^n, \\
 f'(x) &= \begin{pmatrix} \partial_{x_1} f_1(x_1, \dots, x_n) & \dots & \partial_{x_n} f_1(x_1, \dots, x_n) \\ \vdots & & \vdots \\ \partial_{x_1} f_k(x_1, \dots, x_n) & \dots & \partial_{x_n} f_k(x_1, \dots, x_n) \end{pmatrix} \in \mathbb{R}^{k \times n}, \quad x \in \mathbb{R}^n, \\
 f''(x)(y, z) &= \begin{pmatrix} \sum_{i,j=1}^n \partial_{x_i x_j} f_1(x_1, \dots, x_n) y_i z_j \\ \vdots \\ \sum_{i,j=1}^n \partial_{x_i x_j} f_k(x_1, \dots, x_n) y_i z_j \end{pmatrix} \in \mathbb{R}^k, \quad x, y, z \in \mathbb{R}^n.
 \end{aligned}$$

Insbesondere für $k = 1$ ergibt sich (beachte $\partial_{x_i x_j} f = \partial_{x_j x_i} f$ sofern f zweimal stetig partiell differenzierbar)

$$\begin{aligned}
 f &: \mathbb{R}^n \rightarrow \mathbb{R} : x \mapsto f(x), \\
 f'(x) &: \mathbb{R}^n \rightarrow \mathbb{R} : y \mapsto f'(x)y, \quad x \in \mathbb{R}^n, \\
 f''(x) &: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} : (y, z) \mapsto f''(x)(y, z), \quad x \in \mathbb{R}^n, \\
 f(x) &= f(x_1, \dots, x_n) \in \mathbb{R}, \quad x \in \mathbb{R}^n, \\
 f'(x) &= (\partial_{x_1} f(x_1, \dots, x_n), \dots, \partial_{x_n} f(x_1, \dots, x_n)) \in \mathbb{R}^{1 \times n}, \quad x \in \mathbb{R}^n, \\
 f'(x)y &= \sum_{i=1}^n \partial_{x_i} f(x_1, \dots, x_n) y_i \in \mathbb{R}, \quad x, y \in \mathbb{R}^n, \\
 f''(x) &= \begin{pmatrix} \partial_{x_1 x_1} f(x_1, \dots, x_n) & \dots & \partial_{x_1 x_n} f(x_1, \dots, x_n) \\ \vdots & & \vdots \\ \partial_{x_n x_1} f(x_1, \dots, x_n) & \dots & \partial_{x_n x_n} f(x_1, \dots, x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad x \in \mathbb{R}^n, \\
 f''(x)(y, z) &= y^T f''(x)z = \sum_{i,j=1}^n \partial_{x_i x_j} f(x_1, \dots, x_n) y_i z_j \in \mathbb{R}, \quad x, y, z \in \mathbb{R}^n.
 \end{aligned}$$

- **Optimierungsprobleme:** Die Minimierung einer differenzierbaren Funktion

$$g(x) \xrightarrow{!} \min, \quad g: \mathbb{R}^n \rightarrow \mathbb{R},$$

führt auf das nichtlineare Gleichungssystem

$$f(x) = 0, \quad f = g' : \mathbb{R}^n \rightarrow \mathbb{R}^{1 \times n} : x \mapsto f(x) = g'(x).$$

In dieser Situation gibt es zusätzlich alternative numerische Verfahren, die zusätzliche Eigenschaften der Ableitung von $f = g'$ ausnützen (beispielsweise die Symmetrie und positive Definitheit der Hessematrix $f'(x) = g''(x)$). Analoge Überlegungen gelten für Maximierungsprobleme (Betrachtung von $-g$).

- **Inhalt:**

- Grundlegende Begriffe und Resultate
- Verfahren für eindimensionale Probleme (Bisektionsverfahren und Modifikationen, Newton-Verfahren)
- Verfahren für mehrdimensionale Probleme (Modifikationen des Newton-Verfahrens)

7.1. Grundbegriffe

- **Situation:** Es sei $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine hinreichend oft differenzierbare Funktion. Betrachtet wird das nichtlineare Gleichungssystem

$$f(x) = 0$$

mit (lokal) eindeutig bestimmter Lösung $\bar{x} \in \mathbb{R}^n$.

- **Iterationsverfahren, Fixpunktiteration, Konvergenz:**

- Numerische Verfahren zur näherungsweise Berechnung der Lösung des nichtlinearen Gleichungssystems $f(x) = 0$ sind **iterative Verfahren**. Ausgehend von einem (geeignet gewählten) Startwert $x_0 \in \mathbb{R}^n$ wird eine Folge $(x_k)_{k \geq 1}$ von Näherungswerten $x_k \in \mathbb{R}^n$ an \bar{x} mittels einer Rekursion der Form

$$x_{k+1} = \varphi(x_k), \quad k \geq 0,$$

mit **Iterationsfunktion** $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ berechnet. Dabei gilt es sicherzustellen, daß die Iteration (rasch) gegen die gesuchte Lösung konvergiert

$$\lim_{k \rightarrow \infty} x_k = \bar{x}.$$

- Falls die Iterationsfunktion φ stetig ist und die Folge der Näherungswerte gegen \bar{x} konvergiert, folgt

$$\bar{x} = \lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} \varphi(x_k) = \varphi\left(\lim_{k \rightarrow \infty} x_k\right) = \varphi(\bar{x}),$$

d.h. die Lösung des nichtlinearen Gleichungssystems ist ein **Fixpunkt** der Iterationsfunktion

$$f(\bar{x}) = 0 \iff \varphi(\bar{x}) = \bar{x}.$$

Dies motiviert die Bezeichnung **Fixpunktiteration** für die obige Iteration.

- Ein Iterationsverfahren heißt **global konvergent**, wenn man eine Menge $D \subset \mathbb{R}^n$ angeben kann, sodaß die Iteration für beliebige Startwerte $x_0 \in D$ gegen den (eindeutig bestimmten) Fixpunkt \bar{x} konvergiert. Konvergiert die Iteration nur für Startwerte x_0 , die *hinreichend nahe* beim Fixpunkt \bar{x} liegen, so heißt das Iterationsverfahren **lokal konvergent**.
- Es sei $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ eine festgelegte Norm. Eine Funktion $g : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ heißt **Lipschitz-stetig**, wenn es eine Konstante $L > 0$ gibt, sodaß für alle Elemente $x, \tilde{x} \in D$ die folgende Relation gilt

$$\|g(x) - g(\tilde{x})\| \leq L \|x - \tilde{x}\|.$$

Eine Lipschitz-stetige Funktion ist insbesondere stetig.

Eine Funktion $g : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ heißt eine **kontrahierende Abbildung (Kontraktion)**, wenn es eine Konstante $0 < \kappa < 1$ gibt, sodaß für alle $x, \tilde{x} \in D$ die folgende Relation gilt (bezüglich einer festgelegten Norm $\|\cdot\|$)

$$\|g(x) - g(\tilde{x})\| \leq \kappa \|x - \tilde{x}\|, \quad 0 < \kappa < 1,$$

d.h. die Funktion g ist insbesondere Lipschitz-stetig mit Konstante $\kappa < 1$.

Resultat zur Existenz und Eindeutigkeit eines Fixpunktes: Für eine auf einer abgeschlossenen Menge definierte kontrahierende Selbstabbildung $g : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ (d.h. es gilt $g(D) \subset D$) sichert der **Banachsche Fixpunktsatz** die Existenz und Eindeutigkeit eines Fixpunktes $\bar{x} \in D$. Insbesondere konvergiert die Fixpunktiteration $x_{k+1} = g(x_k)$ für beliebige Startwerte $x_0 \in D$ gegen den Fixpunkt \bar{x} .

– Neben **Einschrittverfahren**

$$x_0 \text{ gegeben,} \quad x_{k+1} = \varphi(x_k), \quad k \geq 0,$$

sind (insbesondere im Zusammenhang mit numerischen Verfahren zur Lösung nichtlinearer Differentialgleichungen) auch **Mehrschrittverfahren** gebräuchlich. Dabei verwendet man mehrere bekannte Approximationswerte zur Bestimmung des neuen Näherungswertes ($x_m = \varphi(x_0, \dots, x_{m-1})$, $x_{m+1} = \varphi(x_1, \dots, x_m)$ etc.)

$$x_0, \dots, x_{m-1} \text{ gegeben,} \quad x_{m+k} = \varphi(x_k, \dots, x_{m-1+k}), \quad k \geq 0,$$

Mittels der Umformulierung

$$X_0 = \begin{pmatrix} x_0 \\ \vdots \\ x_{m-1} \end{pmatrix}, \quad X_k = \begin{pmatrix} x_k \\ \vdots \\ x_{k+m-1} \end{pmatrix}, \quad \Phi(X_k) = \begin{pmatrix} x_{k+1} \\ \vdots \\ x_{k+m-1} \\ \varphi(X_k) \end{pmatrix}, \quad k \geq 0,$$

läßt sich jedes Mehrschrittverfahren auf ein Einschrittverfahren zurückführen (für theoretische Untersuchungen)

$$X_0 \text{ gegeben,} \quad X_{k+1} = \Phi(X_k), \quad k \geq 0.$$

– **Konvergenzordnung einer Iteration** (Definition 7.1): Für alle Startwerte $x_0 \in D$ sei die Iteration konvergent

$$x_{k+1} = \varphi(x_k), \quad k \geq 0, \quad \lim_{k \rightarrow \infty} x_k = \bar{x}.$$

Falls es einen Index $K_0 \in \mathbb{N}$ und eine Konstante $c > 0$ gibt, sodaß die folgende Abschätzung mit $p > 0$ (als Supremum) gilt, heißt p die **Konvergenzordnung** des Iterationsverfahrens

$$\|x_{k+1} - \bar{x}\| \leq c \|x_k - \bar{x}\|^p \quad \text{für alle } k \geq K_0.$$

Im Fall $p = 1$ spricht man von **linearer Konvergenz** und die Konstante c heißt **Konvergenzfaktor**. Falls zusätzlich

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|} = 0,$$

spricht man von **superlinearer Konvergenz**. Im Fall $p = 2$ spricht man von **quadratischer Konvergenz**.

Bemerkung: Falls die Iterationsfunktion die folgenden Relationen erfüllt

$$\varphi^{(\ell)}(\bar{x}) = 0, \quad 1 \leq \ell \leq p-1, \quad \varphi^{(p)}(\bar{x}) \neq 0,$$

folgt mittels einer Taylorreihenentwicklung (beispielsweise für den skalaren Fall, ebenso für den allgemeinen Fall)

$$\begin{aligned} x_{k+1} - \bar{x} &= \varphi(x_k) - \varphi(\bar{x}) \\ &= \sum_{\ell=0}^p \frac{1}{\ell!} \varphi^{(\ell)}(\bar{x}) (x_k - \bar{x})^\ell + \mathcal{O}(\|x_k - \bar{x}\|^{p+1}) - \varphi(\bar{x}) \\ &= \frac{1}{p!} \varphi^{(p)}(\bar{x}) (x_k - \bar{x})^p + \mathcal{O}(\|x_k - \bar{x}\|^{p+1}), \\ \|x_{k+1} - \bar{x}\| &\leq \left(\frac{1}{p!} \|\varphi^{(p)}(\bar{x})\| + \mathcal{O}(\|x_k - \bar{x}\|) \right) \|x_k - \bar{x}\|^p, \end{aligned}$$

und somit ist die Konvergenzordnung des Verfahrens p .

- Eine **Linearisierung** der Funktion f und Iteration führt auf das **Newton-Verfahren**

$$\begin{aligned} 0 &= f(\bar{x}) = f(x_k) + f'(x_k)(\bar{x} - x_k) + \mathcal{O}(\|\bar{x} - x_k\|^2), \\ 0 &\approx f(x_k) + f'(x_k)(\bar{x} - x_k) \iff \bar{x} \approx x_k - (f'(x_k))^{-1} f(x_k), \\ x_{k+1} &= x_k - (f'(x_k))^{-1} f(x_k). \end{aligned}$$

– Im **eindimensionalen Fall** ergibt sich

$$x_0 \text{ gegeben,} \quad x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k \geq 0.$$

Veranschaulichung (ersetze Funktion durch lineare Approximation bei x_k und bestimme Nullstelle), vgl. Skriptum, S. 114.

Im **mehrdimensionalen Fall** erfordert das Newton-Verfahren in jedem Iterationsschritt die Lösung eines linearen Gleichungssystems (verwende Hilfsbezeichnung $\xi_k = x_k - x_{k+1}$)

$$x_0 \text{ gegeben,} \quad \begin{cases} f'(x_k) \xi_k = f(x_k), \\ x_{k+1} = x_k - \xi_k, \end{cases} \quad k \geq 0.$$

- Sofern die Matrix $f'(x_k)$ für $k \geq 0$ nicht singulär ist (oder die Matrix $f'(\bar{x})$ invertierbar ist und x_k für $k \geq 0$ *nahe genug* bei der gesuchten Lösung \bar{x} liegt) ist das Newton-Verfahren wohldefiniert.

Die zugehörige Iterationsfunktion lautet

$$\varphi(x) = x - (f'(x))^{-1} f(x).$$

Im Allgemeinen ist das Newton-Verfahren nur lokal konvergent, vgl. auch Abbildung, Skriptum S. 117.

Im Spezialfall $f: \mathbb{R} \rightarrow \mathbb{R}$ ergibt sich (Produktregel, $f(\bar{x}) = 0$)

$$\begin{aligned} \varphi(x) &= x - \frac{f(x)}{f'(x)}, \\ \varphi'(x) &= 1 - \frac{f'(x)}{f'(x)} + \frac{f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}, \quad \varphi'(\bar{x}) = 0, \\ \varphi''(x) &= \frac{f''(x)}{f'(x)} + \frac{f(x)f'''(x)}{(f'(x))^2} - 2 \frac{f(x)(f''(x))^2}{(f'(x))^3}, \quad \varphi''(\bar{x}) = \frac{f''(\bar{x})}{f'(\bar{x})} \stackrel{\text{i.A.}}{\neq} 0, \end{aligned}$$

d.h. das **Newton-Verfahren ist quadratisch konvergent** (in einer geeigneten Umgebung einer einfachen Nullstelle \bar{x}). Dasselbe Resultat gilt im mehrdimensionalen Fall (ohne Begründung).

Beispiel: Quadratische Konvergenz entspricht einer **Verdoppelung der korrekten Dezimalziffern** ($1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16$ korrekte Stellen)

$$\begin{aligned} \|x_j - \bar{x}\| \approx \frac{1}{10} &\longrightarrow \|x_{j+1} - \bar{x}\| \approx \frac{1}{10^2} \longrightarrow \|x_{j+2} - \bar{x}\| \approx \frac{1}{10^4} \\ &\longrightarrow \|x_{j+3} - \bar{x}\| \approx \frac{1}{10^8} \longrightarrow \|x_{j+4} - \bar{x}\| \approx \frac{1}{10^{16}}. \end{aligned}$$

- Vorteil des Newton-Verfahrens

- * Ausgezeichnetes lokales Konvergenzverhalten (quadratische Konvergenz)

Nachteile des Newton-Verfahrens

- * Im Allgemeinen keine globale Konvergenz
- * Notwendigkeit der Berechnung von $f'(x_k)$ in jedem Iterationsschritt

- **Kondition des Nullstellenproblems** (skalare nichtlineare Gleichung): Betrachte die skalare nichtlineare Gleichung

$$f(x) = 0, \quad f: \mathbb{R} \rightarrow \mathbb{R}, \quad \text{Lösung } \bar{x},$$

wobei $\bar{x} \in \mathbb{R}$ eine m -fache Nullstelle von f sei

$$f^{(i)}(\bar{x}) = 0, \quad 0 \leq i \leq m-1, \quad f^{(m)}(\bar{x}) \neq 0.$$

Betrachte weiters die veränderte skalare nichtlineare Gleichung

$$g(x) = 0, \quad g: \mathbb{R} \rightarrow \mathbb{R}, \quad \text{Lösung } \bar{y},$$

unter der Annahme (Abschwächung durch Betrachtung einer Umgebung der Lösung \bar{x})

$$|g(x) - f(x)| \leq \varepsilon, \quad x \in \mathbb{R}.$$

Zu untersuchen ist, wie groß die Abweichung $|\bar{x} - \bar{y}|$ der zugehörigen Lösungen ist. Mittels einer Taylorreihenentwicklung ergibt sich (verwende, daß \bar{x} eine m -fache Nullstelle von f ist)

$$f(\bar{y}) = \sum_{i=0}^m \frac{1}{i!} f^{(i)}(\bar{x}) (\bar{y} - \bar{x})^i + \mathcal{O}(|\bar{y} - \bar{x}|^{m+1}) = \frac{1}{m!} f^{(m)}(\bar{x}) (\bar{y} - \bar{x})^m + \mathcal{O}(|\bar{y} - \bar{x}|^{m+1}).$$

Mittels der Relation $g(\bar{y}) = 0$ folgt

$$\begin{aligned} f(\bar{y}) - g(\bar{y}) &= \frac{1}{m!} f^{(m)}(\bar{x}) (\bar{y} - \bar{x})^m + \mathcal{O}(|\bar{y} - \bar{x}|^{m+1}), \\ \frac{m!}{f^{(m)}(\bar{x})} (f(\bar{y}) - g(\bar{y})) &= \left(1 + \mathcal{O}(|\bar{y} - \bar{x}|)\right) (\bar{y} - \bar{x})^m, \\ (\bar{y} - \bar{x})^m &= \left(1 + \mathcal{O}(|\bar{y} - \bar{x}|)\right) \frac{m!}{f^{(m)}(\bar{x})} (f(\bar{y}) - g(\bar{y})), \\ |\bar{y} - \bar{x}| &= \left(1 + \mathcal{O}(|\bar{y} - \bar{x}|)\right)^{\frac{1}{m}} \left(\frac{m!}{|f^{(m)}(\bar{x})|}\right)^{\frac{1}{m}} |f(\bar{y}) - g(\bar{y})|^{\frac{1}{m}}, \\ |\bar{y} - \bar{x}| &\leq \left(1 + \mathcal{O}(|\bar{y} - \bar{x}|)\right)^{\frac{1}{m}} \left(\frac{m!}{|f^{(m)}(\bar{x})|}\right)^{\frac{1}{m}} \varepsilon^{\frac{1}{m}}. \end{aligned}$$

Dies führt auf die Abschätzung (Vernachlässigung von $\mathcal{O}(|\bar{y} - \bar{x}|)$ für $m \geq 2$)

$$\begin{aligned} m = 1: \quad |\bar{y} - \bar{x}| &\leq \left(1 + \frac{C\varepsilon}{|f'(\bar{x})|}\right) \frac{1}{|f'(\bar{x})|} \varepsilon, \\ m \geq 2: \quad |\bar{y} - \bar{x}| &\leq C \left(\frac{m!}{|f^{(m)}(\bar{x})|}\right)^{\frac{1}{m}} \varepsilon^{\frac{1}{m}}. \end{aligned}$$

Daraus folgt, daß die Berechnung mehrfacher Nullstellen ein schlecht konditioniertes Problem ist (für $0 < \varepsilon \ll 1$ ist $\varepsilon^{\frac{1}{m}} \gg \varepsilon$). Ebenso ist die Berechnung einer einfachen Nullstelle schlecht konditioniert, falls $|f^{(m)}(\bar{x})| \approx 0$ (schleifender Schnitt).

Rundungsfehleranalyse von Fixpunktiterationen: Unter dem Einfluß von Rundungsfehlern wird anstelle der Folge $(x_k)_{k \geq 0}$ eine Folge $(y_k)_{k \geq 0}$ berechnet (mit Startwert $y_0 = x_0 + \delta_0$, die Größen δ_k beschreiben die Rundungsfehler beim Auswerten von φ)

$$x_{k+1} = \varphi(x_k), \quad y_{k+1} = \varphi(y_k) + \delta_k, \quad k \geq 0.$$

Beide Fixpunktiterationen seien konvergent mit Grenzwerten \bar{x} und \bar{y} , d.h. es gilt insbesondere $\varphi(\bar{x}) = \bar{x}$. Weiters gelte

$$|\varphi'(x)| \leq c < 1.$$

Mit Hilfe einer Taylorreihenentwicklung (Mittelwertsatz) folgt (für ein $\zeta_k \in (y_k, \bar{x})$ falls

$y_k < \bar{x}$ oder für ein $\zeta_k \in (\bar{x}, y_k)$ falls $y_k > \bar{x}$, Bezeichnung $d_k = y_k - \bar{x}$)

$$\begin{aligned} d_{k+1} &= \underbrace{y_{k+1}}_{=\varphi(y_k)+\delta_k} - \underbrace{\bar{x}}_{=\varphi(\bar{x})} = \varphi(y_k) - \varphi(\bar{x}) + \delta_k = \varphi'(\zeta_k)(y_k - \bar{x}) + \delta_k \\ &= \varphi'(\zeta_k)(y_k - y_{k+1} + d_{k+1}) + \delta_k, \quad k \geq 0, \\ \Rightarrow d_{k+1} &= \frac{1}{1-\varphi'(\zeta_k)}(\varphi'(\zeta_k)(y_k - y_{k+1}) + \delta_k), \quad k \geq 0, \end{aligned}$$

und weiters (Abschwächung der Voraussetzung $|\varphi'(\zeta_k)| \leq c < 1$ für $k \geq 0$ ausreichend, geometrische Reihe)

$$|d_{k+1}| \leq \frac{1}{|1-\varphi'(\zeta_k)|} (|\varphi'(\zeta_k)||y_k - y_{k+1}| + |\delta_k|) \leq \frac{c}{1-c} |y_k - y_{k+1}| + \frac{1}{1-c} |\delta_k|, \quad k \geq 0.$$

Dies zeigt insbesondere, daß es bei einer Fixpunktiteration zu keiner Akkumulation von Rundungsfehlern kommt (Abhängigkeit vom aktuellen Iterationsschritt)

$$|y_{k+1} - \bar{x}| \leq \frac{c}{1-c} |y_k - y_{k+1}| + \frac{1}{1-c} |\delta_k|, \quad k \geq 0.$$

7.2. Verfahren für den eindimensionalen Fall

- **Bisektionsverfahren:**

- **Situation:** Es sei $f : \mathbb{R} \rightarrow \mathbb{R}$ eine stetige Funktion und $[a, b] \subset \mathbb{R}$ ein Intervall (**Einschließungsintervall**) derart, daß

$$f(a) f(b) < 0.$$

Unter diesen Voraussetzungen existiert (mindestens) eine Nullstelle $\bar{x} \in (a, b)$ (Zwischenwertsatz). Oft wird zudem angenommen, daß das Intervall so gewählt ist, daß es genau eine Nullstelle in (a, b) gibt.

- Das **Bisektionsverfahren** ist ein Zweischnittverfahren. Ausgehend von den Intervallgrenzen a, b und den zugehörigen Funktionswerten $f(a), f(b)$ verwendet es die Berechnung des Intervallmittelpunktes $c = \frac{a+b}{2}$ und des zugehörigen Funktionswerts $f(c)$.
 - * Falls $f(a) f(c) = 0$ ist, ist c die gesuchte Nullstelle.
 - * Falls $f(a) f(c) < 0$ ist, liegt die Nullstelle im Intervall (a, c) und man setzt $b = c$.
 - * Falls $f(a) f(c) > 0$ ist, liegt die Nullstelle im Intervall (c, b) und man setzt $a = c$.

Die Fortführung der Intervallhalbierung ergibt eine Folge von Näherungswerten x_{k+1} (Intervallmittelpunkte) an die gesuchte Nullstelle \bar{x} .

- Zum Nachweis der Konvergenz des Bisektionsverfahrens verwendet man, daß eine streng monotone und beschränkte Folge konvergiert und die Länge der entstehenden Intervalle gegen Null geht. Unter den obigen Voraussetzungen ist das Bisektionsverfahren **global konvergent** mit **Konvergenzordnung** $p = 1$ (lineare Konvergenz) und Konvergenzfaktor $c = \frac{1}{2}$

$$|x_{k+1} - \bar{x}| \leq \frac{1}{2} |x_k - \bar{x}|.$$

- Als **Abbruchkriterium** wählt man meist die Länge des Intervalls

$$b - a < \text{tol},$$

eventuell zusammen mit dem Kriterium $|f(c)| < \text{tol}$ (insbesondere bei schleifenenden Schnitten wäre $|f(c)| < \text{tol}$ als alleiniges Abbruchkriterium jedoch wenig aussagekräftig). Bei der Wahl der Toleranz sollte die Größe der Nullstelle \bar{x} miteinbezogen werden.

- Vgl. Pseudo-Code, Skriptum, S. 120.

- **Regula falsi:**

- Die Regula falsi ist eine Modifikation des Bisektionsverfahren. Anstelle des Intervallmittelpunktes $c = \frac{a+b}{2}$ wird die Nullstelle der Sekante durch die Intervallgrenzen berechnet, d.h. die Bedingung $0 = g(x) = f(a) + \frac{f(b)-f(a)}{b-a}(x-a)$ (insbesondere ist $g(a) = f(a)$ und $g(b) = f(b)$) führt auf $\frac{f(b)-f(a)}{b-a}(c-a) = -f(a)$ und weiters (wegen $f(a)f(b) < 0$ ist Verfahren wohldefiniert)

$$c = a - \frac{b-a}{f(b)-f(a)} f(a).$$

- Unter der Annahme, daß die Funktion f hinreichend oft differenzierbar ist, ist die Regula falsi **global konvergent** mit **Konvergenzordnung** $p = 1$ (lineare Konvergenz).

$$|x_{k+1} - \bar{x}| \leq \kappa |x_k - \bar{x}|.$$

Allerdings kann der Fall eintreten, daß der Konvergenzfaktor κ größer als der Konvergenzfaktor $\kappa = \frac{1}{2}$ des Bisektionsverfahrens ist und die Regula falsi somit langsamer konvergiert.

Denn: Zur einfacheren Untersuchung des Konvergenzverhaltens der Regula falsi wird angenommen, daß die Funktion f konkav (d.h. $f'' < 0$, Graph von f oberhalb jeder Sekante, z.B. $f(x) = -x^2$ und $f''(x) = -2 < 0$) bzw. konvex ist (d.h. $f'' > 0$, Graph von f unterhalb jeder Sekante, z.B. $f(x) = x^2$ und $f''(x) = 2 > 0$). In dieser Situation bleibt eine Intervallgrenze unverändert und die Regula falsi vereinfacht sich zu einem Einschnittverfahren. Man beachte, daß der Schnittpunkt der Sekante gegen die gesuchte Nullstelle konvergiert, die Länge des einschließenden Intervalles konvergiert jedoch *nicht* gegen Null. Beispielsweise für den Fall, daß das linke Intervallende a fest bleibt, sind die Iterationswerte gegeben durch

$$x_{k+1} = \varphi(x_k) = a - \frac{(x_k-a)f(a)}{f(x_k)-f(a)} = \frac{a(f(x_k)-f(a)) - (x_k-a)f(a)}{f(x_k)-f(a)} = \frac{af(x_k) - x_k f(a)}{f(x_k)-f(a)}.$$

Die Ableitung der Iterationsfunktion bei \bar{x} beschreibt die Konvergenzrate des Verfahrens (verwende $f(\bar{x}) = 0$)

$$\begin{aligned} \varphi(x) &= \frac{af(x) - xf(a)}{f(x)-f(a)}, \\ \varphi'(x) &= \frac{af'(x)-f(a)}{f(x)-f(a)} - \frac{(af(x)-xf(a))f'(x)}{(f(x)-f(a))^2}, \\ \varphi'(\bar{x}) &= \frac{f(a)-af'(\bar{x})}{f(a)} + \frac{\bar{x}f(a)f'(\bar{x})}{f(a)^2} = \frac{f(a)+(\bar{x}-a)f'(\bar{x})}{f(a)}. \end{aligned}$$

Mit Hilfe des Mittelwertsatzes folgt damit die Abschätzung (ohne Begründung)

$$\kappa = \left| 1 + \frac{(\bar{x}-a)f'(\bar{x})}{f(a)} \right| < 1$$

und damit die Konvergenz des Verfahrens. \diamond

- Der **Mittelwertsatz der Differentialrechnung** besagt, daß es für eine auf einem abgeschlossenen Intervall definierte und stetige Funktion $f : [a, b] \rightarrow \mathbb{R}$, die im Inneren differenzierbar ist, einen Punkt $\xi \in (a, b)$ gibt, sodaß $(b-a)f'(\xi) = f(b) - f(a)$.

- **Sekantenverfahren:**

- Im Gegensatz zum Bisektionsverfahren und der Regula falsi wird beim **Sekantenverfahren** die Nullstelle der Sekante durch die vorherigen Iterationswerte x_{k-1} und x_k bestimmt, d.h. es gilt

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k).$$

Eine Reduktion dieses Zweischrittverfahrens auf ein Einschrittverfahren ist nicht möglich.

- Das Sekantenverfahren erfüllt die Relation (mit ξ_k zwischen x_{k-1}, x_k, \bar{x} und ζ_k zwischen x_{k-1} und x_k , ohne Begründung)

$$x_{k+1} - \bar{x} = \frac{f''(\xi_k)}{2f'(\zeta_k)} (x_k - \bar{x})(x_{k-1} - \bar{x}), \quad k \geq 0.$$

Sofern die zweite Ableitung von f stetig (und damit beschränkt auf $[a, b]$) und die Nullstelle einfach ist (und somit $f' > 0$ auf $[a, b]$), folgt die Abschätzung

$$\|x_{k+1} - \bar{x}\| \leq c \|x_k - \bar{x}\| \|x_{k-1} - \bar{x}\|, \quad k \geq 0.$$

- Die **Konvergenzordnung** des Sekantenverfahrens ist $p = \lambda_1 = \frac{1}{2}(1 + \sqrt{5})$ (insbesondere keine natürliche Zahl), d.h. es gilt

$$\|x_{k+1} - \bar{x}\| \leq C \|x_k - \bar{x}\|^{\lambda_1}, \quad k \geq 0.$$

Denn: Es bezeichne (ohne Einschränkung der Allgemeinheit sei angenommen, daß $\delta_k > 0$ für $k \geq 0$ und $c > 0$)

$$\delta_k = \|x_k - \bar{x}\|, \quad \eta_k = \ln(c \delta_k), \quad \delta_k = \frac{1}{c} e^{\eta_k}, \quad k \geq 0.$$

Dann gilt (Multiplikation mit $c > 0$, Logarithmieren)

$$\begin{aligned} \delta_{k+1} &\leq c \delta_k \delta_{k-1}, & k \geq 0, \\ c \delta_{k+1} &\leq c \delta_k c \delta_{k-1}, & k \geq 0, \\ \eta_{k+1} &\leq \eta_k + \eta_{k-1}, & k \geq 0. \end{aligned}$$

Die Betrachtung der zugehörigen Gleichung

$$\zeta_{k+1} = \zeta_k + \zeta_{k-1}, \quad k \geq 0,$$

ist ausreichend, weil aus der Abschätzung $\eta_j \leq \zeta_j$ für alle $0 \leq j \leq k$ insbesondere die Relation $\eta_{k+1} \leq \eta_k + \eta_{k-1} \leq \zeta_k + \zeta_{k-1} = \zeta_{k+1}$ folgt. Die Lösung dieser linearen Dreitermrekursion (**Fibonacci-Folge**) verwendet die Umformulierung der Iteration als Einschrittverfahren

$$X_k = \begin{pmatrix} \zeta_k \\ \zeta_{k+1} \end{pmatrix} = \begin{pmatrix} \zeta_k \\ \zeta_k + \zeta_{k-1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \zeta_{k-1} \\ \zeta_k \end{pmatrix} = A X_{k-1} = A^k X_0, \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad k \geq 0,$$

und die Eigenwertzerlegung der Matrix A

$$\chi(\lambda) = \det(A - \lambda I) = \det \begin{pmatrix} -\lambda & 1 \\ 1 & 1 - \lambda \end{pmatrix} = \lambda^2 - \lambda - 1 = 0, \quad \lambda_{1,2} = \frac{1}{2}(1 \pm \sqrt{5}),$$

$$\lambda_1 = \frac{1}{2}(1 - \sqrt{5}): \quad \begin{pmatrix} -\lambda_1 & 1 \\ 1 & 1 - \lambda_1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad v_2 = \lambda_1 v_1, \quad v = v_1 \begin{pmatrix} 1 \\ \lambda_1 \end{pmatrix},$$

$$\lambda_2 = \frac{1}{2}(1 + \sqrt{5}): \quad \begin{pmatrix} -\lambda_2 & 1 \\ 1 & 1 - \lambda_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad v_2 = \lambda_2 v_1, \quad v = v_1 \begin{pmatrix} 1 \\ \lambda_2 \end{pmatrix},$$

$$A = V \Lambda V^{-1}, \quad \Lambda = \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix}, \quad V = \begin{pmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{pmatrix}, \quad V^{-1} = \frac{1}{\lambda_2 - \lambda_1} \begin{pmatrix} \lambda_2 & -1 \\ -\lambda_1 & 1 \end{pmatrix}.$$

Dies führt auf (beachte $\lambda_1 \approx 1.618$ und $\lambda_2 \approx -0.618$, eventuelle Vergrößerung des kleinsten Index)

$$A^k = V \Lambda^k V^{-1}, \quad A^k X_0 = V \Lambda^k V^{-1} X_0,$$

$$\zeta_k = \frac{1}{\lambda_2 - \lambda_1} ((\lambda_2 \zeta_0 - \zeta_1) \lambda_1^k + (\zeta_1 - \lambda_1 \zeta_0) \lambda_2^k) \approx \gamma \lambda_1^k, \quad \gamma = \frac{\lambda_2 \zeta_0 - \zeta_1}{\lambda_2 - \lambda_1}, \quad k \geq 0.$$

Beachte, daß dies dem Ansatz $\zeta_k = \lambda^k$ entspricht. Es folgen die Relationen (für kleinsten Index hinreichend groß)

$$\zeta_k \approx \gamma \lambda_1^k, \quad \zeta_{k+1} \approx \lambda_1 \zeta_k, \quad k \geq 0,$$

$$\tilde{\delta}_k = \frac{1}{c} e^{\zeta_k} \approx \frac{1}{c} e^{\gamma \lambda_1^k}, \quad k \geq 0,$$

$$\tilde{\delta}_{k+1} = \frac{1}{c} e^{\zeta_{k+1}} \approx \frac{1}{c} e^{\gamma \lambda_1^{k+1}} = \frac{1}{c} (e^{\gamma \lambda_1^k})^{\lambda_1} = c^{\lambda_1 - 1} \left(\frac{1}{c} e^{\gamma \lambda_1^k} \right)^{\lambda_1} \approx c^{\lambda_1 - 1} \tilde{\delta}_k^{\lambda_1}, \quad k \geq 0.$$

$$\|x_{k+1} - \bar{x}\| \leq c^{\lambda_1 - 1} \|x_k - \bar{x}\|^{\lambda_1}, \quad k \geq 0.$$

Somit ergibt sich die Behauptung. \diamond

- **Bemerkung:** Ein Vergleich der Konvergenzrate und der benötigten Funktionsauswertungen ergibt, daß das Sekantenverfahren effizienter als das Newton-Verfahren ist, jedoch ebenfalls nicht global konvergent.
- **Bemerkungen:**
 - Der **Dekker-Algorithmus** kombiniert die Idee des Bisektionsverfahrens und des Sekantenverfahrens und besitzt gute lokale und globale Konvergenzeigenschaften (Sekantenschritte, die das Einschließungsintervall verlassen würden bzw. nahezu verlassen würden, werden durch Bisektionsschritte ersetzt).
 - Das **Verfahren von Muller** mit Konvergenzrate $\kappa \approx 1.84$ erweitert das Sekantenverfahren (quadratisches Polynom durch drei aufeinanderfolgende Iterationswerte $(x_j, f(x_j))$ für $j = k - 2, k - 1, k$, Nullstelle ist neuer Iterationswert).
 - Das **Verfahren von Brent** verwendet die Idee der inversen Interpolation (quadratisches Polynom durch $(f(x_j), x_j)$ für $j = k - 2, k - 1, k$, Auswerten bei Null ergibt den neuen Iterationswert) und die des Dekker-Algorithmus.

7.3. Der mehrdimensionale Fall

- **Vorbemerkung:** Bereits bei zwei nichtlinearen Gleichungen in zwei Unbekannten

$$f(x, y) = 0, \quad g(x, y) = 0,$$

zeigen sich die Schwierigkeiten bei der Lösung eines nichtlinearen Gleichungssystems. Vgl. Graphik, Skriptum, S. 124 (Lösungen sind durch den **Schnitt der Höhenlinien** gegeben).

- Sämtliche praktikablen Verfahren zur näherungsweise Lösung eines nichtlinearen Gleichungssystems sind Modifikationen des **Newton-Verfahrens** mit
 - besseren Konvergenzeigenschaften und
 - verringertem Aufwand bei der Berechnung der ersten Ableitung.

- **Quasi-Newton-Verfahren:**

- Das Quasi-Newton-Verfahren beruht auf der Idee, das lineare Gleichungssystem

$$f(x) = 0, \quad f: \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

mit dem Minimierungsproblem

$$F(x) = \frac{1}{2} \|f(x)\|_2^2, \quad F: \mathbb{R}^n \rightarrow \mathbb{R},$$

in Verbindung zu bringen. Eine Lösung des linearen Gleichungssystems ist ein globales Minimum von F (wegen $\|f(x)\|_2 = 0 \Leftrightarrow f(x) = 0$). Deshalb sollten die gewählten Iterationsschritte in Abstiegsrichtung sein, d.h. es sollte gelten

$$F(x_{k+1}) < F(x_k), \quad k \geq 0.$$

Das Newton-Verfahren erfüllt diese Bedingung, denn mit (beachte $F'(x) \in \mathbb{R}^{n \times 1}$)

$$\begin{aligned} F(x) &= \frac{1}{2} \|f(x)\|_2^2 = \frac{1}{2} (f(x))^T f(x) = \frac{1}{2} \sum_{i=1}^n (f_i(x))^2, \\ \partial_{x_\ell} F(x) &= \sum_{i=1}^n f_i(x) \underbrace{\partial_{x_\ell} f_i(x)}_{=(f'(x))_{i\ell}} = \sum_{i=1}^n (f'(x))_{\ell i}^T f_i(x) = ((f'(x))^T f(x))_{\ell 1}, \\ F'(x) &= (f(x))^T f'(x), \end{aligned}$$

und mittels einer Taylorreihenentwicklung ergibt sich (sofern $\mathcal{O}(\|\xi_k\|_2^2)$ hinreichend klein)

$$\begin{aligned} x_{k+1} &= x_k - \xi_k, \quad \xi_k = (f'(x_k))^{-1} f(x_k), \\ F'(x_k) \xi_k &= (f(x_k))^T f'(x_k) (f'(x_k))^{-1} f(x_k) = \|f(x_k)\|_2^2, \\ F(x_{k+1}) &= F(x_k) - F'(x_k) \xi_k + \mathcal{O}(\|\xi_k\|_2^2) = F(x_k) - \|f(x_k)\|_2^2 + \mathcal{O}(\|\xi_k\|_2^2) < F(x_k), \end{aligned}$$

d.h. ein Iterationsschritt des Newtonverfahrens entspricht einem Schritt in Richtung abnehmender Funktionswerte von F .

- Obwohl Iterationsschritte des Newtonverfahrens in Abstiegsrichtung erfolgen, kann es passieren, daß die Schrittlänge zu groß ist und somit nicht die optimale Verkleinerung der Funktionswerte von F erreicht wird. Beim **Quasi-Newton-Verfahren** verwendet man deshalb stattdessen die Iterationsfunktion

$$x_{k+1} = x_k - \lambda_k \xi_k, \quad \xi_k = (f'(x_k))^{-1} f(x_k), \quad k \geq 0.$$

Die zusätzliche **Schrittweite** λ_k wird dabei so bestimmt, daß $F(x_k - \lambda_k \xi_k)$ (zumindest näherungsweise) minimal wird (**Liniensuche**: Quadratischer (oder kubischer) Ansatz $F(x_k - \lambda \xi_k) = p(\lambda) = a_0 + b_0 \lambda + c_0 \lambda^2$ mit Kenntnis der Funktionswerte bei $\lambda = 0, 1$ sowie der Ableitung bei $\lambda = 0$ führt auf die berechenbaren Koeffizienten $a_0 = F(x_k)$, $b_0 = F'(x_k) \xi_k$, $c_0 = F(x_k + \xi_k) - a_0 - b_0$ und die Wahl $\lambda_k = \lambda_{\min}$. **Armijo–Goldstein Bedingung**: $F(x_{k+1}) = F(x_k - \lambda_k \xi_k) \leq F(x_k) - \alpha \lambda_k F'(x_k) \xi_k$ bei vorgegebenem Parameter α).

- Unter gewissen Voraussetzungen an die Funktion f , den Startwert x_0 und die Schrittweitenfolge $(\lambda_k)_{k \geq 0}$ kann gezeigt werden, daß das Quasi-Newton-Verfahren **global konvergent** ist, vgl. Skriptum, S. 126.
- **Vorbemerkung**: Um den erheblichen Aufwand bei der Berechnung der Jacobimatrix $f'(x_k)$ für $k \geq 0$ zu verringern, verwendet man beim **vereinfachten Newton-Verfahren** die Iteration (dies erfordert beispielsweise nur eine einzige LR-Zerlegung von $f'(x_0)$ und jeweils eine Vorwärts- und Rückwärtssubstitution pro Iterationsschritt)

$$x_{k+1} = x_k - \xi_k, \quad f'(x_0) \xi_k = f(x_k), \quad k \geq 0.$$

Allerdings ist dann im Allgemeinen die Konvergenz linear und nicht quadratisch.

Das **Verfahren von Broyden** verwendet folgende Iterationsvorschrift im k -ten Schritt für $k \geq 0$ (üblicherweise mit $J_0 = f'(x_0)$)

$$\begin{cases} \text{Für } J_k \approx f'(x_k) \text{ bestimme } \xi_k \text{ aus } J_k \xi_k = f(x_k) \text{ und berechne } x_{k+1} = x_k - \xi_k. \\ \text{Bestimme } J_{k+1} \text{ aus } J_{k+1} \xi_k = -(f(x_{k+1}) - f(x_k)). \end{cases}$$

Im eindimensionalen Fall entspricht das Verfahren dem Sekantenverfahren und insbesondere J_{k+1} der Steigung der Sekante durch $(x_k, f(x_k))$ und $(x_{k+1}, f(x_{k+1}))$ (wegen $-\xi_k = x_{k+1} - x_k$, sofern $J_k \neq 0$)

$$J_{k+1} = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}.$$

Im mehrdimensionalen Fall ist die Matrix J_{k+1} nicht eindeutig festgelegt (beispielsweise sind die Koeffizienten einer Matrix $A \in \mathbb{R}^{2 \times 2}$ bei Vorgabe von $x \in \mathbb{R}^2$ und $b \in \mathbb{R}^2$ durch die Bedingung $Ax = b \Leftrightarrow a_{11}x_1 + a_{12}x_2 = b_1, a_{21}x_1 + a_{22}x_2 = b_2$ nicht eindeutig festgelegt). Durch die Zusatzforderung (**Rang-1-Modifikation**)

$$J_{k+1} = J_k + uv^T$$

folgt (Einsetzen in $J_{k+1} \xi_k = f(x_k) - f(x_{k+1})$)

$$\underbrace{v^T \xi_k}_{=-\frac{1}{c} \in \mathbb{R}} u = f(x_k) - f(x_{k+1}) - \underbrace{J_k \xi_k}_{=f(x_k)} = -f(x_{k+1}) \iff u = c f(x_{k+1}).$$

Die spezielle Wahl $v = -\xi_k$ führt auf $(-\frac{1}{c} = v^T \xi_k = -\|\xi_k\|_2^2 \Leftrightarrow c = \frac{1}{\|\xi_k\|_2^2}, u = \frac{1}{\|\xi_k\|_2^2} f(x_{k+1}))$

$$J_{k+1} = J_k - \frac{1}{\|\xi_k\|_2^2} f(x_{k+1}) \xi_k^T,$$

das Verfahren von Broyden.

Bemerkung: Falls für eine Matrix $A \in \mathbb{R}^{n \times n}$ die LR-Zerlegung (oder QR-Zerlegung) bekannt ist, benötigt die Berechnung der LR-Zerlegung (oder QR-Zerlegung) einer Rang-1-Modifikation von A

$$A = LR, \quad A + uv^T = (L + \ell)(R + r) = LR + \ell R + Lr + \ell r, \\ uv^T = \ell R + Lr + \ell r,$$

lediglich $\mathcal{O}(n^2)$ Operationen (und nicht $\mathcal{O}(n^3)$ Operationen).

Themenüberblick II

1. Polynominterpolation

1.1. Aufgabenstellung und Notation

1.2. Berechnung des Polynominterpolanten nach Aitken–Neville

1.3. Dividierte Differenzen

1.4. Kondition der Polynominterpolation

1.5. Vor- und Nachteile der Polynominterpolation

2. Polynom-Splines

2.1. Kubische Spline-Interpolanten

2.2. B-Splines

2.3. Linearkombinationen von B-Splines

3. Numerische Integration

3.1. Elementare Quadraturformeln

3.2. Best-Approximation des Integrals

3.3. Romberg-Quadratur

3.4. Adaptive Methoden

4. Anfangswertprobleme für gewöhnliche Differentialgleichungen

4.1. Theoretischer Hintergrund

4.2. Diskretisierungen und Diskretisierungsfehler

4.3. Explizite Einschrittverfahren

4.4. A-Stabilität

5. Randwertprobleme für gewöhnliche Differentialgleichungen

5.1. Problemstellung

5.2. Lösung durch Rückführung auf ein Anfangswertproblem (Schießverfahren)

5.3. Differenzenverfahren

5.4. Kollokationsverfahren

6. Iterative Verfahren für lineare Gleichungssysteme

6.1. Klassische Iterationsverfahren

6.2. Die Idee der Mehrgitter-Verfahren

6.3. Unterraumverfahren CG und GMRES

1. Polynominterpolation

- **Vorbemerkungen:**

- Ein wichtiges Gebiet der Numerischen Mathematik ist die **Interpolation und Approximation** von Funktionen. Man unterscheidet dabei den Fall **univariater Funktionen** (d.h. Funktionen in einer Veränderlichen) und den im Allgemeinen um ein Vielfaches schwierigeren Fall **multivariater Funktionen** (d.h. Funktionen in mehreren Veränderlichen).
- Thema der Vorlesung ist insbesondere die **Interpolation mittels Polynomen** für univariate Funktionen. Vorteile der Verwendung von Polynomfunktionen sind ihre einfache Darstellung und mittels Horner-Schema rasche und stabile Auswertung (vgl. Numerische Mathematik I). Ein Nachteil der Polynominterpolation ist jedoch die schlechte Kondition bei höherem Polynomgrad.

- **Anwendungen:**

- * Numerische Verfahren für nichtlineare Gleichungen (z.B. Interpolation durch Polynome vom Grad 1 zur Herleitung des Sekantenverfahrens)
 - * Numerische Integration (Newton–Côtes Formeln)
 - * Numerische Verfahren für gewöhnliche Differentialgleichungen (Kollokationsverfahren, Adamsverfahren)
 - Eine vorteilhafte Alternative zur Polynominterpolation ist die **Interpolation mittels Splines** (stückweise Polynome, Zusatzbedingungen).
 - Eine Alternative zur Interpolation insbesondere bei fehlerbehafteten Daten ist die Approximation (Lineare Regression, Nichtlineare Regression).
- **Vorsicht!** Notationen unterscheiden sich teilweise von den im Skriptum verwendeten Notationen.

1.1. Aufgabenstellung und Notation

- **Problemstellung:** Zu vorgegebenen **Datenpunkten**, d.h. zu gegebenen reellen Stützstellen und zugehörigen reellen Stützwerten, wird eine Polynomfunktion gesucht, welche die Datenpunkte *interpoliert*. Die Stützwerte werden als Funktionswerte oder Ableitungen einer hinreichend oft differenzierbaren Funktion $f : [a, b] \rightarrow \mathbb{R}$ interpretiert.

Vgl. **Illustration**, Skriptum, S. 15 (Zugrundeliegende Funktion und interpolierende Polynomfunktion bei äquidistanten bzw. nicht äquidistanten Stützstellen sowie interpolierende stückweise kubische Polynomfunktion bei äquidistanten Stützstellen).

- **Bezeichnungen:** Für $n \geq 0$ bezeichnet \mathbb{P}_n den $(n + 1)$ -dimensionalen Vektorraum der reellen Polynome vom **Grad** $\leq n$ (bzw. von der **Ordnung** $n + 1$), d.h. es gilt

$$\mathbb{P}_n = \{p : \mathbb{R} \rightarrow \mathbb{R} \text{ Polynom vom Grad } \leq n\}.$$

Die Monome

$$p_i : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto x^i, \quad 0 \leq i \leq n,$$

bilden die **Taylor-Basis** von \mathbb{P}_n .

Bemerkung: Eine (unwesentliche) Verallgemeinerung ist die Betrachtung der Polynomfunktionen $p_i(x) = (x - x_0)^i$ für $0 \leq i \leq n$ (Reduktion durch Variablentransformation $x \mapsto x - x_0$, vgl. Taylorreihenentwicklung).

- **Fragestellung:** Präzisierung der Problemstellung (Behandlung einfacher und mehrfacher Stützstellen).

Interpolationsaufgabe (Definition 1.1): Es sei $f : [a, b] \rightarrow \mathbb{R}$ eine gegebene Funktion und $x_i \in [a, b]$ für $0 \leq i \leq n$. Ein Polynom $p \in \mathbb{P}_n$ heißt **Polynominterpolant** von f zu den $n + 1$ **Stützstellen** $(x_i)_{0 \leq i \leq n}$, wenn das Restglied $r = p - f$ die Nullstellen $x_i \in [a, b]$ für $0 \leq i \leq n$ besitzt, d.h. es gilt

$$r(x) = p(x) - f(x) = h(x) \prod_{i=0}^n (x - x_i)$$

mit einer (differenzierbaren) Funktion $h : [a, b] \rightarrow \mathbb{R}$. Als **Interpolationsaufgabe** bezeichnet man das Problem, zu vorgegebener Funktion f und vorgegebenen Stützstellen $(x_i)_{0 \leq i \leq n}$ den zugehörigen Polynominterpolanten p zu bestimmen.

Bemerkungen:

- Für theoretische Überlegungen wird manchmal zusätzlich angenommen, daß die Stützstellen angeordnet sind, d.h. es ist $a \leq x_0 \leq \dots \leq x_n \leq b$.
- Im Fall **einfacher Stützstellen**, d.h. paarweise verschiedener Stützstellen

$$x_i \neq x_j, \quad i \neq j, \quad 0 \leq i, j \leq n,$$

folgt mittels Produktregel

$$\begin{aligned} r(x) &= p(x) - f(x) = \tilde{h}(x)(x - x_i), & \tilde{h}(x_i) &\neq 0, \\ r(x_i) &= p(x_i) - f(x_i) = 0, \\ r'(x) &= p'(x) - f'(x) = \tilde{h}'(x)(x - x_i) + \tilde{h}(x), & r'(x_i) &= p'(x_i) - f'(x_i) \neq 0, \end{aligned}$$

und somit

$$p(x_i) = f(x_i), \quad 0 \leq i \leq n.$$

Die Interpolationsaufgabe besteht also darin, zu vorgegebenen Daten $(x_i, y_i)_{0 \leq i \leq n}$ mit **Stützwerten** $y_i = f(x_i)$ für $0 \leq i \leq n$ ein Polynom $p \in \mathbb{P}_n$ zu bestimmen, welches die **Interpolationsbedingungen** erfüllt

$$p(x_i) = y_i, \quad 0 \leq i \leq n.$$

- Im Fall zumindest einer **mehrfachen Stützstelle** spricht man von **Hermite-Interpolation**. Beispielsweise sei x_i eine m -fache Stützstelle (wobei $m \geq 2$, jedoch keine $(m + 1)$ -fache Stützstelle), d.h. bei angeordneten Stützstellen gelte

$$x_{i-1} < x_i = \dots = x_{i+m-1} < x_{i+m}.$$

Da in diesem Fall das Restglied den Term $(x - x_i)^m$ enthält, folgt mittels Produktregel bzw. der Regel von Leibniz

$$\begin{aligned} r(x) &= p(x) - f(x) = \tilde{h}(x)(x - x_i)^m, & \tilde{h}(x_i) &\neq 0, \\ r(x_i) &= p(x_i) - f(x_i) = 0, \\ r'(x) &= p'(x) - f'(x) = \tilde{h}'(x)(x - x_i)^m + m\tilde{h}(x)(x - x_i)^{m-1}, \\ r'(x_i) &= p'(x_i) - f'(x_i) = 0, \\ r^{(k)}(x) &= p^{(k)}(x) - f^{(k)}(x) = \sum_{j=0}^k \frac{k!m!}{j!(k-j)!(m-j)!} \tilde{h}^{(k-j)}(x)(x - x_i)^{m-j}, & 0 \leq k \leq m-1, \\ r^{(k)}(x_i) &= p^{(k)}(x_i) - f^{(k)}(x_i) = 0, & 0 \leq k \leq m-1, \end{aligned}$$

und somit

$$p^{(k)}(x_i) = f^{(k)}(x_i), \quad 0 \leq k \leq m-1.$$

Bei mehrfachen Nullstellen beinhalten die Interpolationsbedingungen neben Funktionswerten auch Ableitungen der zugehörigen Funktion.

- **Fragestellung:** Aussage zur Existenz und Eindeutigkeit des Polynominterpolanten (Konstruktion).

Eindeutige Lösbarkeit der Interpolationsaufgabe (Satz 1.2): Die Lösung der Interpolationsaufgabe ist eindeutig bestimmt.

Denn: Zum Nachweis der Existenz und Eindeutigkeit des Interpolationspolynoms p wird die **Darstellung nach Newton**

$$\begin{aligned} p(x) &= \sum_{i=0}^n c_i \prod_{\ell=0}^{i-1} (x - x_\ell) \\ &= c_0 + c_1 (x - x_0) + c_2 (x - x_0)(x - x_1) + \cdots + c_n (x - x_0) \cdots (x - x_{n-1}) \end{aligned}$$

mit zu bestimmenden Koeffizienten $(c_i)_{0 \leq i \leq n}$ verwendet. In diesem Fall ergibt sich ein lineares Gleichungssystem, dessen eindeutige Lösbarkeit offensichtlich ist. Die *übliche* Darstellung einer Polynomfunktion mittels Taylor-Basis würde auf ein lineares Gleichungssystem für die Koeffizienten führen, dessen Lösbarkeit nicht ersichtlich ist.

- Bei einfachen Stützstellen führen die Interpolationsbedingungen

$$p(x_i) = y_i = f(x_i), \quad 0 \leq i \leq n,$$

auf das lineare Gleichungssystem

$$\begin{aligned} y_0 &= p(x_0) = c_0, \\ y_1 &= p(x_1) = c_0 + c_1 (x_1 - x_0), \\ y_2 &= p(x_2) = c_0 + c_1 (x_2 - x_0) + c_2 (x_2 - x_0)(x_2 - x_1), \\ &\vdots \\ y_n &= p(x_n) = c_0 + c_1 (x_n - x_0) + c_2 (x_n - x_0)(x_n - x_1) + \cdots \\ &\quad + c_n (x_n - x_0) \cdots (x_n - x_{n-1}), \end{aligned}$$

dessen Lösung $c = (c_0, \dots, c_n)$ schrittweise berechnet werden kann ($n + 1$ Gleichungen für $n + 1$ Unbekannte, Lösung mittels Vorwärtseinsetzen, Koeffizient bei c_i gegeben durch $(x_i - x_0) \cdots (x_i - x_{i-1}) \neq 0$).

- Der allgemeine Fall mit mehrfachen Nullstellen beruht auf ähnlichen Ideen, allerdings ist es komplizierter, das resultierende lineare Gleichungssystem anzugeben. Beispielsweise bei einfachen Nullstellen x_0, x_2 und einer dreifachen Nullstelle x_1 führt das Einsetzen der Interpolationsbedingungen

$$\begin{aligned} p(x_i) &= y_i = f(x_i), \quad i = 0, 1, 2, \\ p'(x_1) &= y'_1 = f'(x_1), \quad p''(x_1) = y''_1 = f''(x_1), \end{aligned}$$

in die Darstellung

$$p(x) = c_0 + c_1 (x - x_0) + c_2 (x - x_0)(x - x_1) + c_3 (x - x_0)(x - x_1)^2 + c_4 (x - x_0)(x - x_1)^3$$

auf das lineare Gleichungssystem

$$\begin{aligned} y_0 &= p(x_0) = c_0, \\ y_1 &= p(x_1) = c_0 + c_1(x_1 - x_0), \\ y_1' &= p'(x_1) = c_1 + c_2(x_1 - x_0), \\ y_1'' &= p''(x_1) = 2c_2 + 2c_3(x_1 - x_0), \\ y_2 &= p(x_2) = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) + c_3(x_2 - x_0)(x_2 - x_1)^2 \\ &\quad + c_4(x_2 - x_0)(x_2 - x_1)^3, \end{aligned}$$

dessen Lösung schrittweise bestimmt werden kann (5 Gleichungen für 5 Unbekannte, Lösung mittels Vorwärtseinsetzen, Koeffizient bei c_i ungleich Null). \diamond

Bemerkung: Zur Bestimmung der Koeffizienten des **Interpolationspolynoms nach Newton** verwendet man anstelle der Lösung des im Beweis von Satz 1.2 angegebenen Gleichungssystems üblicherweise das **Schema von Aitken–Neville** (vgl. Abschnitt 1.2).

- **Fragestellung:** Angabe des Polynominterpolanten (einfache Stützstellen).

Lagrange-Polynome (Definition 1.3): Für $n + 1$ paarweise verschiedene Stützstellen $(x_i)_{0 \leq i \leq n}$ (d.h. $x_i \neq x_j$ für $i \neq j$ und $0 \leq i, j \leq n$) sind die **Lagrange-Polynome** $(L_j)_{0 \leq j \leq n}$ definiert durch

$$L_j(x) = \prod_{\substack{0 \leq i \leq n \\ i \neq j}} \frac{x - x_i}{x_j - x_i}, \quad 0 \leq j \leq n.$$

Vgl. **Illustration**, Skriptum, S. 8 (Äquidistante Stützstellen $x_i = i$ für $0 \leq i \leq n = 20$, Lagrange-Polynome L_{10} und L_{11}).

Eigenschaften der Lagrange-Polynome:

- Die Lagrange-Polynome sind Polynome vom Grad n und insbesondere gilt $L_j \in \mathbb{P}_n$ für $0 \leq j \leq n$.
- Es gilt $L_j(x_i) = \delta_{ij}$ für $0 \leq i, j \leq n$. Insbesondere sind die Lagrange-Polynome $(L_j)_{0 \leq j \leq n}$ linear unabhängig und bilden somit eine Basis des Vektorraumes \mathbb{P}_n .

Lagrange-Interpolationsformel (Satz 1.4): Zu **einfachen** Stützstellen $(x_i)_{0 \leq i \leq n}$ und zugehörigen Stützwerten $y_i = f(x_i)$ für $0 \leq i \leq n$ ist die eindeutig bestimmte Lösung der Interpolationsaufgabe gegeben durch

$$p = \sum_{i=0}^n y_i L_i.$$

Denn: Aufgrund der Basiseigenschaft der Lagrange-Polynome folgt für jedes Polynom $p \in \mathbb{P}_n$ die Darstellung

$$p = \sum_{i=0}^n \alpha_i L_i$$

mit eindeutig bestimmten reellen Koeffizienten $(\alpha_i)_{0 \leq i \leq n}$. Einsetzen der Stützstellen

$$y_j = p(x_j) = \sum_{i=0}^n \alpha_i \underbrace{L_i(x_j)}_{=\delta_{ij}} = \alpha_j$$

führt auf die angegebene Relation. \diamond

Bemerkung: Die obige Darstellung des Interpolationspolynoms ist für **theoretische Überlegungen** wesentlich. Für praktische Berechnungen bei einer größeren Anzahl an Datenpunkten sind alternative Darstellungen vorzuziehen, da es aufgrund von stark anwachsenden Funktionswerten mit unterschiedlichem Vorzeichen zu numerisch ungünstigen Operationen und insbesondere zur Auslöschung signifikanter Stellen kommen kann.

1.2. Berechnung des Polynominterpolanten nach Aitken–Neville

- **Situation:** Es sei $f : [a, b] \rightarrow \mathbb{R}$ hinreichend oft differenzierbar, und es seien $(x_i)_{0 \leq i \leq n}$ mit $x_i \in [a, b]$ für $0 \leq i \leq n$ **einfache** Stützstellen (und zusätzlich angeordnet).

Fragestellung: Gesucht ist eine für numerische Berechnungen **vorteilhafte Darstellung** des Polynominterpolanten durch vorgegebene Datenpunkte (nach Satz 1.2 eindeutig bestimmt)

$$p(x_i) = y_i = f(x_i), \quad 0 \leq i \leq n.$$

Der in Abschnitt 1.2 behandelte Zugang wird in Abschnitt 1.3 nochmals betrachtet und führt dann auf den bei praktischen Berechnungen verwendeten Zugang (Darstellung nach Newton, Berechnung der Koeffizienten des Interpolationspolynoms mittels Schema der dividierten Differenzen).

Vorbemerkung: Zur Konstruktion des Polynominterpolanten durch $n + 1$ Datenpunkte

$$p(x_i) = y_i = f(x_i), \quad 0 \leq i \leq n,$$

wird einerseits der Polynominterpolant durch die ersten n Datenpunkte

$$q(x_i) = y_i = f(x_i), \quad 0 \leq i \leq n - 1,$$

und andererseits der Polynominterpolant durch die letzten n Datenpunkte

$$\tilde{q}(x_i) = y_i = f(x_i), \quad 1 \leq i \leq n,$$

betrachtet. Der Ansatz ($p \in \mathbb{P}_n$ wegen $q, \tilde{q} \in \mathbb{P}_{n-1}$)

$$p(x) = c(x - x_n)q(x) + \tilde{c}(x - x_0)\tilde{q}(x)$$

und Einsetzen des ersten bzw. letzten Datenpunktes

$$\begin{aligned} y_0 = p(x_0) = c(x_0 - x_n)q(x_0) = c(x_0 - x_n)y_0 &\Rightarrow c = -\frac{1}{x_n - x_0}, \\ y_n = p(x_n) = \tilde{c}(x_n - x_0)\tilde{q}(x_n) = \tilde{c}(x_n - x_0)y_n &\Rightarrow \tilde{c} = \frac{1}{x_n - x_0}, \end{aligned}$$

führt auf die Darstellung

$$p(x) = \frac{1}{x_n - x_0} \left((x - x_0)\tilde{q}(x) - (x - x_n)q(x) \right).$$

Wie gewünscht gilt in den Zwischenpunkten die Identität

$$\begin{aligned} p(x_i) &= \frac{1}{x_n - x_0} \left((x_i - x_0)\tilde{q}(x_i) - (x_i - x_n)q(x_i) \right) \\ &= \frac{1}{x_n - x_0} \left((x_i - x_0)y_i - (x_i - x_n)y_i \right) = y_i, \quad 1 \leq i \leq n - 1. \end{aligned}$$

Aufgrund der Eindeutigkeit des Polynominterpolanten ist damit die obige Darstellung nachgewiesen.

Das **Schema von Aitken–Neville** ist ein konstruktives Verfahren zur Berechnung des Polynominterpolanten, das auf der Idee beruht, den Interpolanten durch $n + 1$ Datenpunkte auf zwei Interpolanten durch n Datenpunkte zurückzuführen (siehe Vorbemerkung). Es bezeichne P_j^k den Polynominterpolanten durch die Datenpunkte $(x_i, y_i)_{j \leq i \leq k}$

$$P_j^k(x_i) = y_i = f(x_i), \quad j \leq i \leq k.$$

Weiters sei P_j^{k-1} der Polynominterpolant durch die Datenpunkte $(x_i, y_i)_{j \leq i \leq k-1}$

$$P_j^{k-1}(x_i) = y_i = f(x_i), \quad j \leq i \leq k-1,$$

und P_{j+1}^k der Polynominterpolant durch die Datenpunkte $(x_i, y_i)_{j+1 \leq i \leq k}$

$$P_{j+1}^k(x_i) = y_i = f(x_i), \quad j+1 \leq i \leq k.$$

Wie zuvor führt der Ansatz

$$P_j^k(x) = c(x - x_k) P_j^{k-1}(x) + \tilde{c}(x - x_j) P_{j+1}^k(x)$$

und Einsetzen des ersten bzw. letzten Datenpunktes auf die Darstellung

$$P_j^k(x) = \frac{1}{x_k - x_j} ((x - x_j) P_{j+1}^k(x) - (x - x_k) P_j^{k-1}(x)).$$

Wie gewünscht gilt in den Zwischenpunkten die Identität

$$P_j^k(x_i) = y_i, \quad j+1 \leq i \leq k-1,$$

und aufgrund der Eindeutigkeit des Polynominterpolanten ist damit die obige Darstellung nachgewiesen. Diese Rekursion kann verwendet werden, den Polynominterpolanten $p = P_0^n$ ausgehend von den konstanten Funktionen $P_i^i = y_i$ für $0 \leq i \leq n$ schrittweise zu berechnen, vgl. Schema von Aitken–Neville und Algorithmus, Skriptum, S. 10.

Bemerkung: Im Fall einer **mehrfachen Nullstelle** $x_j = \dots = x_k$ wird anstelle der zuvor angegebenen Relation die Darstellung von P_j^k mittels Taylorpolynom verwendet

$$P_j^k(x) = \sum_{\ell=0}^{k-j} \frac{1}{\ell!} f^{(\ell)}(x_j) (x - x_j)^\ell$$

und beim Schema von Aitken–Neville an der entsprechenden Stelle eingesetzt, vgl. Skriptum, S. 11.

1.3. Dividierte Differenzen

- **Vorbemerkung:** Bei Polynomfunktionen $p \in \mathbb{P}_n$ in der *üblichen* Darstellung mittels Taylor-Basis

$$p(x) = \sum_{i=0}^n \tilde{c}_i x^i$$

ist die Kenntnis der Koeffizienten $(\tilde{c}_i)_{0 \leq i \leq n}$ zur Auswertung mittels Horner-Schema ausreichend. Ähnlich ist bei der Darstellung nach Newton

$$p(x) = \sum_{i=0}^n c_i \prod_{\ell=0}^{i-1} (x - x_\ell)$$

die Kenntnis der Stützstellen $(x_i)_{0 \leq i \leq n}$ und der Koeffizienten $(c_i)_{0 \leq i \leq n}$ ausreichend. Die in Abschnitt 1.2 hergeleitete Rekursion wird nun verwendet, um die Koeffizienten in dieser Darstellung des Polynominterpolanten zu berechnen, das Polynom wird dann mittels eines modifizierten Horner-Schemas ausgewertet. Bei **praktischer Berechnung** des Polynominterpolanten (höheren Grades) wird ausschließlich **diese Vorgehensweise** angewendet, die Überlegungen in Abschnitt 1.2 dienen zu Motivation, alternative Darstellungen z.B. mittels Lagrange-Basisfunktionen dienen vorwiegend theoretischen Überlegungen.

Dividierte Differenzen, Interpolationsformel nach Newton (Definition 1.5, Satz 1.6, Satz 1.7): Die Leitkoeffizienten der im Schema von Aitken–Neville auftretenden Polynome P_j^k heißen **dividierte Differenzen**. Bei einfachen Stützstellen $(x_i)_{0 \leq i \leq n}$ sind die dividierten Differenzen rekursiv durch

$$\begin{aligned} \delta y_{i,i+1} &= \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, & 0 \leq i \leq i+1 \leq n, \\ \delta^2 y_{i,i+1,i+2} &= \frac{\delta y_{i+1,i+2} - \delta y_{i,i+1}}{x_{i+2} - x_i}, & 0 \leq i \leq i+2 \leq n, \\ \delta^{j+1} y_{i,\dots,i+j+1} &= \frac{\delta^j y_{i+1,\dots,i+j+1} - \delta^j y_{i,\dots,i+j}}{x_{i+j+1} - x_i}, & 0 \leq i \leq i+j+1 \leq n, \quad 0 \leq j \leq n-1, \end{aligned}$$

gegeben; mit $\delta^0 y_i = y_i$ gilt zudem $\delta y_{i,i+1} = \frac{\delta^0 y_{i+1} - \delta^0 y_i}{x_{i+1} - x_i}$. Im Fall einer mehrfachen Stützstelle setzt man

$$x_i = \dots = x_{i+j}: \quad \delta^j y_{i,\dots,i+j} = \frac{1}{(j-i)!} f^{(j-i)}(x_i), \quad 0 \leq i \leq i+j \leq n, \quad 0 \leq j \leq n.$$

Das **Interpolationspolynom nach Newton** ist durch

$$\begin{aligned} p(x) &= \sum_{i=0}^n \delta^i y_{0,\dots,i} \prod_{\ell=0}^{i-1} (x - x_\ell) \\ &= y_0 + \delta y_{0,1} (x - x_0) + \delta^2 y_{0,1,2} (x - x_0) (x - x_1) + \dots + \delta^n y_{0,\dots,n} (x - x_0) \cdots (x - x_{n-1}) \end{aligned}$$

gegeben.

Schematische Darstellung:

x_i	y_i	$\delta y_{i,i+1}$	$\delta^2 y_{i,i+1,i+2}$	$\delta^3 y_{i,i+1,i+2,i+3}$
x_0	y_0	$\delta y_{0,1} = \frac{y_1 - y_0}{x_1 - x_0}$		
x_1	y_1	$\delta y_{1,2} = \frac{y_2 - y_1}{x_2 - x_1}$	$\delta^2 y_{0,1,2} = \frac{\delta y_{1,2} - \delta y_{0,1}}{x_2 - x_0} = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}$	
x_2	y_2	$\delta y_{2,3} = \frac{y_3 - y_2}{x_3 - x_2}$	$\delta^2 y_{1,2,3} = \frac{\delta y_{2,3} - \delta y_{1,2}}{x_3 - x_1} = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$	$\delta^3 y_{0,1,2,3} = \frac{\delta^2 y_{1,2,3} - \delta^2 y_{0,1,2}}{x_3 - x_0}$
x_3	y_3			
\vdots	\vdots	\vdots	\vdots	\vdots

Einfaches Beispiel: Wählt man als Funktion das kubische Polynom

$$f(x) = 4x^3 + 3x^2 + 2x + 1$$

und die Stützstellen $-2, 0, 1, 2$, ergeben sich die Stützwerte

$$f(-2) = -23, \quad f(0) = 1, \quad f(1) = 10, \quad f(2) = 49.$$

Mittels des Schemas der dividierten Differenzen

x_i	y_i	$\delta y_{i,i+1}$	$\delta^2 y_{i,i+1,i+2}$	$\delta^3 y_{i,i+1,i+2,i+3}$
-2	-23			
0	1	12		
1	10	9	-1	
2	49	39	15	4

ergibt sich als Interpolationspolynom in der **Darstellung nach Newton**

$$p(x) = -23 + 12(x + 2) - (x + 2)x + 4(x + 2)x(x - 1),$$

und eine kleine Rechnung verifiziert $f = p$. Zur Auswertung des Interpolationspolynoms wird ein **modifiziertes Horner-Schema** verwendet. Vgl. **Illustration** (Polynominterpolation, einfaches Beispiel).

Bemerkung: Für theoretische Überlegungen ist es zweckmäßig, den Fall **mehrfacher Nullstellen** durch Einführung eines Inkrements ε und Grenzübergang $\varepsilon \rightarrow 0$ auf

den Fall einfacher Nullstellen zurückzuführen. Beispielsweise betrachtet man bei einer dreifachen Nullstelle die Stützstellen und Stützwerte (Taylorreihenentwicklung)

$$\begin{aligned} & x_i, \quad f(x_i), \\ x_{i+1} &= x_i + \varepsilon, \quad f(x_i + \varepsilon) = f(x_i) + \varepsilon f'(x_i) + \frac{1}{2} \varepsilon^2 f''(x_i) + \mathcal{O}(\varepsilon^3), \\ x_{i+2} &= x_i + 2\varepsilon, \quad f(x_i + 2\varepsilon) = f(x_i) + 2\varepsilon f'(x_i) + 2\varepsilon^2 f''(x_i) + \mathcal{O}(\varepsilon^3), \end{aligned}$$

und dann den Grenzübergang $\varepsilon \rightarrow 0$. Dies ergibt

$$\begin{aligned} \delta y_{i,i+1} &= \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{f(x_i) + \varepsilon f'(x_i) + \frac{1}{2} \varepsilon^2 f''(x_i) + \mathcal{O}(\varepsilon^3) - f(x_i)}{\varepsilon} \\ &= f'(x_i) + \frac{1}{2} \varepsilon f''(x_i) + \mathcal{O}(\varepsilon^2) \xrightarrow{\varepsilon \rightarrow 0} f'(x_i), \\ \delta y_{i+1,i+2} &= \frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}} = \frac{f(x_i) + 2\varepsilon f'(x_i) + 2\varepsilon^2 f''(x_i) - f(x_i) - \varepsilon f'(x_i) - \frac{1}{2} \varepsilon^2 f''(x_i) + \mathcal{O}(\varepsilon^3)}{\varepsilon} \\ &= f'(x_i) + \frac{3}{2} \varepsilon f''(x_i) + \mathcal{O}(\varepsilon^2) \xrightarrow{\varepsilon \rightarrow 0} f'(x_i). \end{aligned}$$

Als Grenzwert der zweiten dividierten Differenz ergibt sich die zweite Ableitung $f''(x_i)$

$$\begin{aligned} \delta^2 y_{i,i+1,i+2} &= \frac{\delta y_{i+1,i+2} - \delta y_{i,i+1}}{x_{i+2} - x_i} = \frac{f'(x_i) + \frac{3}{2} \varepsilon f''(x_i) - f'(x_i) - \frac{1}{2} \varepsilon f''(x_i) + \mathcal{O}(\varepsilon^2)}{2\varepsilon} \\ &= \frac{1}{2} f''(x_i) + \mathcal{O}(\varepsilon) \xrightarrow{\varepsilon \rightarrow 0} \frac{1}{2} f''(x_i). \end{aligned}$$

- **Restglied der Polynominterpolation:** Ziel ist es, eine Aussage über die Güte der Approximation einer hinreichend oft differenzierbaren Funktion $f: [a, b] \rightarrow \mathbb{R}$ durch den Polynominterpolanten $p \in \mathbb{P}_n$ zu treffen, d.h. eine Relation bzw. Abschätzung für den Fehler der Polynominterpolation

$$r(x) = p(x) - f(x), \quad p(x) = \sum_{i=0}^n \delta^i y_{0,\dots,i} \prod_{\ell=0}^{i-1} (x - x_\ell)$$

abzuleiten.

Satz von Rolle (vgl. Analysis): Es sei $f: [a, b] \rightarrow \mathbb{R}$ stetig und auf (a, b) differenzierbar, und es gelte $f(a) = f(b)$. Dann existiert ein Element $c \in (a, b)$ mit $f'(c) = 0$.

Zusammenhang zwischen dividierten Differenzen und Ableitungen (Erweiterter Mittelwertsatz, Satz 1.8): Es sei $f \in \mathcal{C}^n([a, b])$. Dann existiert ein $\xi \in [a, b]$ mit

$$\delta^j y_{i,\dots,i+j} = \frac{1}{j!} f^{(j)}(\xi), \quad 0 \leq i \leq i+j \leq n, \quad 0 \leq j \leq n.$$

Denn: Zum Beweis der Relation

$$\delta^n y_{0,\dots,n} = \frac{1}{n!} f^{(n)}(\xi)$$

wird verwendet, daß die Differenz $r = p - f$ (zumindest) $n + 1$ Nullstellen x_0, \dots, x_n besitzt. Nach dem Satz von Rolle besitzt die erste Ableitung r' (zumindest) n Nullstellen

$\xi_i \in (x_i, x_{i+1})$ für $0 \leq i \leq n-1$. Die wiederholte Anwendung des Satzes von Rolle zusammen mit den Relationen

$$p^{(n)}(\xi) = n! \delta^n y_{0,\dots,n}, \quad \frac{d^n}{dx^n} x^i = 0, \quad 0 \leq i \leq n-1, \quad \frac{d^n}{dx^n} x^n = n!,$$

zeigen, daß die n -te Ableitung (zumindest) eine Nullstelle $\xi \in [a, b]$ besitzt, d.h. es gilt

$$0 = r^{(n)}(\xi) = p^{(n)}(\xi) - f^{(n)}(\xi) = n! \delta^n y_{0,\dots,n} - f^{(n)}(\xi),$$

und damit folgt die Behauptung. \diamond

Fehlerdarstellung der Polynominterpolation (Satz 1.9): Es sei $p \in \mathbb{P}_n$ der Polynominterpolant zu den Datenpunkten $(x_i, f(x_i))_{0 \leq i \leq n}$, und es gelte $f \in \mathcal{C}^{n+1}([x_{\min}, x_{\max}])$ mit $x_{\min} = \min\{x_0, \dots, x_n, \bar{x}\}$ sowie $x_{\max} = \max\{x_0, \dots, x_n, \bar{x}\}$ für ein (fixiertes) Element $\bar{x} \in \mathbb{R}$. Dann existiert ein $\xi \in [x_{\min}, x_{\max}]$ derart, daß das Restglied die folgende Relation erfüllt

$$r(\bar{x}) = p(\bar{x}) - f(\bar{x}) = -\frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^n (\bar{x} - x_i).$$

Denn: Der Polynominterpolant q zu den Datenpunkten $(x_i, f(x_i))_{0 \leq i \leq n+1}$ mit $x_{n+1} = \bar{x}$ ist gegeben durch

$$q(x) = p(x) + \delta^{n+1} y_{0,\dots,n+1} \prod_{i=0}^n (x - x_i).$$

Für $x = \bar{x}$ und mittels des obigen Resultates für die dividierten Differenzen ergibt sich

$$f(\bar{x}) = q(\bar{x}) = p(\bar{x}) + \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^n (\bar{x} - x_i)$$

für ein $\xi \in [x_{\min}, x_{\max}]$ und damit die Behauptung. \diamond

Bemerkungen:

- Das oben angegebene Resultat zeigt, daß die Größe des Restgliedes einerseits durch den funktionsabhängigen Beitrag

$$\frac{1}{(n+1)!} f^{(n+1)}(\xi)$$

und andererseits durch den von den Stützstellen abhängigen Beitrag

$$\prod_{i=0}^n (\bar{x} - x_i)$$

bestimmt ist.

- Als Maß für die Länge einer stetigen Funktion oder auch den Abstand zweier stetiger Funktionen betrachtet man die **Supremumsnorm**

$$\|f\|_{\infty} = \max_{a \leq x \leq b} |f(x)|, \quad \|f - \tilde{f}\|_{\infty} = \max_{a \leq x \leq b} |f(x) - \tilde{f}(x)|, \quad f, \tilde{f} \in \mathcal{C}([a, b]).$$

- Unter der allerdings sehr einschränkenden Voraussetzung, daß **sämtliche Ableitungen** der Funktion $f : [a, b] \rightarrow \mathbb{R}$ **beschränkt** sind, konvergiert das Interpolationspolynom gleichmäßig gegen f . Genauer, falls eine Konstante $C > 0$ existiert, sodaß für alle $j \geq 0$ gilt

$$\|f^{(j)}\|_{\infty} = \max_{x \in [a, b]} |f^{(j)}(x)| \leq M,$$

konvergiert das Interpolationspolynom $p_n \in \mathbb{P}_n$ zu (beliebigen) Stützstellen $(x_{ni})_{0 \leq i \leq n}$ mit $x_{ni} \in [a, b]$ für $0 \leq i \leq n$ bezüglich der Supremumsnorm gegen die Funktion f , d.h. es gilt

$$\|p_n - f\|_{\infty} = \max_{x \in [a, b]} |p_n(x) - f(x)| \xrightarrow{n \rightarrow \infty} 0.$$

Denn: Aus der oben angegebenen Darstellung für das Restglied folgt die Relation

$$|p_n(x) - f(x)| = \frac{1}{(n+1)!} |f^{(n+1)}(\xi)| \prod_{i=0}^n |x - x_{ni}| \leq C \frac{(b-a)^{n+1}}{(n+1)!} \xrightarrow{n \rightarrow \infty} 0, \quad x \in [a, b],$$

und damit die Behauptung. \diamond

- Bei der Interpolation von rationalen Funktionen zu äquidistanten Stützstellen kommt es zum **Phänomen von Runge**. Im Inneren des durch Polstellen von f festgelegten Intervalls konvergiert der Polynominterpolant $p_n \in \mathbb{P}_n$ für $n \rightarrow \infty$ gegen die Funktion f , im Äußeren des Intervalles treten jedoch starke Oszillationen auf und das Interpolationspolynom **divergiert**. Bekanntes Beispiel ist die rationale Funktion

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5].$$

Vgl. **Illustration** (Polynominterpolation, Phänomen von Runge).

- **Vorbemerkung:** Eine Funktion $f : [a, b] \rightarrow \mathbb{R}$ heißt **Lipschitz-stetig**, wenn eine Konstante $L > 0$ existiert, sodaß für alle Elemente $x, \tilde{x} \in [a, b]$ die Abschätzung

$$|f(x) - f(\tilde{x})| \leq L|x - \tilde{x}|$$

gilt. Insbesondere ist jede stetig differenzierbare Funktion $f : [a, b] \rightarrow \mathbb{R}$ Lipschitz-stetig mit Konstante

$$L = \|f'\|_{\infty} = \max_{x \in [a, b]} |f'(x)|.$$

Interpolation basierend auf Chebychev-Knoten: Eine optimale Wahl der Stützstellen sind die Chebychev-Knoten

$$x_i = \frac{1}{2} \left(a + b + (b - a) \cos \frac{(2(i+1)-1)\pi}{2(n+1)} \right), \quad 0 \leq i \leq n,$$

da sie den Beitrag der Stützstellen im Restglied minimieren

$$\max_{x \in [a, b]} \prod_{i=0}^n (x - x_i) \rightarrow \min.$$

In diesem Fall ist bereits für Lipschitz-stetige Funktionen $f : [a, b] \rightarrow \mathbb{R}$ die gleichmäßige Konvergenz des Polynominterpolanten gesichert

$$\|p_n - f\|_\infty = \max_{x \in [a, b]} |p_n(x) - f(x)| \xrightarrow{n \rightarrow \infty} 0.$$

Vgl. **Illustration** (Polynominterpolation, Chebychev-Knoten).

- **Illustration** zur Polynominterpolation mit äquidistanten Stützstellen und Chebychev-Knoten, vgl. Skriptum, S. 15.
- Vgl. Skriptum, S. 16: *In der Praxis — das heißt bei der Berechnung des Interpolanten auf einem Computer mit endlicher Stellenzahl — darf man trotzdem keine Polynome zu hohen Grads (größer 30) verwenden, da diese zu schlecht konditioniert sein können. Der Versuch etwa, im Rungeschen Beispiel ein Interpolationspolynom vom Grad 100 zu den Tschebyscheff-Stützstellen zu berechnen, resultiert in Datenschrott.*

1.4. Kondition der Polynominterpolation

- **Kondition der Polynominterpolation** (Satz 1.10): Die Polynominterpolation ist ein numerisches Problem, das vorgegebenen Stützstellen $(x_i)_{0 \leq i \leq n}$ und Stützwerten $(y_i)_{0 \leq i \leq n}$ mit $y_i = f(x_i)$ für $0 \leq i \leq n$ sowie einem Argument x den Funktionswert des Polynominterpolanten zuordnet

$$P : \mathbb{R}^{n+1} \times \mathbb{R}^{n+1} \times \mathbb{R} \rightarrow \mathbb{R} : (x_0, \dots, x_n, y_0, \dots, y_n, x) \mapsto p(x).$$

Die absoluten Konditionszahlen der Polynominterpolation sind durch

$$\partial_x P = p', \quad \partial_{x_i} P = -p'(x_i) L_i, \quad \partial_{y_i} P = L_i, \quad 0 \leq i \leq n,$$

gegeben.

Denn: Zur Bestimmung der Konditionszahlen $\partial_{y_i} P$ und $\partial_{x_i} P$ wird die Darstellung mittels Lagrange-Basis

$$p(x) = \sum_{i=0}^n y_i L_i(x), \quad L_i(x) = \prod_{\substack{0 \leq \ell \leq n \\ \ell \neq i}} \frac{x - x_\ell}{x_i - x_\ell}, \quad 0 \leq j \leq n,$$

verwendet. \diamond

Bemerkung: Bestimmend für die Kondition der Polynominterpolation ist die Größe L_i . Im ungünstigsten Fall verstärken sich Änderungen in den Stützwerten um den Faktor

$$\kappa = \sum_{i=0}^n |L_i|.$$

Insbesondere bei äquidistanten Stützstellen kann dieser Faktor (vorallem am Rand des betrachteten Intervalls) große Werte annehmen, mit Chebychev-Knoten läßt sich hingegen ein vorteilhafteres Verhalten erzielen.

Illustration (Werte von κ in $[0, 20]$ für äquidistante Stützstellen $x_i = i$ für $0 \leq i \leq 20$ und entsprechende Chebychev-Knoten), vgl. Skriptum, S. 18.

1.5. Vor- und Nachteile der Polynominterpolation

- **Vorteile der Polynominterpolation:**

- Aufgrund der einfachen Handhabung sind Polynome für numerische Berechnungen gut geeignet, insbesondere sind die Berechnung und Auswertung des Polynominterpolanten einfach zu realisieren.
- Polynominterpolanten besitzen gute **lokale Approximationseigenschaften**.

- **Nachteile der Polynominterpolation:**

- Mit $\mathcal{O}(n^2)$ Operationen ist der Rechenaufwand der Polynominterpolation gegenüber $\mathcal{O}(n)$ Operationen bei der Spline-Interpolation erheblich.
- In vielen Fällen ist die Polynominterpolation schlecht konditioniert.
- Im Allgemeinen besitzen Polynominterpolanten schlechte **globale Approximationseigenschaften**.

Die Berechnung eines Polynominterpolanten vom Grad ≥ 15 sowie ein Auswerten des Interpolanten in Randbereichen des betrachteten Intervalles sollten vermieden werden. Im Allgemeinen ist die **Spline-Interpolation** eine bessere Alternative.

2. Polynom-Splines

- Eine vorteilhafte Alternative zur Polynominterpolation, insbesondere in Situationen in denen der Polynominterpolant hohen Grad besitzt oder keine Chebychev-Knoten verwendet werden können, ist die **Interpolation mittels Polynom-Splines**, d.h. mittels stückweisen Polynomfunktionen, die zusätzlichen Stetigkeitsbedingungen genügen.

Illustration (Vergleich Polynominterpolant und kubischer Splineinterpolant), vgl. Skriptum, S. 20.

- **Vorsicht!** Notationen unterscheiden sich teilweise von den im Skriptum verwendeten Notationen.

2.1. Kubische Splineinterpolanten

- **Splinefunktionen** sind stückweise Polynome vom Grad $k \geq 1$ bzw. von der Ordnung $k+1$ auf den jeweiligen Teilintervallen, die global $(k-1)$ -mal stetig differenzierbar sind.

Splinefunktionen (Definition 2.1): Für $m \geq 1$ sei $a = \tau_0 < \dots < \tau_m = b$ eine Zerlegung des Intervalles $[a, b]$ in m Teilintervalle $[\tau_i, \tau_{i+1}]$ für $0 \leq i \leq m-1$. Der **Raum der Splinefunktionen** vom Grad $k \geq 0$ (bzw. von der Ordnung $k+1$) zu den $m+1$ **Knoten** $\tau = \{\tau_0, \dots, \tau_m\}$ ist gegeben durch

$$\mathbb{P}_{0,\tau} = \{s: [a, b] \rightarrow \mathbb{R} : s|_{[\tau_i, \tau_{i+1}]} \in \mathbb{P}_0 \text{ für } 0 \leq i \leq m-1\},$$

$$\mathbb{P}_{k,\tau} = \{s: [a, b] \rightarrow \mathbb{R} \in \mathcal{C}^{k-1}([a, b]) : s|_{[\tau_i, \tau_{i+1}]} \in \mathbb{P}_k \text{ für } 0 \leq i \leq m-1\}, \quad k \geq 1.$$

Bemerkungen:

- Die Summe zweier Splinefunktionen $s, \tilde{s} \in \mathbb{P}_{k,\tau}$ und das skalare Vielfache einer Splinefunktion $s \in \mathbb{P}_{k,\tau}$ liegen in $\mathbb{P}_{k,\tau}$, d.h. der Raum der Splinefunktionen $\mathbb{P}_{k,\tau}$ bildet einen reellen Vektorraum. Geeignete Basisfunktionen sind B-Splines, vgl. Abschnitt 2.2.
- Die stärkere Forderung $s \in \mathbb{P}_{k,\tau} \cap \mathcal{C}^k([a, b])$ würde implizieren, daß die Splinefunktion s global mit einer Polynomfunktion übereinstimmt.

Denn: Für zwei Polynome vom Grad $k \geq 0$

$$p(x) = \sum_{i=0}^k p_i (x-a)^i, \quad q(x) = \sum_{i=0}^k q_i (x-a)^i,$$

deren Funktionswert und deren j -te Ableitung für $1 \leq j \leq k$ in einem Punkt übereinstimmen, folgt die Gleichheit der Koeffizienten (o.E.d.A. wähle $x = a$, ändere ansonsten die Darstellung der Polynomfunktionen)

$$j! p_j = p^{(j)}(a) = q^{(j)}(a) = j! q_j, \quad 0 \leq j \leq k,$$

und damit die Gleichheit der Funktionen. \diamond

- **Splinefunktionen vom Grad 0 bzw. von der Ordnung 1:** Der Spliner Raum

$$\mathbb{P}_{0,\tau} = \{s: [a, b] \rightarrow \mathbb{R} : s|_{[\tau_i, \tau_{i+1}]} \in \mathbb{P}_0 \text{ für } 0 \leq i \leq m-1\}$$

umfaßt stückweise konstante Funktionen

$$s|_{[\tau_i, \tau_{i+1}]}(x) = \alpha_i, \quad 0 \leq i \leq m-1,$$

die bei insgesamt $m+1$ Knoten durch m vorgeschriebene Stützwerte beispielsweise an den Knoten $\tau_0, \dots, \tau_{m-1}$ bestimmt sind. Im Allgemeinen sind die Splinefunktionen unstetig, mit Sprungstellen an den Knoten; der Funktionswert bei $\tau_m = b$ wird nicht miteinbezogen.

Illustration (Splinefunktion vom Grad 0), vgl. Skriptum, S. 22.

- **Splinefunktionen vom Grad 1 bzw. von der Ordnung 2:** Der Splineraum

$$\mathbb{P}_{1,\tau} = \{s: [a, b] \rightarrow \mathbb{R} \in \mathcal{C}([a, b]) : s|_{[\tau_i, \tau_{i+1}]} \in \mathbb{P}_1 \text{ für } 0 \leq i \leq m-1\}$$

umfaßt stückweise lineare Polynome

$$s|_{[\tau_i, \tau_{i+1}]}(x) = \alpha_i + \beta_i(x - \tau_i), \quad 0 \leq i \leq m-1,$$

die bei insgesamt $m+1$ Knoten durch $m+1$ vorgeschriebene Stützwerte beispielsweise an den Knoten τ_0, \dots, τ_m bestimmt sind. Die Funktionen sind auf dem Intervall $[a, b]$ stetig, im Allgemeinen jedoch an den Knoten nicht differenzierbar.

Illustration (Splinefunktion vom Grad 1), vgl. Skriptum, S. 22.

- **Splinefunktionen vom Grad 2 bzw. von der Ordnung 3:** Der Splineraum

$$\mathbb{P}_{2,\tau} = \{s: [a, b] \rightarrow \mathbb{R} \in \mathcal{C}^1([a, b]) : s|_{[\tau_i, \tau_{i+1}]} \in \mathbb{P}_2 \text{ für } 0 \leq i \leq m-1\}$$

umfaßt stückweise quadratische Polynome

$$s_i(x) = s|_{[\tau_i, \tau_{i+1}]}(x) = \alpha_i + \beta_i(x - \tau_i) + \gamma_i(x - \tau_i)(x - \tau_{i+1}), \quad 0 \leq i \leq m-1,$$

die auf dem Intervall $[a, b]$ stetig differenzierbar sind, d.h. an den inneren Knoten die Bedingungen

$$s_{i-1}(\tau_i) = s_i(\tau_i), \quad s'_{i-1}(\tau_i) = s'_i(\tau_i), \quad 1 \leq i \leq m-1,$$

erfüllen.

Bemerkung: Aufgrund der unerwünschten Eigenschaft, daß sich Änderungen in einem einzelnen Teilintervall (ungedämpft) auf die gesamte Splinefunktion auswirken, werden quadratische Splines selten verwendet. Dies zeigt sich beispielsweise an der Splinefunktion mit Funktionswert Null in sämtlichen äquidistant verteilten Knoten (mit $\tau_i = ih$ für $0 \leq i \leq m=3$). Einsetzen der Darstellungen (zusätzliche Skalierungen der Koeffizienten vorteilhaft)

$$\begin{aligned} s_0(x) &= \alpha_0 + \frac{\beta_0}{h}x + \frac{\gamma_0}{h}x(x-h), & x \in [0, h], \\ s_1(x) &= \alpha_1 + \frac{\beta_1}{h}(x-h) + \frac{\gamma_1}{h}(x-h)(x-2h), & x \in [h, 2h], \\ s_2(x) &= \alpha_2 + \frac{\beta_2}{h}(x-2h) + \frac{\gamma_2}{h}(x-2h)(x-3h), & x \in [2h, 3h], \end{aligned}$$

in die Bedingungen an die Funktionswerte

$$\begin{aligned} \alpha_0 = s_0(0) = 0, \quad \beta_0 = s_0(h) = 0 = s_1(h) = \alpha_1, \\ \beta_1 = s_1(2h) = 0 = s_2(2h) = \alpha_2, \quad \beta_2 = s_2(3h) = 0, \end{aligned}$$

führt auf die Relationen

$$\begin{aligned} s_0(x) &= \frac{\gamma_0}{h}x(x-h), & x \in [0, h], \\ s_1(x) &= \frac{\gamma_1}{h}(x-h)(x-2h), & x \in [h, 2h], \\ s_2(x) &= \frac{\gamma_2}{h}(x-2h)(x-3h), & x \in [2h, 3h]. \end{aligned}$$

Die Stetigkeitsbedingungen an die ersten Ableitungen

$$\begin{aligned} s'_0(x) &= \frac{\gamma_0}{h} (2x - h), & x \in [0, h], \\ s'_1(x) &= \frac{\gamma_1}{h} (2x - 3h), & x \in [h, 2h], \\ s'_2(x) &= \frac{\gamma_2}{h} (2x - 5h), & x \in [2h, 3h], \\ \gamma_0 = s'_0(h) = s'_1(h) &= -\gamma_1, & \gamma_1 = s'_1(2h) = s'_2(2h) = -\gamma_2, \end{aligned}$$

ergeben weiters

$$\begin{aligned} s_0(x) &= \frac{\gamma_0}{h} x(x - h), & x \in [0, h], \\ s_1(x) &= -\frac{\gamma_0}{h} (x - h)(x - 2h), & x \in [h, 2h], \\ s_2(x) &= \frac{\gamma_0}{h} (x - 2h)(x - 3h), & x \in [2h, 3h]. \end{aligned}$$

Die Änderung des Koeffizienten γ_0 bei der Funktion im ersten Teilintervall bewirkt somit eine Änderung der Funktionen auf allen anderen Teilintervallen.

- **Splinefunktionen vom Grad 3 bzw. von der Ordnung 4:** Der Splineraum

$$\mathbb{P}_{3,\tau} = \{s : [a, b] \rightarrow \mathbb{R} \in \mathcal{C}^2([a, b]) : s|_{[\tau_i, \tau_{i+1}]} \in \mathbb{P}_3 \text{ für } 0 \leq i \leq m-1\}$$

umfaßt stückweise kubische Polynome, die auf dem Intervall $[a, b]$ zweimal stetig differenzierbar sind.

- Ziel ist es, die Funktionen

$$s_i(x) = s|_{[\tau_i, \tau_{i+1}]}(x), \quad 0 \leq i \leq m-1,$$

beispielsweise bei vorgegebenen Funktionswerten in den Knoten und vorgegebenen Werten für die erste Ableitung in den Randpunkten

$$s_i(\tau_i) = y_i, \quad 0 \leq i \leq m, \quad s'_0(\tau_0) = y'_0, \quad s'_{m-1}(\tau_m) = y'_m,$$

zu bestimmen. Dazu verwendet man zunächst, daß das kubische Polynom s_i durch die Funktionswerte und die (noch unbekannt)en Werte der ersten Ableitung an den Knoten τ_i und τ_{i+1} für $0 \leq i \leq m-1$ eindeutig festgelegt ist

$$s_i(\tau_i) = y_i, \quad s_i(\tau_{i+1}) = y_{i+1}, \quad s'_i(\tau_i) = y'_i, \quad s'_i(\tau_{i+1}) = y'_{i+1}, \quad 0 \leq i \leq m-1.$$

Damit sind die Stetigkeitsbedingungen an den inneren Knoten für die Splinefunktion und deren erste Ableitung erfüllt. Die Darstellung (wobei $h_i = \tau_{i+1} - \tau_i$ für $0 \leq i \leq m-1$)

$$\begin{aligned} s_i(x) &= y_i \left(1 + 2 \frac{x - \tau_i}{h_i}\right) \left(\frac{\tau_{i+1} - x}{h_i}\right)^2 + y_{i+1} \left(3 - 2 \frac{x - \tau_i}{h_i}\right) \left(\frac{x - \tau_i}{h_i}\right)^2 \\ &\quad + y'_i \left(\frac{\tau_{i+1} - x}{h_i}\right)^2 (x - \tau_i) - y'_{i+1} \frac{\tau_{i+1} - x}{h_i} \frac{x - \tau_i}{h_i} (x - \tau_i), \quad 0 \leq i \leq m-1, \end{aligned}$$

verifiziert man leicht durch Einsetzen

$$\begin{aligned} z(x) &= \frac{x-\tau_i}{h_i}, & z(\tau_i) &= 0, & z(\tau_{i+1}) &= 1, \\ \tilde{z}(x) &= \frac{\tau_{i+1}-x}{h_i}, & \tilde{z}(\tau_i) &= 1, & \tilde{z}(\tau_{i+1}) &= 0, \\ s_i &= y_i (\tilde{z}^2 + 2z\tilde{z}^2) + y_{i+1} (3z^2 - 2z^3) + y'_i h_i z\tilde{z}^2 - y'_{i+1} h_i z^2\tilde{z}, \\ s_i(\tau_i) &= y_i, & s_i(\tau_{i+1}) &= y_{i+1}, & 0 \leq i \leq m-1, \end{aligned}$$

bzw. Differenzieren und Einsetzen (wegen $z' = \frac{1}{h_i} = -\tilde{z}'$)

$$\begin{aligned} s'_i &= \frac{2}{h_i} y_i (\tilde{z}^2 - \tilde{z} - 2z\tilde{z}) + \frac{6}{h_i} y_{i+1} (z - z^2) + y'_i (\tilde{z}^2 - 2z\tilde{z}) + y'_{i+1} (z^2 - 2z\tilde{z}), \\ s'_i(\tau_i) &= y'_i, & s'_i(\tau_{i+1}) &= y'_{i+1}, & 0 \leq i \leq m-1. \end{aligned}$$

– Aus der obigen Darstellung erhält man folgende Relationen für die Werte der zweiten Ableitung an den inneren Knoten

$$\begin{aligned} s''_i &= \frac{2}{h_i^2} y_i (1 + 2z - 4\tilde{z}) + \frac{6}{h_i^2} y_{i+1} (1 - 2z) + \frac{2}{h_i} y'_i (z - 2\tilde{z}) + \frac{2}{h_i} y'_{i+1} (2z - \tilde{z}), \\ s''_i(\tau_i) &= \frac{6}{h_i^2} (y_{i+1} - y_i) - \frac{2}{h_i} (2y'_i + y'_{i+1}), & 1 \leq i \leq m-1, \\ s''_i(\tau_{i+1}) &= \frac{6}{h_i^2} (y_i - y_{i+1}) + \frac{2}{h_i} (y'_i + 2y'_{i+1}), & 0 \leq i \leq m-2. \end{aligned}$$

Die Stetigkeitsbedingungen an den inneren Knoten für die zweite Ableitung der Splinefunktion ergeben

$$\begin{aligned} s''_{i-1}(\tau_i) &= s''_i(\tau_i), & 1 \leq i \leq m-1, \\ \frac{3}{h_{i-1}^2} (y_{i-1} - y_i) + \frac{1}{h_{i-1}} (y'_{i-1} + 2y'_i) &= \frac{3}{h_i^2} (y_{i+1} - y_i) - \frac{1}{h_i} (2y'_i + y'_{i+1}), & 1 \leq i \leq m-1, \end{aligned}$$

und weiters

$$\frac{1}{h_{i-1}} y'_{i-1} + 2\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) y'_i + \frac{1}{h_i} y'_{i+1} = \frac{3}{h_{i-1}^2} (y_i - y_{i-1}) + \frac{3}{h_i^2} (y_{i+1} - y_i), \quad 1 \leq i \leq m-1.$$

Aus diesen Relationen lassen sich bei vorgegebenen Funktionswerten an den Knoten und vorgegebenen Werten für die erste Ableitung in den Intervallenden die zu bestimmenden Werte der ersten Ableitung in den inneren Knoten berechnen.

Berechnung eingespannter kubischer Splinefunktionen (Satz 2.2, Satz 2.3): Eine kubische Splinefunktion $s \in \mathbb{P}_{3,\tau}$ ist durch die Darstellung

$$\begin{aligned} s_i(x) = s|_{[\tau_i, \tau_{i+1})}(x) &= y_i \left(1 + 2 \frac{x-\tau_i}{h_i}\right) \left(\frac{\tau_{i+1}-x}{h_i}\right)^2 + y_{i+1} \left(3 - 2 \frac{x-\tau_i}{h_i}\right) \left(\frac{x-\tau_i}{h_i}\right)^2 \\ &+ y'_i \left(\frac{\tau_{i+1}-x}{h_i}\right)^2 (x - \tau_i) - y'_{i+1} \frac{\tau_{i+1}-x}{h_i} \frac{x-\tau_i}{h_i} (x - \tau_i), & 0 \leq i \leq m-1, \end{aligned}$$

gegeben. Bei Vorgabe von $m+1$ zu interpolierenden Funktionswerten

$$s_i(\tau_i) = y_i = f(\tau_i), \quad 0 \leq i \leq m,$$

lauten die $m - 1$ Bedingungen an die $m + 1$ Ableitungswerte an den Knoten

$$s'_i(\tau_i) = y'_i, \quad 0 \leq i \leq m,$$

$$\frac{1}{h_{i-1}} y'_{i-1} + 2\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) y'_i + \frac{1}{h_i} y'_{i+1} = \frac{3}{h_{i-1}^2} (y_i - y_{i-1}) + \frac{3}{h_i^2} (y_{i+1} - y_i), \quad 1 \leq i \leq m - 1.$$

Beispielsweise bei der Festlegung der Werte der ersten Ableitung an den Intervallenden

$$s'_0(\tau_0) = y'_0 = f'(a), \quad s'_{m-1}(\tau_m) = y'_m = f'(b),$$

ist der Vektor $y' = (y'_i)_{1 \leq i \leq m-1}$ als eindeutige Lösung eines linearen Gleichungssystems gegeben und damit die zugehörige kubische Splinefunktion (**eingespannter Spline**) eindeutig bestimmt.

Denn: Bei Vorgabe der Funktionswerte $y_i = f(\tau_i)$ für $0 \leq i \leq m$ und der Ableitungen $y'_0 = f'(a)$ und $y'_m = f'(b)$ sind die restlichen Ableitungen y'_1, \dots, y'_{m-1} als Lösungen der linearen Gleichungen

$$i = 1: \quad 2\left(\frac{1}{h_0} + \frac{1}{h_1}\right) y'_1 + \frac{1}{h_1} y'_2 = \frac{3}{h_0^2} (y_1 - y_0) + \frac{3}{h_1^2} (y_2 - y_1) - \frac{1}{h_0} y'_0,$$

$$2 \leq i \leq m - 2: \quad \frac{1}{h_{i-1}} y'_{i-1} + 2\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) y'_i + \frac{1}{h_i} y'_{i+1} = \frac{3}{h_{i-1}^2} (y_i - y_{i-1}) + \frac{3}{h_i^2} (y_{i+1} - y_i),$$

$$i = m - 1: \quad \frac{1}{h_{m-2}} y'_{m-2} + 2\left(\frac{1}{h_{m-2}} + \frac{1}{h_{m-1}}\right) y'_{m-1} = \frac{3}{h_{m-2}^2} (y_{m-1} - y_{m-2}) + \frac{3}{h_{m-1}^2} (y_m - y_{m-1}) - \frac{1}{h_{m-1}} y'_m,$$

gegeben. In kompakter Form als lineares Gleichungssystem mit symmetrischer Triagonalmatrix ergibt sich

$$Ay' = r, \quad A = (a_{ij})_{1 \leq i, j \leq m-1} \in \mathbb{R}^{(m-1) \times (m-1)}, \quad y', r \in \mathbb{R}^{m-1},$$

$$a_{ii} = 2\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right), \quad 1 \leq i \leq m - 1, \quad a_{i, i+1} = a_{i+1, i} = \frac{1}{h_i}, \quad 1 \leq i \leq m - 2,$$

$$y = \begin{pmatrix} y'_1 \\ \vdots \\ y'_{m-1} \end{pmatrix}, \quad r = \begin{pmatrix} \frac{3}{h_0^2} (y_1 - y_0) + \frac{3}{h_1^2} (y_2 - y_1) \\ \vdots \\ \frac{3}{h_{i-1}^2} (y_i - y_{i-1}) + \frac{3}{h_i^2} (y_{i+1} - y_i) \\ \vdots \\ \frac{3}{h_{m-2}^2} (y_{m-1} - y_{m-2}) + \frac{3}{h_{m-1}^2} (y_m - y_{m-1}) \end{pmatrix} - \begin{pmatrix} \frac{1}{h_0} y'_0 \\ 0 \\ \vdots \\ 0 \\ \frac{1}{h_{m-1}} y'_m \end{pmatrix}.$$

Da die Matrix A strikt diagonaldominant ist, ist das lineare Gleichungssystem eindeutig lösbar und damit die zugehörige kubische Splinefunktion eindeutig bestimmt. \diamond

Bemerkungen:

- Eine Matrix $A \in \mathbb{R}^{n \times n}$ heißt **strikt diagonaldominant**, wenn folgende Relation gilt

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{für alle } 1 \leq i \leq n.$$

Wegen (mit betragsmäßig größtem Koeffizienten x_ℓ)

$$\begin{aligned}
 Ax = 0 &\implies \sum_{j=1}^n a_{ij} x_j = 0 \quad \text{für } 1 \leq i \leq n \\
 &\implies a_{\ell\ell} x_\ell = - \sum_{\substack{j=1 \\ j \neq \ell}}^n a_{\ell j} x_j \quad \text{mit } |x_\ell| = \max_{1 \leq i \leq n} |x_i| \\
 &\implies |a_{\ell\ell}| |x_\ell| \leq \sum_{\substack{j=1 \\ j \neq \ell}}^n |a_{\ell j}| |x_j| \leq |x_\ell| \sum_{\substack{j=1 \\ j \neq \ell}}^n |a_{\ell j}| \\
 &\implies \underbrace{\left(|a_{\ell\ell}| - \sum_{\substack{j=1 \\ j \neq \ell}}^n |a_{\ell j}| \right)}_{>0} |x_\ell| \leq 0 \\
 &\implies x_\ell = 0 \\
 &\implies x = 0
 \end{aligned}$$

ist jede strikt diagonaldominante Matrix invertierbar.

- Beinhaltet ein lineares Gleichungssystem eine symmetrische und positiv definite Matrix $A \in \mathbb{R}^{n \times n}$, so ist das Gaußsche Eliminationsverfahren ohne Spaltenpivot-suche numerisch stabil durchführbar. Ist die Matrix zudem eine Tridiagonalmatrix, so werden zur Elimination der Subdiagonale $\mathcal{O}(n)$ Operationen benötigt.
- Vgl. **Illustration** (Eingespannter Spline, Matrix) für den Spezialfall äquidistanter Knoten ($m = 6$). Für die zugehörige Matrix

$$\begin{aligned}
 Ay' = r, \quad A = (a_{ij})_{1 \leq i, j \leq 5} \in \mathbb{R}^{(m-1) \times (m-1)}, \\
 a_{ii} = \frac{4}{h}, \quad 1 \leq i \leq m-1, \quad a_{i,i+1} = a_{i+1,i} = \frac{1}{h}, \quad 1 \leq i \leq m-2.
 \end{aligned}$$

Das Gaußsche Eliminationsverfahren (bzw. die Cholesky-Zerlegung) führt auf die Zerlegung $A = \tilde{L}D\tilde{L}^T$ mit positiv definiter Diagonalmatrix D (d.h. alle Diagonaleinträge sind positiv) und auf einfache Weise rekursiv berechenbarer Matrix \tilde{L} (Diagonalelemente von \tilde{L} gleich Eins, erste Subdiagonale zu berechnen). Wegen $x^T Ax = x^T \tilde{L}D\tilde{L}^T x = y^T Dy > 0$ mit $y = \tilde{L}^T x$ folgt die positive Definitheit von A . Ähnliche Überlegungen gelten für den allgemeinen Fall.

Fehler der kubischen Splineinterpolation (Satz 2.4): Für $f \in \mathcal{C}^4([a, b])$ sei $s : [a, b] \rightarrow \mathbb{R}$ die eindeutig bestimmte kubische Splinefunktion zu den Funktionswerten

$$s_i(\tau_i) = y_i = f(\tau_i), \quad 0 \leq i \leq m, \quad s'(a) = f'(a), \quad s'(b) = f'(b).$$

Dann gelten folgende Abschätzungen für den Approximationsfehler des Splineinterpolanten (zu $m+1$ Knoten $a = \tau_0 < \dots < \tau_m = b$ mit $h_i = \tau_{i+1} - \tau_i$ für $0 \leq i \leq m-1$ und

$$h_{\max} = \max\{h_i : 0 \leq i \leq m-1\}, \|f\|_{\infty} = \max\{f(x) : a \leq x \leq b\}$$

$$\begin{aligned} |(f-s)(x)| &\leq \frac{3}{64} h_i^2 h_{\max}^2 \|f^{(4)}\|_{\infty}, & x \in [\tau_i, \tau_{i+1}], & 0 \leq i \leq m-1, \\ |(f-s)'(x)| &\leq \frac{3}{16} h_i h_{\max}^2 \|f^{(4)}\|_{\infty}, & x \in [\tau_i, \tau_{i+1}], & 0 \leq i \leq m-1, \\ |(f-s)''(x)| &\leq \frac{3}{8} h_{\max}^2 \max_{x \in [a,b]} \|f^{(4)}\|_{\infty}, & x \in [\tau_i, \tau_{i+1}], & 0 \leq i \leq m-1, \\ |(f-s)'''(x)| &\leq \frac{1}{2} h_i \left(1 + \left(\frac{h_{\max}}{h_i}\right)^2\right) \|f^{(4)}\|_{\infty}, & x \in [\tau_i, \tau_{i+1}], & 0 \leq i \leq m-1. \end{aligned}$$

Bemerkungen:

- Die Approximationseigenschaften von kubischen Splinefunktionen für reguläre Funktionen sind insbesondere bei äquidistanten Stützstellen besser als die Eigenschaften von Polynominterpolanten.
 - Werden die benötigten Werte der Ableitung an den Intervallenden näherungsweise mittels kubischer Polynominterpolation an vier Knoten bestimmt, bleibt der Approximationsfehler von der Größenordnung h_{\max}^4 .
 - Neben eingespannten Splinefunktionen mit vorgegebenen Ableitungen an den Intervallenden sind beispielsweise auch Splinefunktionen zu **natürlichen Randbedingungen** $s_0''(a) = 0 = s_{m-1}''(b)$ gebräuchlich, allerdings besitzen diese weniger gute Approximationseigenschaften.
 - Splinefunktionen kommen ursprünglich aus dem Schiffsbau und sind wesentlich für Anwendungen aus den Bereichen **Computergraphik** und **Design** (u.a. Computeranimationen, Design von Autos und Flugzeugen).
- Vgl. **Illustration** (Splineinterpolation) sowie **Illustration** (Approximationsfehler).

Kurveninterpolation

- Die bisher behandelten Ideen lassen sich auch zur Interpolation von Kurven anwenden. Beispielsweise für eine ebene Kurve beruht die Interpolation mittels kubischen Splinefunktionen auf folgendem Zugang:

- Eine ebene Kurve durch $m + 1$ Punkte der Ebene $v_i = (x_i, y_i)_{0 \leq i \leq m}$ ist durch eine **Parametrisierung** gegeben

$$\gamma : [a, b] \rightarrow \mathbb{R}^2 : t \mapsto \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}.$$

Als Knoten $a = \tau_0 < \dots < \tau_m = b$ wählt man beispielsweise die Zeitpunkte (Zeitinkremente entsprechen der Länge der verbindenden Polygonzüge $\|v_i - v_{i-1}\|_2$)

$$\begin{aligned} \tau_0 &= a, & \tau_i &= \tau_{i-1} + \|v_i - v_{i-1}\|_2, & 1 \leq i \leq m, \\ \|v_i - v_{i-1}\|_2 &= \|(x_i - x_{i-1}, y_i - y_{i-1})^T\|_2 = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \end{aligned}$$

und setzt dann $b = \tau_m$.

Vgl. **Illustration** (Punkte der Ebene, zugehöriges Stützpolygon), Skriptum, S. 27.

- Man bestimmt einerseits die interpolierende Splinefunktion $s_x : [a, b] \rightarrow \mathbb{R}$ zu den Datenpunkten $(\tau_i, x_i)_{0 \leq i \leq m}$ und andererseits die interpolierende Splinefunktion $s_y : [a, b] \rightarrow \mathbb{R}$ zu den Datenpunkten $(\tau_i, y_i)_{0 \leq i \leq m}$. Die interpolierende Kurve ist dann durch

$$s : [a, b] \rightarrow \mathbb{R}^2 : t \mapsto \begin{pmatrix} s_x(t) \\ s_y(t) \end{pmatrix}$$

gegeben.

Vgl. **Illustration** (Tragflächenprofil), Skriptum, S. 28.

- Vgl. **Illustration** (Kurveninterpolation).

2.2. B-Splines

- **Ziel:** Bestimmung einer geeigneten Basis des Vektorraumes der Splinefunktionen

$$\mathbb{P}_{0,\tau} = \{s: [a, b] \rightarrow \mathbb{R} : s|_{[\tau_i, \tau_{i+1})} \in \mathbb{P}_0 \text{ für } 0 \leq i \leq m-1\},$$

$$\mathbb{P}_{k,\tau} = \{s: [a, b] \rightarrow \mathbb{R} \in \mathcal{C}^{k-1}([a, b]) : s|_{[\tau_i, \tau_{i+1})} \in \mathbb{P}_k \text{ für } 0 \leq i \leq m-1\}, \quad k \geq 1.$$

- **Dimension des Raumes der Splinefunktionen und Basisfunktionen** (Lemma 2.5, Definition 2.6, Satz 2.7): Bei $m+1$ Knotenpunkten $(\tau_i)_{0 \leq i \leq m}$ hat der Vektorraum $\mathbb{P}_{k,\tau}$ für $k \geq 0$ die Dimension $m+k$, d.h. es gilt

$$\dim \mathbb{P}_{k,\tau} = m+k, \quad k \geq 0.$$

Die rekursiv definierten **B-Splines** $(N_{ki})_{-k \leq i \leq m-1}$ (mit Einführung zusätzlicher Hilfsknoten $\tau_{-k} < \dots < \tau_{-1} < \tau_0$ und $\tau_m < \tau_{m+1} < \dots < \tau_{m+k}$)

$$k=0: \quad N_{0i}(x) = \begin{cases} 1 & x \in [\tau_i, \tau_{i+1}), \\ 0 & x \notin [\tau_i, \tau_{i+1}), \end{cases} \quad 0 \leq i \leq m-1,$$

$$k \geq 1: \quad N_{ki}(x) = \frac{1}{\tau_{i+k} - \tau_i} (x - \tau_i) N_{k-1,i}(x) + \frac{1}{\tau_{i+k+1} - \tau_{i+1}} (\tau_{i+k+1} - x) N_{k-1,i+1}(x), \quad -k \leq i \leq m-1,$$

bilden eine Basis von $\mathbb{P}_{k,\tau}$. Jede Splinefunktion $s \in \mathbb{P}_{k,\tau}$ ist somit in eindeutiger Weise als Linearkombination der Basisfunktionen

$$s = \sum_{i=-(k-1)}^m c_i N_{ki}$$

darstellbar. **Denn:** Vgl. auch Überlegungen in Abschnitt 2.1.

- $k=0$: Eine stückweise konstante Splinefunktion in $\mathbb{P}_{0,\tau}$ ist durch m Stützwerte an den Knoten $\tau_0, \dots, \tau_{m-1}$ bestimmt, und folglich ist

$$\dim \mathbb{P}_{0,\tau} = m.$$

Offenbar bilden die m Splinefunktionen $(N_{0i})_{0 \leq i \leq m-1}$, definiert durch

$$N_{0i}(x) = \begin{cases} 1, & x \in [\tau_i, \tau_{i+1}), \\ 0, & x \notin [\tau_i, \tau_{i+1}), \end{cases} \quad 0 \leq i \leq m-1,$$

eine Basis von $\mathbb{P}_{0,\tau}$. Da die Splinefunktion N_{0i} am Knoten τ_i den Wert 1 und an den restlichen Knoten den Wert 0 annimmt

$$N_{0i}(\tau_j) = \delta_{ij}, \quad 0 \leq i \leq m-1, \quad 0 \leq j \leq m,$$

ist jede Splinefunktion $s \in \mathbb{P}_{0,\tau}$ in eindeutiger Weise als Linearkombination

$$s = \sum_{i=0}^{m-1} c_i N_{0i}, \quad c_i = s(\tau_i), \quad 0 \leq i \leq m-1,$$

darstellbar. Vgl. Abbildung, Skriptum, S. 30.

- $k = 1$: Eine stückweise lineare Splinefunktion $s \in \mathbb{P}_{1,\tau}$ ist auf dem ersten Teilintervall $[\tau_0, \tau_1]$ durch $k + 1 = 2$ Bedingungen festgelegt, auf jedem weiteren Teilintervall $[\tau_i, \tau_{i+1}]$ für $1 \leq i \leq m - 1$ kommt aufgrund der Stetigkeitsbedingung an s nur jeweils eine Bedingung hinzu. Dies führt auf insgesamt $2 + m - 1 = m + 1$ Bedingungen, und somit ist

$$\dim \mathbb{P}_{1,\tau} = m + 1.$$

Geeignete Basisfunktionen sind die $m + 1$ stückweise linearen Funktionen $(N_{1i})_{0 \leq i \leq m}$, definiert durch (die Einführung von zwei **Hilfsknoten** $\tau_{-1} < \tau_0$ und $\tau_{m+1} > \tau_m$, erlaubt die gleichzeitige Behandlung der inneren Teilintervalle $[\tau_i, \tau_{i+1}]$ für $1 \leq i \leq m - 2$ und der Randintervalle $[\tau_0, \tau_1]$, $[\tau_{m-1}, \tau_m]$; auf $[\tau_i, \tau_{i+1}]$ führt der Ansatz $N_{1i}(x) = \alpha + \beta(x - \tau_i)$ und die Forderungen $N_{1i}(\tau_i) = 0$ und $N_{1i}(\tau_{i+1}) = 1$ auf die angegebene Darstellung, auf dem Teilintervall $[\tau_{i+1}, \tau_{i+2}]$ führt der analoger Ansatz $N_{1i}(x) = \alpha + \beta(x - \tau_{i+2})$ und die Forderungen $N_{1i}(\tau_{i+1}) = 1$ und $N_{1i}(\tau_{i+2}) = 0$ auf die angegebene Darstellung)

$$N_{1i}(x) = \begin{cases} \frac{1}{\tau_{i+1} - \tau_i} (x - \tau_i), & x \in [\tau_i, \tau_{i+1}], \\ \frac{1}{\tau_{i+2} - \tau_{i+1}} (\tau_{i+2} - x), & x \in [\tau_{i+1}, \tau_{i+2}], \\ 0, & \text{sonst,} \end{cases} \quad -1 \leq i \leq m - 1,$$

d.h. N_{1i} nimmt am Knotenpunkt τ_{i+1} den Wert 1 und an den restlichen Knotenpunkten (einschließlich der Hilfsknoten) den Wert 0 an

$$N_{1i}(\tau_j) = \delta_{i+1,j}, \quad -1 \leq i \leq m - 1, \quad -1 \leq j \leq m + 1.$$

Mittels der Basisfunktionen $(N_{0i})_{0 \leq i \leq m}$ ergibt sich die Darstellung

$$N_{1i}(x) = \frac{1}{\tau_{i+1} - \tau_i} (x - \tau_i) N_{0i}(x) + \frac{1}{\tau_{i+2} - \tau_{i+1}} (\tau_{i+2} - x) N_{0,i+1}(x), \quad -1 \leq i \leq m - 1.$$

Offensichtlich ist die Basisfunktion N_{1i} stetig, auf (τ_i, τ_{i+2}) positiv, und es gilt $N_{1i}(x) = 0$ für $x \notin (\tau_i, \tau_{i+2})$, d.h. die Funktion besitzt einen **lokalen Träger**

$$\text{supp } N_{1i} = \overline{\{x \in \mathbb{R} : N_{1i}(x) \neq 0\}} = [\tau_i, \tau_{i+2}].$$

Ähnlich wie zuvor folgt für $s \in \mathbb{P}_{1,\tau}$ die Darstellung

$$s = \sum_{i=-1}^{m-1} c_i N_{1i}, \quad c_i = s(\tau_{i+1}), \quad -1 \leq i \leq m - 1.$$

Vgl. Abbildung, Skriptum, S. 31.

- $k \geq 2$: Ähnliche Überlegungen gelten für Splinefunktionen vom Grad k . Auf dem ersten Teilintervall $[\tau_0, \tau_1]$ ist $s \in \mathbb{P}_{k,\tau}$ durch $k + 1$ Bedingungen festgelegt, und aufgrund der geforderten Stetigkeitsbedingungen an $s^{(j)}(\tau_i)$ für $0 \leq j \leq k - 1$ kommt

auf jedem weiteren Teilintervall $[\tau_i, \tau_{i+1}]$ für $1 \leq i \leq m-1$ nur jeweils eine Bedingung hinzu, was auf insgesamt $k+1+m-1 = m+k$ Bedingungen führt

$$\dim \mathbb{P}_{k,\tau} = m+k.$$

Die Konstruktion geeigneter Basisfunktionen, der **B-Splines**, basiert auf der Rekursion (Einführung weiterer Hilfsknoten, Relation konsistent mit obigen Überlegungen für den Fall $k=1$)

$$N_{ki}(x) = \frac{1}{\tau_{i+k}-\tau_i} (x-\tau_i) N_{k-1,i}(x) + \frac{1}{\tau_{i+k+1}-\tau_{i+1}} (\tau_{i+k+1}-x) N_{k-1,i+1}(x), \quad -k \leq i \leq m-1.$$

Eine wesentliche Eigenschaft der B-Splines ist die **Lokalität** (Trägerintervall $[\tau_i, \tau_{i+1}]$ für N_{0i} , $[\tau_i, \tau_{i+2}]$ für N_{1i} , $[\tau_i, \tau_{i+3}]$ für N_{2i} etc.)

$$\text{supp } N_{ki} = \overline{\{x \in \mathbb{R} : N_{ki}(x) \neq 0\}} = [\tau_i, \tau_{i+k+1}].$$

Wie zuvor folgt für $s \in \mathbb{P}_{k,\tau}$ die Darstellung (Berechnung der Koeffizienten, vgl. Abschnitt 2.3)

$$s = \sum_{i=-k}^{m-1} c_i N_{ki}.$$

Damit folgt die Behauptung. \diamond

Vgl. **Illustration** (Rekursive Berechnung einer quadratischen B-Splinefunktion).

- Bei einer Linearkombination von B-Splinefunktionen führen kleine Änderungen der Koeffizienten zu kleinen Änderungen der Funktionswerte der Splinefunktion. Ebenso bewirken kleine Änderungen der Funktionswerte der Splinefunktion kleine Änderungen der Koeffizienten.

Kondition der B-Splines (Satz 2.8): Es gilt die Abschätzung

$$\gamma \max_{-k \leq i \leq m-1} |c_i| \leq \|s\|_\infty = \max_{a \leq x \leq b} |s(x)| \leq \max_{-k \leq i \leq m-1} |c_i|, \quad s = \sum_{i=-k}^{m-1} c_i N_{ki},$$

mit einer von der Wahl der Knoten und Koeffizienten unabhängigen Konstante γ .

2.3. Linearkombinationen von B-Splines

- **Effiziente Berechnung von Splinefunktionen:** Die effiziente Berechnung einer Splinefunktion beruht auf der Darstellung als Linearkombination von B-Splines, vgl. Abschnitt 2.2.

Einsetzen der Rekursion (Kubische Splinefunktionen): Für eine kubische Splinefunktion

$$s = \sum_{i=-3}^{m-1} c_i N_{3i} \in \mathbb{P}_{3,\tau}$$

ergeben sich bei schrittweisem Einsetzen der in Abschnitt 2.2 angegebenen Rekursion

$$\begin{aligned} N_{ki}(x) &= \frac{1}{\tau_{i+k}-\tau_i} (x - \tau_i) N_{k-1,i}(x) \\ &\quad + \frac{1}{\tau_{i+k+1}-\tau_{i+1}} (\tau_{i+k+1} - x) N_{k-1,i+1}(x), \quad -k \leq i \leq m-1, \quad k \geq 1, \\ N_{3i}(x) &= \frac{x-\tau_i}{\tau_{i+3}-\tau_i} N_{2i}(x) + \frac{\tau_{i+4}-x}{\tau_{i+4}-\tau_{i+1}} N_{2,i+1}(x), \quad -3 \leq i \leq m-1, \\ N_{2i}(x) &= \frac{x-\tau_i}{\tau_{i+2}-\tau_i} N_{1i}(x) + \frac{\tau_{i+3}-x}{\tau_{i+3}-\tau_{i+1}} N_{1,i+1}(x), \quad -2 \leq i \leq m-1, \\ N_{1i}(x) &= \frac{x-\tau_i}{\tau_{i+1}-\tau_i} N_{0i}(x) + \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_{i+1}} N_{0,i+1}(x), \quad -1 \leq i \leq m-1, \\ N_{0i}(x) &= \begin{cases} 1 & x \in [\tau_i, \tau_{i+1}), \\ 0 & x \notin [\tau_i, \tau_{i+1}), \end{cases} \quad 0 \leq i \leq m-1, \end{aligned}$$

die Relationen (Indexverschiebung $j = i + 1 \leftrightarrow i = j - 1$)

$$\begin{aligned} s(x) &= \sum_{i=-3}^{m-1} c_i N_{3i}(x) \\ &= \sum_{i=-3}^{m-1} c_i \frac{x-\tau_i}{\tau_{i+3}-\tau_i} N_{2i}(x) + \sum_{i=-3}^{m-1} c_i \frac{\tau_{i+4}-x}{\tau_{i+4}-\tau_{i+1}} N_{2,i+1}(x) \\ &= \sum_{i=-2}^{m-1} \left(c_i \frac{x-\tau_i}{\tau_{i+3}-\tau_i} + c_{i-1} \frac{\tau_{i+3}-x}{\tau_{i+3}-\tau_i} \right) N_{2i}(x) + \text{Randterme}, \\ s(x) &= \sum_{i=-1}^{m-1} \left(\left(c_i \frac{x-\tau_i}{\tau_{i+3}-\tau_i} + c_{i-1} \frac{\tau_{i+3}-x}{\tau_{i+3}-\tau_i} \right) \frac{x-\tau_i}{\tau_{i+2}-\tau_i} + \left(c_{i-1} \frac{x-\tau_{i-1}}{\tau_{i+2}-\tau_{i-1}} + c_{i-2} \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_{i-1}} \right) \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_i} \right) N_{1i}(x) \\ &\quad + \text{Randterme}, \\ s(x) &= \sum_{i=0}^{m-1} \left(\left(\left(c_i \frac{x-\tau_i}{\tau_{i+3}-\tau_i} + c_{i-1} \frac{\tau_{i+3}-x}{\tau_{i+3}-\tau_i} \right) \frac{x-\tau_i}{\tau_{i+2}-\tau_i} \right. \right. \\ &\quad \left. \left. + \left(c_{i-1} \frac{x-\tau_{i-1}}{\tau_{i+2}-\tau_{i-1}} + c_{i-2} \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_{i-1}} \right) \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_i} \right) \frac{x-\tau_i}{\tau_{i+1}-\tau_i} \right. \\ &\quad \left. + \left(\left(c_{i-1} \frac{x-\tau_{i-1}}{\tau_{i+2}-\tau_{i-1}} + c_{i-2} \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_{i-1}} \right) \frac{x-\tau_{i-1}}{\tau_{i+1}-\tau_{i-1}} \right. \right. \\ &\quad \left. \left. + \left(c_{i-2} \frac{x-\tau_{i-2}}{\tau_{i+1}-\tau_{i-2}} + c_{i-3} \frac{\tau_{i+1}-x}{\tau_{i+1}-\tau_{i-2}} \right) \frac{\tau_{i+1}-x}{\tau_{i+1}-\tau_{i-1}} \right) \frac{\tau_{i+1}-x}{\tau_{i+1}-\tau_i} \right) N_{0i}(x) + \text{Randterme}. \end{aligned}$$

Berechnung der Koeffizienten (Allgemeiner Fall): Die Koeffizienten $(c_i)_{-k \leq i \leq m-1}$ werden so bestimmt, daß die Splinefunktion an Stützstellen $(x_i)_{-k \leq i \leq m-1}$, welche auch von den Knoten verschieden sein können, eine vorgegebene Funktion f interpoliert

$$s(x_i) = f(x_i), \quad -k \leq i \leq m-1.$$

Der **Satz von Schoenberg–Whitney** (Satz 2.9) sichert die Existenz und Eindeutigkeit der Lösung, wenn die Stützstellen die Bedingung $x_i \in (\tau_i, \tau_{i+k})$ für $-k \leq i \leq m-1$ erfüllen.

Berechnung der Koeffizienten (Kubische Splinefunktionen): Beispielsweise im Fall einer kubischen Splinefunktion führen die Interpolationsbedingungen und die obigen Überlegungen

$$\begin{aligned} s(x_i) &= f(x_i), \quad -k \leq i \leq m-1, \\ s(x) &= \sum_{i=0}^{m-1} \left(\left(c_i \frac{x-\tau_i}{\tau_{i+3}-\tau_i} + c_{i-1} \frac{\tau_{i+3}-x}{\tau_{i+3}-\tau_i} \right) \frac{x-\tau_i}{\tau_{i+2}-\tau_i} \right. \\ &\quad + \left(c_{i-1} \frac{x-\tau_{i-1}}{\tau_{i+2}-\tau_{i-1}} + c_{i-2} \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_{i-1}} \right) \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_i} \frac{x-\tau_i}{\tau_{i+1}-\tau_i} \\ &\quad + \left(c_{i-1} \frac{x-\tau_{i-1}}{\tau_{i+2}-\tau_{i-1}} + c_{i-2} \frac{\tau_{i+2}-x}{\tau_{i+2}-\tau_{i-1}} \right) \frac{x-\tau_{i-1}}{\tau_{i+1}-\tau_{i-1}} \\ &\quad \left. + \left(c_{i-2} \frac{x-\tau_{i-2}}{\tau_{i+1}-\tau_{i-2}} + c_{i-3} \frac{\tau_{i+1}-x}{\tau_{i+1}-\tau_{i-2}} \right) \frac{\tau_{i+1}-x}{\tau_{i+1}-\tau_i} \right) N_{0i}(x) + \text{Randterme}, \end{aligned}$$

auf ein lineares Gleichungssystem für die Koeffizienten $(c_i)_{-3 \leq i \leq m-1}$, welches in der Situation des Satzes von Schoenberg–Whitney eindeutig lösbar ist. Aufgrund der Lokalität der B-Splines besitzt die zugehörige Matrix **Bandstruktur** (Diagonale und wenige Nebendiagonale beinhalten Elemente, die verschieden von Null sind) und das Gleichungssystem ist somit mittels LR-Zerlegung (sogar ohne Pivotsuche) effizient lösbar.

3. Numerische Integration

- **Inhalt:** Verfahren zur **numerischen Quadratur**, d.h. zur näherungsweise Berechnung bestimmter Integrale der Form

$$I(f) = \int_a^b f(x) \, dx$$

mit einer (zumindest stetigen) Funktion $f : [a, b] \rightarrow \mathbb{R}$.

Bemerkung:

- Die numerische Berechnung bestimmter Integrale ist erforderlich, wenn es nicht möglich ist, eine Stammfunktion des Integranden in geschlossener Form (d.h. mittels elementarer Funktionen) anzugeben. Es gibt aber auch Situationen, in denen eine **direkte Auswertung** der Stammfunktion aufgrund des Auftretens des Phänomens der Auslöschung signifikanter Stellen **numerisch instabil** ist, numerische Quadratur hingegen ein zufriedenstellendes Ergebnis ergibt.

Illustration (Partielle Integration, Instabilität).

- Numerische Quadraturformeln beruhen auf der Idee, das bestimmte Integral

$$I(f) = \int_a^b f(x) \, dx$$

durch ein einfach zu berechnendes bestimmtes Integral

$$I(\tilde{f}) = \int_a^b \tilde{f}(x) \, dx$$

zu approximieren, d.h. der (komplizierte) Integrand f wird durch eine geeignete Approximation einfacherer Struktur (beispielsweise durch eine (stückweise) Polynomfunktion) ersetzt.

- Falls Funktionswerte des Integranden an beliebigen Stützstellen im Intervall $[a, b]$ berechnet werden können, stehen adaptive Quadraturformeln zur Verfügung, die für hinreichend reguläre Integranden ein Ergebnis hoher Genauigkeit liefern.
- **Erinnerung:** Ein Maß für die Länge einer stetigen Funktion oder auch den Abstand zweier stetiger Funktionen ist die **Supremumsnorm**

$$\|f\|_\infty = \max_{a \leq x \leq b} |f(x)|, \quad \|f - \tilde{f}\|_\infty = \max_{a \leq x \leq b} |f(x) - \tilde{f}(x)|, \quad f, \tilde{f} \in \mathcal{C}([a, b]).$$

Kondition der Integration: Zur Untersuchung der Sensibilität des Wertes eines bestimmten Integrals in Abhängigkeit vom Integranden betrachtet man die Differenz

$$I(f) - I(\tilde{f}) = \int_a^b (f(x) - \tilde{f}(x)) \, dx, \quad f, \tilde{f} \in \mathcal{C}([a, b]).$$

Als Abschätzung für die **absolute Kondition** ergibt sich

$$\begin{aligned} |I(f) - I(\tilde{f})| &\leq \int_a^b |f(x) - \tilde{f}(x)| \, dx \leq \|f - \tilde{f}\|_\infty \int_a^b 1 \, dx \\ &= (b - a) \|f - \tilde{f}\|_\infty, \quad f, \tilde{f} \in \mathcal{C}([a, b]). \end{aligned}$$

Bei der **relativen Kondition** hingegen

$$\begin{aligned} \left| \frac{I(f) - I(\tilde{f})}{I(f)} \right| &\leq (b - a) \frac{\|f - \tilde{f}\|_\infty}{|I(f)|} \\ &= \underbrace{\frac{(b - a) \|f\|_\infty}{|I(f)|}}_{=\kappa} \frac{\|f - \tilde{f}\|_\infty}{\|f\|_\infty}, \quad f, \tilde{f} \in \mathcal{C}([a, b]), \end{aligned}$$

kann der Fall eintreten, daß der Faktor

$$\kappa = \frac{(b - a) \|f\|_\infty}{|I(f)|} \gg 1$$

einen großen Wert annimmt und somit kleine relative Änderungen des Integranden zu großen Änderungen des Ergebnisses führen. Ein typischer Fall eines schlecht konditionierten numerische Problems ist ein stark oszillierender Integrand, der etwa bei Fourier-Integralen auftritt ($I(f) \approx 0$, $(b - a) \|f\|_\infty \approx 1$).

3.1. Elementare Quadraturformeln

- **Fragestellung:** Einführung grundlegender Begriffe und Relationen.

Eine s -stufige **Quadraturformel** $(b_i, c_i)_{1 \leq i \leq s}$ ist durch die **Knoten** $c_i \in [0, 1]$ und die zugehörigen **Gewichte** $b_i \in \mathbb{R}$ für $1 \leq i \leq s$ gegeben. Eine Quadraturformel $(b_i, c_i)_{1 \leq i \leq s}$ heißt **symmetrisch**, falls $c_{s+1-i} = 1 - c_i$ und $b_{s+1-i} = b_i$ für $1 \leq i \leq s$.

- **Lokale Approximation (Einheitsintervall):** Für eine Funktion $g : [0, 1] \rightarrow \mathbb{R}$ erhält man eine **Approximation** an den Wert des bestimmten Integrals mittels

$$Q_0(g) = \sum_{i=1}^s b_i g(c_i) \approx I_0(g) = \int_0^1 g(\xi) d\xi.$$

Die **Konstruktion von (ersten einfachen) Quadraturformeln** beruht auf der Approximation von g durch *einfache* Funktionen \tilde{g} (z.B. Polynominterpolanten) für welche das bestimmte Integral leicht berechnet werden kann (s.u.)

$$Q_0(g) = \sum_{i=1}^s b_i g(c_i) = I_0(\tilde{g}) = \int_0^1 \tilde{g}(\xi) d\xi \approx I_0(g) = \int_0^1 g(\xi) d\xi.$$

Der (lokale) **Verfahrensfehler**

$$Q_0(g) - I_0(g) = \sum_{i=1}^s b_i g(c_i) - \int_0^1 g(\xi) d\xi$$

ist damit durch die Relation

$$Q_0(g) - I_0(g) = \sum_{i=1}^s b_i g(c_i) - \int_0^1 g(\xi) d\xi = I_0(\tilde{g}) - I_0(g) = \int_0^1 (\tilde{g} - g)(\xi) d\xi$$

gegeben.

- **Globale Approximation:** Für eine Funktion $f : [a, b] \rightarrow \mathbb{R}$ erhält man mittels einer Zerlegung des betrachteten Intervalles in *kleine* Teilintervalle

$$a = x_0 < \dots < x_N = b, \quad h_j = x_{j+1} - x_j, \quad 0 \leq j \leq N-1,$$

und der Variablentransformation $x = x_j + \xi h_j \leftrightarrow \xi = \frac{1}{h_j} (x - x_j)$ die **Approximation**

$$\begin{aligned} I(f) &= \int_a^b f(x) dx = \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} f(x) dx = \sum_{j=0}^{N-1} h_j \int_0^1 f(x_j + \xi h_j) d\xi \\ &\approx Q(f) = \sum_{j=0}^{N-1} h_j \sum_{i=1}^s b_i f(x_j + c_i h_j). \end{aligned}$$

Wesentlich für die Abschätzung des (globalen) **Verfahrensfehlers**

$$Q(f) - I(f) = \sum_{j=0}^{N-1} h_j \left(\sum_{i=1}^s b_i f(x_j + c_i h_j) - \int_0^1 f(x_j + \xi h_j) d\xi \right)$$

sind somit Abschätzungen für den lokalen Verfahrensfehler

$$Q_0(g_j) - I_0(g_j) = \sum_{i=1}^s b_i g_j(c_i) - \int_0^1 g_j(\xi) d\xi, \quad g_j(\xi) = f(x_j + \xi h_j), \quad 0 \leq j \leq N-1.$$

Bemerkung: Es ist wünschenswert, Quadraturformeln mit **positiven Gewichten** zu verwenden. In diesem Fall ist sichergestellt, daß aus der Positivität des Integranden die Positivität des bestimmten Integrals und ebenso die Positivität der Quadraturapproximation folgt

$$f(x) \geq 0 \quad \text{für } x \in [a, b] \quad \implies \quad I(f) \geq 0, \quad Q(f) \geq 0.$$

- **Fragestellung:** Untersuchung des Verfahrensfehlers der Quadraturapproximation (Güte der Approximation, Konstruktion von Quadraturformeln)

– Zur Abschätzung des **Verfahrensfehlers der Quadraturapproximation**

$$Q(f) - I(f) = \sum_{j=0}^{N-1} h_j \left(\sum_{i=1}^s b_i f(x_j + c_i h_j) - \int_0^1 f(x_j + \xi h_j) d\xi \right)$$

betrachtet man zunächst den lokalen Verfahrensfehler. Mittels der Taylorreihenentwicklungen

$$f(x_j + c_i h_j) = \sum_{\ell=0}^{p-1} \frac{1}{\ell!} f^{(\ell)}(x_j) c_i^\ell h_j^\ell + \frac{1}{p!} f^{(p)}(\eta_{ij}) c_i^p h_j^p,$$

$$f(x_j + \xi h_j) = \sum_{\ell=0}^{p-1} \frac{1}{\ell!} f^{(\ell)}(x_j) \xi^\ell h_j^\ell + \frac{1}{p!} f^{(p)}(\tilde{\eta}_{ij}) \xi^p h_j^p,$$

ergibt sich die Relation

$$\begin{aligned} & \sum_{i=1}^s b_i f(x_j + c_i h_j) - \int_0^1 f(x_j + \xi h_j) d\xi \\ &= \sum_{i=1}^s b_i \sum_{\ell=0}^{p-1} \frac{1}{\ell!} f^{(\ell)}(x_j) c_i^\ell h_j^\ell - \sum_{\ell=0}^{p-1} \frac{1}{\ell!} f^{(\ell)}(x_j) h_j^\ell \underbrace{\int_0^1 \xi^\ell d\xi}_{=\frac{1}{\ell+1}} \\ & \quad + \sum_{i=1}^s b_i \frac{1}{p!} f^{(p)}(\eta_{ij}) c_i^p h_j^p - \frac{1}{p!} f^{(p)}(\tilde{\eta}_{ij}) h_j^p \underbrace{\int_0^1 \xi^p d\xi}_{=\frac{1}{p+1}} \\ &= \sum_{\ell=0}^{p-1} \frac{1}{\ell!} \left(\sum_{i=1}^s b_i c_i^\ell - \frac{1}{\ell+1} \right) f^{(\ell)}(x_j) h_j^\ell + \frac{1}{p!} \left(\sum_{i=1}^s b_i f^{(p)}(\eta_{ij}) c_i^p - \frac{1}{p+1} f^{(p)}(\tilde{\eta}_{ij}) \right) h_j^p. \end{aligned}$$

Weiters erhält man die Entwicklung

$$Q(f) - I(f) = \sum_{j=0}^{N-1} \sum_{\ell=0}^{p-1} \frac{1}{\ell!} \left(\sum_{i=1}^s b_i c_i^\ell - \frac{1}{\ell+1} \right) f^{(\ell)}(x_j) h_j^{\ell+1} \\ + \frac{1}{p!} \sum_{j=0}^{N-1} \left(\sum_{i=1}^s b_i f^{(p)}(\eta_{ij}) c_i^p - \frac{1}{p+1} f^{(p)}(\tilde{\eta}_{ij}) \right) h_j^{p+1}.$$

– Falls die **Ordnungsbedingungen**

$$\sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}, \quad 1 \leq q \leq p,$$

$$q = 1: \quad b_1 + \dots + b_s = 1$$

$$q = 2: \quad b_1 c_1 + \dots + b_s c_s = \frac{1}{2}$$

$$q = 3: \quad b_1 c_1^2 + \dots + b_s c_s^2 = \frac{1}{3}$$

⋮

$$q = p: \quad b_1 c_1^{p-1} + \dots + b_s c_s^{p-1} = \frac{1}{p}$$

erfüllt sind, ergibt die Quadraturformel eine Approximation der **Ordnung** $p \geq 1$, d.h. für den lokalen Verfahrensfehler gilt die Relation (sofern $f^{(p)}$ auf $[a, b]$ beschränkt ist)

$$\sum_{i=1}^s b_i f(x_j + c_i h_j) - \int_0^1 f(x_j + \xi h_j) d\xi = \mathcal{O}(h_{\max}^p), \quad h_{\max} = \max_{0 \leq j \leq N-1} h_j,$$

und für den globalen Verfahrensfehler folgt die Abschätzung (mit Konstante C abhängig von den Knoten und Gewichten $(b_i, c_i)_{1 \leq i \leq s}$ sowie der Ordnung p)

$$|Q(f) - I(f)| \leq C \|f^{(p)}\|_\infty \sum_{j=0}^{N-1} h_j^{p+1} \leq C(b-a) h_{\max}^p \|f^{(p)}\|_\infty, \quad h_{\max} = \max_{0 \leq j \leq N-1} h_j.$$

– Für eine Quadraturformel $(b_i, c_i)_{1 \leq i \leq s}$ der Ordnung p folgt insbesondere, daß die Quadraturapproximation auf dem Einheitsintervall für die Taylor-Basis

$$g_\ell(x) = x^\ell, \quad 0 \leq \ell \leq p-1,$$

das **exakte Ergebnis** liefert (verwende Linearität der Quadraturformel und des bestimmten Integrals)

$$\sum_{i=1}^s b_i c_i^\ell = \sum_{i=1}^s b_i g_\ell(c_i) = \int_0^1 g_\ell(\xi) d\xi = \int_0^1 \xi^\ell d\xi = \frac{1}{\ell+1}, \quad 0 \leq \ell \leq p-1.$$

- **Beispiele für (einfache) Quadraturformeln:** Mittelpunktsregel, Trapezregel, Simpsonregel

– Mittelpunktsregel

- * Die Forderungen $s = 1$ und $p = 2s = 2$ führen auf die Knoten und Gewichte der Mittelpunktsregel (Lösung der Ordnungsbedingungen $b_1 = 1$, $b_1 c_1 = \frac{1}{2}$, jedoch $b_1 c_1^2 \neq \frac{1}{3}$, symmetrische Quadraturformel, **Superkonvergenz** $p > s$, **Gaußsche Quadraturformel** der maximalen Ordnung $p = 2s$)

$$s = 1, \quad b_1 = 1, \quad c_1 = \frac{1}{2}, \quad p = 2.$$

- * **Historischer Zugang:** Interpolation der Funktion $g : [0, 1] \rightarrow \mathbb{R}$ im Intervallmittelpunkt $\frac{1}{2}$ mittels eines Polynoms vom Grad 0 und Integration führt auf die Quadraturapproximation

$$\begin{aligned} \tilde{g}(\xi) &= g\left(\frac{1}{2}\right) \approx g(\xi), \quad 0 \leq \xi \leq 1, \\ Q_0(g) &= \int_0^1 \tilde{g}(\xi) \, d\xi = \int_0^1 g\left(\frac{1}{2}\right) \, d\xi = g\left(\frac{1}{2}\right) \approx I_0(g) = \int_0^1 g(\xi) \, d\xi, \\ s &= 1, \quad b_1 = 1, \quad c_1 = \frac{1}{2}. \end{aligned}$$

Vgl. Abbildung, Skriptum, S. 39 ($a = 0, b = 1$, Rechtecksfläche).

- * Insgesamt ergibt sich die Quadraturapproximation (wegen $h_j = x_{j+1} - x_j$ und $x_j + \frac{1}{2} h_j = \frac{1}{2} (x_j + x_{j+1})$)

$$Q(f) = \sum_{j=0}^{N-1} h_j f\left(\frac{x_j + x_{j+1}}{2}\right) \approx I(f) = \int_0^1 f(x_j + \xi h_j) \, d\xi.$$

– Trapezregel

- * Die Forderungen $s = 2$ und $p = s$ sowie $c_1 = 0$ und $c_2 = 1$ führen auf die Gewichte der Trapezregel (Lösung der Ordnungsbedingungen $b_1 + b_2 = 1$, $b_1 c_1 + b_2 c_2 = \frac{1}{2}$, jedoch $b_1 c_1^2 + b_2 c_2^2 \neq \frac{1}{3}$, äquidistante Knoten in $[0, 1]$, Newton–Côtes Formel)

$$s = 2, \quad b_1 = \frac{1}{2} = b_2, \quad c_1 = 0, \quad c_2 = 1, \quad p = 2.$$

- * **Historischer Zugang:** Interpolation der Funktion $g : [0, 1] \rightarrow \mathbb{R}$ in den Intervallenden 0, 1 mittels eines Polynoms vom Grad 1 und Integration führt auf die Quadraturapproximation

$$\begin{aligned} \tilde{g}(\xi) &= g(0) + (g(1) - g(0))\xi \approx g(\xi), \quad 0 \leq \xi \leq 1, \\ Q_0(g) &= \int_0^1 \tilde{g}(\xi) \, d\xi = \int_0^1 (g(0) + (g(1) - g(0))\xi) \, d\xi = g(0) + \frac{1}{2} (g(1) - g(0)) \\ &= \frac{1}{2} (g(0) + g(1)) \approx I_0(g) = \int_0^1 g(\xi) \, d\xi, \\ s &= 2, \quad b_1 = \frac{1}{2} = b_2, \quad c_1 = 0, \quad c_2 = 1. \end{aligned}$$

Vgl. Abbildung, Skriptum, S. 41 ($a = 0, b = 1$, Fläche des Trapezes).

* Insgesamt ergibt sich die Quadraturapproximation

$$\begin{aligned}
 Q(f) &= \frac{1}{2} \sum_{j=0}^{N-1} h_j (f(x_j) + f(x_{j+1})) \\
 &= \frac{1}{2} \sum_{j=0}^{N-1} h_j f(x_j) + \frac{1}{2} \sum_{j=0}^{N-1} h_j f(x_{j+1}) \\
 &= \frac{1}{2} \sum_{j=0}^{N-1} h_j f(x_j) + \frac{1}{2} \sum_{j=1}^N h_{j-1} f(x_j) \\
 &= \frac{h_0}{2} f(x_0) + \sum_{j=1}^{N-1} \frac{h_{j-1} + h_j}{2} f(x_j) + \frac{h_{N-1}}{2} f(x_N) \approx I(f) = \int_a^b f(x) dx.
 \end{aligned}$$

* Für periodische Funktionen und äquidistante Stützstellen vereinfacht sich die obige Relation zu $(f(x_N) = f(x_0))$ und $h_j = h$ für $0 \leq j \leq N-1$

$$Q(f) = h \sum_{j=0}^{N-1} f(x_j) \approx I(f) = \int_a^b f(x) dx.$$

Die Trapezregel hat spezielle Bedeutung bei der numerischen Berechnung von **Fourierreihen** (Signalverarbeitung).

– **Simpsonregel** (Faßregel, Johannes Kepler, 1615)

* Die Forderungen $s = 3$ und $p = s$ sowie $c_1 = 0$, $c_2 = \frac{1}{2}$ und $c_3 = 1$ führen auf die Gewichte der Simpsonregel (Lösung der entsprechenden Ordnungsbedingungen $b_1 + b_2 + b_3 = 1$, $b_1 c_1 + b_2 c_2 + b_3 c_3 = \frac{1}{2}$, $b_1 c_1^2 + b_2 c_2^2 + b_3 c_3^2 = \frac{1}{3}$, äquidistante Knoten in $[0, 1]$, Newton-Côtes Formel, aufgrund der Symmetrie der Quadraturformel gilt sogar $b_1 c_1^3 + b_2 c_2^3 + b_3 c_3^3 = \frac{1}{4}$ jedoch $b_1 c_1^4 + b_2 c_2^4 + b_3 c_3^4 = \frac{1}{5}$ und damit $p = 4$)

$$s = 3, \quad b_1 = \frac{1}{6}, \quad b_2 = \frac{2}{3}, \quad b_3 = \frac{1}{6}, \quad c_1 = 0, \quad c_2 = \frac{1}{2}, \quad c_3 = 1, \quad p = 4.$$

* **Historischer Zugang:** Interpolation der Funktion $g : [0, 1] \rightarrow \mathbb{R}$ in den Intervallen $0, 1$ und dem Intervallmittelpunkt $\frac{1}{2}$ mittels eines Polynoms vom Grad 2

0	$g(0)$	
	$2(g(\frac{1}{2}) - g(0))$	
$\frac{1}{2}$	$g(\frac{1}{2})$	$2(g(0) - 2g(\frac{1}{2}) + g(1))$
	$2(g(1) - g(\frac{1}{2}))$	
1	$g(1)$	

$$\begin{aligned}
 \tilde{g}(\xi) &= g(0) + 2(g(\frac{1}{2}) - g(0))\xi + 2(g(0) - 2g(\frac{1}{2}) + g(1))\xi(\xi - \frac{1}{2}) \\
 &\approx g(\xi), \quad 0 \leq \xi \leq 1,
 \end{aligned}$$

und Integration führt auf die Quadraturapproximation

$$\begin{aligned}
 Q_0(g) &= \int_0^1 \tilde{g}(\xi) \, d\xi \\
 &= \int_0^1 \left(g(0) + 2 \left(g\left(\frac{1}{2}\right) - g(0) \right) \xi + \left(g(0) - 2g\left(\frac{1}{2}\right) + g(1) \right) (2\xi^2 - \xi) \right) d\xi \\
 &= g\left(\frac{1}{2}\right) + \frac{1}{6} \left(g(0) - 2g\left(\frac{1}{2}\right) + g(1) \right) \\
 &= \frac{1}{6} \left(g(0) + 4g\left(\frac{1}{2}\right) + g(1) \right), \\
 s &= 3, \quad b_1 = \frac{1}{6}, \quad b_2 = \frac{2}{3}, \quad b_3 = \frac{1}{6}, \quad c_1 = 0, \quad c_2 = \frac{1}{2}, \quad c_3 = 1.
 \end{aligned}$$

* Insgesamt ergibt sich die Quadraturapproximation

$$\begin{aligned}
 Q(f) &= \frac{1}{6} \sum_{j=0}^{N-1} h_j \left(f(x_j) + 4f\left(\frac{x_j+x_{j+1}}{2}\right) + f(x_{j+1}) \right) \\
 &= \frac{h_0}{6} f(x_0) + \sum_{j=1}^{N-1} \frac{h_{j-1}+h_j}{6} f(x_j) + \frac{2}{3} \sum_{j=0}^{N-1} h_j f\left(\frac{x_j+x_{j+1}}{2}\right) + \frac{h_{N-1}}{6} f(x_N) \\
 &\approx I(f) = \int_a^b f(x) \, dx.
 \end{aligned}$$

• **Verschiedene Klassen von Quadraturformeln:**

- Allgemeiner erhält man die **Newton–Côtes Quadraturformeln** bei Vorgabe von s äquidistanten Knoten in $[0, 1]$ und Lösung der Ordnungsbedingungen für $p = s$ (eindeutig lösbares lineares Gleichungssystem mit Vandermonde Matrix, allerdings schlecht konditioniert). Dies entspricht gerade der Integration des Polynominterpolanten. Da für höhere Stufenzahlen negative Gewichte auftreten, werden Newton–Côtes Quadraturformeln höherer Ordnung im Allgemeinen vermieden.
- Bei Verwendung der Chebychev-Knoten ergeben sich die **Clenshaw–Curtis Quadraturformeln** mit stets positiven Gewichten.
- Quadraturformeln hoher Ordnung, deren Konstruktion auf orthogonalen Polynomen beruht, sind beispielsweise die Gaußschen Quadraturformeln ($p = 2s$) und die Radauschen Quadraturformeln ($p = 2s - 1$).

3.2. Best-Approximation des Integrals

- **Fragestellung:** Zu bestimmen ist die **beste Quadraturapproximation** an das Integral

$$\tilde{I}(f) = \int_a^b f(x) dx$$

bei Vorgabe von s Stützstellen und zugehörigen Stützwerten

$$a \leq \gamma_1 < \dots < \gamma_s \leq b, \quad \eta_i = f(\gamma_i), \quad 1 \leq i \leq s.$$

Betrachtet werden dabei **lokale Quadraturapproximationen** der Form (Transformation des Intervalles $[a, b]$ auf das Einheitsintervall $[0, 1]$ mittels $x = a + \xi(b-a) \leftrightarrow \xi = \frac{x-a}{b-a}$)

$$\begin{aligned} \tilde{I}(f) &= \int_a^b f(x) dx = (b-a) \int_0^1 f(a + \xi(b-a)) d\xi \\ &\approx \tilde{Q}(f) = \sum_{i=1}^s \underbrace{(b-a)b_i}_{\beta_i} f(\underbrace{a + c_i(b-a)}_{=\gamma_i}) = \sum_{i=1}^s \beta_i f(\gamma_i). \end{aligned}$$

Als Maß für Funktionen betrachtet man die mittlere quadratische Krümmung (Seminorm, $v(f) = \|f''\|_{L^2}$)

$$v(f)^2 = \int_a^b (f''(x))^2 dx, \quad f \in \mathcal{C}^2([a, b]).$$

Je *glatter* die Funktion f ist, desto kleiner ist $v(f)$, und insbesondere gilt $v(f) = 0$ für lineare Polynomfunktionen $f \in \mathbb{P}_1$. Für beliebige Funktionen $f \in \mathcal{C}^2([a, b]) \setminus \mathbb{P}_1$ sollen die Quadraturgewichte $(\beta_i)_{1 \leq i \leq s}$ so bestimmt werden, daß die Größe

$$\frac{1}{v(f)} |\tilde{Q}(f) - \tilde{I}(f)| \longrightarrow \min$$

minimal wird. Als Lösungen des Minimierungsproblems ergeben sich **natürliche kubische Splinefunktionen** zu den Stützstellen $(\gamma_i)_{1 \leq i \leq s}$ (**Variationsrechnung**)

$$\tilde{Q}(f) = \tilde{I}(s), \quad s \in \mathbb{P}_{3,\gamma}, \quad s''(\gamma_1) = 0 = s''(\gamma_s).$$

3.3. Romberg-Quadratur

- **Vorbemerkung:** Für die folgenden Überlegungen ist es zweckmäßig, einen hinreichend regulären Integranden f zu fixieren und anstelle der Abhängigkeit der Quadraturapproximation vom Integranden die Abhängigkeit von der Schrittweite h (zu einer **äquidistanten Zerlegung** des Integrationsintervalles) anzugeben. Ebenso wird der Wert des bestimmten Integrals kurz mit I (oder auch mit α_0) bezeichnet.

Erinnerung: Bei äquidistanten Stützstellen $a = x_0 < \dots < x_N = b$ mit $x_j = a + jh$ für $0 \leq j \leq N-1$, wobei $h = \frac{b-a}{N}$, führt die Anwendung der Trapezregel auf die globale Quadraturapproximation

$$T(h) = h \left(\frac{1}{2} f(x_0) + \sum_{j=1}^{N-1} f(x_j) + \frac{1}{2} f(x_N) \right) \approx I = \int_a^b f(x) dx.$$

Frühere Überlegungen zeigten die Abschätzung

$$|T(h) - I| \leq C(b-a)h^2 \|f''\|_\infty.$$

Eine zusätzliche Analyse des Verfahrensfehlers führt auf eine **asymptotische Entwicklung** (ohne Begründung). Diese Entwicklung wird dann genutzt, um **verbesserte Approximationen** zu berechnen (Extrapolation, Romberg-Quadratur).

Asymptotische Entwicklung (Trapezregel) (Satz 3.1): Es sei $f \in \mathcal{C}^{2K}([a, b])$. Der Verfahrensfehler der Trapezregel (mit $h = \frac{b-a}{N}$ und $x_j = a + jh$ für $0 \leq j \leq N-1$)

$$T(h) = h \left(\frac{1}{2} f(x_0) + \sum_{j=1}^{N-1} f(x_j) + \frac{1}{2} f(x_N) \right) \approx \alpha_0 = \int_a^b f(x) dx$$

besitzt die asymptotische Entwicklung

$$T(h) - \alpha_0 = \sum_{k=1}^{K-1} \alpha_k h^{2k} + \alpha_K(h) h^{2K}, \quad |\alpha_K(h)| \leq C \int_a^b |f^{(2K)}(x)| dx,$$

mit Koeffizienten $\alpha_k \in \mathbb{R}$ für $1 \leq k \leq K-1$ und einer beschränkten Funktion $\alpha_K: \mathbb{R} \rightarrow \mathbb{R}$ (Schranke unabhängig von h).

Bemerkungen:

- Aus der obigen Relation und insbesondere wegen der Abschätzung für α_K folgt die Konvergenz der Quadraturapproximation gegen den Wert des bestimmten Integrals für $h \rightarrow 0$.
- Für hinreichend kleine Werte der Schrittweite h ist $\alpha_1 h^2$ der dominante Beitrag im Verfahrensfehler

$$T(h) - \alpha_0 = \alpha_1 h^2 + \alpha_2 h^4 + \dots + \alpha_{K-1} h^{2(K-1)} + \alpha_K(h) h^{2K} = \alpha_1 h^2 + \mathcal{O}(h^4).$$

Elimination dominanter Fehlerterme (Richardson-Extrapolation): Die grundlegende Idee der Extrapolation ist, führende Fehlerterme durch geeignete Linearkombinationen zu verschiedenen Schrittweiten zu eliminieren.

- Für zwei verschiedene Schrittweiten $h_1 \neq h_2$ (zu zwei **äquidistanten Zerlegungen** des Integrationsintervalles) lauten die Entwicklungen

$$\begin{aligned} T(h_1) - \alpha_0 &= \alpha_1 h_1^2 + \alpha_2 h_1^4 + \cdots + \alpha_{K-1} h_1^{2(K-1)} + \alpha_K(h_1) h_1^{2K}, \\ T(h_2) - \alpha_0 &= \alpha_1 h_2^2 + \alpha_2 h_2^4 + \cdots + \alpha_{K-1} h_2^{2(K-1)} + \alpha_K(h_2) h_2^{2K}. \end{aligned}$$

Elimination der Unbekannten α_1 führt auf (Multiplikation und Subtraktion analog zum Gaußschen Eliminationsverfahren, Bezeichnung für restliche Terme)

$$\begin{aligned} T(h_1) - \alpha_0 &= \alpha_1 h_1^2 + \alpha_2 h_1^4 + R(h_1, 6), \\ T(h_2) - \alpha_0 &= \alpha_1 h_2^2 + \alpha_2 h_2^4 + R(h_2, 6), \\ h_2^2 (T(h_1) - \alpha_0) - h_1^2 (T(h_2) - \alpha_0) &= \alpha_2 h_1^2 h_2^2 (h_1^2 - h_2^2) + h_2^2 R(h_1, 6) - h_1^2 R(h_2, 6), \\ h_2^2 T(h_1) - h_1^2 T(h_2) + (h_1^2 - h_2^2) \alpha_0 &= \alpha_2 h_1^2 h_2^2 (h_1^2 - h_2^2) + h_2^2 R(h_1, 6) - h_1^2 R(h_2, 6), \\ \frac{h_2^2 T(h_1) - h_1^2 T(h_2)}{h_2^2 - h_1^2} &= \alpha_0 - \alpha_2 h_1^2 h_2^2 + \frac{h_2^2 R(h_1, 6) - h_1^2 R(h_2, 6)}{h_2^2 - h_1^2}. \end{aligned}$$

Somit ist die aus den (berechenbaren) Quadraturapproximationen $T(h_1), T(h_2)$ gebildete Approximation

$$\frac{h_1^2 T(h_2) - h_2^2 T(h_1)}{h_1^2 - h_2^2} = \alpha_0 - \alpha_2 h_1^2 h_2^2 + \frac{h_1^2 h_2^2}{h_1^2 - h_2^2} (R(h_2, 4) - R(h_1, 4)).$$

eine verbesserte Approximation an den Wert des Integrals.

- Speziell für die Wahl $h_1 = h$ und $h_2 = \frac{h}{2}$ erhält man

$$\begin{aligned} \frac{4T(\frac{h}{2}) - T(h)}{3} &= \alpha_0 - \frac{1}{4} \alpha_2 h^4 + \frac{1}{3} h^2 (R(\frac{h}{2}, 4) - R(h, 4)), \\ \frac{4T(\frac{h}{2}) - T(h)}{3} - \alpha_0 &= \mathcal{O}(h^4). \end{aligned}$$

Vgl. **Illustration** (Extrapolation).

- Im **allgemeinen Fall** werden die Approximationen zu K paarweise verschiedenen Schrittweiten

$$T(h_i), \quad 1 \leq i \leq K,$$

geeignet kombiniert. Eine elegante Lösung mittels Polynominterpolation geht auf Romberg zurück (vgl. Satz 3.2). Zur praktischen Durchführung verwendet man das Schema von Aitken–Neville (vgl. Schema der dividierten Differenzen, vgl. Algorithmus zur Romberg-Quadratur, Skriptum, S. 49).

Vgl. **Illustration** (Extrapolation).

– Eine übliche **Wahl der Schrittweiten** ist

$$h_1 = h, \quad h_{i+1} = \frac{1}{2} h_i = \frac{1}{2^{i-1}} h, \quad 1 \leq i \leq K,$$

oder auch die **Bulirsch-Folge**

$$h_1 = h, \quad h_i = \frac{1}{i} h, \quad 2 \leq k \leq K.$$

Bemerkung: Das Interpolationspolynom durch einfache Stützstellen $(x_i)_{0 \leq i \leq n}$ und zugehörige Stützwerte $y_i = f(x_i)$ für $0 \leq i \leq n$ ist durch

$$p = \sum_{i=0}^n y_i L_i \in \mathbb{P}_n, \quad L_i(x) = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j}, \quad 0 \leq i \leq n,$$

gegeben. Im Folgenden werden die Bezeichnungen P (Interpolationspolynom) statt p (Ordnung der Quadraturformel) sowie Indizes $1 \leq k \leq K$ statt $0 \leq i \leq n$ verwendet. Die Darstellung des Interpolationspolynoms zu Datenpunkten $(x_i, y_i)_{1 \leq i \leq K}$ lautet dann

$$P = \sum_{i=1}^K y_i L_i \in \mathbb{P}_{K-1}, \quad L_i(x) = \prod_{\substack{1 \leq j \leq K \\ j \neq i}} \frac{x - x_j}{x_i - x_j}, \quad 1 \leq i \leq K.$$

Extrapolation (Satz 3.2): Es sei $f \in \mathcal{C}^{2K}([a, b])$, und es bezeichne $P \in \mathbb{P}_{K-1}$ das Interpolationpolynom zu den Datenpunkten $(h_i^2, T(h_i))_{1 \leq i \leq K}$ mit positiven und paarweise verschiedenen Schrittweiten $h_i > 0$ für $1 \leq i \leq k$. Dann gilt

$$P(0) - \alpha_0 = \mathcal{O}(h_{\max}^{2K}), \quad \alpha_0 = \int_a^b f(x) dx, \quad h_{\max} = \max_{1 \leq i \leq K} h_i.$$

Denn: Die explizite Darstellung des Interpolationspolynoms $P \in \mathbb{P}_{K-1}$ durch die Datenpunkte $(h_i^2, T(h_i))_{1 \leq i \leq K}$ mittels Lagrange-Basispolynomen lautet

$$P(x) = \sum_{i=1}^K T(h_i) L_i(x) \in \mathbb{P}_{K-1}, \quad L_i(x) = \prod_{\substack{1 \leq j \leq K \\ j \neq i}} \frac{x - h_j^2}{h_i^2 - h_j^2}, \quad 1 \leq i \leq K.$$

Da die Polynomfunktionen $Q_k(x) = x^k$ für $0 \leq k \leq K - 1$ exakt durch das Interpolationspolynom zu den Stützstellen $(h_i^2)_{1 \leq i \leq K}$ dargestellt werden, gilt

$$x^k = Q_k(x) = \sum_{j=1}^K Q(h_j^2) L_j(x) = \sum_{j=1}^K h_j^{2k} L_j(x), \quad 0 \leq k \leq K - 1.$$

Einsetzen der asymptotischen Entwicklung für $T(h)$ führt auf

$$\begin{aligned}
 P(x) &= \sum_{i=1}^K T(h_i) L_i(x) \\
 &= \sum_{i=1}^K \left(\sum_{k=0}^{K-1} \alpha_k h_i^{2k} + \alpha_K(h_i) h_i^{2K} \right) L_i(x) \\
 &= \sum_{k=0}^{K-1} \alpha_k \underbrace{\sum_{i=1}^K h_i^{2k} L_i(x)}_{=x^k} + \sum_{i=1}^K \alpha_K(h_i) h_i^{2K} L_i(x) \\
 &= \sum_{k=0}^{K-1} \alpha_k x^k + \sum_{i=1}^K \alpha_K(h_i) h_i^{2K} L_i(x).
 \end{aligned}$$

Extrapolation bei $x = 0$ ergibt (Auswerten des Interpolationspolynoms an Argumenten außerhalb des betrachteten Intervalles)

$$P(0) = \alpha_0 + \sum_{i=1}^K \alpha_K(h_i) h_i^{2K} L_i(0), \quad L_i(0) = \prod_{\substack{1 \leq j \leq K \\ j \neq i}} \frac{h_j^2}{h_j^2 - h_i^2}, \quad 1 \leq i \leq K,$$

und somit die Abschätzung (Schranke für α_K , positive Schrittweiten $h_i > 0$ für $1 \leq i \leq K$)

$$\begin{aligned}
 |P(0) - \alpha_0| &\leq \sum_{i=1}^K |\alpha_K(h_i)| h_i^{2K} \prod_{\substack{1 \leq j \leq K \\ j \neq i}} \frac{h_j^2}{|h_j^2 - h_i^2|} \\
 &\leq h_{\max}^{2K} \cdot C \int_a^b |f^{(2K)}(x)| dx \sum_{i=1}^K \prod_{\substack{1 \leq j \leq K \\ j \neq i}} \frac{h_j^2}{|h_j^2 - h_i^2|}.
 \end{aligned}$$

Dies zeigt die Behauptung. \diamond

3.4. Adaptive Verfahren

- **Vorsicht!** In Abschnitt 3.3 wurden Schrittweiten h_1, \dots, h_K zu äquidistanten Zerlegungen betrachtet. Im Folgenden geht es um die optimale Wahl der Schrittweiten einer einzigen Zerlegung, und h_j bezeichnet die Schrittweite im j -ten Teilintervall.
- **Adaptivität:**
 - Das Grundprinzip adaptiver Verfahren ist es, eine Balance zwischen **Genauigkeit und Effizienz** zu erreichen.
 - Im Zusammenhang mit Numerischer Integration ist es wünschenswert, in Bereichen wo der **Integrand wenig variiert relativ große Schrittweiten** zuzulassen und in Bereichen wo der **Integrand hingegen stark variiert relativ kleine Schrittweiten** zu wählen. Aufgrund zusätzlicher Funktionsauswertungen und Rechenoperationen hat eine Verkleinerung der Schrittweite eine Verringerung der Effizienz des Verfahrens zur Folge, eine Vergrößerung der Schrittweiten steigert die Effizienz.Vgl. Verlauf des Integranden, Skriptum, S. 50.

- **Adaptive Verfahren zur numerischen Integration** (Trapezregel)

- Es bezeichnet $a = x_0 < \dots < x_N = b$ eine Zerlegung des Integrationsintervalles mit zugehörigen Schrittweiten $h_j = x_{j+1} - x_j$ für $0 \leq j \leq N - 1$. Beispielsweise für die Trapezregel ist die Quadraturapproximation durch

$$Q(f) = \frac{1}{2} \sum_{j=0}^{N-1} h_j (f(x_j) + f(x_{j+1})) \approx I(f) = \int_a^b f(x) dx$$

gegeben.

- Für den Verfahrensfehler der Trapezregel gilt die Abschätzung

$$|Q(f) - I(f)| \leq C (b - a) h_{\max}^p \|f^{(p)}\|_{\infty}, \quad h_{\max} = \max_{0 \leq j \leq N-1} h_j.$$

Die Vorgabe einer sehr kleinen maximalen Schrittweite würde zwar auf eine sehr gute Approximation führen

$$h_{\max} \ll 1 \Rightarrow |Q(f) - I(f)| \ll 1,$$

wäre jedoch sehr ineffizient. Das Ziel adaptiver Verfahren ist es sicherzustellen, daß die Schrittweiten $(h_j)_{0 \leq j \leq N-1}$ so gewählt werden, daß die Approximation eine vorgegebene Toleranz erreicht

$$|Q(f) - I(f)| \leq \text{TOL}.$$

Vgl. **Beispiel**, Skriptum, S. 58

$$\int_{-1}^1 \frac{1}{10^{-4} + x^2} dx.$$

– **Grundlegende Ideen:**

- * Ausgehend von einer groben Zerlegung des Intervalles (z.B. ein oder zwei Teilintervalle) bestimmt man jenes Teilintervall $I_j = [x_j, x_{j+1}]$ in welchem der (absolute oder relative) geschätzte Verfahrensfehler

$$Q_j(f) = \frac{1}{2} h_j (f(x_j) + f(x_{j+1})) \approx I_j(f) = \int_{x_j}^{x_{j+1}} f(x) dx$$

maximal ist. Solange die Forderung (Schätzung $\tilde{I} \approx I$)

$$|Q(f) - \tilde{I}(f)| \leq \text{TOL}$$

verletzt ist, wird das Teilintervall I_j halbiert und die zugehörigen Quadraturapproximationen berechnet (Berechnung der zusätzlich benötigten Funktionswerte, Berechnung der Quadraturapproximationen sowie der geschätzten Verfahrensfehler auf den beiden Teilintervallen).

- * Zur Schätzung des Verfahrensfehlers verwendet man beispielsweise eine Quadraturformel höherer Ordnung wie etwa die Simpsonregel, die wenig zusätzliche Funktionsauswertungen erfordert. Eine Alternative ist auch Richardson Extrapolation.
- * Bei praktischen Berechnungen verwendet man die bessere Approximation (Unterschätzung der Schrittweite).

Vgl. **Illustration** Adaptive Verfahren (Trapezregel, Simpsonregel, ohne Schätzung des Verfahrensfehlers).

4. Anfangswertprobleme für gewöhnliche Differentialgleichungen

- **Inhalte:**

- Grundlegende Begriffe und Resultate zur Theorie von Anfangswertproblemen für gewöhnliche Differentialgleichungen
- Prinzip der Diskretisierung, Verfahrensklassen, Diskretisierungsfehler
- Konvergenzresultat, Adaptivität
- A-Stabilität von Lösungsverfahren, Steife Differentialgleichungen

- **Grundlagen:**

- Quadraturapproximationen, Interpolation
- Lösungsverfahren für lineare und nichtlineare Gleichungssysteme

Weitere Anwendungen:

- Zeitdiskretisierungen partieller Differentialgleichungen

- **Bemerkung:** Die Analyse des Diskretisierungsfehlers unterscheidet sich von der im Skriptum gewählten Vorgehensweise.

4.1. Theoretischer Hintergrund

- **Anfangswertprobleme für gewöhnliche Differentialgleichungen erster Ordnung:** Für vorgegebene Punkte $t_0 \in \mathbb{R}$ (**Anfangszeitpunkt**) und $T \in \mathbb{R}$ (**Endzeitpunkt**) sei $I = [t_0, T]$ falls $t_0 < T$ (bzw. $I = [T, t_0]$ falls $t_0 > T$). Weiters sei $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d : (t, v) \mapsto f(t, v)$ eine vorgegebene stetige Funktion und $y_0 \in \mathbb{R}^d$ ein vorgegebener **Anfangswert** bei t_0 . Eine Lösung des **Anfangswertproblems**

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), & \text{(bzw. } (T, t_0)) \\ y(t_0) = y_0, \end{cases}$$

ist eine Funktion $y : I \rightarrow \mathbb{R}^d$, welche die (gewöhnliche) **Differentialgleichung** (erster Ordnung) und die **Anfangsbedingung** erfüllt. Dabei wird vorausgesetzt, daß y auf I stetig und zumindest in (t_0, T) (bzw. (T, t_0)) differenzierbar ist.

Beispiel, vgl. Skriptum, S. 60 ($d = 2$, nichtautonome nichtlineare Differentialgleichung).

Bemerkungen:

- Falls $d > 1$ spricht man auch von einem **Differentialgleichungssystem**.
- Im Gegensatz zu **gewöhnlichen Differentialgleichungen** für Funktionen in einer Veränderlichen geben **partielle Differentialgleichungen** Zusammenhänge zwischen Funktionen in mehreren Veränderlichen und deren partiellen Ableitungen an.
- In vielen Anwendungssituationen beschreibt die Variable t die Zeit und die Funktion y den zeitabhängigen Zustand eines Systems (z.B. das Wachstum einer Spezies oder die Bahn eines Massenpunktes unter der Einwirkung von Kräften). Zum Zeitpunkt t_0 befindet sich das System in dem bekannten Zustand $y(t_0) = y_0$. Zu bestimmen sind die Zustände $y(t)$ zu späteren (bzw. früheren) Zeitpunkten $t \in (t_0, T)$ (bzw. $t \in (T, t_0)$).
- Die (komponentenweise) Integration der Differentialgleichung

$$\begin{aligned} \frac{d}{dt} y(t) &= y'(t) = f(t, y(t)), & t \in (t_0, T), \\ \int_{t_0}^t y'(\tau) d\tau &= \int_{t_0}^t f(\tau, y(\tau)) d\tau, & t \in (t_0, T), \\ y(t) - y(t_0) &= \int_{t_0}^t f(\tau, y(\tau)) d\tau, & t \in (t_0, T), \end{aligned}$$

und Einsetzen der Anfangsbedingung $y(t_0) = y_0$ führt auf die **Integralgleichung**

$$y(t) = y(t_0) + \int_{t_0}^t f(\tau, y(\tau)) d\tau, \quad t \in (t_0, T).$$

Diese wird einerseits für theoretische Untersuchungen (**Existenzresultate, Picard–Lindelöf**) und andererseits zur **Konstruktion numerischer Verfahren** (u.a. mittels Quadraturapproximationen bzw. Interpolation) genutzt.

- Eine vorteilhafte Alternative, die eine (semilineare) Formulierung der Differentialgleichung mit zeitunabhängiger Matrix $A \in \mathbb{R}^{d \times d}$ nützt, beruht auf der **linearen Variation-der-Konstanten Formel** (Nachprüfen durch Einsetzen bei t_0 und Differenzieren)

$$\begin{aligned} \frac{d}{dt} y(t) &= A y(t) + g(t, y(t)), \quad t \in (t_0, T), \\ y(t) &= e^{(t-t_0)A} y(t_0) + \int_{t_0}^t e^{(t-\tau)A} g(\tau, y(\tau)) d\tau, \quad t \in (t_0, T). \end{aligned}$$

- Im Zusammenhang mit Anwendungen (z.B. aus der Mechanik) haben auch **Differential-Algebraische Gleichungen** besondere Bedeutung

$$\left\{ \begin{array}{l} \left(\begin{array}{c} \frac{d}{dt} y(t) \\ 0 \end{array} \right) = \left(\begin{array}{c} f(t, y(t), z(t)) \\ g(t, y(t), z(t)) \end{array} \right), \quad t \in (t_0, T), \\ \left(\begin{array}{c} y(t_0) \\ z(t_0) \end{array} \right) = \left(\begin{array}{c} y_0 \\ z_0 \end{array} \right), \end{array} \right.$$

die in der allgemeineren **impliziten Form** enthalten sind

$$\left\{ \begin{array}{l} F(t, Y(t), \frac{d}{dt} Y(t)) = 0, \quad t \in (t_0, T), \\ Y(t_0) = Y_0. \end{array} \right.$$

Falls die Funktion F bzgl. des Argumentes $\frac{d}{dt} Y(t)$ auflösbar ist, ergibt sich eine explizite Differentialgleichung der oben angegebenen Form.

- **Autonome Differentialgleichungen:** Falls die die Differentialgleichung erster Ordnung definierende Funktion f nicht explizit von der Variable t abhängt, heißt die Differentialgleichung eine **autonome Differentialgleichung**

$$\left\{ \begin{array}{l} \frac{d}{dt} y(t) = f(y(t)), \quad t \in (t_0, T), \\ y(t_0) = y_0. \end{array} \right.$$

In diesem Fall kann man ohne Einschränkung der Allgemeinheit die Anfangszeit $t_0 = 0$ annehmen.

Reduktion auf autonome Differentialgleichungen: Jedes Anfangswertproblem für eine nichtautonome Differentialgleichung

$$\left\{ \begin{array}{l} \frac{d}{dt} y(t) = f(t, y(t)), \quad t \in (t_0, T), \\ y(t_0) = y_0, \end{array} \right.$$

kann mittels der Definitionen

$$Y(t) = \begin{pmatrix} t \\ y(t) \end{pmatrix}, \quad Y_0 = \begin{pmatrix} t_0 \\ y_0 \end{pmatrix}, \quad F(Y(t)) = \begin{pmatrix} 1 \\ f(t, y(t)) \end{pmatrix},$$

auf ein Anfangswertproblem für eine autonome Differentialgleichung zurückgeführt werden

$$\begin{cases} \frac{d}{dt} Y(t) = F(Y(t)), & t \in (t_0, T), \\ Y(t_0) = Y_0. \end{cases}$$

- **Reduktion auf Differentialgleichungen erster Ordnung:** Jedes Anfangswertproblem für eine Differentialgleichung k -ter Ordnung

$$\begin{cases} \frac{d^k}{dt^k} y(t) = f(t, y(t), \frac{d}{dt} y(t), \dots, \frac{d^{k-1}}{dt^{k-1}} y(t)), & t \in (t_0, T), \\ y(t_0) = y_0, \quad \frac{d}{dt} y(t_0) = y'_0, \quad \dots \quad \frac{d^{k-1}}{dt^{k-1}} y(t_0) = y_0^{(k-1)}, \end{cases}$$

kann mittels der Definitionen

$$Y(t) = \begin{pmatrix} Y_1(t) \\ Y_2(t) \\ \vdots \\ Y_{k-1}(t) \end{pmatrix} = \begin{pmatrix} y(t) \\ \frac{d}{dt} y(t) \\ \vdots \\ \frac{d^{k-1}}{dt^{k-1}} y(t) \end{pmatrix}, \quad Y_0 = \begin{pmatrix} y_0 \\ y'_0 \\ \vdots \\ y_0^{(k-1)} \end{pmatrix},$$

$$F(t, Y(t)) = \begin{pmatrix} Y_2(t) \\ \vdots \\ Y_{k-1}(t) \\ f(t, y(t), \frac{d}{dt} y(t), \dots, \frac{d^{k-1}}{dt^{k-1}} y(t)) \end{pmatrix} = \begin{pmatrix} Y_2(t) \\ \vdots \\ Y_{k-1}(t) \\ f(t, Y(t)) \end{pmatrix},$$

auf ein Anfangswertproblem für eine Differentialgleichung erster Ordnung zurückgeführt werden

$$\begin{cases} \frac{d}{dt} Y(t) = F(t, Y(t)), & t \in (t_0, T), \\ Y(t_0) = Y_0. \end{cases}$$

Beispiel ($d = 2$, autonome lineare Differentialgleichung), vgl. Skriptum, S. 62. Eine Transformation ($Y_1 = y, Y_2 = \frac{d}{dt} y$) der Schwingungsgleichung (Newtonsche Bewegungsgleichung, Masse m , Auslenkung aus der Ruhelage y , Geschwindigkeit $\frac{d}{dt} y$, Beschleunigung $\frac{d^2}{dt^2} y$, Federkraft nach dem Hookschen Gesetz mit Federkonstante k , Reibungskraft durch Dämpfung z.B. in zäher Flüssigkeit mit Reibungskoeffizient r , Vorgabe der Anfangsauslenkung und Anfangsgeschwindigkeit) führt auf

$$m \frac{d^2}{dt^2} y(t) + r \frac{d}{dt} y(t) + k y(t) = 0,$$

$$\frac{d}{dt} \begin{pmatrix} Y_1(t) \\ Y_2(t) \end{pmatrix} = \begin{pmatrix} Y_2(t) \\ -\frac{r}{m} Y_2(t) - \frac{k}{m} Y_1(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{r}{m} \end{pmatrix} \begin{pmatrix} Y_1(t) \\ Y_2(t) \end{pmatrix}.$$

Vgl. **Illustration** (Lösung der Schwingungsgleichung mittels Matrixexponentialfunktion).

- **Lineare Differentialgleichungen:** Falls die rechte Seite der Funktion die spezielle Form $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d : (t, v) \mapsto f(t, v) = A(t)v + g(t)$ mit zeitabhängiger Matrix $A : I \rightarrow \mathbb{R}^{d \times d}$ und **Inhomogenität** $g : I \rightarrow \mathbb{R}^d$ hat (d.h. die Funktion f ist insbesondere linear bzgl. der Variable v), heißt die Differentialgleichung eine **inhomogene lineare Differentialgleichung** (bzw. eine **homogene** lineare Differentialgleichung, falls speziell $g = 0$)

$$\begin{cases} \frac{d}{dt} y(t) = A(t)y(t) + g(t), & t \in (t_0, T), \\ y(t_0) = y_0. \end{cases}$$

Ähnlich wie bei Gleichungssystemen ist auch bei Differentialgleichungen der nichtlineare Fall wesentlich schwieriger als der lineare Fall.

- **Vorbemerkung:** Eine Funktion $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ heißt **Lipschitz-stetig**, wenn eine Konstante $L > 0$ existiert, sodaß für alle Elemente $x, \tilde{x} \in D$ die folgende Abschätzung mit **Lipschitz-Konstante** $L > 0$ gilt

$$\|f(x) - f(\tilde{x})\| \leq L \|x - \tilde{x}\|.$$

Resultat zur Existenz und Eindeutigkeit (Satz von Picard–Lindelöf): Falls die Funktion $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d : (t, v) \mapsto f(t, v)$ stetig und bezüglich der Variable v Lipschitz-stetig ist, d.h. für alle $t \in I$ und $v, \tilde{v} \in \mathbb{R}^d$ gilt die Relation

$$\|f(t, v) - f(t, \tilde{v})\| \leq L \|v - \tilde{v}\|$$

mit Konstante $L > 0$, existiert eine eindeutig bestimmte stetig differenzierbare Funktion $y : I \rightarrow \mathbb{R}^d$, welche das Anfangswertproblem

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ y(t_0) = y_0, \end{cases}$$

erfüllt.

Bemerkung: Eine Abschwächung des Satzes von Picard–Lindelöf nützt die **lokale Lipschitz-Stetigkeit** in einer Umgebung der Anfangsbedingung (t_0, y_0) und sichert die **Existenz und Eindeutigkeit einer lokalen Lösung** $y : [t_0, t_0 + \delta] \rightarrow \mathbb{R}^d$.

Beispiel: Ein bekanntes Beispiel einer Funktion, die insbesondere bei $x = 0$ nicht (lokal) Lipschitz-stetig ist, ist die Abbildung

$$f : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto f(x) = \sqrt{x}, \quad \frac{f(x) - f(0)}{x - 0} = \frac{1}{\sqrt{x}}.$$

Zudem gilt $f : \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R} : x \mapsto \frac{1}{2\sqrt{x}}$ wegen

$$\frac{f(x+\Delta x) - f(x)}{\Delta x} = \frac{\sqrt{x+\Delta x} - \sqrt{x}}{\Delta x} = \frac{1}{\sqrt{x+\Delta x} + \sqrt{x}} \xrightarrow{\Delta x \rightarrow 0} f'(x) = \frac{1}{2\sqrt{x}}.$$

Erweitern mit $\sqrt{x+\Delta x} + \sqrt{x}$,
Verwendung von $(a-b)(a+b) = a^2 - b^2$

- Wesentlich in Hinblick auf die Theorie dynamischer Systeme ist das folgende Resultat zur Sensitivität der Lösung eines Anfangswertproblems bezüglich Änderungen in den Anfangswerten.

Stetige Abhängigkeit der Lösung von den Anfangswerten (Kondition, vgl. Satz 4.2): Falls die Funktion $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d : (t, v) \mapsto f(t, v)$ stetig und bezüglich der Variable v Lipschitz-stetig mit Konstante $L > 0$ ist, gilt für zwei Lösungen y, \tilde{y} der Differentialgleichung

$$\frac{d}{dt} z(t) = f(t, z(t)), \quad t \in (t_0, T),$$

zu den Anfangswerten y_0, \tilde{y}_0 die Abschätzung

$$\|y(t) - \tilde{y}(t)\| \leq e^{L|t-t_0|} \|y_0 - \tilde{y}_0\|, \quad t \in (t_0, T).$$

Bemerkung: Das exponentielle Auseinanderdriften von Lösungen zeigt sich beispielsweise bei der folgenden skalaren Differentialgleichung mit $\lambda \geq 0$

$$\frac{d}{dt} z(t) = \lambda z(t), \quad z(t) = e^{\lambda(t-t_0)} z(t_0), \quad t \geq t_0.$$

4.2. Diskretisierungen und Diskretisierungsfehler

- **Zeitintegrationsverfahren** (Zeitdiskretisierungsverfahren) für Anfangswertprobleme der Form

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ y(t_0) \text{ gegeben,} \end{cases}$$

beruhen auf folgendem Zugang (**time-stepping approach**). Ausgehend von einem näherungsweise Anfangswert

$$y_0 \approx y(t_0)$$

berechnet man an gewissen Zeitpunkten mit zugehörigen Zeitschrittweiten

$$t_0 < t_1 < \dots < t_N = T, \quad h_i = t_{i+1} - t_i, \quad 0 \leq i \leq N-1,$$

mittels einer Rekursion Näherungswerte an die exakten Lösungswerte

$$y_n \approx y(t_n), \quad 1 \leq n \leq N.$$

Bei **Einschrittverfahren** hat die Rekursion die Form

$$y_n = \Phi(h_{n-1}, t_{n-1}, y_{n-1}), \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben.}$$

Zur Konstruktion und Analyse von Zeitintegrationsverfahren für (lineare) Differentialgleichungen ist es vorteilhaft, den numerischen Lösungsoperator Φ mit dem exakten Lösungsoperator E zu vergleichen

$$y_n = \Phi(h_{n-1}, t_{n-1}, y_{n-1}) \approx y(t_n) = E(h_{n-1}, t_{n-1}, y(t_{n-1})), \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben.}$$

Allgemeiner hängt bei **Mehrschrittverfahren** (k -Schrittverfahren) der numerische Lösungsoperator von zuvor berechneten Approximationen ab

$$y_n = \Phi(h_{n-1}, \dots, h_{n-k}, t_{n-1}, y_{n-1}, \dots, y_{n-k}), \quad k \leq n \leq N.$$

Neben dem Startwert sind dann geeignete Approximationen y_1, \dots, y_{k-1} (z.B. mittels eines Einschrittverfahrens) zu berechnen.

- **Beispiele für (einfache) Zeitintegrationsverfahren:**

- **Explizites Eulerverfahren** (*Explicit Euler, Forward Euler*):

- * Das explizite Eulerverfahren beruht auf der Idee, in der Differentialgleichung (Auswerten bei $t = t_{n-1}$)

$$\left. \frac{d}{dt} y(t) \right|_{t=t_{n-1}} = f(t_{n-1}, y(t_{n-1})), \quad 1 \leq n \leq N,$$

den Differentialquotienten durch den Differenzenquotienten (Vorwärtsdifferenz, *forward*)

$$\frac{y(t_n) - y(t_{n-1})}{t_n - t_{n-1}} \approx \left. \frac{d}{dt} y(t) \right|_{t=t_{n-1}} = \lim_{t \rightarrow t_{n-1}} \frac{y(t) - y(t_{n-1})}{t - t_{n-1}}$$

zu ersetzen. Dies führt auf die Rekursion (explizite Relation)

$$y_n = y_{n-1} + h_{n-1} f(t_{n-1}, y_{n-1}), \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben.}$$

- * Eine alternative Herleitung des expliziten Eulerverfahrens beruht auf der Integralgleichung (Integration der Differentialgleichung mit Integrationsintervall $[t_{n-1}, t_n]$)

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(\tau, y(\tau)) d\tau,$$

und Polynominterpolation (Grad 0, Stützstelle bei t_{n-1})

$$y(t_n) = y(t_{n-1}) + h_{n-1} f(t_{n-1}, y(t_{n-1})).$$

- * Vgl. **Skizze**, Skriptum, S.66.

– **Implizites Eulerverfahren** (*Implicit Euler, Backward Euler*):

- * Das implizite Eulerverfahren beruht auf der Idee, in der Differentialgleichung (Auswerten bei $t = t_n$)

$$\frac{d}{dt} y(t) \Big|_{t=t_n} = f(t_n, y(t_n)), \quad 1 \leq n \leq N,$$

den Differentialquotienten durch den Differenzenquotienten (Rückwärtsdifferenz, *backward*)

$$\frac{y(t_n) - y(t_{n-1})}{t_n - t_{n-1}} \approx \frac{d}{dt} y(t) \Big|_{t=t_n} = \lim_{t \rightarrow t_n} \frac{y(t) - y(t_n)}{t - t_n}$$

zu ersetzen. Dies führt auf die Rekursion

$$y_n = y_{n-1} + h_{n-1} f(t_n, y_n), \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben.}$$

- * Im Gegensatz zum expliziten Eulerverfahren ist beim impliziten Eulerverfahren in jedem Zeitschritt ein **nichtlineares Gleichungssystem** zu lösen (vgl. Numerische Mathematik I). Aufgrund des besseren Stabilitätsverhaltens, lohnt sich im Zusammenhang mit **steifen Differentialgleichungen** (Abschnitt 4.4) der Mehraufwand.

- * Eine alternative Herleitung des impliziten Eulerverfahrens beruht auf der Integralgleichung (Integration der Differentialgleichung mit Integrationsintervall $[t_{n-1}, t_n]$)

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(\tau, y(\tau)) d\tau,$$

und Polynominterpolation (Grad 0, Stützstelle bei t_n)

$$y(t_n) = y(t_{n-1}) + h_{n-1} f(t_n, y(t_n)).$$

– **Mittelpunktsregeln** (*Einschrittverfahren, Zweischrittverfahren*):

- * Eine Herleitung einer Mittelpunktsregel beruht ebenfalls auf der Integralgleichung (Integration der Differentialgleichung mit Integrationsintervall $[t_{n-1}, t_n]$)

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(\tau, y(\tau)) d\tau,$$

und Polynominterpolation (Grad 1, Mittelpunkt $t_{n-1} + \frac{1}{2} h_{n-1}$, vgl. entsprechende Quadraturapproximation)

$$y(t_n) = y(t_{n-1}) + h_{n-1} f\left(t_{n-1} + \frac{1}{2} h_{n-1}, y\left(t_{n-1} + \frac{1}{2} h_{n-1}\right)\right).$$

Zur Approximation des unbekanntes Lösungswertes kann man einerseits das explizite Eulerverfahren verwenden

$$y\left(t_{n-1} + \frac{1}{2} h_{n-1}\right) \approx y_{n-1} + \frac{1}{2} h_{n-1} f(t_{n-1}, y_{n-1})$$

und erhält dann die Rekursion für die **(explizite) Mittelpunktsregel**

$$y_n = y_{n-1} + h_{n-1} f\left(t_{n-1} + \frac{1}{2} h_{n-1}, y_{n-1} + \frac{1}{2} h_{n-1} f(t_{n-1}, y_{n-1})\right), \quad 1 \leq n \leq N.$$

Verwendet man hingegen das implizite Eulerverfahren (Lösung eines nicht-linearen Gleichungssystems zur Berechnung der Hilfsapproximation $Y_{n-1,1}$)

$$y\left(t_{n-1} + \frac{1}{2} h_{n-1}\right) \approx Y_{n-1,1} = y_{n-1} + \frac{1}{2} h_{n-1} f\left(t_{n-1} + \frac{1}{2} h_{n-1}, Y_{n-1,1}\right),$$

ergibt sich die **(implizite) Mittelpunktsregel**

$$Y_{n-1,1} = y_{n-1} + \frac{1}{2} h_{n-1} f\left(t_{n-1} + \frac{1}{2} h_{n-1}, Y_{n-1,1}\right),$$

$$y_n = y_{n-1} + h_{n-1} f\left(t_{n-1} + \frac{1}{2} h_{n-1}, Y_{n-1,1}\right), \quad 1 \leq n \leq N, \quad y_0 \text{ geg.}$$

- * **Bemerkung:** Die implizite Mittelpunktsregel ist ein erstes Beispiel eines impliziten **Runge-Kutta Verfahrens** (Definition, s.u.), das sich speziell für die Wahl der folgenden Verfahrenskoeffizienten ergibt

$$s = 1, \quad c_1 = \frac{1}{2}, \quad a_{11} = \frac{1}{2}, \quad b_1 = 1.$$

- * Betrachtet man die Integralgleichung (Integration der Differentialgleichung mit Integrationsintervall $[t_{n-2}, t_n]$, zur Vereinfachung konstante Zeitschrittweite h)

$$y(t_n) = y(t_{n-2}) + \int_{t_{n-2}}^{t_n} f(\tau, y(\tau)) d\tau,$$

und Polynominterpolation (Grad 1, Mittelpunkt t_{n-1} , vgl. entsprechende Quadraturapproximation)

$$y(t_n) = y(t_{n-2}) + 2h f(t_{n-1}, y(t_{n-1})),$$

so ergibt sich ein **explizites Zweischrittverfahren** (vgl. Skriptum, S. 67)

$$y_n = y_{n-2} + 2h f(t_{n-1}, y_{n-1}), \quad 2 \leq n \leq N, \quad y_0 \text{ gegeben.}$$

Zur Bestimmung der unbekanntes Approximation y_1 verwendet man üblicherweise ein Einschrittverfahren (z.B. explizites Eulerverfahren mit hinreichend kleiner Zeitschrittweite zur Erhaltung der Konvergenzordnung).

- **Trapezregel:** Die Trapezregel beruht auf der Integralgleichung (Integration der Differentialgleichung mit Integrationsintervall $[t_{n-1}, t_n]$)

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(\tau, y(\tau)) d\tau,$$

und Polynominterpolation (Grad 1, Intervallenden t_{n-1} und t_n als Stützstellen, vgl. entsprechende Quadraturapproximation)

$$y(t_n) = y(t_{n-1}) + \frac{1}{2} h_{n-1} \left(f(t_{n-1}, y(t_{n-1})) + f(t_n, y(t_n)) \right).$$

Dies führt auf die Rekursion für die Trapezregel

$$y_n = y_{n-1} + \frac{1}{2} h_{n-1} \left(f(t_{n-1}, y_{n-1}) + f(t_n, y_n) \right), \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben.}$$

Die Trapezregel ist ebenfalls ein Beispiel eines (impliziten) Runge–Kutta Verfahrens (s.u.).

- **Klassifizierung von gebräuchlichen Zeitintegrationsverfahren (Runge–Kutta Verfahren, Lineare Mehrschrittverfahren):**

- Ein **Runge–Kutta Verfahren** besitzt die Form

$$\begin{cases} Y'_{n-1,i} = f(t_{n-1} + c_i h_{n-1}, Y_{n-1,i}), \\ Y_{n-1,i} = y_{n-1} + h_{n-1} \sum_{j=1}^s a_{ij} Y'_{n-1,j}, \end{cases}$$

$$y_n = y_{n-1} + h_{n-1} \sum_{i=1}^s b_i Y'_{n-1,i}, \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben,}$$

mit Hilfsapproximationen $(Y_{n-1,i})_{1 \leq i \leq s}$ und $(Y'_{n-1,i})_{1 \leq i \leq s}$ (**Stufen**) an die Funktionswerte bzw. Ableitungen der exakten Lösung $Y_{n-1,i} \approx y(t_{n-1} + c_i h_{n-1})$ bzw. $Y'_{n-1,i} \approx y'(t_{n-1} + c_i h_{n-1}) = f(t_{n-1} + c_i h_{n-1}, y(t_{n-1} + c_i h_{n-1}))$. Ersetzt man die internen Stufen, so ergibt sich die alternative Darstellung

$$Y'_{n-1,i} = f\left(t_{n-1} + c_i h_{n-1}, y_{n-1} + h_{n-1} \sum_{j=1}^s a_{ij} Y'_{n-1,j}\right),$$

$$y_n = y_{n-1} + h_{n-1} \sum_{i=1}^s b_i Y'_{n-1,i}, \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben.}$$

Ein Runge–Kutta Verfahren (und damit die Approximationsgüte der Näherungslösung) ist durch die Angabe der Verfahrenskoeffizienten im (John) **Butcher Tableau**

$$\frac{c = (c_i)_{1 \leq i \leq s} \quad \bigg| \quad A = (a_{ij})_{1 \leq i, j \leq s}}{b = (b_i)_{1 \leq i \leq s}}$$

festgelegt. Das Verfahren ist **explizit**, falls $a_{ij} = 0$ für alle $1 \leq i \leq j \leq s$, ansonsten **implizit**.

- **Lineare Mehrschrittverfahren** benützen die Kenntnis von Approximationen zu früheren Zeitpunkten. Ein lineares k -Schrittverfahren ist durch eine Rekursion der Form (zur Vereinfachung für konstante Zeitschrittweiten h formuliert, bei variablen Schrittweiten hängen die Koeffizienten des Verfahrens von Schrittweitenverhältnissen ab)

$$\sum_{i=0}^k \alpha_i y_{n-k+i} = h \sum_{i=0}^k \beta_i f(t_{n-k+i}, y_{n-k+i})$$

mit Verfahrenskoeffizienten $(\alpha_i, \beta_i)_{0 \leq i \leq k}$ gegeben. Falls der Koeffizient β_k nicht auftritt (d.h. $\beta_k = 0$), ist das Verfahren **explizit**, ansonsten **implizit**.

Die Konstruktion der bekanntesten Linearen Mehrschrittverfahren beruht auf Interpolation.

- * Bei den **BDF-Verfahren (Backward Differentiation Formulae)** betrachtet man das Interpolationspolynom $P \in \mathbb{P}_k$ durch die Datenpunkte $(t_{n-k+i}, y_{n-k+i})_{0 \leq i \leq k}$ für $k \geq 1$ und bestimmt den unbekanntes Approximationswert y_n so, daß die Forderung

$$\frac{d}{dt} P(t) \Big|_{t=t_n} = f(t_n, P(t_n)) = f(t_n, y_n)$$

erfüllt ist, d.h. das Interpolationspolynom P erfüllt die Differentialgleichung in t_n . Offensichtlich führt dieser Zugang auf ein **implizites k -Schrittverfahren**.

- (i) Für $k = 1$ führt der Ansatz

$$P(t) = y_{n-1} + (t - t_{n-1}) \frac{1}{h} (y_n - y_{n-1}), \quad \frac{d}{dt} P(t) = \frac{1}{h} (y_n - y_{n-1}),$$

und die Forderung

$$\frac{1}{h} (y_n - y_{n-1}) = \frac{d}{dt} P(t) \Big|_{t=t_n} = f(t_n, y_n)$$

auf das **implizite Eulerverfahren**

$$y_n = y_{n-1} + h f(t_n, y_n), \quad 1 \leq n \leq N.$$

- (ii) Für $k = 2$ führt beispielsweise der Ansatz (Interpolation nach Newton, Schema dividierter Differenzen)

$$\begin{aligned} P(t) &= y_{n-2} + (t - t_{n-2}) \frac{1}{h} (y_{n-1} - y_{n-2}) \\ &\quad + (t - t_{n-2})(t - t_{n-1}) \frac{1}{h^2} (y_n - 2y_{n-1} + y_{n-2}), \\ \frac{d}{dt} P(t) &= \frac{1}{h} (y_{n-1} - y_{n-2}) + (2t - t_{n-1} - t_{n-2}) \frac{1}{2h^2} (y_n - 2y_{n-1} + y_{n-2}), \end{aligned}$$

und die Forderung

$$\begin{aligned} & \frac{1}{h} (y_{n-1} - y_{n-2}) + (2t_n - t_{n-1} - t_{n-2}) \frac{1}{2h^2} (y_n - 2y_{n-1} + y_{n-2}) \\ &= \frac{1}{h} (y_{n-1} - y_{n-2}) + \frac{3}{2h} (y_n - 2y_{n-1} + y_{n-2}) \\ &= \frac{1}{2h} (3y_n - 4y_{n-1} + y_{n-2}) \\ &= \frac{d}{dt} P(t) \Big|_{t=t_n} = f(t_n, y_n) \end{aligned}$$

auf das bekannte Zweischrittverfahren **BDF 2** (zuvor angegebene Form mit Koeffizienten $\alpha_0 = \frac{1}{2}$, $\alpha_1 = -2$, $\alpha_2 = \frac{3}{2}$, $\beta_0 = 0 = \beta_1$, $\beta_2 = 1$)

$$\frac{3}{2} y_n - 2y_{n-1} + \frac{1}{2} y_{n-2} = h f(t_n, y_n), \quad 2 \leq n \leq N.$$

* Bei den **Adamsverfahren** betrachtet man die Integralgleichung

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(\tau, y(\tau)) d\tau$$

und ersetzt den Integranden durch ein Interpolationspolynom zu gewissen Datenpunkten. Im Fall der **expliziten Adams-Verfahren** (Adams–Bashforth Verfahren) bestimmt man das Interpolationspolynom $P \in \mathbb{P}_{k-1}$ durch $(t_{n-k+i}, f(t_{n-k+i}, y_{n-k+i}))_{0 \leq i \leq k-1}$, was auf ein explizites k -Schrittverfahren der Form

$$y_n = y_{n-1} + h \sum_{i=0}^{k-1} \beta_i f(t_{n-k+i}, y_{n-k+i}), \quad k \leq n \leq N,$$

führt. Bei den **impliziten Adams-Verfahren** (Adams–Moulton Verfahren) bestimmt man das Interpolationspolynom $P \in \mathbb{P}_k$ durch $(t_{n-k+i}, f(t_{n-k+i}, y_{n-k+i}))_{0 \leq i \leq k}$ (beinhaltet die unbekannt Approximation y_n) und erhält damit ein implizites k -Schrittverfahren der Form

$$y_n = y_{n-1} + h \sum_{i=0}^k \beta_i f(t_{n-k+i}, y_{n-k+i}), \quad k \leq n \leq N.$$

Bemerkung: Zur näherungsweisen Lösung eines nichtlinearen Gleichungssystems kann man die Idee der Fixpunktiteration verwenden. Ausgehend von einem geeigneten Startwert werden Approximationen an eine Lösung des Problems

$$\eta = F(\eta)$$

mittels der Iteration

$$\eta_i = F(\eta_{i-1}), \quad i = 1, 2, \dots$$

berechnet. Im Zusammenhang mit impliziten Adamsverfahren führt dieser Zugang auf **Prädiktor-Korrektor Verfahren**, d.h. zur Bestimmung geeigneter Startwerte wendet man das explizite Adamsverfahren an und berechnet anschließend (einige wenige) Werte mittels Fixpunktiteration.

Illustrationen (Explizite Verfahren der Ordnungen $p = 1, 2, 4$ für eine skalare Testgleichung mit bekannter Lösung, implizite Verfahren der Ordnung $p = 1, 2$ für eine skalare **lineare** Testgleichung, Explizites Zweischrittverfahren der Ordnung $p = 2$).

- **Fragestellung:** Güte der Approximation mittels Zeitintegrationsverfahren.

Vorbemerkung: Zur Vereinfachung werden im Folgenden äquidistante Zeitgitter betrachtet, d.h. es gelte

$$t_n = t_0 + n h, \quad 0 \leq n \leq N, \quad h = \frac{T-t_0}{N}.$$

Ansonsten ist in den globalen Fehlerabschätzungen die konstante Schrittweite h durch die maximale Schrittweite

$$h_{\max} = \max_{0 \leq n \leq N-1} h_n, \quad h_n = t_{n+1} - t_n, \quad 0 \leq n \leq N-1,$$

zu ersetzen.

Globaler und lokaler Fehler:

- Um die Güte der mittels eines Zeitintegrationsverfahrens bestimmten Approximationen an die exakten Lösungswerte

$$y_0, y_1, y_2, \dots, y_N, \quad y_n \approx y(t_n), \quad 0 \leq n \leq N,$$

beurteilen zu können, definiert man den **globalen Fehler**

$$e_N = y_N - y(t_N).$$

Unter der Annahme, daß Anfangswertprobleme betrachtet werden, die durch eine hinreichend reguläre Funktion $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ gegeben sind, heißt ein Zeitintegrationsverfahren **konvergent** mit **Konvergenzordnung** $p \geq 1$, falls eine Abschätzung der Form (für hinreichend kleine Schrittweiten $0 < h \leq \bar{h}$)

$$\|y_N - y(t_N)\| \leq C h^p$$

bzw. in Kurzschreibweise die Relation

$$e_N = \mathcal{O}(h^p)$$

gilt. Wesentlich ist dabei, daß die Konstante C weder von der Schrittweite h noch von der Anzahl der Zeitschritte N abhängt. Bei nichtlinearen Problemen ist die Existenz der numerischen Lösung im Allgemeinen nur für Schrittweiten $0 < h \leq \bar{h}$ sichergestellt.

- Zur Analyse des globalen Verfahrensfehlers (und auch zur Konstruktion von Verfahren) ist es im Allgemeinen vorteilhaft, den **lokalen Fehler** eines Zeitintegrationsverfahrens zu untersuchen. Bei Einschrittverfahren betrachtet man die Differenz zwischen numerischer Approximation und exaktem Lösungswert, ausgehend

von einem gemeinsamen (beliebigen) Anfangswert (zur Vereinfachung der Notation wird der Startzeitpunkt t_0 nicht angegeben)

$$y_1 - y(t_1) = \Phi(h, y_0) - E(h, y_0).$$

Ein Einschrittverfahren hat Konsistenzordnung $p \geq 0$ (im Unterschied zur Konvergenzordnung), wenn

$$y_1 - y(t_1) = \mathcal{O}(h^{p+1}).$$

Bei einem k -Schrittverfahren geht man von einem (beliebigen) Anfangswert y_0 und Startwerten y_1, \dots, y_{k-1} aus, die *exakten* Lösungswerten entsprechen (insbesondere gilt $y_{k-1} = y(t_{k-1})$), und betrachtet dann die Differenz

$$y_k - y(t_k) = \Phi(h, y_0, \dots, y_{k-1}) - E(h, y_{k-1}).$$

- Ein Hauptresultat der Numerik von Zeitintegrationsverfahren besagt, daß unter gewissen **Stabilitätsforderungen** aus einer **lokalen Fehlerentwicklung**

$$\mathbf{y}_1 - \mathbf{y}(t_1) = \mathcal{O}(\mathbf{h}^{p+1})$$

die **globale Fehlerentwicklung**

$$\mathbf{y}_N - \mathbf{y}(t_N) = \mathcal{O}(\mathbf{h}^p)$$

folgt (bei hinreichend regulärer Funktion f und geeignet gewählten Startwerten).
Kurz gefaßt

$$\text{Konsistenz} + \text{Stabilität} = \text{Konvergenz}$$

Ein wesentlicher Schritt ist somit die Analyse des lokalen Fehlers für verschiedene Verfahrensklassen.

Beispiele von lokalen Fehlerentwicklungen für (einfache) Zeitintegrationsverfahren:

- Vorbemerkungen:

- * Bei der Konvergenzanalyse eines Zeitintegrationsverfahrens betrachtet man als ersten Schritt eine skalare lineare Testgleichung (**Dahlquist Testgleichung**, autonome Differentialgleichung, Anfangszeitpunkt $t_0 = 0$)

$$\frac{d}{dt} y(t) = f(t, y(t)) = \lambda y(t), \quad t \geq 0, \quad y(0) = y_0, \quad \lambda \in \mathbb{R}, \quad (\text{bzw. } \lambda \in \mathbb{C})$$

mit exakter Lösung

$$y(t) = e^{t\lambda} y_0, \quad t \geq 0.$$

- * Aufgrund der Linearität der Testgleichung ist es zudem ausreichend, den numerischen und exakten Lösungsoperator zu betrachten, d.h. man verwendet die einfache Abhängigkeit der numerischen und exakten Lösung vom Anfangswert $y_1 = \Phi(h, y_0) = \Phi(h)y_0$ und $y(h) = E(h, y_0) = E(h)y_0$.

- * Eine Abschätzung des globalen Verfahrensfehlers mittels einer Abschätzung für lokale Verfahrensfehler und Stabilitätsabschätzungen basiert auf der Teleskopsumme (selbes Prinzip für Skalare $\alpha, \beta \in \mathbb{R}$ bzw. Matrizen $\alpha, \beta \in \mathbb{R}^{d \times d}$ bzw. sogar Lösungsoperatoren)

$$\begin{aligned} \alpha^k - \beta^k &= \underbrace{\alpha^k - \alpha^{k-1}\beta}_{=\alpha^{k-1}(\alpha-\beta)} + \underbrace{\alpha^{k-1}\beta - \alpha^{k-2}\beta^2}_{\alpha^{k-2}(\alpha-\beta)\beta} + \underbrace{\alpha^{k-2}\beta^2 - \alpha^{k-3}\beta^3}_{=\alpha^{k-3}(\alpha-\beta)\beta^2} \pm \dots + \underbrace{\alpha\beta^{k-1} - \beta^k}_{=(\alpha-\beta)\beta^{k-1}} \\ &= \alpha^{k-1}(\alpha-\beta) + \alpha^{k-2}(\alpha-\beta)\beta + \alpha^{k-3}(\alpha-\beta)\beta^2 \pm \dots + (\alpha-\beta)\beta^{k-1} \\ &= \sum_{j=0}^{k-1} \alpha^{k-1-j}(\alpha-\beta)\beta^j. \end{aligned}$$

- * **Vorsicht!** Bei der Analyse der lokalen Fehlers nimmt man an, daß die Zeitschrittweite h hinreichend klein ist und somit eine Entwicklung des Wertes der Exponentialfunktion bei h sinnvoll ist (die Anfangsglieder der Entwicklung sind dominant)

$$e^{h\lambda} = 1 + h\lambda + \frac{1}{2}h^2\lambda^2 + \underbrace{\frac{1}{6}h^3\lambda^3 + \dots}_{\text{vergleichsweise klein}}$$

Die Verwendung einer Entwicklung beim Endzeitpunkt T

$$e^{T\lambda} = 1 + T\lambda + \frac{1}{2}T^2\lambda^2 + \frac{1}{6}T^3\lambda^3 + \dots$$

ist im Allgemeinen jedoch nicht sinnvoll (die Anfangsglieder der Entwicklung sind *nicht* dominant).

– Explizites Eulerverfahren:

- * Das explizite Eulerverfahren (einzelner Zeitschritt der Länge h)

$$y_1 = y_0 + h f(t_0, y_0)$$

angewendet auf die Testgleichung ergibt die Approximation (Taylorreihenentwicklung der Exponentialfunktion)

$$\begin{aligned} y_1 = \Phi(h)y_0 &\approx y(h) = E(h)y_0, \\ \Phi(h) = 1 + h\lambda &\approx E(h) = e^{h\lambda} = 1 + h\lambda + \mathcal{O}(h^2). \end{aligned}$$

- * Der **lokale Verfahrensfehler** des expliziten Eulerverfahrens

$$y_1 - y(h) = (\Phi(h) - E(h))y_0$$

erfüllt somit die Abschätzung (Konsistenzordnung $p = 1$, d.h. $p + 1 = 2$)

$$|y_1 - y(h)| \leq Ch^2.$$

- * Die Approximation zum Endzeitpunkt $t_N = T$ ist durch

$$y_N = (1 + h\lambda)^N y_0 \approx y(t_N) = e^{Nh\lambda} y_0$$

gegeben.

- * Einfacher als die direkte Analyse des globalen Verfahrensfehlers (insbesondere für allgemeine nichtlineare Differentialgleichungen)

$$y_N - y(t_N) = (\Phi(h)^N - E(t_N)) y_0 = \left((1 + h\lambda)^N - e^{Nh\lambda} \right) y_0$$

ist die Verwendung der Relation (Teleskopsumme)

$$\alpha^k - \beta^k = \sum_{j=0}^{k-1} \alpha^{k-1-j} (\alpha - \beta) \beta^j,$$

die auf die Relation (**Lady Winderemere Fächer**, für den exakten Lösungsoperator gilt die Gleichheit $E(h)^N = E(Nh)$)

$$y_N - y(t_N) = (\Phi(h)^N - E(h)^N) y_0 = \sum_{j=0}^{N-1} \Phi(h)^{N-1-j} (\Phi(h) - E(h)) E(jh) y_0$$

und damit auf die Abschätzung

$$\begin{aligned} |y_N - y(t_N)| &\leq \sum_{j=0}^{N-1} \underbrace{|\Phi(h)|^{N-1-j}}_{=|1+h\lambda|^{N-1-j} \leq C} \underbrace{|\Phi(h) - E(h)|}_{\leq Ch^2} \underbrace{|E(jh) y_0|}_{=|y(t_j)| \leq C} \leq Ch \underbrace{\sum_{j=0}^{N-1} h}_{=Nh=T} \\ &\leq Ch \end{aligned}$$

führt.

Bemerkung: Das exponentielle Anwachsen der Lösungen für $\lambda > 0$ spiegelt sich in der Stabilitätsschranke (Annahme $h > 0$)

$$|\Phi(h)|^n = |1 + h\lambda|^n \leq e^{\lambda nh}.$$

In dieser Situation ist ein exponentielles Auseinanderdriften (Fehlerschranke im wesentlichen $|y_N - y(t_N)| \leq C e^{\lambda T} h^p$) der numerischen und exakten Lösung unvermeidbar. Insbesondere über lange Zeiten $T \gg 1$ ist die Bestimmung genauer Approximationen mit sehr hohem Rechenaufwand verbunden bzw. man kann sich nicht erwarten, daß die Ergebnisse quantitativ korrekt sind (chaotische Systeme, qualitative Theorie).

- **Implizites Eulerverfahren:** Das implizite Eulerverfahren

$$y_1 = y_0 + h f(h, y_1)$$

angewendet auf die Testgleichung ergibt die Approximation (Taylorreihenentwicklung der Exponentialfunktion, geometrische Reihe für $|h\lambda| < 1$ anwendbar)

$$y_1 = \Phi(h)y_0 \approx y(h) = E(h)y_0,$$

$$\Phi(h) = (1 - h\lambda)^{-1} = 1 + h\lambda + \mathcal{O}(h^2) \approx E(h) = e^{h\lambda} = 1 + h\lambda + \mathcal{O}(h^2).$$

Der **lokale Verfahrensfehler** des impliziten Eulerverfahrens erfüllt somit die Abschätzung (Ordnung $p = 1$, d.h. $p+1 = 2$, ebenso wie das explizite Eulerverfahren)

$$|y_1 - y(h)| \leq Ch^2.$$

Die Approximation zum Endzeitpunkt $t_N = T$ ist durch

$$y_N = (1 - h\lambda)^{-N} y_0 \approx y(t_N) = e^{Nh\lambda} y_0$$

gegeben. Wie zuvor beruht die Analyse des globalen Verfahrensfehlers auf der Relation

$$y_N - y(t_N) = (\Phi(h)^N - E(h)^N) y_0 = \sum_{j=0}^{N-1} \Phi(h)^{N-1-j} (\Phi(h) - E(h)) E(jh) y_0$$

und führt damit auf die Abschätzung (für $\lambda \leq 0$ und wegen $h > 0$ ist die Schranke $|1 - h\lambda|^{-1} \leq 1$ immer gültig!)

$$|y_N - y(t_N)| \leq \sum_{j=0}^{N-1} \underbrace{|\Phi(h)|^{N-1-j}}_{=|1-h\lambda|^{-(N-1-j)} \leq C} \underbrace{|\Phi(h) - E(h)|}_{\leq Ch^2} \underbrace{|E(jh)y_0|}_{=|y(t_j)| \leq C} \leq Ch \underbrace{\sum_{j=0}^{N-1} h}_{=Nh=T}$$

$$\leq Ch.$$

- **Verallgemeinerung:** Zur Konvergenzanalyse von Einschritt- und Mehrschrittverfahren für allgemeine nichtlineare Differentialgleichungen mit (lokal) Lipschitzstetiger definierender Funktion verwendet man dasselbe Prinzip wie beim expliziten und impliziten Eulerverfahren

$$\underbrace{\|y_N - y(t_N)\|}_{\text{Globaler Fehler}} \leq \sum_{j=0}^{N-1} \underbrace{\|\Phi(h)\|^{N-1-j}}_{\leq C} \underbrace{\|\Phi(h) - E(h)\|}_{\leq Ch^{p+1}} \underbrace{\|E(jh)y_0\|}_{\leq C}$$

Stabilitätsabschätzung **Lokale Fehlerabschätzung** Exakter Lösungswert

$$\leq Ch^p,$$

vgl. **Resultat zur Konvergenz von Einschrittverfahren** (Satz 4.5).

Ähnlich wie bei Quadraturapproximationen beruht die Konstruktion von expliziten und impliziten Runge–Kutta Verfahren auf der Lösung von Ordnungsbedingungen, die sich aus der Forderung

$$y_1 - y(h) = \mathcal{O}(h^{p+1})$$

ergeben. Die Herleitung der Ordnungsbedingungen für autonome Differentialgleichungen mit hinreichend oft differenzierbarer Funktion f beruht auf Taylorreihenentwicklungen der exakten Lösung (Differentialgleichung $y'(t) = f(y(t))$, mittels Kettenregel $y''(t) = f'(y(t)) f(y(t))$)

$$y(h) = y_0 + h \underbrace{y'(0)}_{=f(y_0)} + \frac{1}{2} h^2 \underbrace{y''(0)}_{=f'(y_0)f(y_0)} + \dots$$

unter der Verwendung graphentheoretischer Mittel wie *Bäume*) und analoger Entwicklungen der Runge–Kutta Lösung (komplizierter Teil). Beispielsweise führen die Gaußschen Quadraturformeln auf implizite Runge–Kutta Verfahren, die Gaußschen Verfahren.

4.3. Konvergenzresultat für Einschrittverfahren

- **Fragestellung:** Konvergenzresultat für Zeitintegrationsverfahren (vgl. Abschnitt 4.2)

Stabilität von Zeitintegrationsverfahren: Ein Zeitintegrationsverfahren (insbesondere ein Einschrittverfahren) mit Verfahrensfunktion Φ heißt **stabil**, wenn die mehrfache Hintereinanderausführung von Φ durch eine Konstante beschränkt ist, wobei die Konstante unabhängig von der Größe und der Anzahl der Zeitschritte ist.

Vorsicht! Bei Zeitintegrationsverfahren unterscheidet man die Begriffe **Stabilität** und u.a. **A-Stabilität** (vgl. Abschnitt 4.4), weiters zu unterscheiden von verschiedenen Stabilitätsbegriffen bei Differentialgleichungen oder der numerischen Stabilität eines Algorithmus.

Resultat zur Konvergenz von Einschrittverfahren (Satz 4.5): Die das Anfangswertproblem definierende Funktion f sei hinreichend regulär

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ y(t_0) \text{ gegeben.} \end{cases}$$

Weiters sei das Einschrittverfahren mit Verfahrensfunktion Φ wohldefiniert und stabil (für hinreichend kleine Zeitschrittweiten $0 < h_i \leq \bar{h}$, $0 \leq i \leq N-1$)

$$\begin{aligned} t_0 < t_1 < \dots < t_N = T, & \quad h_i = t_{i+1} - t_i, \quad 0 \leq i \leq N-1, \\ y_n = \Phi(h_{n-1}, t_{n-1}, y_{n-1}), & \quad 1 \leq n \leq N, \quad y_0 \text{ gegeben,} \end{aligned}$$

und es besitze die Konsistenzordnung $p \geq 1$. Dann gilt die globale Fehlerabschätzung

$$\|y_N - y(t_N)\| \leq C (\|y_0 - y(t_0)\| + h_{\max}^p), \quad h_{\max} = \max_{0 \leq i \leq N-1} h_i,$$

mit Konstante $C > 0$ unabhängig von der Größe und der Anzahl der Zeitschritte.

Bemerkung: Die Aussage des Konvergenzresultates gilt im Wesentlichen auch für Lineare Mehrschrittverfahren. Für ein stabiles k -Schriftverfahren mit Startwerten y_0, \dots, y_{k-1} ergibt sich die globale Fehlerabschätzung

$$\|y_N - y(t_N)\| \leq C \left(\max_{0 \leq i \leq k-1} \|y_i - y(t_i)\| + h_{\max}^p \right).$$

Da Mehrschrittverfahren durch Mehrschrittrekursionen definiert sind und diese im Allgemeinen instabil sind, sind Mehrschrittverfahren höherer Ordnung zu vermeiden.

Illustrationen (Einfache stabile Zeitintegrationsverfahren, Numerische Berechnung der Konvergenzordnungen):

- Explizites Eulerverfahren (Ordnung 1)
 - Explizite Mittelpunktsregel (Ordnung 2)
 - Explizites Runge–Kutta Verfahren der Ordnung 4
- Implizites Eulerverfahren (Ordnung 1)
 - Implizite Mittelpunktsregel (Ordnung 2)
 - (Implizite) Trapezregel (Ordnung 2)
- Mittelpunktsregel (explizites Zweischrittverfahren, Ordnung 2)

Bemerkungen (Beweis des Konvergenzresultates):

- Die Herleitung eines Konvergenzresultates für Einschrittverfahren angewendet auf nichtlineare Differentialgleichungen beruht auf dem zuvor für das explizite Eulerverfahren angegebenen Zugang. Zur Abschätzung des globalen Fehlers wird die folgende Relation verwendet (zur Vereinfachung Formulierung für lineare Differentialgleichungen und äquidistante Schrittweiten)

$$\underbrace{\|y_N - y(t_N)\|}_{\text{Globaler Fehler}} \leq \sum_{j=0}^{N-1} \underbrace{\|\Phi(h)\|^{N-1-j}}_{\leq C \text{ Stabilitätsabschätzung}} \underbrace{\|\Phi(h) - E(h)\|}_{\leq C h^{p+1} \text{ Lokale Fehlerabschätzung}} \underbrace{\|E(jh)y_0\|}_{\leq C \text{ Exakter Lösungswert}} \leq C h^p.$$

Wesentlich ist es somit, Stabilitätsabschätzungen und lokale Fehlerabschätzungen (Taylorreihenentwicklungen) abzuleiten.

Beispiele:

- * Mittels der folgenden Taylorreihenentwicklung der exakten Lösung

$$\begin{aligned} \frac{d}{dt} y(t) &= f(y(t)), \\ y(h) &= y(0) + h \underbrace{y'(0)}_{=f(y_0)} + \frac{1}{2} h^2 \underbrace{y''(0)}_{=f'(y_0)f(y_0)} + \mathcal{O}(h^3), \end{aligned}$$

ergibt sich für das **explizite Eulerverfahren** die lokale Fehlerentwicklung (Konsistenzordnung $p = 1$)

$$\begin{aligned} y_1 &= y_0 + h f(y_0), & y(h) &= y_0 + h f(y_0) + \mathcal{O}(h^2), \\ y_1 - y(h) &= \mathcal{O}(h^2), \end{aligned}$$

und für die **explizite Mittelpunktsregel** die lokale Fehlerentwicklung (Konsistenzordnung $p = 2$)

$$\begin{aligned} y_1 &= y_0 + h \underbrace{f\left(y_0 + \frac{h}{2} f(y_0)\right)}_{=f(y_0) + \frac{1}{2} h f'(y_0) f(y_0) + \mathcal{O}(h^2)} = y_0 + h f(y_0) + \frac{1}{2} h^2 f'(y_0) f(y_0) + \mathcal{O}(h^3), \\ y(h) &= y_0 + h f(y_0) + \frac{1}{2} h^2 f'(y_0) f(y_0) + \mathcal{O}(h^3), \\ y_1 - y(h) &= \mathcal{O}(h^3). \end{aligned}$$

- * Diffiziler ist die Vorgehensweise beim impliziten Eulerverfahren

$$y_1 = y_0 + h f(y_1),$$

da hier zuerst der Nachweis der Existenz der numerischen Lösung erbracht werden muß. Dazu verwendet man beispielsweise den Banachsche Fixpunktsatz (vgl. Numerische Mathematik I). Für Lipschitz-stetige definierende Funktionen f mit Lipschitz-Konstante $L > 0$ ist für hinreichend kleine

Zeitschrittweiten $0 < h \leq \bar{h}$ die Kontraktivität der Abbildung F sichergestellt

$$y_1 = F(y_1) = y_0 + h f(y_1),$$

$$\|F(z) - F(\tilde{z})\| = \|h(f(z) - f(\tilde{z}))\| \leq \underbrace{L\bar{h}}_{=\kappa < 1} \|z - \tilde{z}\|.$$

Damit folgt die Existenz und Eindeutigkeit des Fixpunktes y_1 sowie die Beschränktheit der numerischen Lösung

$$\begin{aligned} \|y_1\| &\leq \|y_0\| + h \|f(y_1)\| \\ &\leq \|y_0\| + h \|f(y_0)\| + h \|f(y_1) - f(y_0)\| \\ &\leq \|y_0\| + h \|f(y_0)\| + hL \|y_1 - y_0\| \\ &\leq (1 + hL) \|y_0\| + h \|f(y_0)\| + hL \|y_1\| \\ \Rightarrow \|y_1\| &\leq C = \frac{1}{1-\kappa} \left((1 + hL) \|y_0\| + h \|f(y_0)\| \right), \end{aligned}$$

Nun ist ein Taylorreihenentwicklung der numerischen Lösung sinnvoll und führt auf die lokale Fehlerabschätzung (Konsistenzordnung $p = 1$)

$$\begin{aligned} y_1 &= y_0 + h f(y_1) = y_0 + h f(y_0 + h f(y_1)) = y_0 + h f(y_0) + \mathcal{O}(h^2), \\ y(h) &= y_0 + h f(y_0) + \mathcal{O}(h^2), \\ y_1 - y(h) &= \mathcal{O}(h^2), \end{aligned}$$

- Ein alternativer Zugang verwendet (vgl. Skriptum S. 76)

$$\underbrace{\|y_N - y(t_N)\|}_{\text{Globaler Fehler}} \leq \sum_{j=0}^{N-1} \underbrace{\|E(t_{N-1-j})\|}_{\leq C} \underbrace{\|\Phi(h) - E(h)\|}_{\leq C h^{p+1}} \underbrace{\|\Phi(h)^j y_0\|}_{\leq C} \leq C h^p.$$

Abschätzung mittels Satz 4.2 Lokale Fehlerabschätzung Numerische Lösung

Stabilitätsabschätzungen gehen hier bei der Abschätzung der numerischen Lösung ein.

- Zur Konvergenanalyse von Linearen Mehrschrittverfahren nützt man die Formulierung als Einschrittverfahren (vgl. Numerische Mathematik I, Abschnitt 7.1).
- Bereits bei einfachen Anfangswertproblemen kann es zu **Ordnungsreduktionen** kommen, wenn beispielsweise die definierende Funktion f und damit die exakte Lösung y nicht hinreichend oft differenzierbar ist.

Illustration (Ordnungsreduktion): Für das triviale Testbeispiel (direkter Zusammenhang mit bestimmten Integralen und Quadraturapproximationen)

$$\frac{d}{dt} y(t) = f(t) = \frac{3}{2} \sqrt{t}, \quad 0 < t < T, \quad y(T) = y(0) + \int_0^T f(t) dt = y(0) + T^{\frac{3}{2}},$$

beobachtet man aufgrund der Singularität bei $t = 0$

$$\frac{d^2}{dt^2} y(t) = \frac{d}{dt} f(t) = \frac{3}{4} \frac{1}{\sqrt{t}}, \quad 0 < t < T,$$

für Ein- und Mehrschrittverfahren der Konsistenzordnung $p \geq 2$ eine Reduktion auf die Konvergenzordnung $\frac{3}{2}$.

- Bei Miteinbeziehung des Einflusses von Rundungsfehlern (zusätzliche Inhomogenität in der Differentialgleichung) ergibt sich insgesamt die Fehlerabschätzung (wegen $Nh = T$ folgt $N = \frac{C}{h}$ für äquidistante Zeitschrittweiten)

$$\text{Gesamtfehler} \leq \underbrace{C h^p}_{\text{Verfahrensfehler}} + \underbrace{C \varepsilon_{\text{mach}} \frac{1}{h}}_{\text{Rundungsfehler}}.$$

Bei zu groß gewählten Zeitschrittweiten dominiert der Verfahrensfehler, bei zu klein gewählten Schrittweiten beeinträchtigt der Fehler aufgrund der Akkumulation von Rundungsfehlern die erreichbare Genauigkeit. Für die optimale Wahl (Vernachlässigung der Konstanten, $\varepsilon_{\text{mach}} \approx 10^{-16}$)

$$\begin{aligned} g(h) &= h^p + \varepsilon_{\text{mach}} \frac{1}{h} \longrightarrow \min, \\ 0 = g'(h) &= p h^{p-1} - \varepsilon_{\text{mach}} \frac{1}{h^2} = \frac{1}{h^2} (p h^{p+1} - \varepsilon_{\text{mach}}) \implies h = \sqrt[p+1]{\frac{\varepsilon_{\text{mach}}}{p}}, \\ g(h) &\approx \varepsilon_{\text{mach}}^{\frac{p}{p+1}}, \\ p = 1: \quad g(h) &\approx \sqrt{\varepsilon_{\text{mach}}} \approx 10^{-8}, \\ p = 2: \quad g(h) &\approx \varepsilon_{\text{mach}}^{\frac{2}{3}} \approx 10^{-11}, \\ p = 4: \quad g(h) &\approx \varepsilon_{\text{mach}}^{\frac{4}{5}} \approx 10^{-12}, \\ p = 10: \quad g(h) &\approx \varepsilon_{\text{mach}}^{\frac{10}{11}} \approx 10^{-15}, \end{aligned}$$

zeigt sich der Vorteil bei der Anwendung eines Zeitintegrationsverfahrens höherer Ordnung.

- **Richardson Extrapolation (Explizites Eulerverfahren):** Im Zusammenhang mit Zeitintegrationsverfahren beruhen Extrapolationsverfahren auf der Idee, Linearkombinationen von numerische Approximationen zu verschiedenen Zeitschrittverfahren zu berechnen, die eine höhere Ordnung besitzen. Bestimmt man beispielsweise mittels explizitem Eulerverfahren numerische Approximationen zu den Schrittweiten h und $\frac{1}{2} h$ (Ordnung $p = 1$, einzelner Zeitschritt)

$$y_1 = y_0 + h f(t_0, y_0), \quad z_{1/2} = y_0 + \frac{1}{2} h f(t_0, y_0), \quad z_1 = z_{1/2} + \frac{1}{2} h f(t_0 + \frac{1}{2} h, z_{1/2}),$$

so zeigt eine einfache Rechnung, daß die Linearkombination

$$2 z_1 - y_1$$

Ordnung $p + 1 = 2$ besitzt. Der allgemeine Zugang beruht ähnlich wie bei Quadraturapproximationen auf asymptotischen Entwicklungen der numerischen Lösung.

Vgl. **Illustration** (Extrapolation).

- Entsprechend adaptiven Quadraturformeln verwendet man u.a. Paare von Runge–Kutta Verfahren verschiedener Ordnung zur vorteilhaften Wahl der Zeitschrittweiten (Verfahren zur Integration, Verfahren zur Schätzung des lokalen Fehlers). Optimale Verfahren in Hinblick auf die Anzahl der benötigten Funktionsauswertungen sind **eingebettete Runge–Kutta Verfahren** der Ordnungen $p \neq \hat{p}$

$$\frac{c \mid A}{\mid b} \quad \frac{c \mid A}{\mid \hat{b}}$$

mit gleichen Knoten und internen Stufen, jedoch unterschiedlich gewählten Gewichten.

Beispiel: DOPRI (Dormand – Prince Verfahren, 1980), explizites Runge–Kutta Verfahren der Ordnung 4(5), Standard-Löser ODE45

- **Adaptive Zeitintegrationsverfahren:** Ähnlich dem Prinzip adaptiver Verfahren zur numerischer Berechnung bestimmter Integrale sollen bei adaptiven Zeitintegrationsverfahren die Zeitschrittweiten dem Lösungsverlauf optimal angepaßt werden. Die Idee ist es, bei Anwendung eines Verfahrens der Konsistenzordnung p die Schrittweite so zu modifizieren, daß eine vorgegebene Toleranz erreicht wird

$$\text{ERR}_{\text{lokal}} = \text{err}(h) \approx C h^{p+1}, \quad \text{err}(h_{\text{optimal}}) \approx C h_{\text{optimal}}^{p+1} \approx \text{TOL}.$$

Die Division beider Relationen führt auf folgende Faustregel für die optimale Zeitschrittweite

$$\left(\frac{h_{\text{optimal}}}{h}\right)^{p+1} \approx \frac{\text{TOL}}{\text{ERR}_{\text{lokal}}} \iff h_{\text{optimal}} \approx h^{p+1} \sqrt[p+1]{\frac{\text{TOL}}{\text{ERR}_{\text{lokal}}}}.$$

Dabei bezeichnet $\text{ERR}_{\text{lokal}}$ den mittels eines Verfahrens höherer Ordnung (siehe auch Eingebettete Verfahren, Extrapolation) geschätzten **lokalen Fehler**.

Vgl. **Illustration** (Schrödingergleichung, Adaptive Verfahren).

4.4. A-Stabilität

- **Situation:** Betrachtet wird eine autonome Differentialgleichung mit hinreichend regulärer definierender Funktion $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ (unendliches Zeitintervall)

$$\frac{d}{dt} y(t) = f(y(t)), \quad t > t_0.$$

Asymptotisch stabile stationäre Lösungen: Eine Lösung $z: I = [t_0, \infty) \rightarrow \mathbb{R}^d$ der Differentialgleichung mit Anfangswert $z_0 = z(t_0)$ heißt **asymptotisch stabil**, wenn es eine Konstante $\delta > 0$ gibt, sodaß für jede andere Lösung $\tilde{z}: I \rightarrow \mathbb{R}^d$ der Differentialgleichung mit Anfangswert $\tilde{z}_0 = \tilde{z}(t_0)$ folgende Eigenschaft gilt

$$\|z_0 - \tilde{z}_0\| < \delta \implies \lim_{t \rightarrow \infty} \|z(t) - \tilde{z}(t)\| = 0.$$

Besondere Bedeutung haben asymptotisch stabile **stationäre Lösungen** (bzw. Gleichgewichtslösungen), d.h. zeitunabhängige Lösungen der Differentialgleichung

$$\frac{d}{dt} z(t) = 0 = f(z(t)), \quad t > t_0.$$

Beispiel: Die skalare lineare Differentialgleichung

$$\frac{d}{dt} y(t) = \lambda y(t), \quad t \geq 0, \quad \lambda < 0,$$

besitzt die asymptotisch stabile stationäre Lösung $z = 0$. Für beliebige Anfangswerte $\tilde{z}_0 \in \mathbb{R}$ gilt nämlich

$$\tilde{z}(t) = e^{\lambda t} \tilde{z}_0, \quad t \geq 0, \quad \lim_{t \rightarrow \infty} \|\tilde{z}(t)\| = \lim_{t \rightarrow \infty} e^{\lambda t} \|\tilde{z}_0\| = 0.$$

- **Fragestellung:** Approximationsgüte eines Zeitintegrationsverfahrens bei der Berechnung asymptotisch stabiler Lösungen.

Bemerkung: Aus dem zuvor angegebenen Resultat zum Konvergenzverhalten von Zeitintegrationsverfahren folgt keine Aussage über die Approximationsgüte bei der Berechnung asymptotisch stabiler Lösungen, da die Konvergenzabschätzung nur auf beschränkten Zeitintervallen gültig ist (Aussage für hinreichend kleine Zeitschrittweiten, aufgrund der auftretenden Konstante $C = C(T)$ ist die Abschätzung nur auf vergleichsweise kurzen Zeitintervallen sinnvoll).

- **Vorbemerkung:** Es sei z eine stationäre Lösung der Differentialgleichung

$$\frac{d}{dt} y(t) = f(y(t)), \quad t > t_0.$$

Theoretische Resultate sichern, daß das asymptotische Verhalten der stationären Lösung durch die linearisierte Differentialgleichung bestimmt ist

$$\begin{aligned} \frac{d}{dt} y(t) &= f(y(t)) = f(z) + f'(z)(y(t) - z) + \mathcal{O}(\|y(t) - z\|^2), \\ \frac{d}{dt} \tilde{y}(t) &= A \tilde{y}(t), \quad \tilde{y}(t) = e^{(t-t_0)A} \tilde{y}(t_0), \quad t \geq t_0, \quad A = f'(z). \end{aligned}$$

Falls alle Eigenwerte der Matrix A negativen Realteil haben, ist die Lösung asymptotisch stabil, falls jedoch ein Eigenwert mit positivem Realteil existiert, ist die Lösung asymptotisch instabil. Dies rechtfertigt die Betrachtung der Testgleichung von Dahlquist.

Im Folgenden wird wieder die Bezeichnung $y: I \rightarrow \mathbb{R}^d$ für die betrachtete Lösung verwendet, und z bezeichnet eine stationäre Lösung der Differentialgleichung.

Bemerkung: Für die folgenden Überlegungen ist es zweckmäßig, die exakte und numerische Lösung mittels des exakten und numerischen Lösungsoperators anzugeben (für die Testgleichung wird die Abhängigkeit vom Zeitpunkt bzw. von der Zeitschrittweite und dem Parameter λ angegeben)

$$y_1 = \Phi(h\lambda) y_0 \approx y(h) = E(h\lambda) y(0).$$

Testgleichung (Dahlquist): Betrachte die skalare lineare Testgleichung mit bekannter exakter Lösung (vgl. Abschnitt 4.2, Annahme $y_0 \neq 0$)

$$\begin{aligned} \frac{d}{dt} y(t) &= \lambda y(t), \quad t \geq 0, \quad y(0) = y_0, \quad \lambda < 0, \quad (\text{bzw. } \lambda \in \mathbb{C} \text{ mit } \Re \lambda < 0) \\ y(t) &= E(t\lambda) y_0 = e^{t\lambda} y_0, \quad t \geq 0, \quad \lim_{t \rightarrow \infty} E(t\lambda) = 0. \end{aligned}$$

A-Stabilität (vgl. Definition 4.8): Ähnlich dem Verhalten der exakten Lösung der Testgleichung fordert man von einem **A-stabilen** numerischen Verfahren mit Verfahrensfunktion Φ (angewendet mit äquidistanten Zeitschritten $h > 0$), daß in der obigen Situation die Approximationswerte gegen die asymptotisch stabile Lösung konvergieren

$$y_n = \Phi(h\lambda) y_{n-1} = \Phi(h\lambda)^n y_0, \quad n \geq 0, \quad \lim_{n \rightarrow \infty} \Phi(h\lambda)^n = 0.$$

Dies ist gleichbedeutend mit der Forderung

$$|\Phi(h\lambda)| < 1$$

bzw. auch damit, daß die linke Halbebene im (absoluten) **Stabilitätsbereich** des Verfahrens enthalten ist

$$\mathbb{C}_{<0} = \{\mu \in \mathbb{C} : \Re \mu < 0\} \subset S = \{\mu \in \mathbb{C} : |\Phi(\mu)| < 1\}.$$

Bemerkungen:

- Abhängig von der betrachteten Problemklasse ist auch eine Abschwächung des Begriffes der A-Stabilität ausreichend.
- Bei Anwendung eines Zeitintegrationsverfahrens auf die Testgleichung ist es naheliegend, die Bezeichnungen $\Phi(h\lambda) \approx E(h\lambda)$ zu verwenden, welche die wesentlichen Größen beinhalten.

Beispiele (Explizites Eulerverfahren, Implizites Eulerverfahren):

- Das **explizite Eulerverfahren** angewendet auf die Testgleichung ist durch

$$\begin{aligned}\Phi(h\lambda) &= 1 + h\lambda \approx E(h\lambda) = e^{h\lambda}, \\ y_n &= \Phi(h\lambda)^n y_0 = (1 + h\lambda)^n y_0 \approx y(nh) = E(nh\lambda) y_0 = e^{nh\lambda} y_0,\end{aligned}$$

gegeben. Somit folgt

$$\lim_{n \rightarrow \infty} y_n = 0 \iff |\Phi(h\lambda)| = |1 + h\lambda| < 1$$

und weiters ($\lambda < 0$, Fall $|1 - h|\lambda|| = 1 - h|\lambda| < 1$ für $1 - h|\lambda| > 0$ ist abgedeckt)

$$|1 + h\lambda| = |h|\lambda| - 1| = h|\lambda| - 1 < 1 \iff h|\lambda| < 2.$$

Z.B. für $\lambda = -10^j$ führt dies auf die Schrittweitereinschränkung $h < 2 \cdot 10^{-j}$. Bei zu groß gewählten Zeitschrittweiten oszilliert die numerische Lösung und wächst betragsmäßig stark an, folglich ist das numerische Ergebnis wertlos.

Der Stabilitätsbereich

$$S = \{\mu \in \mathbb{C} : |1 + \mu| < 1\}$$

beschreibt das Innere eines Kreises mit Mittelpunkt -1 und Radius 1 , d.h. das explizite Eulerverfahren ist *nicht* A-stabil.

- Das **implizite Eulerverfahren** angewendet auf die Testgleichung ist durch

$$\Phi(h\lambda) = \frac{1}{1 - h\lambda} \approx E(h\lambda) = e^{h\lambda}$$

gegeben. Die Bedingung

$$|\Phi(h\lambda)| = \frac{1}{|1 - h\lambda|} < 1 \iff 1 < |1 - h\lambda| = 1 + h|\lambda|$$

ist für alle Schrittweiten $h > 0$ erfüllt, und insbesondere ist das implizite Eulerverfahren A-stabil.

Fazit: Bei der numerischen Lösung der Testgleichung mittels explizitem Eulerverfahren sind aus Stabilitätsgründen (für $\lambda \gg 1$ extrem starke) Schrittweitereinschränkungen erforderlich. Die numerische Lösung der Testgleichung mittels implizitem Eulerverfahren bleibt hingegen für beliebige Zeitschritte stabil und führt bereits bei vergleichsweise großen Schrittweiten auf ein zufriedenstellendes Ergebnis.

Allgemein gilt, daß nur implizite Zeitintegrationsverfahren die Eigenschaft der A-Stabilität besitzen können.

- Implizite Runge–Kutta Verfahren wie Radau IIA Verfahren sind A-stabil.
- Das implizite lineare Mehrschrittverfahren BDF 2 ist A-stabil. Für $3 \leq k \leq 6$ ist das Verfahren BDF k zwar nicht A-stabil, jedoch A(ϑ)-stabil (Sektor anstelle Halbebene im Stabilitätsbereich enthalten).

- Stabilitätseigenschaften numerischer Verfahren wie A-Stabilität sind im Zusammenhang mit **steifen Differentialgleichungen** (insbesondere partiellen Differentialgleichungen wie Diffusionsgleichungen) wesentlich.

Curtiss & Hirschfelder (1952): ... *stiff equations are equations where certain implicit methods ... perform better, usually tremendously better, than explicit ones.*

Illustration (Zeitintegration der Diffusionsgleichung mittels explizitem und implizitem Eulerverfahren)

5. Randwertprobleme für gewöhnliche Differentialgleichungen

- **Inhalte:**

- Problemstellung
- Schießverfahren (Lösung von Anfangswertproblemen, Newtonverfahren)
Differenzenverfahren
Kollokationsverfahren

Bemerkung: Vertauschen der Abschnitte 5.3 und 5.4

- **Ausblick:** Ortsdiskretisierungsverfahren für partielle Differentialgleichungen

- Finite Differenzen Methode
- Finite Elemente Methode (Galerkin Verfahren)

5.1. Problemstellung

- **Erinnerung:**

- Bei einem Anfangswertproblem sind das Zeitintervall $I = [t_0, T]$, die (hinreichend reguläre) definierende Funktion $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ und der **Anfangswert** $y_0 \in \mathbb{R}^d$ vorgegeben. Gesucht ist eine (hinreichend reguläre) Funktion $y : I \rightarrow \mathbb{R}^d$, die die Differentialgleichung sowie die Anfangsbedingung erfüllt

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ y(t_0) = y_0. \end{cases}$$

- Speziell bei der Schwingungsgleichung sind neben der Differentialgleichung

$$m \frac{d^2}{dt^2} y(t) + r \frac{d}{dt} y(t) + k y(t) = 0, \quad t \in (t_0, T),$$

die **Anfangsauslenkung und Anfangsgeschwindigkeit** vorgegeben

$$y(t_0), \quad \left. \frac{d}{dt} y(t) \right|_{t=t_0}.$$

Bei der Formulierung der Differentialgleichung zweiter Ordnung als Differentialgleichungssystem erster Ordnung, entsprechen die erste bzw. zweite Komponente des Vektors $Y(t) = (y(t), \frac{d}{dt} y(t))^T$ der Auslenkung bzw. Geschwindigkeit zum aktuellen Zeitpunkt

$$\frac{d}{dt} \begin{pmatrix} Y_1(t) \\ Y_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{r}{m} \end{pmatrix} \begin{pmatrix} Y_1(t) \\ Y_2(t) \end{pmatrix}, \quad t \in (t_0, T), \quad Y(t_0) \text{ gegeben.}$$

- **Vorbemerkung:** Bei der Schwingungsgleichung kann man anstelle der Anfangsauslenkung und Anfangsgeschwindigkeit beispielsweise auch die **Auslenkung zum Anfangs- und Endzeitpunkt** vorgegeben, was auf ein Randwertproblem führt. Allgemeiner ist ein Randwertproblem durch eine gewöhnliche Differentialgleichung und eine algebraische Bedingung für die Randwerte $y(t_0)$ und $y(T)$ gegeben.

Randwertprobleme für gewöhnliche Differentialgleichungen erster Ordnung: Für vorgegebene Punkte $t_0 \in \mathbb{R}$ und $T \in \mathbb{R}$ sei $I = [t_0, T]$ falls $t_0 < T$ (bzw. $I = [T, t_0]$ falls $t_0 > T$). Weiters seien $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d : (t, v) \mapsto f(t, v)$ und $r : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d : (v, w) \mapsto r(t, v)$ vorgegebene (reguläre) Funktionen. Eine Lösung des **Randwertproblems**

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), & (\text{bzw. } (T, t_0)) \\ r(y(t_0), y(T)) = 0, \end{cases}$$

ist eine Funktion $y : I \rightarrow \mathbb{R}^d$, welche die (gewöhnliche) **Differentialgleichung** (erster Ordnung) und die **Randbedingung** erfüllt. Dabei wird vorausgesetzt, daß y auf I stetig und zumindest in (t_0, T) (bzw. (T, t_0)) differenzierbar ist.

- Falls die Funktion r durch Matrizen $A, B \in \mathbb{R}^{d \times d}$ und eine Spalte $c \in \mathbb{R}^d$ definiert ist, d.h. es ist $r(v, w) = Av + Bw - c$, spricht man von einer **linearen Randbedingung**

$$Ay(t_0) + By(T) = c.$$

- Falls die Funktion r von der Form $r(v, w) = (\varrho(v), \tilde{\varrho}(w))$ mit Funktionen $\varrho : \mathbb{R}^d \rightarrow \mathbb{R}^k$ und $\tilde{\varrho} : \mathbb{R}^d \rightarrow \mathbb{R}^{d-k}$ für $1 \leq k \leq d-1$ ist, spricht man von einer **separierten Randbedingung**. Insbesondere ist dies für Randbedingungen der einfachen Form (nach eventueller Umordnung der Komponenten von y)

$$y_1(t_0) = c_1, \dots, y_k(t_0) = c_k, \quad y_{k+1}(T) = c_{k+1}, \dots, y_d(T) = c_d,$$

der Fall.

Bemerkung: Im Gegensatz zu Anfangswertproblemen ist es bei (nichtlinearen) Randwertproblemen (noch) schwierig(er), allgemeine Aussagen zur Existenz und Eindeutigkeit zu machen.

- Dies zeigt sich beispielsweise an der einfachen linearen Differentialgleichung (Schwingungsgleichung mit $m = k = 1, r = 0$)

$$\frac{d^2}{dt^2} y(t) + y(t) = 0, \quad t \in (0, \pi),$$

mit exakter Lösung

$$y(t) = C_1 \sin t + C_2 \cos t, \quad t \in [0, \pi].$$

Wegen $y(0) = C_2$ und $y(\pi) = -C_2$ existieren bei Vorgabe der Randbedingungen

$$y(0) = 0, \quad y(\pi) = 0, \quad y(t) = C_1 \sin t,$$

unendlich viele Lösungen, bei Vorgabe der Randbedingungen

$$y(0) = 1, \quad y(\pi) = 1,$$

existiert jedoch keine Lösung.

- **Verallgemeinerung:** Die Lösung $y : I \rightarrow \mathbb{R}$ einer skalaren linearen Differentialgleichung zweiter Ordnung ist von der folgenden Form mit **homogenen Lösungen** $y_{h,1}$ und $y_{h,2}$ (zur Differentialgleichung mit $\gamma = 0$) und einer **partikulären Lösung** y_p (spezielle Lösung).

$$\frac{d^2}{dt^2} y(t) + \alpha(t) \frac{d}{dt} y(t) + \beta(t) y(t) = \gamma(t), \quad t \in (t_0, T),$$

$$y = C_1 y_{h,1} + C_2 y_{h,2} + y_p.$$

Bei der spezieller Wahl der Anfangsbedingungen (Satz von Picard–Lindelöf sichert die Existenz und Eindeutigkeit der Lösungen)

$$y_{h,1}(t_0) = 1, \quad \frac{d}{dt} y_{h,1}(t) \Big|_{t=t_0} = 0, \quad y_{h,2}(t_0) = 0, \quad \frac{d}{dt} y_{h,2}(t) \Big|_{t=t_0} = 1, \\ y_p(t_0) = 0, \quad \frac{d}{dt} y_p(t) \Big|_{t=t_0} = 0,$$

folgen bei Vorgabe der Randbedingungen

$$y(t_0) = y_0, \quad y(T) = y_T,$$

die folgenden Relationen für die zu bestimmenden Konstanten C_1, C_2

$$\begin{aligned} y(t_0) &= C_1 \underbrace{y_{h,1}(t_0)}_{=1} + C_2 \underbrace{y_{h,2}(t_0)}_{=0} + \underbrace{y_p(t_0)}_{=0} = C_1 \stackrel{!}{=} y_0 \quad \implies \quad C_1 = y_0, \\ y(T) &= \underbrace{C_1}_{=y_0} y_{h,1}(T) + C_2 y_{h,2}(T) + y_p(T) = y_0 y_{h,1}(T) + C_2 y_{h,2}(T) + y_p(T) = y_T \\ &\implies \quad C_2 y_{h,2}(T) = y_T - y_p(T) - y_0 y_{h,1}(T). \end{aligned}$$

Somit können drei unterschiedliche Fälle eintreten:

- * $y_{h,2}(T) \neq 0$: Eindeutige Lösung mit $C_2 = \frac{1}{y_{h,2}(T)} (y_T - y_p(T) - y_0 y_{h,1}(T))$.
 - * $y_{h,2}(T) = 0$ und $y(T) = y_p(T) + y_0 y_{h,1}(T) = y_T$: Unendlich viele Lösungen.
 - * $y_{h,2}(T) = 0$ und $y(T) = y_p(T) + y_0 y_{h,1}(T) \neq y_T$: Keine Lösung.
- **Freie Randwertprobleme:** Beispielsweise beim *Re-Entry Problem* (Wiedereintritt einer Rakete in die Atmosphäre) tritt ein Randwertproblem auf, wo der Endzeitpunkt nicht festgelegt ist und stattdessen eine zusätzliche Randbedingung vorgegeben ist (d.h. es ist $r: \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^{d+1}$)

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \quad (\text{bzw. } (T, t_0)) \\ r(y(t_0), y(T), T) = 0, \end{cases}$$

Solche **freien Randwertprobleme** lassen sich mittels einer Transformation des Zeitintervalles auf das Einheitsintervall

$$\begin{aligned} [t_0, T] &\longleftrightarrow [0, 1] \\ t = t_0 + s(T - t_0) &\longleftrightarrow s = \frac{t - t_0}{T - t_0} \end{aligned}$$

auf die Standardform eines Randwertproblems für den Lösungsvektor $Y(s) = (\tilde{Y}(s), T)^T$ mit $\tilde{Y}(s) = y(t)$ reduzieren, nämlich (wegen $\frac{d}{ds} \tilde{Y}(s) = \frac{d}{dt} y(t) \frac{dt}{ds} = f(t, y(t)) T$) folgt $\frac{d}{ds} \tilde{Y}(s) = T f(t_0 + s(T - t_0), \tilde{Y}(s))$, weiters ist $Y(0) = (y(t_0), T)$ und $Y(1) = (y(T), T)$

$$\begin{cases} \frac{d}{ds} Y(s) = \begin{pmatrix} T f(t_0 + s(T - t_0), \tilde{Y}(s)) \\ 0 \end{pmatrix}, & s \in (0, 1), \\ r(Y(0), Y(1)) = 0. \end{cases}$$

Illustration: Wurfparabel als Anfangswertproblem, Randwertproblem und Freies Randwertproblem (zwei physikalisch sinnvolle Lösungen zu den Endzeitpunkten $T_1, T_2 > 0$)

5.2. Lösung durch Rückführung auf ein Anfangswertproblem (Schießverfahren)

- **Einfaches Schießverfahren:**

- **Vereinfachung:** Zur Vereinfachung wird zunächst ein Randwertproblem für eine Funktion $y = (y_1, y_2)^T : I = [t_0, T] \rightarrow \mathbb{R}^2$ mit speziellen separierten Randbedingungen betrachtet

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ y_1(t_0) = y_{01}, \quad y_1(T) = y_{T1}. \end{cases}$$

Beispiele: Schwingungsgleichung, Wurfparabel

- **Idee:** Die Idee des (einfachen) **Schießverfahrens** ist es, das Randwertproblem auf ein Anfangswertproblem zurückzuführen

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ y(t_0) = y_0 = (y_{01}, y_{02})^T. \end{cases}$$

Die unbekannte Komponente des Anfangswertes soll dabei so bestimmt werden, daß die zugehörige exakte Lösung (Angabe der Abhängigkeit des Lösungsoperators E vom aktuellen Zeitpunkt, Anfangszeit und Anfangswert)

$$y(t) = E(t, t_0, y_0)$$

im Endzeitpunkt T die geforderte Randbedingung erfüllt (dabei bezeichnet $y_1(T) = (E(T, t_0, y_0))_1$ die erste Komponente der exakten Lösung bei $t = T$)

$$y_1(T) = (E(T, t_0, y_0))_1 = y_{T1},$$

d.h. es ist die nichtlineare Gleichung

$$F(y_{02}) = y_1(T) - y_{T1} = (E(T, t_0, y_0))_1 - y_{T1} = 0$$

zu lösen.

Illustration (Verschiedene Anfangsgeschwindigkeiten und zugehörige Lösungen), vgl. Skriptum, S. 99.

- **Verallgemeinerung:** Die Lösung eines Randwertproblems allgemeiner Form für eine Funktion $y : I \rightarrow \mathbb{R}^d$

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ r(y(t_0), y(T)) = 0, \end{cases}$$

beruht auf der Betrachtung des zugehörigen Anfangswertproblems

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ y(t_0) = y_0, \end{cases}$$

und der Lösung der nichtlinearen Gleichung (für die unbekanntenen Lösungskomponenten)

$$F(y_0) = r(y(t_0), y(T)) = r(y_0, E(T, t_0, y_0)) = 0.$$

– **Näherungsweise Lösung:**

- * Im Allgemeinen wird die nichtlineare Gleichung

$$F(y_0) = r(y(t_0), y(T)) = 0$$

mittels eines iterativen Verfahrens näherungsweise gelöst, und eine naheliegende Wahl ist das Newtonverfahren oder Modifikationen davon (vgl. Numerische Mathematik I). Bei Verwendung des Newtonverfahrens wird (zumindest näherungsweise) die erste Ableitung der Funktion F benötigt (Kettenregel, partielle Ableitungen $\partial_1 r$ und $\partial_2 r$, Ableitung der exakten Lösung nach dem Anfangswert $\partial_{y_0} y$)

$$F'(y_0) = \partial_1 r(y_0, y(T)) + \partial_2 r(y_0, y(T)) \partial_{y_0} y(T).$$

Zur Bestimmung der Ableitung der exakten Lösung nach dem Anfangswert verwendet man die **Variationsgleichung** (zeitabhängige Matrix $Y = \partial_{y_0} y$ erfüllt Differentialgleichung $\frac{d}{dt} \partial_{y_0} y(t) = \partial_2 f(t, y(t)) \partial_{y_0} y(t)$ und Anfangsbedingung $\partial_{y_0} y(t_0) = I$, $\partial_2 f(t, v)$ bezeichnet die partielle Ableitung von f bezüglich des zweiten Argumentes v)

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), & y(t_0) = y_0, \\ \frac{d}{dt} Y(t) = \partial_2 f(t, y(t)) Y(t), & t \in (t_0, T), & Y(t_0) = I, \end{cases}$$

vgl. Dynamische Systeme und insbesondere Stabilität von Differentialgleichungen.

- * Zur näherungsweisen Lösung eines Randwertproblems wird im Allgemeinen die folgende Vorgehensweise gewählt:
 - (i) Wahl eines geeigneten Startwertes y_0 .
 - (ii) Anwendung eines Zeitintegrationsverfahren zur numerischen Lösung des zugehörigen Anfangswertproblems, und insbesondere Berechnung einer Approximation an den Funktionswert $y_N \approx y(T) = E(T, t_0, y_0)$ (vgl. Abschnitt 4). Berechnung von $\tilde{F}(y_0) = r(y_0, y_N) \approx F(y_0)$.
 - (iii) Anwendung eines Zeitintegrationsverfahren zur numerischen Lösung der zugehörigen Variationsgleichung, und insbesondere Berechnung einer Approximation an den Funktionswert $Y_N \approx \partial_{y_0} y(T)$. Berechnung von $G(y_0) = \partial_1 r(y_0, y_N) + \partial_2 r(y_0, y_N) Y_N \approx F'(y_0)$.
 - (iv) In einem Newtonschritt, ersetze y_0 durch $y_0 - G(y_0)^{-1} \tilde{F}(y_0)$ und iteriere.
- * In Situationen, wo die obige Vorgehensweise versagt, weil das Newtonverfahren schlechte Konvergenzeigenschaften besitzt (Startwert ungeeignet, Differentialgleichung mit exponentiell anwachsenden Lösungen, großes Integrationsintervall) wird anstelle des einfachen Schießverfahrens das **mehrfache Schießverfahren** (Einfügen zusätzlicher Stützstellen) verwendet, vgl. Skriptum, S. 100.

- Speziell für ein Randwertproblem, das von einem Randwertproblem für eine skalare lineare Differentialgleichung zweiter Ordnung mit speziellen separierten Randbedingungen her stammt (Lösung $z : I \rightarrow \mathbb{R}$)

$$\begin{cases} \frac{d^2}{dt^2} z(t) + \alpha(t) \frac{d}{dt} z(t) + \beta(t) z(t) = \gamma(t), & t \in (t_0, T), \\ z(t_0) = z_0, \quad z(T) = z_T, \end{cases}$$

kann man die gesuchte Komponente des Anfangswertes bestimmen (sofern spezielle homogene und eine spezielle partikuläre Lösung bekannt sind). In diesem Fall ist das zugehörige Anfangswertproblem (setze $y = (y_1, y_2)^T = (z, \frac{d}{dt} z)^T : I \rightarrow \mathbb{R}^2$)

$$\begin{cases} \frac{d}{dt} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} y_2(t) \\ -\beta(t) y_1(t) - \alpha(t) y_2(t) + \gamma(t) \end{pmatrix}, & t \in (t_0, T), \\ \begin{pmatrix} y_1(t_0) \\ y_2(t_0) \end{pmatrix} = \begin{pmatrix} y_{01} \\ y_{02} \end{pmatrix}. \end{cases}$$

Die unbekannte Komponente y_{02} des Anfangswertes soll dabei so bestimmt werden, daß die zugehörige exakte Lösung im Endzeitpunkt die geforderte Randbedingung erfüllt

$$y_1(T) = z(T) = z_T.$$

Falls homogene Lösungen $z_{h,1}$, $z_{h,2}$ und eine partikuläre Lösung z_p der Differentialgleichung zweiter Ordnung bekannt sind (zu den in Abschnitt 5.1 angegebenen Anfangsbedingungen), folgt die Lösungsdarstellung

$$z = C_1 z_{h,1} + C_2 z_{h,2} + z_p.$$

Einsetzen der bekannten Funktionswerte $y_1(t_0) = z(t_0) = z_0$, $y_1(T) = z(T) = z_T$ bzw. der geforderten Anfangsbedingung $y_2(t_0) = \frac{d}{dt} z(t)|_{t=t_0} = y_{02}$ führt auf

$$\begin{aligned} z &= C_1 z_{h,1} + C_2 z_{h,2} + z_p, \\ \frac{d}{dt} z &= C_1 \frac{d}{dt} z_{h,1} + C_2 \frac{d}{dt} z_{h,2} + \frac{d}{dt} z_p, \\ z(t_0) &= C_1 \underbrace{z_{h,1}(t_0)}_{=1} + C_2 \underbrace{z_{h,2}(t_0)}_{=0} + \underbrace{z_p(t_0)}_{=0} = C_1 = z_0 \implies C_1 = z_0, \\ \frac{d}{dt} z(t)|_{t=t_0} &= C_1 \underbrace{\frac{d}{dt} z_{h,1}(t)|_{t=t_0}}_{=0} + C_2 \underbrace{\frac{d}{dt} z_{h,2}(t)|_{t=t_0}}_{=1} + \underbrace{\frac{d}{dt} z_p(t)|_{t=t_0}}_{=0} = C_2 = y_{02} \\ &\implies C_2 = y_{02}, \\ z(T) &= C_1 z_{h,1}(T) + C_2 z_{h,2}(T) + z_p(T) = z_0 z_{h,1}(T) + y_{02} z_{h,2}(T) + z_p(T) = z_T \\ &\implies y_{02} = \frac{z_T - z_p(T) - z_0 z_{h,1}(T)}{z_{h,2}(T)}. \end{aligned}$$

Sofern die Bedingung $z_{h,2}(T) \neq 0$ für die Lösbarkeit des Randwertproblems erfüllt ist, ist die Lösung des Randwertproblems gerade die Lösung des zugehörigen Anfangswertproblems mit

$$y_{02} = \frac{z_T - z_p(T) - z_0 z_{h,1}(T)}{z_{h,2}(T)}.$$

- **Illustration** (Einfaches Schießverfahren für homogene lineare Differentialgleichung $y' = Ay$).

5.4. Kollokationsverfahren

- **Situation:** Betrachtet wird ein Randwertproblem der Form für eine Funktion $y: I \rightarrow \mathbb{R}^d$

$$\begin{cases} \frac{d}{dt} y(t) = f(t, y(t)), & t \in (t_0, T), \\ r(y(t_0), y(T)) = 0. \end{cases}$$

Bemerkung: Mit **Kollokation** bezeichnet man die Interpolation von Funktionswerten und Ableitungen.

Idee: Bei einem Kollokationsverfahren für ein Randwertproblem betrachtet man einen Raum von Funktionen (z.B. Polynomfunktionen oder Splinefunktionen, zusätzliche Eigenschaften) und fordert, daß die Differentialgleichung in gewissen Stützstellen sowie die Randbedingungen erfüllt sind. Im Allgemeinen ist der Funktionenraum als lineare Hülle $\langle \nu_1, \dots, \nu_K \rangle$ von Basisfunktionen gegeben und man wählt den Ansatz

$$z = \sum_{i=1}^K c_i \nu_i \approx y,$$

wobei die Koeffizienten $c_i \in \mathbb{R}$ so bestimmt werden, daß für vorgegebene Stützstellen $t_0 < t_1 < \dots < t_{N-1} < t_N = T$ die Bedingungen (Differentialgleichung in den inneren Punkten, Randbedingung)

$$\begin{cases} \frac{d}{dt} z(t)|_{t=t_n} = f(t_n, z(t_n)), & 1 \leq n \leq N-1, \\ r(z(t_0), z(t_N)) = 0, \end{cases}$$

erfüllt sind.

Bemerkungen:

- Oft ist es vorteilhaft, die Basisfunktionen so zu wählen, daß spezielle Randbedingungen wie etwa die homogene Randbedingungen $y(a) = 0 = y(b)$ automatisch erfüllt sind (Linearkombinationen der Basisfunktionen erfüllen dann ebenfalls die homogenen Randbedingungen).
- Vgl. Konstruktion der BDF-Verfahren.

Illustration (Einfache Differentialgleichung, Kollokation mittels Polynomfunktionen), vgl. Skriptum, S. 110.

5.3. Differenzenverfahren

- **Vorbemerkung:** Differenzenverfahren (Relaxationsverfahren) sind *globale* Verfahren (kein *time-stepping approach*) zur näherungsweise Lösung von Randwertproblemen für gewöhnliche Differentialgleichungen und werden auch zur Ortsdiskretisierung von partiellen Differentialgleichungen verwendet. Aus diesem Grund wird die unabhängige Variable in diesem Abschnitt mit x bezeichnet.
- **Situation:** Betrachtet wird ein Randwertproblem, das von einem Randwertproblem für eine **skalare lineare Differentialgleichung zweiter Ordnung** mit speziellen separierten Randbedingungen her stammt (Lösung $y: [a, b] \rightarrow \mathbb{R}$, (zumindest) stetige Funktionen $\alpha, \beta, \gamma: [a, b] \rightarrow \mathbb{R}$, **Voraussetzung** $\beta \geq 0$ (punktweise), d.h. $\beta(x) \geq 0$ für $x \in [a, b]$)

$$\begin{cases} -\frac{d^2}{dx^2} y(x) + \alpha(x) \frac{d}{dx} y(x) + \beta(x) y(x) = \gamma(x), & x \in (a, b), \\ y(a) = y_a, \quad y(b) = y_b. \end{cases}$$

Kurzschreibweise mittels Differentialoperator: Eine übliche Kurzschreibweise mittels eines (linearen) Differentialoperators (zweiter Ordnung) ist (Bemerkung zu Definitionsbereich, s.u.)

$$L: \mathcal{C}^2(a, b) \longrightarrow \mathcal{C}(a, b) : z \longmapsto Lz = -\frac{d^2}{dx^2} z + \alpha \frac{d}{dx} z + \beta z,$$

d.h. es ist $(Lz)(x) = -\frac{d^2}{dx^2} z(x) + \alpha(x) \frac{d}{dx} z(x) + \beta(x) z(x)$ (oft auch kurz $Lz(x)$ statt $(Lz)(x)$). Die Differentialgleichung läßt sich dann in der kompakten Form (ähnlich einem linearen Gleichungssystem, s.u.)

$$Ly = \gamma \quad \text{bzw.} \quad (Ly)(x) = \gamma(x), \quad x \in (a, b),$$

angeben.

Differenzenverfahren: Die Idee von **Differenzenverfahren** (bzw. **Finiten Differenzenverfahren**) ist es, in einer Differentialgleichung die auftretenden Differentialquotienten durch **Differenzenquotienten** zu ersetzen. Bei einer linearen Differentialgleichung führt dies auf ein lineares Gleichungssystem für die Näherungswerte an den vorgegebenen Stützwerten.

Beispiele (Symmetrische Differenzen): Häufig verwendete Approximationen der ersten und zweiten Ableitung sind (Vorwärtsdifferenz, Rückwärtsdifferenz, Symmetrische Differenzen)

$$\begin{aligned} \frac{y(x+h)-y(x)}{h} &= \frac{d}{dx} y(x) + \mathcal{O}(h), & \frac{y(x-h)-y(x)}{-h} &= \frac{y(x)-y(x-h)}{h} = \frac{d}{dx} y(x) + \mathcal{O}(h), \\ \frac{y(x+h)-y(x-h)}{2h} &= \frac{d}{dx} y(x) + \mathcal{O}(h^2), \\ \frac{y(x+h)-2y(x)+y(x-h)}{h^2} &= \frac{d^2}{dx^2} y(x) + \mathcal{O}(h^2). \end{aligned}$$

Genauer: Falls die Funktion y hinreichend oft differenzierbar ist, führen Taylorreihenentwicklungen auf folgende Relationen für die symmetrischen Differenzen (jeweils mit $\xi \in [x-h, x+h]$)

$$\begin{aligned} y \in \mathcal{C}^3(a, b) : \quad & \frac{y(x+h) - y(x-h)}{2h} = \frac{d}{dx} y(x) + \frac{1}{6} h^2 \frac{d^3}{dx^3} y(x) \Big|_{x=\xi}, \\ y \in \mathcal{C}^4(a, b) : \quad & \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} = \frac{d^2}{dx^2} y(x) + \frac{1}{12} h^2 \frac{d^4}{dx^4} y(x) \Big|_{x=\xi}. \end{aligned}$$

Gitterfunktion (Grid function): Zur Vereinfachung werden im Folgenden äquidistante **Gitterpunkte** zur **Gitterweite** $h > 0$ betrachtet (M innere Punkte x_1, \dots, x_M)

$$\bar{\Omega} = \{x_m = a + mh : 0 \leq m \leq M+1\}, \quad h = \frac{b-a}{M+1}.$$

Als diskretes Analogon der exakten Lösung $y : [a, b] \rightarrow \mathbb{R}$ des Randwertproblems ist die **Gitterfunktion** auf dem Ortsgitter definiert

$$\tilde{y} : \bar{\Omega} \rightarrow \mathbb{R} : x_m \rightarrow \tilde{y}(x_m) = \tilde{y}_m, \quad \tilde{y}_m \approx y(x_m), \quad 0 \leq m \leq M+1.$$

Die Randwerte $\tilde{y}_0 = y(x_0) = y_a$ und $\tilde{y}_{M+1} = y(x_{M+1}) = y_b$ sind vorgegeben (zur Vereinfachung exakte Randwerte), zu bestimmen sind die Werte der Gitterfunktion an den inneren Gitterpunkten

$$\Omega = \{x_m = a + mh : 1 \leq m \leq M\}.$$

Bemerkungen:

- Im vorliegenden eindimensionalen Fall ist die Gitterfunktion \tilde{y} durch die Funktionswerte an den (fixierten) inneren Gitterpunkten x_1, \dots, x_M bestimmt, und deshalb wird auch die Vektorschreibweise (Matrix in 2D, Tensorstruktur in 3D)

$$\tilde{y} = (\tilde{y}_m)_{1 \leq m \leq M} = (\tilde{y}_1, \dots, \tilde{y}_M)^T$$

verwendet.

- Nach Einschränkung der exakten Lösung auf die Gitterpunkte (wie zuvor Vektorschreibweise mit $y_m = y(x_m)$ für $0 \leq m \leq M+1$)

$$y|_{\Omega} = (y_1, \dots, y_M)^T,$$

ist es sinnvoll, die Differenz (Fehler des Differenzenverfahrens, kein Beitrag der Randwerte unter der Annahme $\tilde{y}_0 = y(x_0) = y_a$ und $\tilde{y}_{M+1} = y(x_{M+1}) = y_b$)

$$\tilde{y} - y|_{\Omega} = (\tilde{y}_1 - y_1, \dots, \tilde{y}_M - y_M)^T$$

zu betrachten.

Approximation mittels symmetrischer Differenzen: Mittels der oben angegebenen symmetrischen Differenzen ergibt sich als diskretes Analogon des Differentialoperators

$$L: z \mapsto Lz = -\frac{d^2}{dx^2} z + \alpha \frac{d}{dx} z + \beta z$$

der Differenzenoperator

$$\tilde{L}: \tilde{z} = (\tilde{z}_m)_{1 \leq m \leq M} \mapsto \tilde{L}\tilde{z} = \left(-\frac{\tilde{z}_{m+1} - 2\tilde{z}_m + \tilde{z}_{m-1}}{h^2} + \alpha(x_m) \frac{\tilde{z}_{m+1} - \tilde{z}_{m-1}}{2h} + \beta(x_m) \tilde{z}_m \right)_{1 \leq m \leq M}.$$

Beachte! Der Operator L ist vorerst für auf dem offenen Intervall zweimal differenzierbare Funktionen $z: (a, b) \rightarrow \mathbb{R}$ definiert, und es ergibt sich eine Funktion $Lz: (a, b) \rightarrow \mathbb{R}$. Durch die Voraussetzung $z \in \mathcal{C}^2(a, b)$ kann man Lz (in eindeutiger Weise) auf das abgeschlossene Intervall $[a, b]$ fortsetzen. Der Operator \tilde{L} ist für Gitterfunktionen $\tilde{z}: \bar{\Omega} \rightarrow \mathbb{R}$ definiert, und ergibt eine auf den inneren Gitterpunkten definierte Funktion $\tilde{L}\tilde{z}: \Omega \rightarrow \mathbb{R}$. Schreibt man für $\tilde{L}\tilde{z}$ dieselben Randwerte wie für \tilde{z} vor, erhält man eine auf allen Gitterpunkten definierte Funktion $\tilde{L}\tilde{z}: \bar{\Omega} \rightarrow \mathbb{R}$, bei der Funktionsvorschrift gibt man üblicherweise jedoch nur die Werte der inneren Gitterpunkte an.

Kompakte Schreibweise: Mittels Matrix- und Vektorschreibweise ergibt sich

$$(\beta(x_m) \tilde{z}_m)_{1 \leq m \leq M} = \begin{pmatrix} \beta(x_1) \tilde{z}_1 \\ \vdots \\ \beta(x_m) \tilde{z}_m \\ \vdots \\ \beta(x_M) \tilde{z}_M \end{pmatrix} = \underbrace{\begin{pmatrix} \beta(x_1) & & & & \\ & \ddots & & & \\ & & \beta(x_m) & & \\ & & & \ddots & \\ & & & & \beta(x_M) \end{pmatrix}}_{=A_0} \begin{pmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_m \\ \vdots \\ \tilde{z}_M \end{pmatrix}$$

sowie

$$\begin{aligned} (\alpha(x_m) \frac{\tilde{z}_{m+1} - \tilde{z}_{m-1}}{2h})_{1 \leq m \leq M} &= \frac{1}{2h} \begin{pmatrix} \alpha(x_1) (\tilde{z}_2 - \tilde{z}_0) \\ \vdots \\ \alpha(x_m) (\tilde{z}_{m+1} - \tilde{z}_{m-1}) \\ \vdots \\ \alpha(x_M) (\tilde{z}_{M+1} - \tilde{z}_{M-1}) \end{pmatrix} \\ &= \frac{1}{2h} \underbrace{\begin{pmatrix} 0 & \alpha(x_1) & & & \\ & \ddots & & & \\ & & -\alpha(x_m) & 0 & \alpha(x_m) \\ & & & \ddots & \\ & & & & -\alpha(x_M) & 0 \end{pmatrix}}_{=A_1} \begin{pmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_m \\ \vdots \\ \tilde{z}_M \end{pmatrix} + \frac{1}{2h} \underbrace{\begin{pmatrix} -\alpha(x_1) \tilde{z}_0 \\ 0 \\ \vdots \\ 0 \\ \alpha(x_M) \tilde{z}_{M+1} \end{pmatrix}}_{=b_1} \end{aligned}$$

und weiters

$$\begin{aligned} \left(\frac{\tilde{z}_{m+1} - 2\tilde{z}_m + \tilde{z}_{m-1}}{h^2} \right)_{1 \leq m \leq M} &= \frac{1}{h^2} \begin{pmatrix} \tilde{z}_2 - 2\tilde{z}_1 + \tilde{z}_0 \\ \vdots \\ \tilde{z}_{M+1} - 2\tilde{z}_M + \tilde{z}_{M-1} \\ \vdots \\ \tilde{z}_{M+1} - 2\tilde{z}_M + \tilde{z}_{M-1} \end{pmatrix} \\ &= \frac{1}{h^2} \underbrace{\begin{pmatrix} -2 & 1 & & & \\ & \ddots & & & \\ & & 1 & -2 & 1 \\ & & & \ddots & \\ & & & & 1 & -2 \end{pmatrix}}_{=A_2} \underbrace{\begin{pmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_m \\ \vdots \\ \tilde{z}_M \end{pmatrix}}_{=b_2} + \frac{1}{h^2} \underbrace{\begin{pmatrix} \tilde{z}_0 \\ 0 \\ \vdots \\ 0 \\ \tilde{z}_{M+1} \end{pmatrix}}_{=b_2}. \end{aligned}$$

Insgesamt führt dies auf die kompakte Darstellung (affin-lineare Abbildung)

$$\tilde{L}: \mathbb{R}^M \longrightarrow \mathbb{R}^M: \tilde{z} \longmapsto \tilde{L}\tilde{z} = A\tilde{z} + b,$$

mit

$$A = \begin{pmatrix} \beta(x_1) + \frac{2}{h^2} & \frac{1}{2h} \alpha(x_1) - \frac{1}{h^2} & & & \\ & \ddots & & & \\ & & -\frac{1}{2h} \alpha(x_m) - \frac{1}{h^2} & \beta(x_m) + \frac{2}{h^2} & \frac{1}{2h} \alpha(x_m) - \frac{1}{h^2} \\ & & & \ddots & \\ & & & & -\frac{1}{2h} \alpha(x_M) - \frac{1}{h^2} & \beta(x_M) + \frac{2}{h^2} \end{pmatrix},$$

und (Einsetzen der vorgegebenen Randwerte)

$$b = b_1 - b_2 = \begin{pmatrix} -\left(\frac{1}{2h} \alpha(x_1) + \frac{1}{h^2}\right) y_a \\ 0 \\ \vdots \\ 0 \\ \left(\frac{1}{2h} \alpha(x_M) - \frac{1}{h^2}\right) y_b \end{pmatrix} \in \mathbb{R}^M.$$

Differenzenverfahren für Randwertprobleme: Die **näherungsweise Lösung des Randwertproblems**

$$Ly = \gamma$$

mittels symmetrischer Differenzen entspricht der **Lösung des linearen Gleichungssystems** (wegen $\tilde{L}\tilde{z} = A\tilde{z} + b$, mit **Tridiagonalmatrix** A)

$$\tilde{L}\tilde{y} = \tilde{\gamma} \iff A\tilde{y} = \tilde{\gamma} - b,$$

wobei die Approximation $\tilde{y} = (\tilde{y}(x_m))_{1 \leq m \leq M} = (\tilde{y}_m)_{1 \leq m \leq M}$ bei vorgegebener rechter Seite $\tilde{\gamma} = (\tilde{\gamma}(x_m))_{1 \leq m \leq M}$ zu bestimmen ist.

• **Fragestellungen:**

- Lösbarkeit des linearen Gleichungssystems, d.h. Existenz und Eindeutigkeit der diskreten Lösung
- Konvergenz der diskreten Lösung \tilde{y} gegen die exakte Lösung y für $h \rightarrow 0$, Approximationsgüte

- **Vorbemerkung:** Anstelle der direkten Betrachtung der Matrix A wird ein Maximumsprinzip verwendet (nützlich in Hinblick auf Verallgemeinerungen für Finite Differenzen Verfahren und Finite Elemente Verfahren zur Ortsdiskretisierung von partiellen Differentialgleichungen).

Erinnerung: Für eine (zumindest stetige) Funktion $z : [a, b] \rightarrow \mathbb{R}$ bezeichnet

$$\|\tilde{z}\|_{\infty, [a, b]} = \|\tilde{z}\|_{\infty} = \max_{a \leq x \leq b} |z(x)|.$$

Bezeichnung: Für eine Gitterfunktion $\tilde{z} : \bar{\Omega} \rightarrow \mathbb{R}$ bezeichnet

$$\|\tilde{z}\|_{\infty, \bar{\Omega}} = \max_{0 \leq m \leq M+1} |\tilde{z}_m|, \quad \|\tilde{z}\|_{\infty, \Omega} = \max_{1 \leq m \leq M} |\tilde{z}_m|.$$

Diskretes Maximumsprinzip (Lemma 5.1): Es sei $\tilde{z} : \bar{\Omega} \rightarrow \mathbb{R}$ eine Gitterfunktion mit

$$(\tilde{L}\tilde{z})_m \leq 0, \quad 1 \leq m \leq M.$$

Weiters sei die Gitterweite $h > 0$ so gewählt, daß die Bedingungen $1 + \frac{h}{2} \alpha(x_m) \geq 0$ sowie $1 - \frac{h}{2} \alpha(x_m) \geq 0$ für $1 \leq m \leq M$ erfüllt sind. Falls $\beta \geq 0$ folgt

$$\|\tilde{z}\|_{\infty, \bar{\Omega}} = \max\{|\tilde{z}_0|, |\tilde{z}_{M+1}|\}.$$

Denn: (i) Falls $\beta = 0$ gilt (Definition von \tilde{L})

$$(\tilde{L}\tilde{z})_m = -\frac{\tilde{z}_{m+1} - 2\tilde{z}_m + \tilde{z}_{m-1}}{h^2} + \alpha(x_m) \frac{\tilde{z}_{m+1} - \tilde{z}_{m-1}}{2h}, \quad 1 \leq m \leq M.$$

Sollte das Maximum in einem inneren Punkt angenommen werden

$$\|\tilde{z}\|_{\infty, \bar{\Omega}} = |\tilde{z}_j| \quad \text{mit} \quad 1 \leq j \leq M,$$

folgt durch Umformen der Relation (Anwendung der Voraussetzungen $(\tilde{L}\tilde{z})_j \leq 0$ und $1 \pm \frac{h}{2} \alpha(x_m) \geq 0$)

$$\begin{aligned}
(\tilde{L}\tilde{z})_j &= -\frac{\tilde{z}_{j+1}-2\tilde{z}_j+\tilde{z}_{j-1}}{h^2} + \alpha(x_j) \frac{\tilde{z}_{j+1}-\tilde{z}_{j-1}}{2h} \\
\iff \frac{1}{2} h^2 (\tilde{L}\tilde{z})_j &= -\frac{1}{2} \tilde{z}_{j+1} + \tilde{z}_j - \frac{1}{2} \tilde{z}_{j-1} + \frac{1}{4} h \alpha(x_j) (\tilde{z}_{j+1} - \tilde{z}_{j-1}) \\
\iff \frac{1}{2} h^2 (\tilde{L}\tilde{z})_j &= \tilde{z}_j - \frac{1}{2} \left(1 - \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j+1} - \frac{1}{2} \left(1 + \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j-1} \\
\iff \tilde{z}_j &= \underbrace{\frac{1}{2} h^2 (\tilde{L}\tilde{z})_j}_{\leq 0} + \underbrace{\frac{1}{2} \left(1 - \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j+1} + \frac{1}{2} \left(1 + \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j-1}}_{\leq \max\{\tilde{z}_{j-1}, \tilde{z}_{j+1}\}} \\
\iff \tilde{z}_j &\leq \max\{\tilde{z}_{j-1}, \tilde{z}_{j+1}\} \\
\implies \tilde{z}_{j-1} &= \tilde{z}_j = \tilde{z}_{j+1}.
\end{aligned}$$

Eine wiederholte Anwendung der Argumentation zeigt, daß \tilde{z} notwendigerweise konstant ist (und insbesondere das Maximum am Rand angenommen wird)

$$\tilde{z}_0 = \tilde{z}_1 = \dots = \tilde{z}_{M+1}.$$

(ii) Wie zuvor wird angenommen, daß das Maximum an einem inneren Gitterpunkt angenommen wird, d.h. es gelte $\|\tilde{z}\|_{\infty, \bar{\Omega}} = |\tilde{z}_j|$ mit $1 \leq j \leq M$. Unter der Voraussetzung $\beta \geq 0$ (punktweise), folgt ähnlich wie zuvor

$$\begin{aligned}
(\tilde{L}\tilde{z})_j &= -\frac{\tilde{z}_{j+1}-2\tilde{z}_j+\tilde{z}_{j-1}}{h^2} + \alpha(x_j) \frac{\tilde{z}_{j+1}-\tilde{z}_{j-1}}{2h} + \beta(x_j) \tilde{z}_j \\
\iff \frac{1}{2} h^2 (\tilde{L}\tilde{z})_j &= (1 + \beta(x_j)) \tilde{z}_j - \frac{1}{2} \left(1 - \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j+1} - \frac{1}{2} \left(1 + \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j-1} \\
\iff \tilde{z}_j &= \underbrace{\frac{1}{2} h^2 (\tilde{L}\tilde{z})_j}_{\leq 0} + \underbrace{\frac{1}{2} \left(1 - \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j+1} + \frac{1}{2} \left(1 + \frac{h}{2} \alpha(x_j)\right) \tilde{z}_{j-1}}_{\leq \max\{\tilde{z}_{j-1}, \tilde{z}_{j+1}\}} \\
\iff \tilde{z}_j &\leq \max\{\tilde{z}_{j-1}, \tilde{z}_{j+1}\} - \beta(x_j) \tilde{z}_j.
\end{aligned}$$

Falls $\tilde{z}_j \geq 0$ erhält man die Abschätzung

$$\tilde{z}_j \leq \max\{\tilde{z}_{j-1}, \tilde{z}_{j+1}\} - \underbrace{\beta(x_j) \tilde{z}_j}_{\leq 0} \leq \max\{\tilde{z}_{j-1}, \tilde{z}_{j+1}\} \implies \tilde{z}_{j-1} = \tilde{z}_j = \tilde{z}_{j+1},$$

und falls $\tilde{z}_j \leq 0$ verwendet man

$$\tilde{z}_j \leq \tilde{z}_j + \underbrace{\beta(x_j) \tilde{z}_j}_{\geq 0} \leq \max\{\tilde{z}_{j-1}, \tilde{z}_{j+1}\} \implies \tilde{z}_{j-1} = \tilde{z}_j = \tilde{z}_{j+1}.$$

Eine wiederholte Anwendung der Argumentation zeigt wiederum, daß \tilde{z} notwendigerweise konstant ist (und insbesondere das Maximum am Rand angenommen wird). \diamond

Bemerkung: Lemma 5.1 ist das diskrete Analogon zum Maximumsprinzip für Funktionen, welches besagt, daß eine Funktion $z \in \mathcal{C}^2(a, b)$ mit $Lz \leq 0$ (punktweise) ihr Maximum am Rand des Definitionsbereiches $[a, b]$ (d.h. im Punkt a oder b) annimmt.

Insbesondere für den Differentialoperator $L = -\frac{d^2}{dx^2}$ entspricht die Bedingung $Lz \leq 0$ der Eigenschaft, daß die Funktion z konvex ist und die Gültigkeit des Maximumsprinzipes ist dann offensichtlich.

Beispiel: Für die Funktion $z : [a, b] \rightarrow \mathbb{R} : x \mapsto z(x) = x^2$ (nach oben geöffnete Parabel mit Maximum am Rand) folgt $\frac{d^2}{dx^2}z = 2$ und damit $Lz = -\frac{d^2}{dx^2}z \leq 0$.

Erinnerung: Im Folgenden wird weiterhin die Voraussetzung $\beta \geq 0$ verwendet.

Abschätzung für Gitterfunktionen (Lemma 5.2): Falls die Gitterweite $h > 0$ hinreichend klein gewählt ist, erfüllt jede Gitterfunktion $\tilde{z} : \bar{\Omega} \rightarrow \mathbb{R}$ die Abschätzung

$$\|\tilde{z}\|_{\infty, \bar{\Omega}} \leq C \|\tilde{L}\tilde{z}\|_{\infty, \Omega} + \max\{|\tilde{z}_0|, |\tilde{z}_{M+1}|\}.$$

Denn: (i) Zur (nichttrivialen) Konstruktion einer Gitterfunktion $\tilde{\zeta} : \bar{\Omega} \rightarrow \mathbb{R} : x_m \rightarrow \tilde{\zeta}_m$, die den folgenden Eigenschaften genügt

$$\tilde{\zeta}_m \geq 0, \quad 0 \leq m \leq M+1, \quad (\tilde{L}\tilde{\zeta})_m \geq 1, \quad 1 \leq m \leq M,$$

verwendet man den Ansatz (mit Konstante $\lambda > 0$)

$$\tilde{\zeta}_m = e^\lambda - e^{\lambda \frac{x_m - a}{b-a}}, \quad 0 \leq m \leq M+1.$$

Die erste Bedingung ist offensichtlich erfüllt (Abschätzung $\frac{x_m - a}{b-a} \leq 1$, Monotonie der Exponentialfunktion)

$$\tilde{\zeta}_m = e^\lambda - e^{\lambda \frac{x_m - a}{b-a}} \geq 0, \quad 0 \leq m \leq M+1.$$

Andererseits folgt (mit $\mu = \frac{\lambda}{b-a}$, verwende $\cosh x = \frac{1}{2}(e^x + e^{-x})$ und $\sinh x = \frac{1}{2}(e^x - e^{-x})$)

$$\begin{aligned} (\tilde{L}\tilde{\zeta})_m &= -\frac{\tilde{\zeta}_{m+1} - 2\tilde{\zeta}_m + \tilde{\zeta}_{m-1}}{h^2} + \alpha(x_m) \frac{\tilde{\zeta}_{m+1} - \tilde{\zeta}_{m-1}}{2h} + \beta(x_m) \tilde{\zeta}_m \\ &= -\frac{1}{h^2} \left(e^\lambda - e^{\lambda \frac{x_{m+1} - a}{b-a}} - 2e^\lambda + 2e^{\lambda \frac{x_m - a}{b-a}} + e^\lambda - e^{\lambda \frac{x_{m-1} - a}{b-a}} \right) \\ &\quad + \alpha(x_m) \frac{1}{2h} \left(e^\lambda - e^{\lambda \frac{x_{m+1} - a}{b-a}} - e^\lambda + e^{\lambda \frac{x_{m-1} - a}{b-a}} \right) \\ &\quad + \underbrace{\beta(x_m) \left(e^\lambda - e^{\lambda \frac{x_m - a}{b-a}} \right)}_{\geq 0} \\ &\geq e^{\lambda \frac{x_m - a}{b-a}} \left(\frac{1}{h^2} (e^{h\mu} + e^{-h\mu} - 2) - \alpha(x_m) \frac{1}{2h} (e^{h\mu} - e^{-h\mu}) \right) \\ &= \underbrace{e^{\lambda \frac{x_m - a}{b-a}}}_{\geq 1} \underbrace{\left(2(\cosh(h\mu) - 1) - h\alpha(x_m) \sinh(h\mu) \right)}_{\geq 1}. \end{aligned}$$

Bei geeigneter Wahl von $h > 0$ (hinreichend klein) und $\mu > 0$ (d.h. λ hinreichend groß) läßt sich auch die zweite Bedingung erfüllen (setze $x = h\mu$ und verwende die Abschätzung $\alpha(x_m) \leq \alpha_{\max}$ durch den Maximalwert)

$$\begin{aligned} &\frac{2}{h^2} \left(2(\cosh(h\mu) - 1) - h\alpha(x_m) \sinh(h\mu) \right) \\ &\geq \begin{cases} \frac{4\mu^2}{x^2} (\cosh x - 1), & \alpha \leq 0, \\ \frac{2\mu^2}{x^2} \left(2(\cosh x - 1) - h\alpha_{\max} \sinh x \right), & \text{sonst.} \end{cases} \end{aligned}$$

(ii) Für die Gitterfunktion $\tilde{z} : \overline{\Omega} \rightarrow \mathbb{R}$ definiert man mittels der zuvor konstruierten Gitterfunktion $\tilde{\zeta} : \overline{\Omega} \rightarrow \mathbb{R}$ in Abhängigkeit vom Vorzeichen von \tilde{z}_0

$$\tilde{v} = \tilde{z} - \|\tilde{L}\tilde{z}\|_{\infty,\Omega} \tilde{\zeta} \quad \text{oder} \quad \tilde{v} = -\tilde{z} - \|\tilde{L}\tilde{z}\|_{\infty,\Omega} \tilde{\zeta},$$

derart daß $|\tilde{v}_0| = |\tilde{z}_0 - \|\tilde{L}\tilde{z}\|_{\infty,\Omega} e^\lambda| \leq |\tilde{z}_0|$ (falls $\tilde{z}_0 \geq 0$) oder $|\tilde{v}_0| = |\tilde{z}_0 + \|\tilde{L}\tilde{z}\|_{\infty,\Omega} e^\lambda| \leq |\tilde{z}_0|$ (falls $\tilde{z}_0 \leq 0$). Es gilt (verwende Linearität von \tilde{L} , o.E.d.A. $\|\tilde{L}\tilde{z}\|_{\infty,\Omega} > 0$)

$$\begin{aligned} \tilde{L}\tilde{v} &= \pm \tilde{L}\tilde{z} - \|\tilde{L}\tilde{z}\|_{\infty,\Omega} \tilde{L}\tilde{\zeta} = -\|\tilde{L}\tilde{z}\|_{\infty,\Omega} \left(\tilde{L}\tilde{\zeta} \mp \frac{1}{\|\tilde{L}\tilde{z}\|_{\infty,\Omega}} \tilde{L}\tilde{z} \right), \\ (\tilde{L}\tilde{v})_m &= -\|\tilde{L}\tilde{z}\|_{\infty,\Omega} \underbrace{\left((\tilde{L}\tilde{\zeta})_m \mp \frac{1}{\|\tilde{L}\tilde{z}\|_{\infty,\Omega}} (\tilde{L}\tilde{z})_m \right)}_{\geq 0} \leq 0, \quad 1 \leq m \leq M. \end{aligned}$$

Mittels Lemma 5.1 folgt (für $h > 0$ hinreichend klein, Gitterfunktionen \tilde{v} erfüllen Voraussetzung $\tilde{L}\tilde{v} \leq 0$ (komponentenweise), nach Konstruktion ist $|\tilde{v}_0| \leq |\tilde{z}_0|$ und wegen $\tilde{\zeta}_{M+1} = 0$ ist $|\tilde{v}_{M+1}| = |\tilde{z}_{M+1}|$)

$$|\tilde{v}_m| \leq \|\tilde{v}\|_{\infty,\overline{\Omega}} = \max\{|\tilde{v}_0|, |\tilde{v}_{M+1}|\} \leq \max\{|\tilde{z}_0|, |\tilde{z}_{M+1}|\},$$

und damit wegen $(\tilde{z}_m = \pm \tilde{v}_m \pm \|\tilde{L}\tilde{z}\|_{\infty,\Omega} \tilde{\zeta}_m)$

$$|\tilde{z}_m| \leq |\tilde{v}_m| + \|\tilde{L}\tilde{z}\|_{\infty,\Omega} \underbrace{|\tilde{\zeta}_m|}_{\leq C} \leq \max\{|\tilde{z}_0|, |\tilde{z}_{M+1}|\} + C \|\tilde{L}\tilde{z}\|_{\infty,\Omega}$$

die Behauptung. \diamond

Bemerkung: Lemma 5.2 ist das diskrete Analogon zu einem Resultat, welches besagt, daß eine Funktion $z \in \mathcal{C}^2(a, b)$ die Abschätzung

$$\max_{a \leq x \leq b} |z(x)| \leq C \max_{a \leq x \leq b} |Lz(x)| + \max\{|z(a)|, |z(b)|\}$$

erfüllt. Insbesondere im Zusammenhang mit Randwertproblemen

$$Ly = \gamma \quad \text{bzw.} \quad \tilde{L}\tilde{y} = \tilde{\gamma}$$

bezeichnet man die Abschätzung von Lemma 5.2 als **Stabilitätsabschätzung** (Schranke für die Werte der Lösung in Abhängigkeit von den Eingabedaten, d.h. von den Randwerten und der rechten Seite γ).

Bemerkung: Bei Differentialgleichungen, wo $\beta = 0$ oder $\beta \geq 0$ bzw. $\beta \leq 0$ (kein Vorzeichenwechsel) kann man die Einschränkung an die Gitterweite $h > 0$ vermeiden. Im letzteren Fall verwendet man anstelle der symmetrischen ersten Differenzen die einseitigen Differenzen (Rückwärtsdifferenz $\frac{y(x-h)-y(x)}{-h} = \frac{y(x)-y(x-h)}{h} = \frac{d}{dx} y(x) + \mathcal{O}(h)$ falls $\beta \geq 0$ bzw. Vorwärtsdifferenz $\frac{y(x+h)-y(x)}{h} = \frac{d}{dx} y(x) + \mathcal{O}(h)$ falls $\beta \leq 0$, vgl. Upwindverfahren bei Diffusions-Advektionsgleichungen).

- **Eindeutige Lösbarkeit:** Das lineare Gleichungssystem

$$\tilde{L}\tilde{y} = \tilde{\gamma}$$

besitzt eine eindeutige Lösung (für hinreichend kleine Schrittweiten $h > 0$).

Denn: Frühere Überlegungen zeigten (mit $\tilde{y} = (\tilde{y}(x_1), \dots, \tilde{y}(x_m))^T$)

$$\tilde{L}\tilde{y} = A\tilde{y} + b = \tilde{\gamma}.$$

Es reicht aus, das homogene lineare Gleichungssystem

$$A\tilde{y} = 0$$

zu betrachten und zu zeigen, daß nur die triviale Lösung $\tilde{y} = 0$ existiert. Dies entspricht dem Fall $\tilde{\gamma} = 0$ und $\tilde{y}_0 = \tilde{y}(x_0) = y_a = 0$ sowie $\tilde{y}_{M+1} = \tilde{y}(x_{M+1}) = y_b = 0$ (und somit $b = 0$). Mittels Lemma 5.2 folgt

$$\|\tilde{y}\|_{\infty, \bar{\Omega}} \leq C \underbrace{\|\tilde{L}\tilde{y}\|_{\infty, \Omega}}_{=0} + \underbrace{\max\{|\tilde{y}_0|, |\tilde{y}_{M+1}|\}}_{=0} = 0 \iff \tilde{y}_m = 0, \quad 1 \leq m \leq M,$$

was auf $\tilde{y} = 0$ führt. \diamond

Bemerkung: Die Lösung des linearen Gleichungssystems

$$\tilde{L}\tilde{y} = \tilde{\gamma}$$

mit Tridiagonalmatrix $\tilde{L} \in \mathbb{R}^{M \times M}$ benötigt $\mathcal{O}(M)$ Operationen (vgl. Abschnitt 2.1. zu Kubischen Splineinterpolanten).

- **Konvergenz der diskreten Lösung:** Wie zuvor bezeichnet $y : [a, b] \rightarrow \mathbb{R}$ die Lösung des Randwertproblems und $\tilde{y} : \Omega \rightarrow \mathbb{R}$ die Lösung des mittels symmetrischer Differenzen erhaltenen linearen Gleichungssystems (an den inneren Gitterpunkten, zu den exakten Randwerten)

$$Ly = \gamma \quad \text{bzw.} \quad \tilde{L}\tilde{y} = \tilde{\gamma}.$$

Die Differenz (Einschränkung der exakten Lösung auf die inneren Gitterpunkte, in den Randpunkten ist nach Annahme $d_0 = 0 = d_{M+1}$)

$$d = \tilde{y} - y|_{\Omega} : \Omega \longrightarrow \mathbb{R} : x_m \longmapsto d_m = \tilde{y}_m - y_m$$

erfüllt an den inneren Gitterpunkten die Relation

$$\begin{aligned} \tilde{L}d &= \tilde{L}\tilde{y} - \tilde{L}y|_{\Omega} = \tilde{\gamma} - \tilde{L}y|_{\Omega} = \gamma|_{\Omega} - \tilde{L}y|_{\Omega} = (Ly)|_{\Omega} - \tilde{L}y|_{\Omega}, \\ (Ly)(x_m) &= -\frac{d^2}{dx^2} y(x)|_{x=x_m} + \alpha(x_m) \frac{d}{dx} y(x)|_{x=x_m} + \beta(x_m) y(x_m), \\ (\tilde{L}y|_{\Omega})_m &= -\frac{1}{h^2} (y(x_{m+1}) - 2y(x_m) + y(x_{m-1}))) + \alpha(x_m) \frac{1}{2h} (y(x_{m+1}) - y(x_{m-1})) \\ &\quad + \beta(x_m) y(x_m), \quad 1 \leq m \leq M. \end{aligned}$$

Frühere Überlegungen zeigten (Taylorreihenentwicklungen symmetrischer Differenzen, jeweils mit $\xi \in [x - h, x + h]$)

$$y \in \mathcal{C}^3(a, b) : \quad \frac{y(x+h) - y(x-h)}{2h} - \frac{d}{dx} y(x) = \frac{1}{6} h^2 \frac{d^3}{dx^3} y(x) \Big|_{x=\xi},$$

$$y \in \mathcal{C}^4(a, b) : \quad \frac{y(x+h) - 2y(x) + y(x-h))}{h^2} - \frac{d^2}{dx^2} y(x) = \frac{1}{12} h^2 \frac{d^4}{dx^4} y(x) \Big|_{x=\xi}.$$

und da die Koeffizienten α, β auf $[a, b]$ stetig und damit beschränkt sind ergibt sich die Abschätzung

$$\|\tilde{L}(\tilde{y} - y|_{\Omega})\|_{\infty, \Omega} \leq C h^2 \|y^{(4)}\|_{\infty}.$$

Mittels Lemma 5.2 erhält man somit die (globale) Fehlerabschätzung (vergleichsweise einschränkende Regularitätsvoraussetzungen für Ordnung 2)

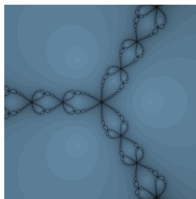
$$\|\tilde{y} - y|_{\Omega}\|_{\infty, \Omega} \leq C h^2 \|y^{(4)}\|_{\infty}.$$

Bemerkungen:

- Die Idee der Extrapolation mit Lösungen \tilde{y}_h und $\tilde{y}_{h/2}$ zu den Gitterweiten h und $\frac{h}{2}$ führt auf die verbesserte Approximation (Ordnung 4)

$$\frac{1}{3} (4\tilde{y}_{h/2, m} - \tilde{y}_{h, m})$$

- Die Idee der Adaptivität führt auf nichtuniforme Gitter.
- Die Anwendung von Differenzenverfahren auf nichtlineare Differentialgleichungen führt auf nichtlineare Gleichungssysteme (Lösbarkeit und Konvergenz deutlich schwieriger zu analysieren).



Informationen

Covid-19. Die Handhabung der aktuell an der Universität Innsbruck geltenden Regelungen wird per Email oder im Rahmen der Lehrveranstaltung geklärt.

Virtueller Konferenzraum. Die Lehrveranstaltung soll zum überwiegenden Teil in Präsenz stattfinden. Der BigBlueButton-Raum

webconference.uibk.ac.at/b/mec-est-pln (Link anklicken)

wird gegebenenfalls bei virtuellen Einheiten oder zur Online-Teilnahme verwendet. Um Störeffekte zu vermeiden, bitte daran denken, zunächst das Mikrofon zu deaktivieren.

Curriculum. Detaillierte Informationen zum Curriculum des Bachelorstudiums Mathematik finden sich unter

www.uibk.ac.at/studium/angebot/ba-mathematik.

Das in die Gebiete Numerik und Wissenschaftliches Rechnen einführende Modul

Numerische Mathematik 1

(VO 3, 4,5 ECTS, LV-Nr. 702331 & PS 2, 3 ECTS, LV-Nr. 702332)

wird im Wintersemester angeboten und im Sommersemester durch das Modul

Numerische Mathematik 2

(VO 3, 4,5 ECTS, LV-Nr. 702431 & PS 2, 3 ECTS, LV-Nr. 702432)

fortgesetzt. Die Anrechenbarkeit für andere Studien bitte abklären!

Terminübersicht. Im Wintersemester 2022/2023 finden die Lehrveranstaltungen an folgenden Terminen statt.

Vorlesung Montag 10.15 – 11.00 Uhr (HS D) / Mittwoch 14.15 – 16.00 (HS E)
Proseminar (Gruppe 1) Dienstag 8.15 – 9.45 (RR 25)
Proseminar (Gruppe 2) Dienstag 16.15 – 17.45 (RR 21)

Woche 1	3./5. & 4. Oktober 2022
Woche 2	10./12. & 11. Oktober 2022
Woche 3	17./19. & 18. Oktober 2022
Woche 4	24. & 25. Oktober 2022
Woche 5	31. Oktober 2022
Woche 6	7./9. & 8. November 2022
Woche 7	14./16. & 15. November 2022
Woche 8	21./23. & 22. November 2022
Woche 9	28./30. & 29. November 2022
Woche 10	5./7. & 6. Dezember 2022
Woche 11	12./14. & 13. Dezember 2022
Woche 12	9./11. & 10. Jänner 2023
Woche 13	16./18. & 17. Jänner 2023
Woche 14	23./25. & 24. Jänner 2023
Woche 15	30. Jänner/1. Feber & 31. Jänner 2023
Ersatztermin	27. Jänner 2023 (Präsentation von Projektarbeiten)

Inhalte. Das Gebiet der Numerik ist ein Teilgebiet der Mathematik, das sich insbesondere der Konstruktion und Analyse von Algorithmen widmet. In den einführenden Vorlesungen *Numerische Mathematik 1 & 2* werden Methoden zur näherungsweise Lösung von grundlegenden Problemen der Analysis und Linearen Algebra angegeben. Dies umfasst die Themenbereiche

- **Denkansätze und Grundbegriffe** (Wintersemester)
- **Vektoren und Matrizen** (Wintersemester)
- **Direkte Verfahren für lineare Gleichungssysteme** (Wintersemester)
- **Lineare Ausgleichsrechnung** (Wintersemester)
- Eigenwerte und SVD (Sommersemester)
- Nichtlineare Gleichungssysteme (Sommersemester)
- **Polynominterpolation** (Wintersemester)
- Polynom-Splines (Sommersemester)
- **Numerische Integration** (Wintersemester)
- Anfangswertprobleme für gewöhnliche Differentialgleichungen (Sommersemester)
- Randwertprobleme für gewöhnliche Differentialgleichungen (Sommersemester)
- Iterative Verfahren für lineare Gleichungssysteme (Sommersemester)

und gegebenenfalls weiterführende Themen. Als Illustrationen werden einfache Modellprobleme und Implementierungen betrachtet. In den begleitenden Proseminaren geht es

insbesondere um eine Vertiefung der Inhalte der Vorlesung, die Auswahl und Anwendung geeigneter numerischer Methoden sowie die Umsetzung mittels MATLAB.

Lernziele (laut Curriculum).

Absolventinnen und Absolventen der Lehrveranstaltungen Numerische Mathematik 1 & 2 (Pflichtmodule 11 & 14) verstehen die Inhalte der Vorlesung und können diese wiedergeben und anwenden. Sie haben die Fertigkeit erworben, sich ähnliche Inhalte selbständig zu erarbeiten. Sie sind in der Lage, grundlegende (Numerische Mathematik 1) bzw. fortgeschrittene (Numerische Mathematik 2) Methoden der Numerischen Mathematik situationsgerecht anzuwenden und mit MATLAB zu implementieren und zu demonstrieren. Weiters haben sie ein Grund- (Numerische Mathematik 1) bzw. vertieftes Verständnis (Numerische Mathematik 2) für die Methoden der Numerischen Mathematik erlangt.

Beurteilung. Detaillierte Informationen zur inhaltlichen Schwerpunkten und Beurteilungskriterien werden im Rahmen der Lehrveranstaltungen bekannt gegeben.

- Mündliche Vorlesungsprüfungen finden ab dem Semesterende statt (6./13. Feber 2023 oder Termine nach Vereinbarung, Dauer ca. 30 Minuten). Der Fokus liegt auf den besprochenen Begriffen, Resultaten, Methoden und Algorithmen.
- Die Beurteilung des Proseminares erfolgt anhand von Präsentationen und Projektarbeiten. Zentral sind die erworbenen Lösungskompetenzen in Hinblick auf die Implementierung von Algorithmen.

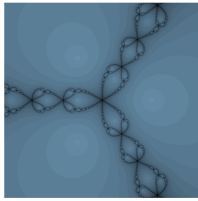
Unterlagen. Auf Skripten, Literaturquellen und ergänzende Unterlagen wird im Rahmen der Lehrveranstaltungen hingewiesen. Eine Vielzahl an Informationen zu numerischen Methoden und Algorithmen finden sich online – man sollte deren Richtigkeit aber kritisch hinterfragen.

Teamwork. Ein Austausch mit StudienkollegInnen und die Zusammenarbeit in kleinen Gruppen wird empfohlen.

Kontakt. Bei Fragen und für andere Anliegen bin ich im Anschluß an die Lehrveranstaltungen oder per Email

mechthild.thalhammer@uibk.ac.at

zu erreichen.



Informationen zum Proseminar

Modus.

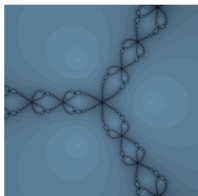
- In Proseminaren gilt Anwesenheitspflicht. Zweimaliges begründetes Fehlen wird toleriert. Gegebenenfalls sind Ersatzleistungen zu erbringen.
- Studierende sind dazu aufgefordert, sich zu melden und ihre Lösungen zu wöchentlich vorgegebenen Aufgaben und Problemstellungen zu präsentieren. Wesentlich sind neben der eigenständigen Wahl von geeigneten Testproblemen und den richtigen Resultaten korrekte mathematische Schreibweisen, schlüssige Begründungen sowie nachvollziehbare Kommentare in Programmen. Die Implementierungen können in MATLAB oder mittels Alternativen wie MATHEMATICA, PYTHON etc. erstellt werden.
- Für die bis Semesterende auszuarbeitenden Projekte gibt es eine Vielzahl an inhaltlichen Möglichkeiten, etwa die Einführung in ein neues Thema (z.B. Toolbox von MATLAB) oder die Vertiefung eines Themas (z.B. Standards für binäre Gleitkommazahlen IEEE 754, spezielle orthogonale Basisfunktionen, globalisiertes Newton-Verfahren). Es gelten folgende Richtlinien:
 - Beitrag jeder Person ersichtlich
 - Präsentation (ca. 15 Minuten pro Person)
 - * Inhaltliche Aspekte
 - * Komplexe Anwendung
 - * Sorgfältig kommentierter und ausführbarer Code ;-)
 - Folien und Programme hochladen
- Unterlagen zu den Lehrveranstaltungen sind im (aktuellen) OLAT-Kurs mit der Bezeichnung *Numerik 1 – Thalhammer Mechthild* o.ä. zu finden.
 - Bitte im Hauptordner (*Unterlagen* o.ä.) einen Ordner mit der Bezeichnung *NachnameVorname* generieren und darin sämtliche schriftliche Ausarbeitungen sowie Implementierungen hochladen, vorzugsweise wöchentlich eine einzige pdf-Datei und ein einziges Skript mit der Bezeichnung

Blattx_Nachname_Vorname_Aufgabenxxx.

Bitte Dateien NIEMALS per Email zusenden! ;-)

- Die Mindestanforderungen für eine positive Beurteilung des Proseminares sind:
 - Schriftliche Ausarbeitung von 60% der Proseminar-Aufgaben
 - Sorgfältige Nachbereitungen und nachvollziehbare Implementierungen
 - Präsentation von Aufgaben (je nach Gruppengröße) und Projektarbeit

In die Gesamtbeurteilung wird die Quantität und Qualität der Ausarbeitungen und Präsentationen sowie das Projekt einbezogen.



Numerische Mathematik 1 & 2
Mechthild Thalhammer
Illustrationen (Kapitel *Einführung*)

Illustrationen zum Kapitel *Einführung*

- **Binärdarstellung (Ergänzung, MATLAB)**
- **Hornerschema zur Auswertung von Polynomen (Ergänzung, MATLAB)**
- **Nullstellen von Polynomen / Eigenwerte von Matrizen (Illustration 1 und Modifikation, MAPLE / MATLAB)**


```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Beispiel: Polynom vom Grad d = 4 (übliche Darstellung, Horner Schema)
% p(x) = a(1) + a(2) x + a(3) x^2 + a(4) x^3 + a(5) x^4
% = (((a(5) x + a(4)) x + a(3)) x + a(2)) x + a(1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Direkte Auswertung:
% d Additionen
% 1 + 2 + ... + d = d*(d+1)/2 Multiplikationen (quadratische Komplexität)
% Auswertung mittels Horner Schema:
% d Additionen
% d Multiplikationen (lineare Komplexität)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% VORSICHT! Optimierungsstrategie von Matlab hier nicht berücksichtigt!
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc
format long

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Grad = 4;
a = [1:Grad+1];
AnzahlArgumente = 10^3;
x = rand(AnzahlArgumente,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Direkte Auswertung
tic
AnzahlAdditionDirekt = 0;
AnzahlMultiplikationDirekt = 0;
yDirekt = zeros(size(x));
for j = 1:length(x)
    yDirekt(j) = a(1);
    for i = 1:Grad
        aux = x(j);
        for k = 2:i
            aux = aux*x(j);
            AnzahlMultiplikationDirekt = AnzahlMultiplikationDirekt + 1;
        end
        yDirekt(j) = yDirekt(j) + a(i+1)*aux;
        AnzahlAdditionDirekt = AnzahlAdditionDirekt + 1;
        AnzahlMultiplikationDirekt = AnzahlMultiplikationDirekt + 1;
    end
end
% ErwarteteAnzahlAddition = Grad*length(x)
% ErwarteteAnzahlMultiplikation = Grad*(Grad+1)/2*length(x)
AnzahlAdditionDirekt
AnzahlMultiplikationDirekt
ZeitDirekt = toc
% Horner Schema
tic
AnzahlAdditionHorner = 0;
AnzahlMultiplikationHorner = 0;
yHorner = zeros(size(x));
for j = 1:length(x)
    yHorner(j) = a(end);
    for i = Grad:-1:1
        yHorner(j) = yHorner(j)*x(j) + a(i);
        AnzahlAdditionHorner = AnzahlAdditionHorner + 1;
        AnzahlMultiplikationHorner = AnzahlMultiplikationHorner + 1;
    end
end
end

```


Bekanntes Phänomen bei der Berechnung der Nullstellen von Polynomen höheren Grades / Eigenwerten einer Matrix höherer Dimension:

In Relation kleine Änderungen der Koeffizienten bewirken signifikante Änderungen des Resultates.

Beim gewählten Beispiel erhält man insbesondere komplexe anstelle von reellen Werten.

```
> restart;
```

```
> with(LinearAlgebra) :
```

```
Dimension der Matrix
```

```
> d := 12;
```

```
d := 12
```

(1)

```
Definition der Matrix
```

```
(Ausgabe für kleinere Dimensionen)
```

```
> A := Matrix(d, d, 0) :
```

```
for i from 1 to d - 1 do
```

```
A[i, i + 1] := 1;
```

```
A[i, i] := i;
```

```
A[i + 1, i] := 1;
```

```
od:
```

```
A[d, d] := d:
```

```
if d < 11 then
```

```
'A' = A;
```

```
fi;
```

```
Charakteristisches Polynom, Zugehörige Koeffizienten
```

```
> chi := lambda -> CharacteristicPolynomial(A, lambda) :
```

```
'chi(lambda)' = chi(lambda);
```

```
for i from 0 to d do
```

```
a[i] := coeff(chi(lambda), lambda, i);
```

```
od;
```

```

$$\chi(\lambda) = \lambda^{12} - 78 \lambda^{11} + 2706 \lambda^{10} - 55055 \lambda^9 + 729048 \lambda^8 - 6590064 \lambda^7 + 41449694 \lambda^6$$

$$- 181301679 \lambda^5 + 540869521 \lambda^4 - 1053129168 \lambda^3 + 1227435449 \lambda^2 - 711234043 \lambda$$

$$+ 116601199$$

```

```
a0 := 116601199
```

```
a1 := -711234043
```

```
a2 := 1227435449
```

```
a3 := -1053129168
```

```
a4 := 540869521
```

```
a5 := -181301679
```

```
a6 := 41449694
```

```
a7 := -6590064
```

```
a8 := 729048
```

```
 $a_9 := -55055$ 
```

```
 $a_{10} := 2706$ 
```

```
 $a_{11} := -78$ 
```

```
 $a_{12} := 1$ 
```

(2)

Binärdarstellung des k-ten Koeffizienten, Approximation

```
> k := 0;
c := 'c';
z := abs(a[k]);
j := -1;
while z > 0 do
j := j + 1;
c[j] := z mod 2;
z := simplify((z - c[j]) / 2);
od;
c = convert(c, list);
J := j;
a[k] := signum(a[k]) * sum(c['j'] * 2j, j'= 0 .. J);
J0 := 24;
Approximation := signum(a[k]) * sum(c['j'] * 2j, j'= max(0, J - J0) .. J);
k := 0
z := 116601199
c = [1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1]
J := 26
a0 := 116601199
J0 := 24
Approximation := 116601196
```

(3)

Leichte Modifikation von Koeffizienten, Zugehöriges charakteristisches Polynom
(Vorsicht! Speziell gewählt für den Fall d = 12!)

```
> b[0] := a[0] + 1;
b[1] := a[1] - 5;
b[2] := a[2] - 57;
b[3] := a[3] + 16;
b[4] := a[4] - 17;
b[5] := a[5] - 1;
b[6] := a[6] + 2;
for i from 7 to d do
b[i] := a[i];
od;
chiModifiziert := lambda → sum(b['i'] * lambda'i, i'= 0 .. d) :
'chiModifiziert(lambda)' = chiModifiziert(lambda);
b0 := 116601200
```

```

b1 := -711234048
b2 := 1227435392
b3 := -1053129152
b4 := 540869504
b5 := -181301680
b6 := 41449696
b7 := -6590064
b8 := 729048
b9 := -55055
b10 := 2706
b11 := -78
b12 := 1

```

$$\begin{aligned}
\text{chiModifiziert}(\lambda) = & \lambda^{12} - 78 \lambda^{11} + 2706 \lambda^{10} - 55055 \lambda^9 + 729048 \lambda^8 - 6590064 \lambda^7 \\
& + 41449696 \lambda^6 - 181301680 \lambda^5 + 540869504 \lambda^4 - 1053129152 \lambda^3 + 1227435392 \lambda^2 \\
& - 711234048 \lambda + 116601200
\end{aligned} \tag{4}$$

Änderung der Koeffizienten, Relativer Fehler $|b_i - a_i|/|a_i|$

```

> for i from 0 to d do
  RelativerFehler[i] := evalf(abs(b[i]-a[i])/abs(a[i]));
od;

```

```

RelativerFehler0 := 8.576241141 × 10-9
RelativerFehler1 := 7.030034697 × 10-9
RelativerFehler2 := 4.643828728 × 10-8
RelativerFehler3 := 1.519281821 × 10-8
RelativerFehler4 := 3.143087074 × 10-8
RelativerFehler5 := 5.515668721 × 10-9
RelativerFehler6 := 4.825126091 × 10-8
RelativerFehler7 := 0.
RelativerFehler8 := 0.
RelativerFehler9 := 0.
RelativerFehler10 := 0.
RelativerFehler11 := 0.

```

(5)

*RelativerFehler*₁₂ := 0. (5)

Gewählte Genauigkeit (Anzahl der Stellen)

> *Digits* := 20;

Digits := 20 (6)

Nullstellen des charakteristischen Polynoms (Eigenwerte)

> *Nullstellen* := *evalf*(*solve*(*chi*(*lambda*) = 0)) :

Nullstellen := *sort*([*Nullstellen*]);

Nullstellen := [0.2538058170966424315, 1.7893213526669562328, 2.9610588806941174082, (7)

3.9960479973887434338, 4.9997743236220860120, 5.9999920463969422866,

7.0000079536030577134, 8.0002256763779139880, 9.0039520026112565662,

10.038941119305882592, 11.210678647333043767, 12.746194182903357568]

Nullstellen des modifizierten charakteristischen Polynoms

> *NullstellenModifiziert* := *evalf*(*solve*(*chiModifiziert*(*lambda*) = 0)) :

NullstellenModifiziert := *sort*([*NullstellenModifiziert*]);

NullstellenModifiziert := [0.25380580266317534635, 1.7893425070603660049, (8)

2.9608597243156826114, 3.9889286969745980017, 5.1326912818009883634,

5.6153927283014304826, 11.453800408874389962, 12.714268818181762132,

7.3491378382439050165 – 0.82758926327591072723 I, 7.3491378382439050165

+ 0.82758926327591072723 I, 9.6963171776698985314 – 0.82180189154917339006 I,

9.6963171776698985314 + 0.82180189154917339006 I]

Numerischer Vergleich der Resultate

> **for** *i* **from** 1 **to** *d* **do**

print(*Nullstellen*[*i*], *NullstellenModifiziert*[*i*], *abs*(*NullstellenModifiziert*[*i*] – *Nullstellen*[*i*]));

end;

0.2538058170966424315, 0.25380580266317534635, 1.443346708515 × 10⁻⁸

1.7893213526669562328, 1.7893425070603660049, 0.0000211543934097721

2.9610588806941174082, 2.9608597243156826114, 0.0001991563784347968

3.9960479973887434338, 3.9889286969745980017, 0.0071193004141454321

4.9997743236220860120, 5.1326912818009883634, 0.1329169581789023514

5.9999920463969422866, 5.6153927283014304826, 0.3845993180955118040

7.0000079536030577134, 11.453800408874389962, 4.4537924552713322486

8.0002256763779139880, 12.714268818181762132, 4.7140431418038481440

9.0039520026112565662, 7.3491378382439050165 – 0.82758926327591072723 I,

1.8502199618640970309

10.038941119305882592, 7.3491378382439050165 + 0.82758926327591072723 I,

2.8142398049031544129

11.210678647333043767, 9.6963171776698985314 – 0.82180189154917339006 I,

1.7229767873520874831

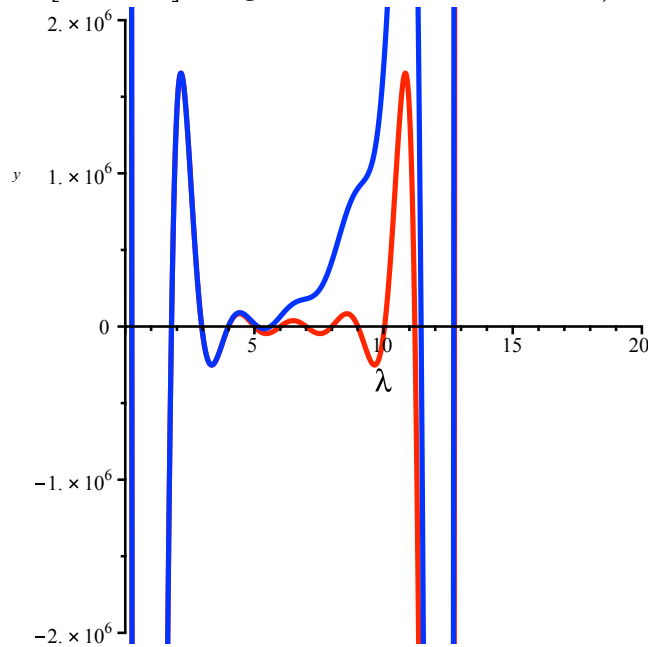
(9)

12.746194182903357568, 9.6963171776698985314 + 0.82180189154917339006 I,
 3.1586560585169148900

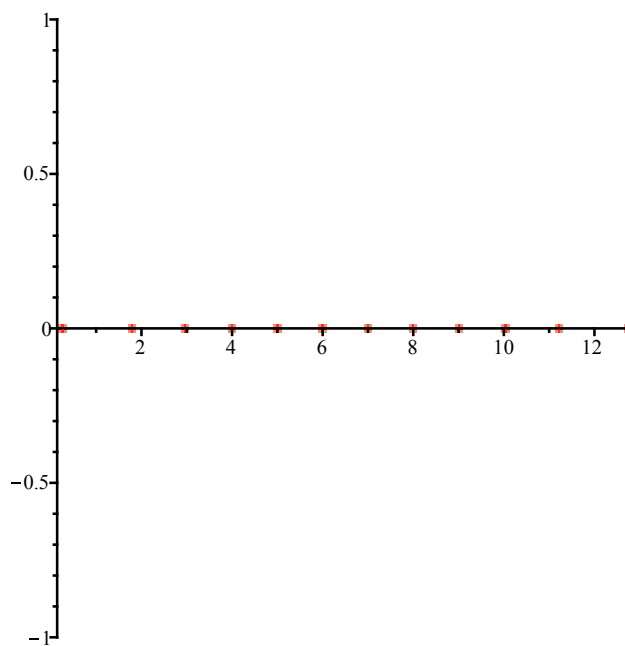
(9)

Vergleich der Graphen der Polynome und der Nullstellen (Imaginärteil versus Realteil)

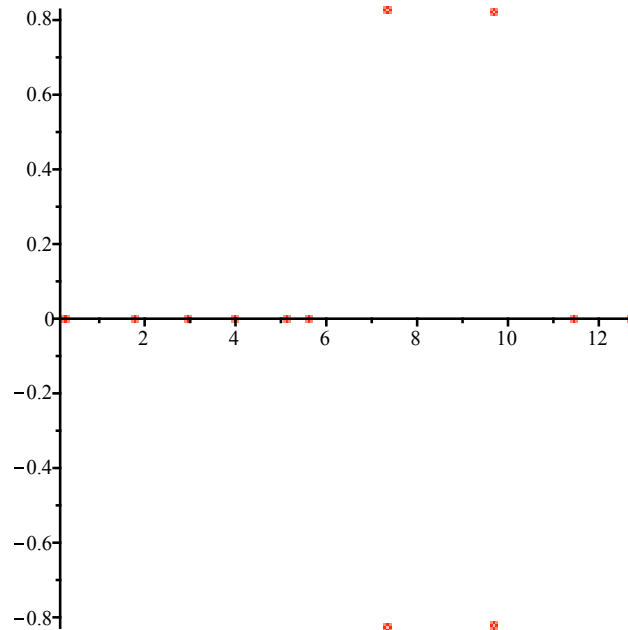
```
> plot([chi(lambda), chiModifiziert(lambda)], lambda = 0..20, y = -2 * 10^6 .. 2 * 10^6, color
    = [red, blue], style = [line, line], numpoints = 1000, thickness = 2);
```



```
> z := [[Re(Nullstellen['j']), Im(Nullstellen['j'])] $j' = 1..d];
plot(z, color = red, style = point, symbol = circle);
z := [[0.2538058170966424315, 0.], [1.7893213526669562328, 0.], [2.9610588806941174082, 0.],
    [3.9960479973887434338, 0.], [4.9997743236220860120, 0.], [5.9999920463969422866, 0.],
    [7.0000079536030577134, 0.], [8.0002256763779139880, 0.], [9.0039520026112565662, 0.],
    [10.038941119305882592, 0.], [11.210678647333043767, 0.], [12.746194182903357568, 0.]]
```



```
> zModifiziert := [[Re(NullstellenModifiziert['j']), Im(NullstellenModifiziert['j'])]$j'= 1..d];  
plot(zModifiziert, color = red, style = point, symbol = circle);  
zModifiziert := [[0.25380580266317534635, 0.], [1.7893425070603660049, 0.],  
[2.9608597243156826114, 0.], [3.9889286969745980017, 0.], [5.1326912818009883634, 0.],  
[5.6153927283014304826, 0.], [11.453800408874389962, 0.], [12.714268818181762132, 0.],  
[7.3491378382439050165, -0.82758926327591072723], [7.3491378382439050165,  
0.82758926327591072723], [9.6963171776698985314, -0.82180189154917339006],  
[9.6963171776698985314, 0.82180189154917339006]]
```



```
>  
>  
>  
>  
>  
>
```

**Modifikation der Überlegungen und Beobachtung:
Die Berechnung der Eigenwerte als Nullstellen des charakteristischen Polynomes
führt bei Matrizen höherer Dimensionen auf falsche Resultate.**

Setze beispielsweise $d = 12, 20$, $\text{eps} = 1/10^8, 1/10^{10}, 1/10^{16}$

> restart;

> with(LinearAlgebra):

Genauigkeit, Relative Störung

> Digits := 40;

$Digits := 40$ (1)

> epsilon := evalf(10^{-8});

$\epsilon := 1.000 \times 10^{-8}$ (2)

Dimension der Matrix

> d := 20;

$d := 20$ (3)

Definition der Matrix

(Ausgabe für kleinere Dimensionen)

> A := Matrix(d, d, 0):

for i from 1 to d - 1 do

A[i, i + 1] := 1;

A[i, i] := i;

A[i + 1, i] := 1;

od:

A[d, d] := d:

if d < 11 then

'A' = A;

fi;

Charakteristisches Polynom, Zugehörige Koeffizienten

> chi := lambda -> CharacteristicPolynomial(A, lambda):

'chi(lambda)' = chi(lambda);

for i from 0 to d do

a[i] := coeff(chi(lambda), lambda, i);

od;

$$\begin{aligned}\chi(\lambda) = & \lambda^{20} - 210 \lambda^{19} + 20596 \lambda^{18} - 1253259 \lambda^{17} + 53012205 \lambda^{16} - 1655131968 \lambda^{15} \\ & + 39527521712 \lambda^{14} - 738347797698 \lambda^{13} + 10938088624879 \lambda^{12} - 129531966351606 \lambda^{11} \\ & + 1230153370283045 \lambda^{10} - 9359939483770512 \lambda^9 + 56768939326385766 \lambda^8 \\ & - 271815930536588046 \lambda^7 + 1011836239345010647 \lambda^6 - 2861594301676338585 \lambda^5 \\ & + 5938910441528866024 \lambda^4 - 8567280674633006970 \lambda^3 + 7833712471890744650 \lambda^2 \\ & - 3778910598454300491 \lambda + 572669728886769911\end{aligned}$$

$$a_0 := 572669728886769911$$

$$a_1 := -3778910598454300491$$

$$\begin{aligned}
a_2 &:= 7833712471890744650 \\
a_3 &:= -8567280674633006970 \\
a_4 &:= 5938910441528866024 \\
a_5 &:= -2861594301676338585 \\
a_6 &:= 1011836239345010647 \\
a_7 &:= -271815930536588046 \\
a_8 &:= 56768939326385766 \\
a_9 &:= -9359939483770512 \\
a_{10} &:= 1230153370283045 \\
a_{11} &:= -129531966351606 \\
a_{12} &:= 10938088624879 \\
a_{13} &:= -738347797698 \\
a_{14} &:= 39527521712 \\
a_{15} &:= -1655131968 \\
a_{16} &:= 53012205 \\
a_{17} &:= -1253259 \\
a_{18} &:= 20596 \\
a_{19} &:= -210 \\
a_{20} &:= 1
\end{aligned} \tag{4}$$

Leichte Modifikation der Koeffizienten (Relativer Fehler $b_i = a_i(1 + \text{eps})$), Zugehöriges Polynom

```

> for i from 0 to d do
  b[i] := a[i] * (1 + (-1)^i * epsilon);
od:
chiModifiziert := lambda -> sum(b['i'] * lambda^('i'), 'i' = 0 .. d) :
'chiModifiziert(lambda)' = chiModifiziert(lambda);

```

$$\begin{aligned}
\text{chiModifiziert}(\lambda) &= 5.7266973461346719986769911000000000000000 \times 10^{17} \\
&- 3.7789105606651945064569950900000000000000 \times 10^{18} \lambda \\
&+ 7.8337125502278693689074465000000000000000 \times 10^{18} \lambda^2 \\
&- 8.567280588960200223669930300000000000000 \times 10^{18} \lambda^3 \\
&+ 5.9389105009179704392886602400000000000000 \times 10^{18} \lambda^4 \\
&- 2.8615942730603955682366141500000000000000 \times 10^{18} \lambda^5 \\
&+ 1.0118362494633730404501064700000000000000 \times 10^{18} \lambda^6
\end{aligned} \tag{5}$$

$$\begin{aligned}
& - 2.71815927818428740634119540000000000000000 \times 10^{17} \lambda^7 \\
& + 5.6768939894075159263857660000000000000000 \times 10^{16} \lambda^8 \\
& - 9.3599393901711171622948800000000000000000 \times 10^{15} \lambda^9 \\
& + 1.23015338258457870283045000000000000000000 \times 10^{15} \lambda^{10} \\
& - 1.29531965056286336483940000000000000000000 \times 10^{14} \lambda^{11} \\
& + 1.09380887342598862487900000000000000000000 \times 10^{13} \lambda^{12} \\
& - 7.38347790314522023020000000000000000000000 \times 10^{11} \lambda^{13} \\
& + 3.9527522107275217120000000000000000000000 \times 10^{10} \lambda^{14} \\
& - 1.65513195144868032000000000000000000000000 \times 10^9 \lambda^{15} \\
& + 5.30122055301220500000000000000000000000000 \times 10^7 \lambda^{16} \\
& - 1.25325898746741000000000000000000000000000 \times 10^6 \lambda^{17} \\
& + 20596.0002059600000000000000000000000000000 \lambda^{18} \\
& - 209.999979000000000000000000000000000000000 \lambda^{19} \\
& + 1.00000010000000000000000000000000000000000 \lambda^{20}
\end{aligned}$$

Nullstellen des charakteristischen Polynoms (Eigenwerte)

> *Nullstellen* := *evalf*(*solve*(*chi*(*lambda*) = 0)) :

Nullstellen := *sort*([*Nullstellen*]);

Nullstellen := [0.25380581709664242941311603232794425975,

(6)

1.789321352666953511672300366066380167435,
2.961058880693559110232594097363218914355,
3.996047997334638671633885431999871352700,
4.999774319814829655874727295812634621935,
5.999991841327054989535957324577387937562,
6.99999794929562199682945194668776152316,
7.99999996191873116750109421679207337617,
8.99999999945511936201645709244080343873,
9.9999999999385952891638690358916764786,
11.0000000000061404710836130964108323521,
12.00000000005448806379835429075591965613,
13.00000000380812688324989057832079266238,
14.00000020507043780031705480533122384768,
15.00000815867294501046404267542261206244,
16.00022568018517034412527270418736537806,
17.00395200266536132836611456800012864730,

18.03894111930644088976740590263678108564,
19.21067864733304648832769963393361983256,
20.74619418290335757058688396767205574025]

Verifizierung der k-ten Nullstelle mittels Hornerschema

```
> k := 1;  
Argument := Nullstellen[k];  
Funktionswert := a[d]:  
for i from 1 to d do  
Funktionswert := Funktionswert * Argument + a[d - i];  
od:  
Funktionswert := Funktionswert;  
  
k := 1  
Argument := 0.25380581709664242941311603232794425975  
Funktionswert := 2.6 × 10-21
```

(7)

Nullstellen des modifizierten charakteristischen Polynoms

```
> NullstellenModifiziert := evalf(solve(chiModifiziert(lambda) = 0)) :  
NullstellenModifiziert := sort([NullstellenModifiziert]);  
NullstellenModifiziert := [0.2538058367240358501598174762272422344279,  
1.789197123824442305407057842773802036618,  
2.987503059174360922283154898360695942208,  
3.618632365774889248924477811963241524847,  
4.366851181350590397790857841593497015118  
- 1.127827332943978373693812600662852100123 I,  
4.366851181350590397790857841593497015118  
+ 1.127827332943978373693812600662852100123 I,  
5.501730631933793053333316162561911867307  
- 2.453290561405601508668703486088823451023 I,  
5.501730631933793053333316162561911867307  
+ 2.453290561405601508668703486088823451023 I,  
6.996891117286442982709703038938042430790  
- 3.985545066067076158183697922909942693875 I,  
6.996891117286442982709703038938042430790  
+ 3.985545066067076158183697922909942693875 I,  
9.053189930035292927557073524369912388627  
- 5.667881592986541604491937290217397991118 I,  
9.053189930035292927557073524369912388627  
+ 5.667881592986541604491937290217397991118 I,  
11.96935774505686487215709753995697590024  
- 7.293154035306373116392694609846660944305 I,  
11.96935774505686487215709753995697590024
```

(8)

```

+ 7.293154035306373116392694609846660944305 I,
16.07236074660884881911199945133524312985
- 8.277188943176294719226300433957164843587 I,
16.07236074660884881911199945133524312985
+ 8.277188943176294719226300433957164843587 I,
21.21849284720851628163515460311992346754
- 7.351795791770108914057816978702575844887 I,
21.21849284720851628163515460311992346754
+ 7.351795791770108914057816978702575844887 I,
25.49655450777080750231733382346410293146
- 3.119744327637346674018044294921560025018 I,
25.49655450777080750231733382346410293146
+ 3.119744327637346674018044294921560025018 I]

```

Numerischer Vergleich der Resultate

```

> for i from 1 to d do
  print(Nullstellen[i], NullstellenModifiziert[i], abs(NullstellenModifiziert[i]-Nullstellen[i]));
end;
0.25380581709664242941311603232794425975,
  0.2538058367240358501598174762272422344279, 1.96273934207467014438992979746779
  × 10-8
1.789321352666953511672300366066380167435,
  1.789197123824442305407057842773802036618,
  0.000124228842511206265242523292578130817
2.961058880693559110232594097363218914355,
  2.987503059174360922283154898360695942208,
  0.026444178480801812050560800997477027853
3.996047997334638671633885431999871352700,
  3.618632365774889248924477811963241524847,
  0.377415631559749422709407620036629827853
4.999774319814829655874727295812634621935,
  4.366851181350590397790857841593497015118
  - 1.127827332943978373693812600662852100123 I,
  1.293285039014582440964297336189929937573
5.999991841327054989535957324577387937562,
  4.366851181350590397790857841593497015118
  + 1.127827332943978373693812600662852100123 I,
  1.984727414080807667495932302811503264407
6.999999794929562199682945194668776152316,

```


5.501730631933793053333316162561911867307
– 2.453290561405601508668703486088823451023 I,
2.874620855602674792598841673326063116248

7.999999996191873116750109421679207337617,
5.501730631933793053333316162561911867307
+ 2.453290561405601508668703486088823451023 I,
3.501426051635573661741521015672013642173

8.99999999945511936201645709244080343873,
6.996891117286442982709703038938042430790
– 3.985545066067076158183697922909942693875 I,
4.460606984418071115591803923874618900306

9.9999999999385952891638690358916764786,
6.996891117286442982709703038938042430790
+ 3.985545066067076158183697922909942693875 I,
4.990313861379962854022241718208543925575

11.00000000000061404710836130964108323521,
9.053189930035292927557073524369912388627
– 5.667881592986541604491937290217397991118 I,
5.992908409164454682247727245599260320352

12.00000000005448806379835429075591965613,
9.053189930035292927557073524369912388627
+ 5.667881592986541604491937290217397991118 I,
6.388158681567184599408167269060152759563

13.00000000380812688324989057832079266238,
11.96935774505686487215709753995697590024
– 7.293154035306373116392694609846660944305 I,
7.365617370474082480096838787028138240446

14.00000020507043780031705480533122384768,
11.96935774505686487215709753995697590024
+ 7.293154035306373116392694609846660944305 I,
7.570574917608015390190375845565347788983

15.00000815867294501046404267542261206244,
16.07236074660884881911199945133524312985
– 8.277188943176294719226300433957164843587 I,
8.346364290749160498271144841811106292063

16.00022568018517034412527270418736537806,
16.07236074660884881911199945133524312985
+ 8.277188943176294719226300433957164843587 I,

```

8.277503262992280763363129126146232160112
17.00395200266536132836611456800012864730,
21.21849284720851628163515460311992346754
- 7.351795791770108914057816978702575844887 I,
8.474152222742473985596942803548732432258
18.03894111930644088976740590263678108564,
21.21849284720851628163515460311992346754
+ 7.351795791770108914057816978702575844887 I,
8.009897037683677735026150996486798176473
19.21067864733304648832769963393361983256,
25.49655450777080750231733382346410293146
- 3.119744327637346674018044294921560025018 I,
7.017481029732788857378628022265080295519
20.74619418290335757058688396767205574025,
25.49655450777080750231733382346410293146
+ 3.119744327637346674018044294921560025018 I,
5.683196977573466503264551164725464925794

```

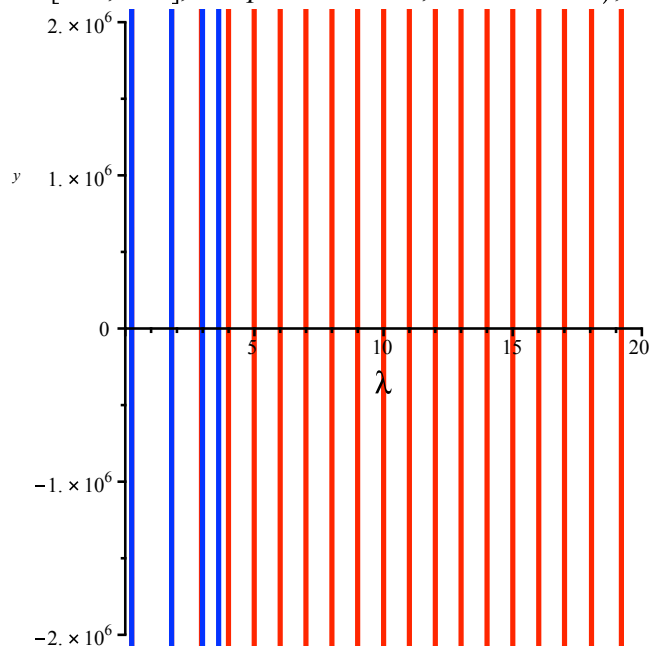
(9)

Vergleich der Graphen der Polynome und der Nullstellen (Imaginärteil versus Realteil)

```

> plot([chi(lambda), chiModifiziert(lambda)], lambda = 0..20, y = -2 * 10^6 .. 2 * 10^6, color
= [red, blue], style = [line, line], numpoints = 1000, thickness = 2);

```



```

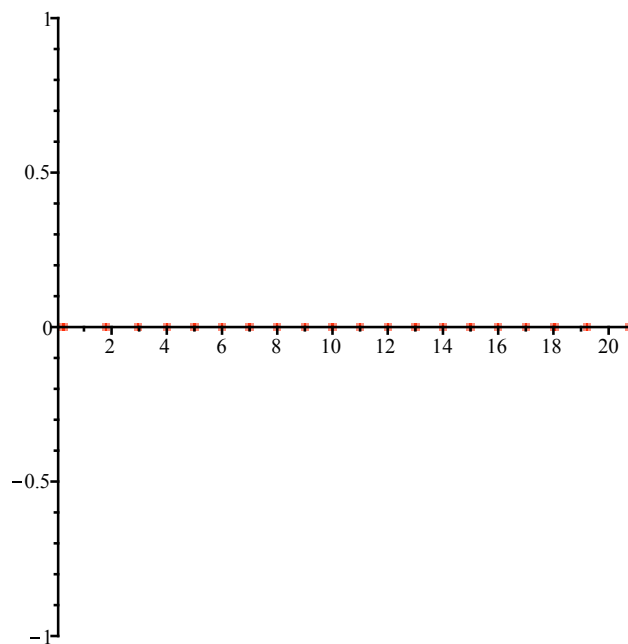
> z := [[Re(Nullstellen['j']), Im(Nullstellen['j'])]$j'= 1 .. d];
plot(z, color = red, style = point, symbol = circle);
z := [[0.25380581709664242941311603232794425975, 0.],
[1.789321352666953511672300366066380167435, 0.],
[2.961058880693559110232594097363218914355, 0.],

```

```

[3.996047997334638671633885431999871352700, 0. ],
[4.999774319814829655874727295812634621935, 0. ],
[5.999991841327054989535957324577387937562, 0. ],
[6.999999794929562199682945194668776152316, 0. ],
[7.999999996191873116750109421679207337617, 0. ],
[8.99999999945511936201645709244080343873, 0. ],
[9.9999999999385952891638690358916764786, 0. ],
[11.00000000000061404710836130964108323521, 0. ],
[12.00000000005448806379835429075591965613, 0. ],
[13.00000000380812688324989057832079266238, 0. ],
[14.00000020507043780031705480533122384768, 0. ],
[15.00000815867294501046404267542261206244, 0. ],
[16.00022568018517034412527270418736537806, 0. ],
[17.00395200266536132836611456800012864730, 0. ],
[18.03894111930644088976740590263678108564, 0. ],
[19.21067864733304648832769963393361983256, 0. ],
[20.74619418290335757058688396767205574025, 0. ]]

```



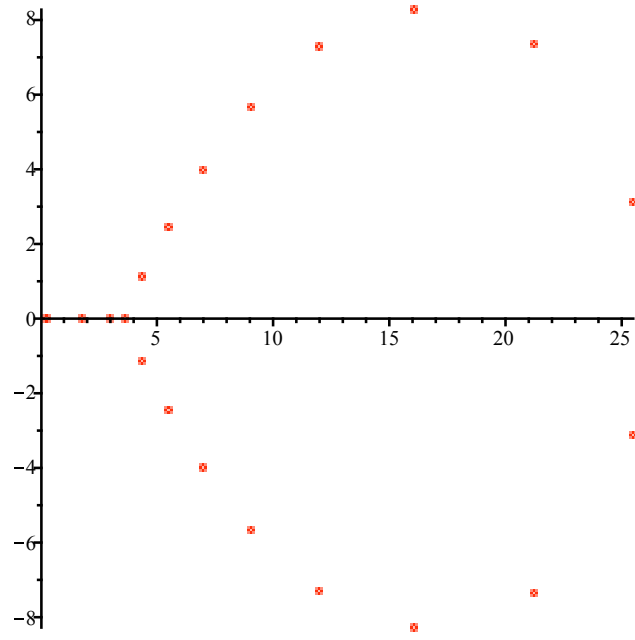
```

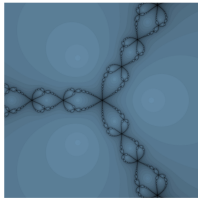
> zModifiziert := [[Re(NullstellenModifiziert['j']), Im(NullstellenModifiziert['j']) ]$j'= 1 .. d];
plot(zModifiziert, color = red, style = point, symbol = circle);
zModifiziert := [[0.2538058367240358501598174762272422344279, 0. ],
[1.789197123824442305407057842773802036618, 0. ],
[2.987503059174360922283154898360695942208, 0. ],
[3.618632365774889248924477811963241524847, 0. ],
[4.366851181350590397790857841593497015118,

```

— 1.127827332943978373693812600662852100123],
[4.366851181350590397790857841593497015118,
1.127827332943978373693812600662852100123],
[5.501730631933793053333316162561911867307,
— 2.453290561405601508668703486088823451023],
[5.501730631933793053333316162561911867307,
2.453290561405601508668703486088823451023],
[6.996891117286442982709703038938042430790,
— 3.985545066067076158183697922909942693875],
[6.996891117286442982709703038938042430790,
3.985545066067076158183697922909942693875],
[9.053189930035292927557073524369912388627,
— 5.667881592986541604491937290217397991118],
[9.053189930035292927557073524369912388627,
5.667881592986541604491937290217397991118],
[11.96935774505686487215709753995697590024,
— 7.293154035306373116392694609846660944305],
[11.96935774505686487215709753995697590024,
7.293154035306373116392694609846660944305],
[16.07236074660884881911199945133524312985,
— 8.277188943176294719226300433957164843587],
[16.07236074660884881911199945133524312985,
8.277188943176294719226300433957164843587],
[21.21849284720851628163515460311992346754,
— 7.351795791770108914057816978702575844887],
[21.21849284720851628163515460311992346754,
7.351795791770108914057816978702575844887],
[25.49655450777080750231733382346410293146,
— 3.119744327637346674018044294921560025018],
[25.49655450777080750231733382346410293146,
3.119744327637346674018044294921560025018]]

|||
|>
|>
|>





Numerische Mathematik 1 & 2
Mechthild Thalhammer
Illustrationen (Verfahren von Bairstow)

Illustrationen zum Verfahren von Bairstow

- **Division mit Rest (MAPLE / MATLAB)**
- **Verfahren von Bairstow (MAPLE / MATLAB)**

Bairstow-Verfahren zur näherungsweise Berechnung der Nullstellen eines Polynomes

Vgl. de.wikipedia.org/wiki/Bairstowverfahren

Adaption der Bezeichnungsweise

Normierung der Leitkoeffizienten

Explizite Angabe der Bedingungen

Reduktion auf zwei lineare Gleichungen für die wesentlichen Größen d_0, d_1

(Code wegen Neudefinitionen neu starten)

```
> restart;
```

```
> Grad := 10;
```

$$\text{Grad} := 10$$

(1)

```
> # Polynom, dessen Nullstellen berechnet werden
```

```
p[Grad] := 1;
```

```
P := sum(p[j]·xj, j = 0.. Grad);
```

$$p_{10} := 1$$
$$P := x^{10} + p_9 x^9 + p_8 x^8 + p_7 x^7 + p_6 x^6 + p_5 x^5 + p_4 x^4 + p_3 x^3 + p_2 x^2 + p_1 x + p_0$$

(2)

```
> # Idee: Faktorisierung von P, jedoch nicht in (möglicherweise komplexe) Linearfaktoren, sondern in (reelle) quadratische Polynome.
```

```
# Für jedes quadratische Polynom können die (konjugierten) Nullstellen mittels Satz von Vieta leicht berechnet werden.
```

```
# Daher: Ansatz mit quadratischem Polynom A
```

```
# P = A B + R (Division mit Rest, R(x) = r0 + r1 x)
```

```
# Zusätzlicher Ansatz
```

```
# B = C A + S (Division mit Rest, S(x) = s0 + s1 x)
```

```
a[2] := 1;
```

```
A := sum(a[j]·xj, j = 0.. 2);
```

```
b[Grad - 2] := 1;
```

```
B := sum(b[j]·xj, j = 0.. Grad - 2);
```

```
c[Grad - 4] := 1;
```

```
C := sum(c[j]·xj, j = 0.. Grad - 4);
```

$$a_2 := 1$$
$$A := x^2 + a_1 x + a_0$$
$$b_8 := 1$$
$$B := x^8 + b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0$$
$$c_6 := 1$$
$$C := x^6 + c_5 x^5 + c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0$$

(3)

```
> # Division mit Rest
```

```
# R Polynom vom Grad 1
```

```
# Koeffizienten von P durch Koeffizienten von A,B,R ausdrücken
```

```
R := expand(P - A·B);
```

```
for k from 0 to 1 do
```

```
Aux := coeff(R, x, k) = r[k];
```



```

p[k] := solve(Aux, p[k]);
od;
for k from 2 to Grad - 1 do
Aux := coeff(R, x, k) = 0;
p[k] := solve(Aux, p[k]);
od;

```

$$\begin{aligned}
R := & -x^7 a_1 b_6 - x^4 a_0 b_4 - x^7 a_0 b_7 - x^3 a_1 b_2 - x a_1 b_0 - x a_0 b_1 - x^2 a_0 b_2 - x^4 a_1 b_3 \\
& - x^3 a_0 b_3 - x^8 a_1 b_7 - x^2 a_1 b_1 - x^5 a_1 b_4 - x^6 a_1 b_5 - x^5 a_0 b_5 - x^6 a_0 b_6 - x^9 a_1 - x^9 b_7 \\
& - x^8 a_0 - x^8 b_6 - x^7 b_5 - x^6 b_4 - x^5 b_3 - x^4 b_2 - x^3 b_1 - x^2 b_0 - a_0 b_0 + p_0 + p_2 x^2 \\
& + p_3 x^3 + p_4 x^4 + p_5 x^5 + p_6 x^6 + p_7 x^7 + p_8 x^8 + p_9 x^9 + p_1 x
\end{aligned}$$

$$Aux := -a_0 b_0 + p_0 = r_0$$

$$p_0 := a_0 b_0 + r_0$$

$$Aux := -a_0 b_1 - a_1 b_0 + p_1 = r_1$$

$$p_1 := a_0 b_1 + a_1 b_0 + r_1$$

$$Aux := -a_0 b_2 - a_1 b_1 - b_0 + p_2 = 0$$

$$p_2 := a_0 b_2 + a_1 b_1 + b_0$$

$$Aux := -a_0 b_3 - a_1 b_2 - b_1 + p_3 = 0$$

$$p_3 := a_0 b_3 + a_1 b_2 + b_1$$

$$Aux := -a_0 b_4 - a_1 b_3 - b_2 + p_4 = 0$$

$$p_4 := a_0 b_4 + a_1 b_3 + b_2$$

$$Aux := -a_0 b_5 - a_1 b_4 - b_3 + p_5 = 0$$

$$p_5 := a_0 b_5 + a_1 b_4 + b_3$$

$$Aux := -a_0 b_6 - a_1 b_5 - b_4 + p_6 = 0$$

$$p_6 := a_0 b_6 + a_1 b_5 + b_4$$

$$Aux := -a_0 b_7 - a_1 b_6 - b_5 + p_7 = 0$$

$$p_7 := a_0 b_7 + a_1 b_6 + b_5$$

$$Aux := -a_1 b_7 - a_0 - b_6 + p_8 = 0$$

$$p_8 := a_1 b_7 + a_0 + b_6$$

$$Aux := -a_1 - b_7 + p_9 = 0$$

$$p_9 := a_1 + b_7$$

(4)

```
> simplify(P);
```

$$x^{10} + x^9 (a_1 + b_7) + (a_1 b_7 + a_0 + b_6) x^8 + (a_0 b_7 + a_1 b_6 + b_5) x^7 + (a_0 b_6 + a_1 b_5$$

(5)

$$+ b_4) x^6 + (a_0 b_5 + a_1 b_4 + b_3) x^5 + (a_0 b_4 + a_1 b_3 + b_2) x^4 + (a_0 b_3 + a_1 b_2 + b_1) x^3 \\ + (a_0 b_2 + a_1 b_1 + b_0) x^2 + (a_0 b_1 + a_1 b_0 + r_1) x + a_0 b_0 + r_0$$

> # Analoge Vorgehensweise für B

$S := \text{expand}(B - A \cdot C);$

for k from 0 to 1 do

$\text{Bedingung} := \text{coeff}(S, x, k) = s[k];$

$b[k] := \text{solve}(\text{Bedingung}, b[k]);$

od;

for k from 2 to Grad - 3 do

$\text{Bedingung} := \text{coeff}(S, x, k) = 0;$

$b[k] := \text{solve}(\text{Bedingung}, b[k]);$

od;

$$S := -x^7 a_1 + x^7 b_7 - x^7 c_5 - x^6 a_1 c_5 - x^6 a_0 + x^6 b_6 - x^6 c_4 - x^5 a_0 c_5 - x^5 a_1 c_4 + x^5 b_5 \\ - x^5 c_3 - x^4 a_0 c_4 - x^4 a_1 c_3 + x^4 b_4 - x^4 c_2 - x^3 a_0 c_3 - x^3 a_1 c_2 + x^3 b_3 - x^3 c_1 - x^2 a_0 c_2 \\ - x^2 a_1 c_1 + x^2 b_2 - x^2 c_0 - x a_0 c_1 - x a_1 c_0 + x b_1 - a_0 c_0 + b_0$$

$$\text{Bedingung} := -a_0 c_0 + b_0 = s_0$$

$$b_0 := a_0 c_0 + s_0$$

$$\text{Bedingung} := -a_0 c_1 - a_1 c_0 + b_1 = s_1$$

$$b_1 := a_0 c_1 + a_1 c_0 + s_1$$

$$\text{Bedingung} := -a_0 c_2 - a_1 c_1 + b_2 - c_0 = 0$$

$$b_2 := a_0 c_2 + a_1 c_1 + c_0$$

$$\text{Bedingung} := -a_0 c_3 - a_1 c_2 + b_3 - c_1 = 0$$

$$b_3 := a_0 c_3 + a_1 c_2 + c_1$$

$$\text{Bedingung} := -a_0 c_4 - a_1 c_3 + b_4 - c_2 = 0$$

$$b_4 := a_0 c_4 + a_1 c_3 + c_2$$

$$\text{Bedingung} := -a_0 c_5 - a_1 c_4 + b_5 - c_3 = 0$$

$$b_5 := a_0 c_5 + a_1 c_4 + c_3$$

$$\text{Bedingung} := -a_1 c_5 - a_0 + b_6 - c_4 = 0$$

$$b_6 := a_1 c_5 + a_0 + c_4$$

$$\text{Bedingung} := b_7 - a_1 - c_5 = 0$$

$$b_7 := a_1 + c_5$$

(6)

> $\text{simplify}(P);$

$$x^{10} + x^9 (2 a_1 + c_5) + (a_1^2 + 2 a_1 c_5 + 2 a_0 + c_4) x^8 + (a_1^2 c_5 + (2 c_4 + 2 a_0) a_1 + 2 a_0 c_5 \\ + c_3) x^7 + (a_1^2 c_4 + (2 a_0 c_5 + 2 c_3) a_1 + 2 a_0 c_4 + a_0^2 + c_2) x^6 + (a_1^2 c_3 + (2 a_0 c_4$$

(7)

$$\begin{aligned}
& + 2c_2) a_1 + a_0^2 c_5 + 2a_0 c_3 + c_1) x^5 + (a_1^2 c_2 + (2a_0 c_3 + 2c_1) a_1 + a_0^2 c_4 + 2a_0 c_2 \\
& + c_0) x^4 + (a_1^2 c_1 + (2a_0 c_2 + 2c_0) a_1 + a_0^2 c_3 + 2a_0 c_1 + s_1) x^3 + (a_1^2 c_0 + (2a_0 c_1 \\
& + s_1) a_1 + a_0^2 c_2 + 2a_0 c_0 + s_0) x^2 + ((2a_0 c_0 + s_0) a_1 + a_0^2 c_1 + a_0 s_1 + r_1) x + a_0^2 c_0 \\
& + a_0 s_0 + r_0
\end{aligned}$$

> # Ziel wäre Faktorisierung von P

$$\# P = (A + D)(B + E)$$

A vorgegeben, B damit bestimmt durch Division mit Rest

D, E (Wunsch: kleine) Änderungen von A, B

Vernachlässigung von Termen höherer Ordnung DE

$$\# P = AB + AE + DB$$

$$DD := \text{sum}(d[j] \cdot x^j, j = 0..1);$$

$$E := \text{sum}(e[j] \cdot x^j, j = 0.. \text{Grad} - 3);$$

$$\text{Differenz} := \text{expand}(P - A \cdot B - A \cdot E - DD \cdot B);$$

$$DD := d_1 x + d_0$$

$$E := e_7 x^7 + e_6 x^6 + e_5 x^5 + e_4 x^4 + e_3 x^3 + e_2 x^2 + e_1 x + e_0$$

$$\text{Differenz} := -x^7 a_1 c_5 d_1 - x^6 a_0 c_5 d_1 - x^6 a_1 c_4 d_1 - x^6 a_1 c_5 d_0 - x^5 a_0 c_4 d_1 - x^5 a_0 c_5 d_0 \quad (8)$$

$$- x^5 a_1 c_3 d_1 - x^5 a_1 c_4 d_0 - x^4 a_0 c_3 d_1 - x^4 a_0 c_4 d_0 - x^4 a_1 c_2 d_1 - x^4 a_1 c_3 d_0$$

$$- x^3 a_0 c_2 d_1 - x^3 a_0 c_3 d_0 - x^3 a_1 c_1 d_1 - x^3 a_1 c_2 d_0 - x^2 a_0 c_1 d_1 - x^2 a_0 c_2 d_0$$

$$- x^2 a_1 c_0 d_1 - x^2 a_1 c_1 d_0 - x a_0 c_0 d_1 - x a_0 c_1 d_0 - x a_1 c_0 d_0 + r_0 - x^9 e_7 - x^8 e_6 - x^7 e_5$$

$$- x^6 e_4 - x^5 e_3 - x^4 e_2 - x^3 e_1 - x^2 e_0 - a_0 e_0 - x^9 d_1 - x^8 d_0 - d_0 s_0 - x^2 c_0 d_0 - x^6 a_0 e_6$$

$$- x^6 c_4 d_0 - x^7 a_1 e_6 - x^6 a_1 e_5 - x^4 c_1 d_1 - x^7 c_5 d_0 - x^6 a_0 d_0 - x^4 a_0 e_4 - x a_1 e_0$$

$$- x^2 a_0 e_2 - x^8 a_1 d_1 - x^2 d_1 s_1 - x d_0 s_1 - x d_1 s_0 - a_0 c_0 d_0 - x^7 a_0 d_1 - x^4 c_2 d_0$$

$$- x^3 a_1 e_2 - x^7 c_4 d_1 - x^2 a_1 e_1 - x^8 a_1 e_7 - x^3 c_0 d_1 - x^5 c_2 d_1 - x^3 a_0 e_3 - x^5 a_1 e_4$$

$$- x^8 c_5 d_1 - x a_0 e_1 - x^7 a_1 d_0 - x^6 c_3 d_1 - x^7 a_0 e_7 - x^4 a_1 e_3 - x^3 c_1 d_0 - x^5 c_3 d_0$$

$$- x^5 a_0 e_5 + x r_1$$

> # Alle erhaltenen Bedingungen

for k from 0 to Grad do

$$\text{Coeffs}[k] := \text{expand}(\text{coeff}(\text{Differenz}, x, k));$$

od;

$$\text{Coeffs}_0 := -a_0 c_0 d_0 - a_0 e_0 - d_0 s_0 + r_0$$

$$\text{Coeffs}_1 := -a_0 c_0 d_1 - a_0 c_1 d_0 - a_1 c_0 d_0 - a_0 e_1 - a_1 e_0 - d_0 s_1 - d_1 s_0 + r_1$$

$$\text{Coeffs}_2 := -a_0 c_1 d_1 - a_0 c_2 d_0 - a_1 c_0 d_1 - a_1 c_1 d_0 - a_0 e_2 - a_1 e_1 - c_0 d_0 - d_1 s_1 - e_0$$

$$\text{Coeffs}_3 := -a_0 c_2 d_1 - a_0 c_3 d_0 - a_1 c_1 d_1 - a_1 c_2 d_0 - a_0 e_3 - a_1 e_2 - c_0 d_1 - c_1 d_0 - e_1$$

$$\begin{aligned} \text{Coeffs}_4 &:= -a_0 c_3 d_1 - a_0 c_4 d_0 - a_1 c_2 d_1 - a_1 c_3 d_0 - a_0 e_4 - a_1 e_3 - c_1 d_1 - c_2 d_0 - e_2 \\ \text{Coeffs}_5 &:= -a_0 c_4 d_1 - a_0 c_5 d_0 - a_1 c_3 d_1 - a_1 c_4 d_0 - a_0 e_5 - a_1 e_4 - c_2 d_1 - c_3 d_0 - e_3 \\ \text{Coeffs}_6 &:= -a_0 c_5 d_1 - a_1 c_4 d_1 - a_1 c_5 d_0 - a_0 d_0 - a_0 e_6 - a_1 e_5 - c_3 d_1 - c_4 d_0 - e_4 \\ \text{Coeffs}_7 &:= -a_1 c_5 d_1 - a_0 d_1 - a_0 e_7 - a_1 d_0 - a_1 e_6 - c_4 d_1 - c_5 d_0 - e_5 \\ \text{Coeffs}_8 &:= -a_1 d_1 - a_1 e_7 - c_5 d_1 - d_0 - e_6 \\ \text{Coeffs}_9 &:= -e_7 - d_1 \\ \text{Coeffs}_{10} &:= 0 \end{aligned}$$

(9)

```

> # Schrittweise Elimination der Koeffizienten von E
# (aufgrund der speziellen Struktur der Gleichungen möglich)
j := Grad - 3;
k := Grad - 1;
for counter from 1 to Grad - 2 do
  Aux := simplify(Coeffs[k]);
  e[j] := solve(Aux, e[j]);
  j := j - 1;
  k := k - 1;
od;

      j := 7
      k := 9
      Aux := -e7 - d1
      e7 := -d1
      j := 6
      k := 8
      Aux := -c5 d1 - d0 - e6
      e6 := -c5 d1 - d0
      j := 5
      k := 7
      Aux := -c4 d1 - c5 d0 - e5
      e5 := -c4 d1 - c5 d0
      j := 4
      k := 6
      Aux := -c3 d1 - c4 d0 - e4
      e4 := -c3 d1 - c4 d0
      j := 3
      k := 5
      Aux := -c2 d1 - c3 d0 - e3

```

```

e3 := -c2 d1 - c3 d0
j := 2
k := 4
Aux := -c1 d1 - c2 d0 - e2
e2 := -c1 d1 - c2 d0
j := 1
k := 3
Aux := -c0 d1 - c1 d0 - e1
e1 := -c0 d1 - c1 d0
j := 0
k := 2
Aux := -c0 d0 - d1 s1 - e0
e0 := -c0 d0 - d1 s1
j := -1
k := 1

```

(10)

```

> # Verbleibende Gleichungen
for k from 0 to Grad do
simplify(Coeffs[k]);
od;

```

```

a0 d1 s1 - d0 s0 + r0
(a1 s1 - s0) d1 - d0 s1 + r1
0
0
0
0
0
0
0
0
0
0
0

```

(11)

```

> # Gleichungen für wesentliche Größen d0,d1
# gegeben durch vorgegebene Größen a0,a1
# und in Folge berechenbare Größen r0,r1,s0,s1
Gleichung_d0d1[1] := simplify(Coeffs[0]);
Gleichung_d0d1[2] := simplify(Coeffs[1]);
solve({Gleichung_d0d1[1], Gleichung_d0d1[2]}, {d[0], d[1]});
Gleichung_d0d1_1 := a0 d1 s1 - d0 s0 + r0

```

$$\begin{aligned} \text{Gleichung_d0d1}_2 &:= (a_1 s_1 - s_0) d_1 - d_0 s_1 + r_1 \\ \left\{ d_0 = \frac{a_0 r_1 s_1 - a_1 r_0 s_1 + r_0 s_0}{a_0 s_1^2 - a_1 s_0 s_1 + s_0^2}, d_1 = -\frac{r_0 s_1 - r_1 s_0}{a_0 s_1^2 - a_1 s_0 s_1 + s_0^2} \right\} \end{aligned} \quad (12)$$

Beispiel (Vgl. Wikipedia)

Bestimmung eines quadratischen Polynomes, das p teilt

Beobachtung: Quadratische Konvergenz

> restart;

> Digits := 40;

Digits := 40 (13)

> p := 6·x⁵ + 11·x⁴ - 33·x³ - 33·x² + 11·x + 6;

Grad := 5;

p := $\frac{p}{\text{coeff}(p, x, \text{Grad})}$;

$$\begin{aligned} p &:= 6x^5 + 11x^4 - 33x^3 - 33x^2 + 11x + 6 \\ \text{Grad} &:= 5 \end{aligned}$$

$$p := x^5 + \frac{11}{6}x^4 - \frac{11}{2}x^3 - \frac{11}{2}x^2 + \frac{11}{6}x + 1 \quad (14)$$

> roots(p);

$$\left[[-1, 1], [-3, 1], [2, 1], \left[-\frac{1}{3}, 1\right], \left[\frac{1}{2}, 1\right] \right] \quad (15)$$

> Nullstelle[1] := -1;

Nullstelle[2] := -3;

Nullstelle[3] := 2;

Nullstelle[4] := $-\frac{1}{3}$;

Nullstelle[5] := $\frac{1}{2}$;

p = expand(product(x - Nullstelle[j], j = 1 .. 5));

Nullstelle₁ := -1

Nullstelle₂ := -3

Nullstelle₃ := 2

Nullstelle₄ := $-\frac{1}{3}$

Nullstelle₅ := $\frac{1}{2}$

$$x^5 + \frac{11}{6}x^4 - \frac{11}{2}x^3 - \frac{11}{2}x^2 + \frac{11}{6}x + 1 = x^5 + \frac{11}{6}x^4 - \frac{11}{2}x^3 - \frac{11}{2}x^2 + \frac{11}{6}x + 1 \quad (16)$$

Start (Vorgabe der Koeffizienten a0, a1 bzw. d0, d1)

Iteration

$$B = \frac{p_6 x^4}{a_2} - \frac{(a_1 p_6 - a_2 p_5) x^3}{a_2^2} - \frac{(a_0 a_2 p_6 - a_1^2 p_6 + a_1 a_2 p_5 - a_2^2 p_4) x^2}{a_2^3} \quad (21)$$

$$+ \frac{(2 a_0 a_1 a_2 p_6 - a_0 a_2^2 p_5 - a_1^3 p_6 + a_1^2 a_2 p_5 - a_1 a_2^2 p_4 + a_2^3 p_3) x}{a_2^4} + \frac{1}{a_2^5} (a_0^2 a_2 p_6 - 3 a_0 a_1^2 a_2 p_6 + 2 a_0 a_1 a_2^2 p_5 - a_0 a_2^3 p_4 + a_1^4 p_6 - a_1^3 a_2 p_5 + a_1^2 a_2^2 p_4 - a_1 a_2^3 p_3 + a_2^4 p_2)$$

> 'R' = rem(P, A, x);

$$R = -\frac{1}{a_2^5} \left((3 a_0^2 a_1 a_2^2 p_6 - a_0^2 a_2^3 p_5 - 4 a_0 a_1^3 a_2 p_6 + 3 a_0 a_1^2 a_2^2 p_5 - 2 a_0 a_1 a_2^3 p_4 + a_0 a_2^4 p_3 + a_1^5 p_6 - a_1^4 a_2 p_5 + a_1^3 a_2^2 p_4 - a_1^2 a_2^3 p_3 + a_1 a_2^4 p_2 - a_2^5 p_1) x - \frac{1}{a_2^5} (a_0^3 a_2^2 p_6 - 3 a_0^2 a_1^2 a_2 p_6 + 2 a_0^2 a_1 a_2^2 p_5 - a_0^2 a_2^3 p_4 + a_0 a_1^4 p_6 - a_0 a_1^3 a_2 p_5 + a_0 a_1^2 a_2^2 p_4 - a_0 a_1 a_2^3 p_3 + a_0 a_2^4 p_2 - a_2^5 p_0) \right) \quad (22)$$

> # Division mit Rest

R Polynom vom Grad l

Koeffizienten von P durch Koeffizienten von A,B,R ausdrücken

Sehr einfache Form der Koeffizienten r0, r1

R := expand(P - A·B);

for k from 0 to 1 do

Aux := coeff(R, x, k) = r[k];

p[k] := solve(Aux, p[k]);

od;

for k from 2 to Grad - 1 do

Aux := coeff(R, x, k) = 0;

p[k] := solve(Aux, p[k]);

od;

$$R := -x^6 a_2 b_4 + p_6 x^6 - x^5 a_1 b_4 - x^5 a_2 b_3 + p_5 x^5 - x^4 a_0 b_4 - x^4 a_1 b_3 - x^4 a_2 b_2 + p_4 x^4$$

$$- x^3 a_0 b_3 - x^3 a_1 b_2 - x^3 a_2 b_1 + p_3 x^3 - x^2 a_0 b_2 - x^2 a_1 b_1 - x^2 a_2 b_0 + p_2 x^2 - x a_0 b_1$$

$$- x a_1 b_0 + p_1 x - a_0 b_0 + p_0$$

$$Aux := -a_0 b_0 + p_0 = r_0$$

$$p_0 := a_0 b_0 + r_0$$

$$Aux := -a_0 b_1 - a_1 b_0 + p_1 = r_1$$

$$p_1 := a_0 b_1 + a_1 b_0 + r_1$$

$$Aux := -a_0 b_2 - a_1 b_1 - a_2 b_0 + p_2 = 0$$

$$p_2 := a_0 b_2 + a_1 b_1 + a_2 b_0$$

$$\begin{aligned}
Aux &:= -a_0 b_3 - a_1 b_2 - a_2 b_1 + p_3 = 0 \\
p_3 &:= a_0 b_3 + a_1 b_2 + a_2 b_1 \\
Aux &:= -a_0 b_4 - a_1 b_3 - a_2 b_2 + p_4 = 0 \\
p_4 &:= a_0 b_4 + a_1 b_3 + a_2 b_2 \\
Aux &:= -a_1 b_4 - a_2 b_3 + p_5 = 0 \\
p_5 &:= a_1 b_4 + a_2 b_3
\end{aligned} \tag{23}$$

Einfache Testbeispiele (Division mit Rest, Nullstellen)

> restart;

> A := 1 + 2·x + 3·x²;

B := 4 + 5·x + 6·x² + 7·x³;

R := -1 - 2·x;

$$A := 3x^2 + 2x + 1$$

$$B := 7x^3 + 6x^2 + 5x + 4$$

$$R := -1 - 2x$$

(24)

> P := expand(A·B + R);

$$P := 21x^5 + 32x^4 + 34x^3 + 28x^2 + 11x + 3$$

(25)

> restart;

> P := expand((x + 2)·(x + 1)·x·(x - 1)·(x - 2));

$$P := x^5 - 5x^3 + 4x$$

(26)

> P := expand((x + 2)·(x - 2)·(x + I)·(x - I)·(x + 1)²);

$$P := x^6 + 2x^5 - 2x^4 - 6x^3 - 7x^2 - 8x - 4$$

(27)

>

>

>

Bairstow-Verfahren zur näherungsweise Berechnung der Nullstellen eines Polynomes

Vgl. de.wikipedia.org/wiki/Bairstowverfahren

Adaption der Bezeichnungsweise

Normierung der Leitkoeffizienten

Explizite Angabe der Bedingungen

Reduktion auf zwei lineare Gleichungen für die wesentlichen Größen d_0, d_1

(Code wegen Neudefinitionen neu starten)

```
> restart;
```

```
> Grad := 10;
```

$$\text{Grad} := 10$$

(1)

```
> # Polynom, dessen Nullstellen berechnet werden
```

```
p[Grad] := 1;
```

```
P := sum(p[j]·xj, j = 0.. Grad);
```

$$p_{10} := 1$$
$$P := x^{10} + p_9 x^9 + p_8 x^8 + p_7 x^7 + p_6 x^6 + p_5 x^5 + p_4 x^4 + p_3 x^3 + p_2 x^2 + p_1 x + p_0$$

(2)

```
> # Idee: Faktorisierung von P, jedoch nicht in (möglicherweise komplexe) Linearfaktoren, sondern in (reelle) quadratische Polynome.
```

```
# Für jedes quadratische Polynom können die (konjugierten) Nullstellen mittels Satz von Vieta leicht berechnet werden.
```

```
# Daher: Ansatz mit quadratischem Polynom A
```

```
# P = A B + R (Division mit Rest, R(x) = r0 + r1 x)
```

```
# Zusätzlicher Ansatz
```

```
# B = C A + S (Division mit Rest, S(x) = s0 + s1 x)
```

```
a[2] := 1;
```

```
A := sum(a[j]·xj, j = 0.. 2);
```

```
b[Grad - 2] := 1;
```

```
B := sum(b[j]·xj, j = 0.. Grad - 2);
```

```
c[Grad - 4] := 1;
```

```
C := sum(c[j]·xj, j = 0.. Grad - 4);
```

$$a_2 := 1$$
$$A := x^2 + a_1 x + a_0$$
$$b_8 := 1$$
$$B := x^8 + b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0$$
$$c_6 := 1$$
$$C := x^6 + c_5 x^5 + c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0$$

(3)

```
> # Division mit Rest
```

```
# R Polynom vom Grad 1
```

```
# Koeffizienten von P durch Koeffizienten von A,B,R ausdrücken
```

```
R := expand(P - A·B);
```

```
for k from 0 to 1 do
```

```
Aux := coeff(R, x, k) = r[k];
```

```

p[k] := solve(Aux, p[k]);
od;
for k from 2 to Grad - 1 do
Aux := coeff(R, x, k) = 0;
p[k] := solve(Aux, p[k]);
od;

```

$$\begin{aligned}
R := & -x^7 a_1 b_6 - x^4 a_0 b_4 - x^7 a_0 b_7 - x^3 a_1 b_2 - x a_1 b_0 - x a_0 b_1 - x^2 a_0 b_2 - x^4 a_1 b_3 \\
& - x^3 a_0 b_3 - x^8 a_1 b_7 - x^2 a_1 b_1 - x^5 a_1 b_4 - x^6 a_1 b_5 - x^5 a_0 b_5 - x^6 a_0 b_6 - x^9 a_1 - x^9 b_7 \\
& - x^8 a_0 - x^8 b_6 - x^7 b_5 - x^6 b_4 - x^5 b_3 - x^4 b_2 - x^3 b_1 - x^2 b_0 - a_0 b_0 + p_0 + p_2 x^2 \\
& + p_3 x^3 + p_4 x^4 + p_5 x^5 + p_6 x^6 + p_7 x^7 + p_8 x^8 + p_9 x^9 + p_1 x
\end{aligned}$$

$$Aux := -a_0 b_0 + p_0 = r_0$$

$$p_0 := a_0 b_0 + r_0$$

$$Aux := -a_0 b_1 - a_1 b_0 + p_1 = r_1$$

$$p_1 := a_0 b_1 + a_1 b_0 + r_1$$

$$Aux := -a_0 b_2 - a_1 b_1 - b_0 + p_2 = 0$$

$$p_2 := a_0 b_2 + a_1 b_1 + b_0$$

$$Aux := -a_0 b_3 - a_1 b_2 - b_1 + p_3 = 0$$

$$p_3 := a_0 b_3 + a_1 b_2 + b_1$$

$$Aux := -a_0 b_4 - a_1 b_3 - b_2 + p_4 = 0$$

$$p_4 := a_0 b_4 + a_1 b_3 + b_2$$

$$Aux := -a_0 b_5 - a_1 b_4 - b_3 + p_5 = 0$$

$$p_5 := a_0 b_5 + a_1 b_4 + b_3$$

$$Aux := -a_0 b_6 - a_1 b_5 - b_4 + p_6 = 0$$

$$p_6 := a_0 b_6 + a_1 b_5 + b_4$$

$$Aux := -a_0 b_7 - a_1 b_6 - b_5 + p_7 = 0$$

$$p_7 := a_0 b_7 + a_1 b_6 + b_5$$

$$Aux := -a_1 b_7 - a_0 - b_6 + p_8 = 0$$

$$p_8 := a_1 b_7 + a_0 + b_6$$

$$Aux := -a_1 - b_7 + p_9 = 0$$

$$p_9 := a_1 + b_7$$

(4)

```
> simplify(P);
```

$$x^{10} + x^9 (a_1 + b_7) + (a_1 b_7 + a_0 + b_6) x^8 + (a_0 b_7 + a_1 b_6 + b_5) x^7 + (a_0 b_6 + a_1 b_5$$

(5)

$$+ b_4) x^6 + (a_0 b_5 + a_1 b_4 + b_3) x^5 + (a_0 b_4 + a_1 b_3 + b_2) x^4 + (a_0 b_3 + a_1 b_2 + b_1) x^3 \\ + (a_0 b_2 + a_1 b_1 + b_0) x^2 + (a_0 b_1 + a_1 b_0 + r_1) x + a_0 b_0 + r_0$$

> # Analoge Vorgehensweise für B

$S := \text{expand}(B - A \cdot C);$

for k from 0 to 1 do

$\text{Bedingung} := \text{coeff}(S, x, k) = s[k];$

$b[k] := \text{solve}(\text{Bedingung}, b[k]);$

od;

for k from 2 to Grad - 3 do

$\text{Bedingung} := \text{coeff}(S, x, k) = 0;$

$b[k] := \text{solve}(\text{Bedingung}, b[k]);$

od;

$$S := -x^7 a_1 + x^7 b_7 - x^7 c_5 - x^6 a_1 c_5 - x^6 a_0 + x^6 b_6 - x^6 c_4 - x^5 a_0 c_5 - x^5 a_1 c_4 + x^5 b_5 \\ - x^5 c_3 - x^4 a_0 c_4 - x^4 a_1 c_3 + x^4 b_4 - x^4 c_2 - x^3 a_0 c_3 - x^3 a_1 c_2 + x^3 b_3 - x^3 c_1 - x^2 a_0 c_2 \\ - x^2 a_1 c_1 + x^2 b_2 - x^2 c_0 - x a_0 c_1 - x a_1 c_0 + x b_1 - a_0 c_0 + b_0$$

$$\text{Bedingung} := -a_0 c_0 + b_0 = s_0$$

$$b_0 := a_0 c_0 + s_0$$

$$\text{Bedingung} := -a_0 c_1 - a_1 c_0 + b_1 = s_1$$

$$b_1 := a_0 c_1 + a_1 c_0 + s_1$$

$$\text{Bedingung} := -a_0 c_2 - a_1 c_1 + b_2 - c_0 = 0$$

$$b_2 := a_0 c_2 + a_1 c_1 + c_0$$

$$\text{Bedingung} := -a_0 c_3 - a_1 c_2 + b_3 - c_1 = 0$$

$$b_3 := a_0 c_3 + a_1 c_2 + c_1$$

$$\text{Bedingung} := -a_0 c_4 - a_1 c_3 + b_4 - c_2 = 0$$

$$b_4 := a_0 c_4 + a_1 c_3 + c_2$$

$$\text{Bedingung} := -a_0 c_5 - a_1 c_4 + b_5 - c_3 = 0$$

$$b_5 := a_0 c_5 + a_1 c_4 + c_3$$

$$\text{Bedingung} := -a_1 c_5 - a_0 + b_6 - c_4 = 0$$

$$b_6 := a_1 c_5 + a_0 + c_4$$

$$\text{Bedingung} := b_7 - a_1 - c_5 = 0$$

$$b_7 := a_1 + c_5$$

(6)

> $\text{simplify}(P);$

$$x^{10} + x^9 (2 a_1 + c_5) + (a_1^2 + 2 a_1 c_5 + 2 a_0 + c_4) x^8 + (a_1^2 c_5 + (2 c_4 + 2 a_0) a_1 + 2 a_0 c_5 \\ + c_3) x^7 + (a_1^2 c_4 + (2 a_0 c_5 + 2 c_3) a_1 + 2 a_0 c_4 + a_0^2 + c_2) x^6 + (a_1^2 c_3 + (2 a_0 c_4$$

(7)

$$\begin{aligned}
& + 2c_2) a_1 + a_0^2 c_5 + 2a_0 c_3 + c_1) x^5 + (a_1^2 c_2 + (2a_0 c_3 + 2c_1) a_1 + a_0^2 c_4 + 2a_0 c_2 \\
& + c_0) x^4 + (a_1^2 c_1 + (2a_0 c_2 + 2c_0) a_1 + a_0^2 c_3 + 2a_0 c_1 + s_1) x^3 + (a_1^2 c_0 + (2a_0 c_1 \\
& + s_1) a_1 + a_0^2 c_2 + 2a_0 c_0 + s_0) x^2 + ((2a_0 c_0 + s_0) a_1 + a_0^2 c_1 + a_0 s_1 + r_1) x + a_0^2 c_0 \\
& + a_0 s_0 + r_0
\end{aligned}$$

> # Ziel wäre Faktorisierung von P

$$\# P = (A + D)(B + E)$$

A vorgegeben, B damit bestimmt durch Division mit Rest

D, E (Wunsch: kleine) Änderungen von A, B

Vernachlässigung von Termen höherer Ordnung DE

$$\# P = AB + AE + DB$$

$$DD := \text{sum}(d[j] \cdot x^j, j = 0..1);$$

$$E := \text{sum}(e[j] \cdot x^j, j = 0.. \text{Grad} - 3);$$

$$\text{Differenz} := \text{expand}(P - A \cdot B - A \cdot E - DD \cdot B);$$

$$DD := d_1 x + d_0$$

$$E := e_7 x^7 + e_6 x^6 + e_5 x^5 + e_4 x^4 + e_3 x^3 + e_2 x^2 + e_1 x + e_0$$

$$\text{Differenz} := -x^7 a_1 c_5 d_1 - x^6 a_0 c_5 d_1 - x^6 a_1 c_4 d_1 - x^6 a_1 c_5 d_0 - x^5 a_0 c_4 d_1 - x^5 a_0 c_5 d_0 \quad (8)$$

$$- x^5 a_1 c_3 d_1 - x^5 a_1 c_4 d_0 - x^4 a_0 c_3 d_1 - x^4 a_0 c_4 d_0 - x^4 a_1 c_2 d_1 - x^4 a_1 c_3 d_0$$

$$- x^3 a_0 c_2 d_1 - x^3 a_0 c_3 d_0 - x^3 a_1 c_1 d_1 - x^3 a_1 c_2 d_0 - x^2 a_0 c_1 d_1 - x^2 a_0 c_2 d_0$$

$$- x^2 a_1 c_0 d_1 - x^2 a_1 c_1 d_0 - x a_0 c_0 d_1 - x a_0 c_1 d_0 - x a_1 c_0 d_0 + r_0 - x^9 e_7 - x^8 e_6 - x^7 e_5$$

$$- x^6 e_4 - x^5 e_3 - x^4 e_2 - x^3 e_1 - x^2 e_0 - a_0 e_0 - x^9 d_1 - x^8 d_0 - d_0 s_0 - x^2 c_0 d_0 - x^6 a_0 e_6$$

$$- x^6 c_4 d_0 - x^7 a_1 e_6 - x^6 a_1 e_5 - x^4 c_1 d_1 - x^7 c_5 d_0 - x^6 a_0 d_0 - x^4 a_0 e_4 - x a_1 e_0$$

$$- x^2 a_0 e_2 - x^8 a_1 d_1 - x^2 d_1 s_1 - x d_0 s_1 - x d_1 s_0 - a_0 c_0 d_0 - x^7 a_0 d_1 - x^4 c_2 d_0$$

$$- x^3 a_1 e_2 - x^7 c_4 d_1 - x^2 a_1 e_1 - x^8 a_1 e_7 - x^3 c_0 d_1 - x^5 c_2 d_1 - x^3 a_0 e_3 - x^5 a_1 e_4$$

$$- x^8 c_5 d_1 - x a_0 e_1 - x^7 a_1 d_0 - x^6 c_3 d_1 - x^7 a_0 e_7 - x^4 a_1 e_3 - x^3 c_1 d_0 - x^5 c_3 d_0$$

$$- x^5 a_0 e_5 + x r_1$$

> # Alle erhaltenen Bedingungen

for k **from** 0 **to** Grad **do**

$$\text{Coeffs}[k] := \text{expand}(\text{coeff}(\text{Differenz}, x, k));$$

od;

$$\text{Coeffs}_0 := -a_0 c_0 d_0 - a_0 e_0 - d_0 s_0 + r_0$$

$$\text{Coeffs}_1 := -a_0 c_0 d_1 - a_0 c_1 d_0 - a_1 c_0 d_0 - a_0 e_1 - a_1 e_0 - d_0 s_1 - d_1 s_0 + r_1$$

$$\text{Coeffs}_2 := -a_0 c_1 d_1 - a_0 c_2 d_0 - a_1 c_0 d_1 - a_1 c_1 d_0 - a_0 e_2 - a_1 e_1 - c_0 d_0 - d_1 s_1 - e_0$$

$$\text{Coeffs}_3 := -a_0 c_2 d_1 - a_0 c_3 d_0 - a_1 c_1 d_1 - a_1 c_2 d_0 - a_0 e_3 - a_1 e_2 - c_0 d_1 - c_1 d_0 - e_1$$

$$\begin{aligned} \text{Coeffs}_4 &:= -a_0 c_3 d_1 - a_0 c_4 d_0 - a_1 c_2 d_1 - a_1 c_3 d_0 - a_0 e_4 - a_1 e_3 - c_1 d_1 - c_2 d_0 - e_2 \\ \text{Coeffs}_5 &:= -a_0 c_4 d_1 - a_0 c_5 d_0 - a_1 c_3 d_1 - a_1 c_4 d_0 - a_0 e_5 - a_1 e_4 - c_2 d_1 - c_3 d_0 - e_3 \\ \text{Coeffs}_6 &:= -a_0 c_5 d_1 - a_1 c_4 d_1 - a_1 c_5 d_0 - a_0 d_0 - a_0 e_6 - a_1 e_5 - c_3 d_1 - c_4 d_0 - e_4 \\ \text{Coeffs}_7 &:= -a_1 c_5 d_1 - a_0 d_1 - a_0 e_7 - a_1 d_0 - a_1 e_6 - c_4 d_1 - c_5 d_0 - e_5 \\ \text{Coeffs}_8 &:= -a_1 d_1 - a_1 e_7 - c_5 d_1 - d_0 - e_6 \\ \text{Coeffs}_9 &:= -e_7 - d_1 \\ \text{Coeffs}_{10} &:= 0 \end{aligned}$$

(9)

```

> # Schrittweise Elimination der Koeffizienten von E
# (aufgrund der speziellen Struktur der Gleichungen möglich)
j := Grad - 3;
k := Grad - 1;
for counter from 1 to Grad - 2 do
  Aux := simplify(Coeffs[k]);
  e[j] := solve(Aux, e[j]);
  j := j - 1;
  k := k - 1;
od;

      j := 7
      k := 9
      Aux := -e7 - d1
      e7 := -d1
      j := 6
      k := 8
      Aux := -c5 d1 - d0 - e6
      e6 := -c5 d1 - d0
      j := 5
      k := 7
      Aux := -c4 d1 - c5 d0 - e5
      e5 := -c4 d1 - c5 d0
      j := 4
      k := 6
      Aux := -c3 d1 - c4 d0 - e4
      e4 := -c3 d1 - c4 d0
      j := 3
      k := 5
      Aux := -c2 d1 - c3 d0 - e3

```



```

e3 := -c2 d1 - c3 d0
j := 2
k := 4
Aux := -c1 d1 - c2 d0 - e2
e2 := -c1 d1 - c2 d0
j := 1
k := 3
Aux := -c0 d1 - c1 d0 - e1
e1 := -c0 d1 - c1 d0
j := 0
k := 2
Aux := -c0 d0 - d1 s1 - e0
e0 := -c0 d0 - d1 s1
j := -1
k := 1

```

(10)

```

> # Verbleibende Gleichungen
for k from 0 to Grad do
simplify(Coeffs[k]);
od;

```

```

a0 d1 s1 - d0 s0 + r0
(a1 s1 - s0) d1 - d0 s1 + r1
0
0
0
0
0
0
0
0
0
0
0
0
0

```

(11)

```

> # Gleichungen für wesentliche Größen d0,d1
# gegeben durch vorgegebene Größen a0,a1
# und in Folge berechenbare Größen r0,r1,s0,s1
Gleichung_d0d1[1] := simplify(Coeffs[0]);
Gleichung_d0d1[2] := simplify(Coeffs[1]);
solve({Gleichung_d0d1[1], Gleichung_d0d1[2]}, {d[0], d[1]});
Gleichung_d0d1_1 := a0 d1 s1 - d0 s0 + r0

```

$$\begin{aligned} \text{Gleichung_d0d1}_2 &:= (a_1 s_1 - s_0) d_1 - d_0 s_1 + r_1 \\ \left\{ d_0 = \frac{a_0 r_1 s_1 - a_1 r_0 s_1 + r_0 s_0}{a_0 s_1^2 - a_1 s_0 s_1 + s_0^2}, d_1 = -\frac{r_0 s_1 - r_1 s_0}{a_0 s_1^2 - a_1 s_0 s_1 + s_0^2} \right\} \end{aligned} \quad (12)$$

Beispiel (Vgl. Wikipedia)

Bestimmung eines quadratischen Polynomes, das p teilt

Beobachtung: Quadratische Konvergenz

> restart;

> Digits := 40;

Digits := 40 (13)

> p := 6·x⁵ + 11·x⁴ - 33·x³ - 33·x² + 11·x + 6;

Grad := 5;

p := $\frac{p}{\text{coeff}(p, x, \text{Grad})}$;

$$\begin{aligned} p &:= 6x^5 + 11x^4 - 33x^3 - 33x^2 + 11x + 6 \\ \text{Grad} &:= 5 \end{aligned}$$

$$p := x^5 + \frac{11}{6}x^4 - \frac{11}{2}x^3 - \frac{11}{2}x^2 + \frac{11}{6}x + 1 \quad (14)$$

> roots(p);

$$\left[[-1, 1], [-3, 1], [2, 1], \left[-\frac{1}{3}, 1\right], \left[\frac{1}{2}, 1\right] \right] \quad (15)$$

> Nullstelle[1] := -1;

Nullstelle[2] := -3;

Nullstelle[3] := 2;

Nullstelle[4] := $-\frac{1}{3}$;

Nullstelle[5] := $\frac{1}{2}$;

p = expand(product(x - Nullstelle[j], j = 1 .. 5));

Nullstelle₁ := -1

Nullstelle₂ := -3

Nullstelle₃ := 2

Nullstelle₄ := $-\frac{1}{3}$

Nullstelle₅ := $\frac{1}{2}$

$$x^5 + \frac{11}{6}x^4 - \frac{11}{2}x^3 - \frac{11}{2}x^2 + \frac{11}{6}x + 1 = x^5 + \frac{11}{6}x^4 - \frac{11}{2}x^3 - \frac{11}{2}x^2 + \frac{11}{6}x + 1 \quad (16)$$

Start (Vorgabe der Koeffizienten a0, a1 bzw. d0, d1)

Iteration

```

> # a
a0 :=  $\frac{\text{coeff}(p, x, \text{Grad} - 2)}{\text{coeff}(p, x, \text{Grad})}$ ;
a1 :=  $\frac{\text{coeff}(p, x, \text{Grad} - 1)}{\text{coeff}(p, x, \text{Grad})}$ ;
d0 := 0;
d1 := 0;

LoopMax := 10;
for loop from 1 to LoopMax do
a0 := evalf(a0 + d0);
a1 := evalf(a1 + d1);
a := a0 + a1·x + x2;
# p = a b + r
b := quo(p, a, x);
r := rem(p, a, x);
r0 := coeff(r, x, 0);
r1 := coeff(r, x, 1);
# b = a c + s
s := rem(b, a, x);
s0 := coeff(s, x, 0);
s1 := coeff(s, x, 1);
# Gleichungen für Koeffizienten des "Inkrementes" D, siehe oben
d0 :=  $\frac{a0 r1 s1 - a1 r0 s1 + r0 s0}{a0 s1^2 - a1 s0 s1 + s0^2}$ ;
d1 :=  $-\frac{r0 s1 - r1 s0}{a0 s1^2 - a1 s0 s1 + s0^2}$ ;
print(evalf(a0));
print(evalf(a1));
print(evalf(solve(a0 + a1·x + x2)));
od:

a0 := -  $\frac{11}{2}$ 
a1 :=  $\frac{11}{6}$ 
d0 := 0
d1 := 0
LoopMax := 10
-5.50000000000000000000000000000000
1.833333333333333333333333333333333333
1.601324154956354614869137283853624425621,
-3.434657488289687948202470617186957758954
-0.039896784438269154426359666534339023408
2.979026068545719200741034802170173349217

```


Vgl. Angabe in Wikipedia

'B' = quo(P, A, x);

$$B = x^4 + (-a_1 + p_5) x^3 + (a_1^2 - a_1 p_5 - a_0 + p_4) x^2 + (-a_1^3 + a_1^2 p_5 + 2 a_0 a_1 - a_0 p_5 - a_1 p_4 + p_3) x + a_1^4 - a_1^3 p_5 - 3 a_0 a_1^2 + 2 a_0 a_1 p_5 + a_1^2 p_4 + a_0^2 - a_0 p_4 - a_1 p_3 + p_2 \quad (21)$$

> 'R' = rem(P, A, x);

$$R = (-a_1^5 + a_1^4 p_5 + 4 a_0 a_1^3 - 3 a_0 a_1^2 p_5 - a_1^3 p_4 - 3 a_0^2 a_1 + a_0^2 p_5 + 2 a_0 a_1 p_4 + a_1^2 p_3 - a_0 p_3 - a_1 p_2 + p_1) x - a_0 a_1^4 + a_0 a_1^3 p_5 + 3 a_0^2 a_1^2 - 2 a_0^2 a_1 p_5 - a_0 a_1^2 p_4 - a_0^3 + a_0^2 p_4 + a_0 a_1 p_3 - a_0 p_2 + p_0 \quad (22)$$

> # Division mit Rest

R Polynom vom Grad 1

Koeffizienten von P durch Koeffizienten von A,B,R ausdrücken

Sehr einfache Form der Koeffizienten r0, r1

R := expand(P - A·B);

for k from 0 to 1 do

Aux := coeff(R, x, k) = r[k];

p[k] := solve(Aux, p[k]);

od;

for k from 2 to Grad - 1 do

Aux := coeff(R, x, k) = 0;

p[k] := solve(Aux, p[k]);

od;

$$R := -x^5 a_1 - x^5 b_3 + p_5 x^5 - x^4 a_1 b_3 - x^4 a_0 - x^4 b_2 + p_4 x^4 - x^3 a_0 b_3 - x^3 a_1 b_2 - x^3 b_1 + p_3 x^3 - x^2 a_0 b_2 - x^2 a_1 b_1 - x^2 b_0 + p_2 x^2 - x a_0 b_1 - x a_1 b_0 + p_1 x - a_0 b_0 + p_0$$

$$Aux := -a_0 b_0 + p_0 = r_0$$

$$p_0 := a_0 b_0 + r_0$$

$$Aux := -a_0 b_1 - a_1 b_0 + p_1 = r_1$$

$$p_1 := a_0 b_1 + a_1 b_0 + r_1$$

$$Aux := -a_0 b_2 - a_1 b_1 - b_0 + p_2 = 0$$

$$p_2 := a_0 b_2 + a_1 b_1 + b_0$$

$$Aux := -a_0 b_3 - a_1 b_2 - b_1 + p_3 = 0$$

$$p_3 := a_0 b_3 + a_1 b_2 + b_1$$

$$Aux := -a_1 b_3 - a_0 - b_2 + p_4 = 0$$

$$p_4 := a_1 b_3 + a_0 + b_2$$

$$Aux := -a_1 - b_3 + p_5 = 0$$

$$p_5 := a_1 + b_3$$

(23)

>

L>

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Division mit Rest von Polynom p (Grad d) durch quadratisches Polynom a
%  $p = a b + r$  mit  $\text{gr}(r) \leq 1$ 
% Bestimmung der Koeffizienten von b,r aus Koeffizienten von p,a
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  $p(x) = p_d x^d + p_{(d-1)} x^{d-1} + \dots + p_1 x + p_0$ 
%  $a(x) = a_2 x^2 + a_1 x + a_0$ 
%  $b(x) = b_{(d-2)} x^{d-2} + b_{(d-3)} x^{d-3} + \dots + b_1 x + b_0$ 
%  $r(x) = r_1 x + r_0$ 
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

clear all
close all
format long e
clc

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Division mit Rest
% VORSICHT!  $a(x) = 1 + 2x + 3x^2$ 
% ! Befehl roots verwendet umgekehrte Reihenfolge der Koeffizienten!
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Einfaches Beispiel
p = [3,11,28,34,32,21];
a = [1,2,3];
bExakt = [4,5,6,7]
rExakt = [-1,-2]
[b,r] = DivisonMitRest(p,a)
pause

```

```

% Zufällige Wahl von Koeffizienten
d = 10;
p = -1 * 2*rand(d+1,1)
a = -1 * 2*rand(3,1)
[b,r] = DivisonMitRest(p,a);
Probe_DivisonMitRest(p,a,b,r);
pause

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Faktorisierung in quadratische Polynome
% Nullstellen (Vorsicht! Anordnung der Koeffizienten für "roots")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Einfaches Beispiel

clear all
close all
clc

p = [0,4,0,-5,0,1]
NullstellenExakt = roots(flip1r(p))

b{1} = p;
Nullstellen = [];
loop = 0;
while length(b{loop+1}) > 2

```



```

    loop = loop + 1
    [a{loop},b{loop+1}] = Iteration(b{loop});
    Probe_DivisonMitRest(b{loop},a{loop},b{loop+1},[0,0]);
    Nullstellen = [Nullstellen;roots(fliplr(a{loop}))]
    pause
end
Nullstellen = [Nullstellen;roots(fliplr(b{loop+1}))]
NullstellenExakt
pause

```

% Weiteres Beispiel mit komplexen Nullstellen

```

clear all
close all
clc

% p = [0,-4,0,-3,0,1]
p = [-4,-8,-7,-6,-2,2,1]

NullstellenExakt = roots(fliplr(p))

b{1} = p;
Nullstellen = [];
loop = 0;
while length(b{loop+1}) > 2
    loop = loop + 1
    [a{loop},b{loop+1}] = Iteration(b{loop});
    Probe_DivisonMitRest(b{loop},a{loop},b{loop+1},[0,0]);
    Nullstellen = [Nullstellen;roots(fliplr(a{loop}))]
    pause
end
Nullstellen = [Nullstellen;roots(fliplr(b{loop+1}))]
NullstellenExakt
plot(real(Nullstellen),imag(Nullstellen),'ro')
hold on
plot(real(NullstellenExakt),imag(NullstellenExakt),'b*')

```

%%%

function [b,r] = DivisonMitRest(p,a)

```

d = length(p)-1;
Shift = 1;

if d > 2
    % Quotient
    i = d-2;
    b(i+Shift) = p(i+2+Shift)/a(2+Shift);
    i = d-3;
    b(i+Shift) = (p(i+2+Shift) - a(1+Shift)*b(i+1+Shift))/a(2+Shift);
    for i = d-4:-1:0
        b(i+Shift) = (p(i+2+Shift) - a(0+Shift)*b(i+2+Shift) -
            a(1+Shift)*b(i+1+Shift))/a(2+Shift);
    end
    % Rest

```

```

    r(1+Shift) = p(1+Shift) - a(0+Shift)*b(1+Shift) - a(1+Shift)*b(0+Shift);
    r(0+Shift) = p(0+Shift) - a(0+Shift)*b(0+Shift);
end
if d == 2
    b = p(2+Shift)/a(2+Shift);
    r(1+Shift) = p(1+Shift) - a(1+Shift)*b(0+Shift);
    r(0+Shift) = p(0+Shift) - a(0+Shift)*b(0+Shift);
end
if d < 2
    b = 0;
    r = p;
end
end

```

```
function Probe_DivisonMitRest(p,a,b,r)
```

```

Shift = 1;
Bereich = [-1;1];
M = 10;

x = linspace(Bereich(1),Bereich(2),M);

```

```

% p = a b + r
P = 0;
for i = 0:length(p)-1
    P = P + p(i+Shift)*x.^i;
end
A = 0;
for i = 0:length(a)-1
    A = A + a(i+Shift)*x.^i;
end
B = 0;
for i = 0:length(b)-1
    B = B + b(i+Shift)*x.^i;
end
R = 0;
for i = 0:length(r)-1
    R = R + r(i+Shift)*x.^i;
end
Probe = norm(A.*B + R - P,inf)
plot(x,P,'b.-',x,A.*B + R,'r')
end

```

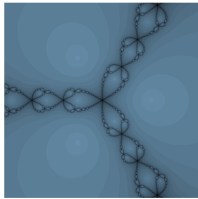
```
function [a,b] = Iteration(p)
```

```

d = length(p)-1;
Shift = 1;

% Ansatz für quadratisches Polynom (Startwert)
a(0+Shift) = p(d-2+Shift)/p(d+Shift);
a(1+Shift) = p(d-1+Shift)/p(d+Shift);

```

Numerische Mathematik 1 & 2
Mechthild Thalhammer
Illustrationen (Kapitel *Grundbegriffe*)

Illustrationen zum Kapitel *Grundbegriffe*

- **Maschinenzahlen (MAPLE)**
- **Sonderoperanden (MATLAB)**
- **Rundung (MAPLE)**
- **Kondition (MAPLE)**
- **Kondition von Matrizen, Stabilität des Gauß-Algorithmus (MAPLE)**

Bereich des Signifikanden bei normalisierten t-stelligen Gleitpunktzahlen zur Basis B (setze z.B. $B = 2$, 10 , $t = 1, 2, 3$)

```
> restart;  
> B := 10;  
t := 2;  
  
B := 10  
t := 2
```

(1)

```
> Menge_S_positiv := { }:  
for i from B^(t-1) to B^t-1 do  
  Menge_S_positiv := Menge_S_positiv union {i};  
od:  
'Menge_S_positiv' = Menge_S_positiv;
```

```
Menge_S_positiv = {10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99}
```

(2)

```
> Menge_S := {0 }:  
for i from B^(t-1) to B^t-1 do  
  Menge_S := Menge_S union { + i, -i};  
od:  
'Menge_S' = Menge_S;
```

```
Menge_S = { -99, -98, -97, -96, -95, -94, -93, -92, -91, -90, -89, -88, -87, -86, -85, -84, -83, -82, -81, -80, -79, -78, -77, -76, -75, -74, -73, -72, -71, -70, -69, -68, -67, -66, -65, -64, -63, -62, -61, -60, -59, -58, -57, -56, -55, -54, -53, -52, -51, -50, -49, -48, -47, -46, -45, -44, -43, -42, -41, -40, -39, -38, -37, -36, -35, -34, -33, -32, -31, -30, -29, -28, -27, -26, -25, -24, -23, -22, -21, -20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10, 0, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99}
```

(3)

Maschinenzahlen

```
> restart;  
> B := 2;  
t := 3;  
alpha := -3;  
beta := 0;  
  
B := 2  
t := 3  
alpha := -3
```

$$\beta := 0$$

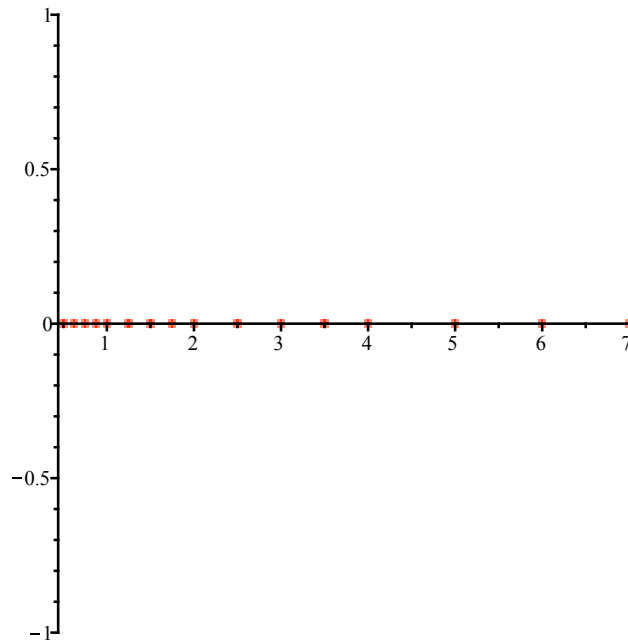
(4)

```
> M_positiv := { } :
for E from alpha to beta do
  for S from B^(t-1) to B^t-1 do
    g := S·B^E;
    M_positiv := M_positiv union {g};
  od;
od:
'M_positiv' = M_positiv;
```

$$M_{\text{positiv}} = \left\{ 1, 2, 3, 4, 5, 6, 7, \frac{1}{2}, \frac{3}{2}, \frac{3}{4}, \frac{5}{2}, \frac{5}{4}, \frac{5}{8}, \frac{7}{2}, \frac{7}{4}, \frac{7}{8} \right\}$$

(5)

```
> z := convert(M_positiv, list) :
plot([ [z['j'], 0] $j'= 1 ..nops(z) ], color = red, style = point, symbol = circle);
```

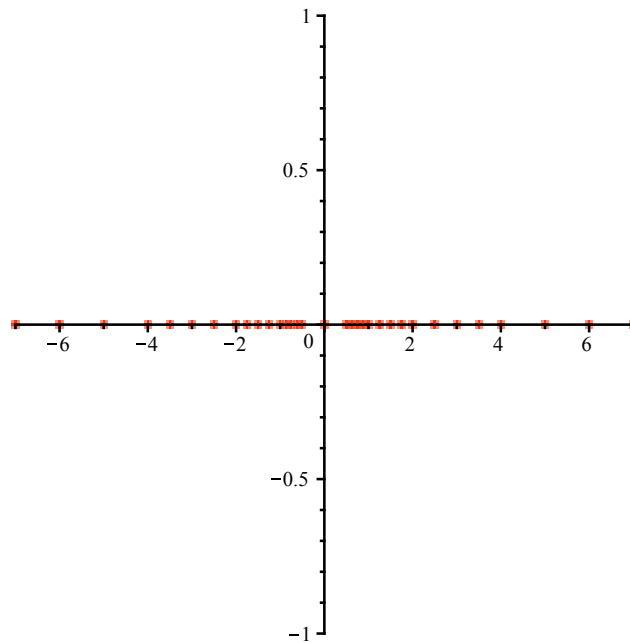


```
> M := {0} :
for E from alpha to beta do
  for S from B^(t-1) to B^t-1 do
    g := S·B^E;
    M := M union {g, -g};
  od;
od:
'M' = M;
```

$$M = \left\{ -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, -\frac{7}{2}, -\frac{7}{4}, -\frac{7}{8}, -\frac{5}{2}, -\frac{5}{4}, -\frac{5}{8}, \right. \\ \left. -\frac{3}{2}, -\frac{3}{4}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{3}{4}, \frac{5}{2}, \frac{5}{4}, \frac{5}{8}, \frac{7}{2}, \frac{7}{4}, \frac{7}{8} \right\}$$

(6)

```
> z := convert(M, list) :
plot([ [z['j'], 0] $j'= 1 ..nops(z) ], color = red, style = point, symbol = circle);
```



Auflösung, kleinste positive und größte Maschinenzahl für die Genauigkeitsstufen
single/double/extended precision

> restart;

> rho := (B, t, alpha, beta) → B^(1-t);

sigma := (B, t, alpha, beta) → B^(t-1 + alpha);

lambda := (B, t, alpha, beta) → (B^t-1) · B^beta;

$$\rho := (B, t, \alpha, \beta) \mapsto B^{1-t}$$

$$\sigma := (B, t, \alpha, \beta) \mapsto B^{t-1+\alpha}$$

$$\lambda := (B, t, \alpha, \beta) \mapsto (B^t - 1) \cdot B^\beta$$

(7)

> B := 2;

$$B := 2$$

(8)

Single precision

> t := 24;

alpha := -149;

beta := 104;

rho('B', t, alpha, beta) = evalf(rho(B, t, alpha, beta));

sigma('B', t, alpha, beta) = evalf(sigma(B, t, alpha, beta));

lambda('B', t, alpha, beta) = evalf(lambda(B, t, alpha, beta));

$$t := 24$$

$$\alpha := -149$$

$$\beta := 104$$

$$\frac{1}{B^{23}} = 1.192092896 \times 10^{-7}$$

$$\frac{1}{B^{126}} = 1.175494351 \times 10^{-38}$$

$$(B^{24} - 1) B^{104} = 3.402823466 \times 10^{38} \quad (9)$$

> # Größte Maschinenzahl im Wesentlichen $B^{t + \text{beta}}$

```
digits := 8;
evalf((B^t-1)·B^beta, digits);
evalf(B^(t + beta), digits);
```

$$\text{digits} := 8$$

$$3.4028235 \times 10^{38}$$

$$3.4028237 \times 10^{38} \quad (10)$$

Double precision

```
> t := 53;
alpha := -1074;
beta := 971;
```

```
rho('B', t, alpha, beta) = evalf(rho(B, t, alpha, beta));
sigma('B', t, alpha, beta) = evalf(sigma(B, t, alpha, beta));
lambda('B', t, alpha, beta) = evalf(lambda(B, t, alpha, beta));
```

$$t := 53$$

$$\alpha := -1074$$

$$\beta := 971$$

$$\frac{1}{B^{52}} = 2.220446049 \times 10^{-16}$$

$$\frac{1}{B^{1022}} = 2.225073858 \times 10^{-308}$$

$$(B^{53} - 1) B^{971} = 1.797693135 \times 10^{308} \quad (11)$$

```
> digits := 16;
evalf((B^t-1)·B^beta, digits);
evalf(B^(t + beta), digits);
```

$$\text{digits} := 16$$

$$1.797693134862316 \times 10^{308}$$

$$1.797693134862316 \times 10^{308} \quad (12)$$

Extended precision

```
> t := 64;
alpha := -16445;
beta := 16320;
```

```
rho('B', t, alpha, beta) = evalf(rho(B, t, alpha, beta));
sigma('B', t, alpha, beta) = evalf(sigma(B, t, alpha, beta));
```



```
lambda('B', t, alpha, beta) = evalf(lambda(B, t, alpha, beta));
```

```
t := 64
```

```
alpha := -16445
```

```
beta := 16320
```

$$\frac{1}{B^{63}} = 1.084202172 \times 10^{-19}$$

$$\frac{1}{B^{16382}} = 3.362103143 \times 10^{-4932}$$

$$(B^{64} - 1) B^{16320} = 1.189731495 \times 10^{4932}$$

(13)

```
> digits := 19;
```

```
evalf((B^t-1)·B^beta, digits);
```

```
evalf(B^(t+beta), digits);
```

```
digits := 19
```

$$1.189731495357231765 \times 10^{4932}$$

$$1.189731495357231765 \times 10^{4932}$$

(14)

```
>
```

```
>
```

```
>
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Illustration von in Matlab verfügbaren Sonderoperanden
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear all
close all
clc
format long e
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bereichsüberschreitung
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Double precision
```

```
B = 2;
t = 53;
alpha = 1074;
beta = 971;
```

```
% Größte Maschinenzahl
```

```
lambda = (B^t-1)*B^beta;
```

```
% Addition von Zahlen der Form 2^j
```

```
x = lambda;
counter = 0;
while x < inf
    counter = counter + 1
    AddierteZahl = 2^counter
    xDavor = x
    x = x + AddierteZahl;
    xDanach = x
    DifferenzDanachDavor = xDanach - xDavor
end
pause
```

```
% Zusammenfassung
```

```
clc
GroessteMaschinenzahl = lambda
AddierteZahl = 2^(counter-1)
KeineBereichsueberschreitung = lambda + 2^(counter-1)
Differenz = KeineBereichsueberschreitung - lambda
Bereichsueberschreitung = lambda + 2^counter
pause
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rechnen mit Sonderoperanden
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear all
close all
clc
format long e
```

```
x = inf
xPlus1 = x + 1
xMal2 = 2*x
```

xMalMinus2 = $-2 \cdot x$
xHalbe = $x/2$
xMinus = $x - x$
xKehrwert = $1/x$
xMalNull = $0 \cdot x$

Ergebnisse von elementaren arithmetischen Operationen liegen i.a. nicht im Bereich der darstellbaren Zahlen

(zudem treten Bereichsüberschreitungen auf)

```
> restart;
```

```
> B := 2;
```

```
t := 3;
```

```
alpha := -3;
```

```
beta := 0;
```

```
B := 2
```

```
t := 3
```

```
α := -3
```

```
β := 0
```

(1)

```
> M := {0}:
```

```
for E from alpha to beta do
```

```
  for S from B^(t-1) to B^t-1 do
```

```
    g := S·B^E;
```

```
    M := M union {g, -g};
```

```
  od;
```

```
od:
```

```
'M' = M;
```

```
M = { -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, - $\frac{7}{2}$ , - $\frac{7}{4}$ , - $\frac{7}{8}$ , - $\frac{5}{2}$ , - $\frac{5}{4}$ , - $\frac{5}{8}$ ,  
- $\frac{3}{2}$ , - $\frac{3}{4}$ , - $\frac{1}{2}$ ,  $\frac{1}{2}$ ,  $\frac{3}{2}$ ,  $\frac{3}{4}$ ,  $\frac{5}{2}$ ,  $\frac{5}{4}$ ,  $\frac{5}{8}$ ,  $\frac{7}{2}$ ,  $\frac{7}{4}$ ,  $\frac{7}{8}$  }
```

(2)

```
> x := 5·2^(-3);
```

```
y := 4·2^(-3);
```

```
evalb(x in M);
```

```
evalb(y in M);
```

```
x :=  $\frac{5}{8}$ 
```

```
y :=  $\frac{1}{2}$ 
```

```
true
```

```
true
```

(3)

```
> xPlusy = x + y;
```

```
evalb(x + y in M);
```

```
xMinusy = x - y;
```

```
evalb(x - y in M);
```

```
yMinusx = y - x;
```

```
evalb(y - x in M);
```

```
xMaly = x·y;
```

```
evalb(x·y in M);
```

```
xDurchy = x/y;  
evalb(x/y in M);  
yDurchx = y/x;  
evalb(y/x in M);
```

$$xPlusy = \frac{9}{8}$$

false

$$xMinusy = \frac{1}{8}$$

false

$$yMinusx = -\frac{1}{8}$$

false

$$xMaly = \frac{5}{16}$$

false

$$xDurchy = \frac{5}{4}$$

true

$$yDurchx = \frac{4}{5}$$

false

(4)

Addition zweier Zahlen: Exaktes Ergebnis, im Rahmen der Genauigkeit korrekt gerundetes Ergebnis.
Exakte Berechnung von $x + y$ nicht nötig, um im Rahmen der Genauigkeit korrekt gerundetes Ergebnis zu berechnen.

Wenige weitere Stellen ausreichend.

```
> restart;
```

```
> Digits := 30;
```

```
digits := 3;
```

Digits := 30

digits := 3

(5)

```
> x := 215·10^10;
```

```
y := evalf(-125·10^(-7));
```

```
xPlusy := x + y;
```

```
xPlusy_Truncated := 0.2149·10^13;
```

```
evalf(xPlusy, digits);
```

```
evalf(xPlusy_Truncated, digits);
```

x := 215000000000

y := -0.0000125000000000000000000000000000000000

xPlusy := 2.149999999999999987500000000000000000000000 × 10¹²

$$xPlusy_Truncated := 2.1490000000000000 \times 10^{12}$$

$$2.15 \times 10^{12}$$

$$2.15 \times 10^{12}$$

(6)

Größe des Rundungsfehlers bei Multiplikation beschränkt durch Maschinengenauigkeit epsilon

```
> restart;
> B := 10;
  t := 4;
  rho := B1-t;
  # Maschinengenauigkeit bei "korrektem" Runden
  epsilon := evalf( $\left(\frac{\rho}{2}\right)$ );
```

$$B := 10$$

$$t := 4$$

$$\rho := \frac{1}{1000}$$

$$\epsilon := 0.0005000000000000$$

(7)

```
> a := 20.29;
  b := 49.31;
  aMalb := a·b;
  evalf(aMalb, 4);
  Rundung_aMalb := 1000;
  RelativerRundungsfehler :=  $\frac{Rundung\_aMalb - aMalb}{aMalb}$ ;
  a := 20.29
  b := 49.31
  aMalb := 1000.4999
  1000.
```

$$Rundung_aMalb := 1000$$

$$RelativerRundungsfehler := -0.0004996502249$$

(8)

```
> # Abrunden versus Aufrunden
  a := 28.75;
  b := 34.80;
  aMalb := a·b;
  Rundung_aMalb := 1000;
  RelativerRundungsfehler :=  $\frac{Rundung\_aMalb - aMalb}{aMalb}$ ;
  Rundung_aMalb := 1001;
  RelativerRundungsfehler :=  $\frac{Rundung\_aMalb - aMalb}{aMalb}$ ;
```

$$a := 28.75$$

```

        b := 34.80
        aMalb := 1000.5000
        Rundung_aMalb := 1000
        RelativerRundungsfehler := -0.0004997501249
        Rundung_aMalb := 1001
        RelativerRundungsfehler := 0.0004997501249

```

(9)

Auswirkung von Rundungsfehlern bei Berechnung von a + b + c (Assoziativität nicht gültig)

```

> restart;
> Digits := 20;
  digits := 8;

```

```

        Digits := 20
        digits := 8

```

(10)

```

> a := 0.23371258·10(-4);
  b := 0.33678429·102;
  c := -0.33677811·102;
  Ergebnis := a + b + c;
  Rundung_Ergebnis := evalf(Ergebnis, digits);

```

```

        a := 0.0000233712580000000000000000
        b := 33.67842900
        c := -33.67781100
        Ergebnis := 0.000641371258000000
        Rundung_Ergebnis := 0.00064137126

```

(11)

```

> aPlusb := a + b;
  Rundung_aPlusb := evalf(aPlusb, digits);
  Ergebnis1 := evalf(Rundung_aPlusb + c, digits);
  RelativerFehlerErgebnis1 := abs( (Ergebnis1 - Ergebnis) / Ergebnis );
  GroessenordnungZwischenergebnis := aPlusb / Ergebnis;
  Differenz := Ergebnis1 - Rundung_Ergebnis;

```

```

        aPlusb := 33.678452371258000000
        Rundung_aPlusb := 33.678452
        Ergebnis1 := 0.000641
        RelativerFehlerErgebnis1 := 0.00057885038559055603954
        GroessenordnungZwischenergebnis := 52510.074237311706912
        Differenz := -3.7126 × 10-7

```

(12)

```

> bPlusc := b + c;
  Rundung_bPlusc := evalf(bPlusc, digits);
  Ergebnis2 := evalf(a + Rundung_bPlusc, digits);
  RelativerFehlerErgebnis2 := abs( (Ergebnis2 - Ergebnis) / Ergebnis );

```

```

GroessenordnungZwischenergebnis :=  $\frac{bPlusc}{Ergebnis}$ ;
Differenz := Ergebnis2 - Rundung_Ergebnis;

      bPlusc := 0.00061800
      Rundung_bPlusc := 0.00061800
      Ergebnis2 := 0.00064137126
      RelativerFehlerErgebnis2 :=  $3.1183187195457392947 \times 10^{-9}$ 
      GroessenordnungZwischenergebnis := 0.96356048433963344207
      Differenz := 0.

```

(13)

Berechnung von $1 + 1/2 + \text{etc.}$

Schlussfolgerung: Günstige Reihenfolge bei wiederholtem Summieren beachten!

```

> restart;
> Digits := 20;
  n := 1000;
  Summe1 := evalf(sum(1/i, i=1..n));
  Summe2 := evalf(sum((-1)^i/i, i=1..n));
      Digits := 20
      n := 1000
      Summe1 := 7.4854708605503449126
      Summe2 := -0.69264743055982030967

```

(14)

```

> digits := 4;
      digits := 4

```

(15)

```

> Resultat := 0 :
  for i from 1 to n do
    Resultat := evalf(Resultat + 1/i, digits);
  od:
  Exakt := Summe1;
  Numerisch := Resultat;
  AbsoluterFehler := abs(Resultat - Summe1);

      Exakt := 7.4854708605503449126
      Numerisch := 7.446
      AbsoluterFehler := 0.0394708605503449126

```

(16)

```

> Resultat := 0 :
  for i from n to 1 by -1 do
    Resultat := evalf(Resultat + 1/i, digits);
  od:
  Exakt := Summe1;
  Numerisch := Resultat;
  AbsoluterFehler := abs(Resultat - Summe1);

      Exakt := 7.4854708605503449126

```



```

                                Numerisch := 7.484
                                AbsoluterFehler := 0.0014708605503449126
(17)
```

```

> Resultat := 0 :
  for i from 1 to n do
    Resultat := evalf(Resultat + (-1)i/i, digits);
  od:
  Exakt := Summe2;
  Numerisch := Resultat;
  AbsoluterFehler := abs(Resultat - Summe2);
                                Exakt := -0.69264743055982030967
                                Numerisch := -0.6930
                                AbsoluterFehler := 0.00035256944017969033
(18)
```

```

> Resultat := 0 :
  for i from n to 1 by -1 do
    Resultat := evalf(Resultat + (-1)i/i, digits);
  od:
  Exakt := Summe2;
  Numerisch := Resultat;
  AbsoluterFehler := abs(Resultat - Summe2);
                                Exakt := -0.69264743055982030967
                                Numerisch := -0.6926
                                AbsoluterFehler := 0.00004743055982030967
(19)
```

```

>
```

Addition bzw. Subtraktion zweier annähernd gleich großer Zahlen sowie der gerundeten Zahlen
(Korrekte Rundung bei Addition bzw. Abrunden bei Subtraktion)
Auslöschung signifikanter Stellen

```
> restart;  
> Digits := 20;  
  
Digits := 20 (1)
```

```
> # Beispiel 1 (Rundung(a) = Rundung(b))  
a := 123456789.1;  
b := 123456788.5;  
rd_a := 123456789;  
rd_b := 123456789;  
  
a := 1.234567891 × 108  
b := 1.234567885 × 108  
rd_a := 123456789  
rd_b := 123456789 (2)
```

```
> # Beispiel 2  
a := 123456789.51;  
b := 123456789.42;  
rd_a := 123456789.5;  
rd_b := 123456789.4;  
  
a := 1.2345678951 × 108  
b := 1.2345678942 × 108  
rd_a := 1.234567895 × 108  
rd_b := 1.234567894 × 108 (3)
```

```
> # Relativer Fehler der Rundung  
Fehler_a := (rd_a - a) / a;  
Fehler_b := (rd_b - b) / b;  
  
Fehler_a := -8.1000000402489002000 × 10-11  
Fehler_b := -1.6200000092307600526 × 10-10 (4)
```

```
> # Relativer Fehler der Addition  
a_plus_b := a + b;  
rda_plus_rdb := rd_a + rd_b;  
Fehler_a_plus_b := (rda_plus_rdb - a_plus_b) / a_plus_b;  
  
a_plus_b := 2.4691357893 × 108  
rda_plus_rdb := 2.469135789 × 108  
Fehler_a_plus_b := -1.2150000064802025346 × 10-10 (5)
```

```
> # Relativer Fehler der Subtraktion (bei beiden Beispielen signifikant!)  
a_minus_b := a - b;  
rda_minus_rdb := rd_a - rd_b;  
Fehler_a_minus_b := (rda_minus_rdb - a_minus_b) / a_minus_b;
```

$$\begin{aligned}
a_minus_b &:= 0.09 \\
rda_minus_rdb &:= 0.1 \\
Fehler_a_minus_b &:= 0.11111111111111111111
\end{aligned}
\tag{6}$$

Kondition der Formel von Vieta
(unabhängig von Formulierung!)
(schlecht konditioniertes Problem, wenn $a^2 + b$ klein)

```

> restart;
> # Übliche Formulierung
p := -a + sqrt(a^2 + b);
da_p := diff(p, a);
db_p := diff(p, b);

```

$$\begin{aligned}
p &:= -a + \sqrt{a^2 + b} \\
da_p &:= -1 + \frac{a}{\sqrt{a^2 + b}} \\
db_p &:= \frac{1}{2\sqrt{a^2 + b}}
\end{aligned}
\tag{7}$$

```

> # Umformulierung (siehe Skriptum)
# Gleiche Kondition
p_neu := b / (a + sqrt(a^2 + b));
simplify(p - p_neu);
da_p_neu := simplify(diff(p_neu, a));
simplify(da_p - da_p_neu);
db_p_neu := simplify(diff(p_neu, b));
simplify(db_p - db_p_neu);

```

$$\begin{aligned}
p_neu &:= \frac{b}{a + \sqrt{a^2 + b}} \\
da_p_neu &:= -\frac{b}{\sqrt{a^2 + b} (a + \sqrt{a^2 + b})} \\
db_p_neu &:= \frac{2a^2 + b + 2a\sqrt{a^2 + b}}{2(a + \sqrt{a^2 + b})^2 \sqrt{a^2 + b}}
\end{aligned}
\tag{8}$$

Formel von Vieta
Ursprüngliche Formulierung (Auslöschung für b "relativ klein", instabiler Algorithmus)
Umformulierung (stabiler Algorithmus)
Setze etwa digits = 8,10,12
Relativer Fehler O(Verstärkungsfaktor*digits)

```

> restart;
> Digits := 30;

```

```
digits := 8;
```

```
Digits := 30
```

```
digits := 8
```

(9)

```
> # Beispiel 1
```

```
a := 1000;
```

```
b := 0.018000000081;
```

```
a := 1000
```

```
b := 0.018000000081
```

(10)

```
> # Beispiel 2
```

```
a := evalf(Pi);
```

```
b := evalf(exp(1)/10^7);
```

```
a := 3.14159265358979323846264338328
```

```
b := 2.71828182845904523536028747135 × 10-7
```

(11)

```
> rda := evalf(a, digits);
```

```
rdb := evalf(b, digits);
```

```
rda := 1000.
```

```
rdb := 0.018000000
```

(12)

Berechnung sämtlicher Zwischenergebnisse und entsprechender Rundungsfehler für ursprüngliche Formulierung und entsprechenden Algorithmus

$p = -a + \sqrt{a^2 + b}$

Signifikanter Fehler bei Subtraktion

```
> aa := a^2;
```

```
rdaa := evalf(a^2, digits);
```

```
fehler := (rdaa - aa) / aa;
```

```
aab := aa + b;
```

```
rdaab := evalf(rdaa + b, digits);
```

```
fehler := (rdaab - aab) / aab;
```

```
sqртаab := sqrt(aab);
```

```
rdsqртаab := evalf(sqrt(rdaab), digits);
```

```
fehler := (rdsqртаab - sqртаab) / sqртаab;
```

```
sqртаab_minus_a := sqртаab - a;
```

```
rd_sqртаab_minus_a := evalf(rdsqртаab - a, digits);
```

```
fehler := (rd_sqртаab_minus_a - sqртаab_minus_a) / sqртаab_minus_a;
```

```
aa := 1000000
```

```
rdaa := 1.000000 × 106
```

```
fehler := 0.
```

```
aab := 1.000000018000000081 × 106
```

```
rdaab := 1.0000000 × 106
```

```
fehler := -1.79999997570000029159999671950 × 10-8
```

```

sqртаab := 1000.0000090000000000000000000000
rdsqртаab := 1000.0000
fehler := -8.999999919000000072899999343900 × 10-9
sqртаab_minus_a := 9.00000000000000000000 × 10-6
rd_sqртаab_minus_a := 0.
fehler := -1.00000000000000000000000000000000

```

(13)

```

> Verstaerkungsfaktor := sqrt(a^2 + b) / (-a + sqrt(a^2 + b));
Verstaerkungsfaktor := 1.11111121111111111111111111111111 × 108

```

(14)

Berechnung sämtlicher Zwischenergebnisse und entsprechender Rundungsfehler für Umformulierung und entsprechenden Algorithmus

```

p_neu = b / (a + sqrt(a^2 + b))
> sqртаab_plus_a := sqртаab + a;
rd_sqртаab_plus_a := evalf(rdsqртаab + a, digits);
fehler := (rd_sqртаab_plus_a - sqртаab_plus_a) / sqртаab_plus_a;

b_sqртаabplusa := b / sqртаab_plus_a;
rd_b_sqртаabplusa := evalf(b / rd_sqртаab_plus_a, digits);
fehler := (rd_b_sqртаabplusa - b_sqртаabplusa) / b_sqртаabplusa;
sqртаab_plus_a := 2000.000009000000000000000000000000
rd_sqртаab_plus_a := 2000.0000
fehler := -4.49999997975000009112499958994 × 10-9
b_sqртаabplusa := 9.00000000000000000000000000000000 × 10-6
rd_b_sqртаabplusa := 9.0000000 × 10-6
fehler := 0.

```

(15)

Lösung des zugehörigen linearen Gleichungssystems

Gut konditioniertes Problem für epsilon *sehr klein*

(i) Instabiler Algorithmus

(ii) Stabiler Algorithmus (Vertauschen Gleichungen, entspricht Pivotwahl)

> restart;

> with(LinearAlgebra) :

> A := Matrix(2, 2, [epsilon, 1, 1, 1]);

$$A := \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$$

(1)

> Determinante = Determinant(A);

Kondition = ConditionNumber(A);

P, L, R := LUDecomposition(A);

UntereDreiecksmatrix = L;

ObereDreiecksmatrix = R;

subs(epsilon = 0, A) = subs(epsilon = 0, L) • subs(epsilon = 0, R);

$$\text{Determinante} = \epsilon - 1$$

$$\text{Kondition} = \max\left(\frac{2}{|\epsilon - 1|}, \frac{1}{|\epsilon - 1|} + \left|\frac{\epsilon}{\epsilon - 1}\right|\right) \max(2, 1 + |\epsilon|)$$

$$P, L, R := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ \frac{1}{\epsilon} & 1 \end{bmatrix}, \begin{bmatrix} \epsilon & 1 \\ 0 & \frac{\epsilon - 1}{\epsilon} \end{bmatrix}$$

$$\text{UntereDreiecksmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\epsilon} & 1 \end{bmatrix}$$

$$\text{ObereDreiecksmatrix} = \begin{bmatrix} \epsilon & 1 \\ 0 & \frac{\epsilon - 1}{\epsilon} \end{bmatrix}$$

Error, numeric exception: division by zero

> A := Matrix(2, 2, [1, 1, epsilon, 1]);

$$A := \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix}$$

(2)

> Determinante = Determinant(A);

Kondition = ConditionNumber(A);

P, L, R := LUDecomposition(A);

UntereDreiecksmatrix = L;

ObereDreiecksmatrix = R;

subs(epsilon = 0, A) = subs(epsilon = 0, L) • subs(epsilon = 0, R);

$$\text{Determinante} = -\epsilon + 1$$

$$\text{Kondition} = \max\left(\frac{2}{|\epsilon - 1|}, \frac{1}{|\epsilon - 1|} + \left|\frac{\epsilon}{\epsilon - 1}\right|\right) \max(2, 1 + |\epsilon|)$$

$$P, L, R := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & -\epsilon + 1 \end{bmatrix}$$

$$\text{UntereDreiecksmatrix} = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}$$

$$\text{ObereDreiecksmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & -\epsilon + 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

(3)

Schlecht konditionierte Probleme

> restart;

> with(LinearAlgebra) :

> A := Matrix(2, 2, [1, epsilon, 1, -epsilon]);

$$A := \begin{bmatrix} 1 & \epsilon \\ 1 & -\epsilon \end{bmatrix}$$

(4)

> Determinante = Determinant(A);

Kondition = ConditionNumber(A);

P, L, R := LUDecomposition(A);

UntereDreiecksmatrix = L;

ObereDreiecksmatrix = R;

subs(epsilon = 0, A) = subs(epsilon = 0, L) • subs(epsilon = 0, R);

$$\text{Determinante} = -2 \epsilon$$

$$\text{Kondition} = \max\left(1, \frac{1}{|\epsilon|}\right) (1 + |\epsilon|)$$

$$P, L, R := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & \epsilon \\ 0 & -2 \epsilon \end{bmatrix}$$

$$\text{UntereDreiecksmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\text{ObereDreiecksmatrix} = \begin{bmatrix} 1 & \epsilon \\ 0 & -2 \epsilon \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

(5)

> A := Matrix(2, 2, [epsilon, 1, -epsilon, 1]);

$$A := \begin{bmatrix} \epsilon & 1 \\ -\epsilon & 1 \end{bmatrix}$$

(6)

> Determinante = Determinant(A);

$Kondition = ConditionNumber(A);$
 $P, L, R := LUdecomposition(A);$
 $UntereDreiecksmatrix = L;$
 $ObereDreiecksmatrix = R;$
 $subs(\epsilon = 0, A) = subs(\epsilon = 0, L) \cdot subs(\epsilon = 0, R);$

$$Determinante = 2 \epsilon$$

$$Kondition = \max\left(1, \frac{1}{|\epsilon|}\right) (1 + |\epsilon|)$$

$$P, L, R := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} \epsilon & 1 \\ 0 & 2 \end{bmatrix}$$

$$UntereDreiecksmatrix = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

$$ObereDreiecksmatrix = \begin{bmatrix} \epsilon & 1 \\ 0 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

(7)

> $A := Matrix(2, 2, [\epsilon, -\epsilon, 1, 1]);$

$$A := \begin{bmatrix} \epsilon & -\epsilon \\ 1 & 1 \end{bmatrix}$$

(8)

> $Determinante = Determinant(A);$
 $Kondition = ConditionNumber(A);$
 $P, L, R := LUdecomposition(A);$
 $UntereDreiecksmatrix = L;$
 $ObereDreiecksmatrix = R;$
 $subs(\epsilon = 0, A) = subs(\epsilon = 0, L) \cdot subs(\epsilon = 0, R);$

$$Determinante = 2 \epsilon$$

$$Kondition = \left(\frac{1}{2} + \frac{1}{2|\epsilon|}\right) \max(2, 2|\epsilon|)$$

$$P, L, R := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ \frac{1}{\epsilon} & 1 \end{bmatrix}, \begin{bmatrix} \epsilon & -\epsilon \\ 0 & 2 \end{bmatrix}$$

$$UntereDreiecksmatrix = \begin{bmatrix} 1 & 0 \\ \frac{1}{\epsilon} & 1 \end{bmatrix}$$

$$ObereDreiecksmatrix = \begin{bmatrix} \epsilon & -\epsilon \\ 0 & 2 \end{bmatrix}$$

Error, numeric exception: division by zero

> $A := \text{Matrix}(2, 2, [1, 1, \text{epsilon}, -\text{epsilon}]);$

$$A := \begin{bmatrix} 1 & 1 \\ \epsilon & -\epsilon \end{bmatrix}$$

(9)

> $\text{Determinante} = \text{Determinant}(A);$

$\text{Kondition} = \text{ConditionNumber}(A);$

$P, L, R := \text{LUdecomposition}(A);$

$\text{UntereDreiecksmatrix} = L;$

$\text{ObereDreiecksmatrix} = R;$

$\text{subs}(\text{epsilon} = 0, A) = \text{subs}(\text{epsilon} = 0, L) \cdot \text{subs}(\text{epsilon} = 0, R);$

$$\text{Determinante} = -2 \epsilon$$

$$\text{Kondition} = \left(\frac{1}{2} + \frac{1}{2|\epsilon|} \right) \max(2, 2|\epsilon|)$$

$$P, L, R := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & -2\epsilon \end{bmatrix}$$

$$\text{UntereDreiecksmatrix} = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}$$

$$\text{ObereDreiecksmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & -2\epsilon \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

(10)

Beispiel von Kahan

Geometrische Veranschaulichung (Geraden mit näherungsweise gleichen Steigungen, "schleifender Schnitt")

Vgl. etwa www.aleph1.info/?call=Puc&permalink=grundbegriffe_3_3_Z2

> $\text{restart};$

> $\text{Digits} := 8;$

$\text{with}(\text{LinearAlgebra}) :$

$$\text{Digits} := 8$$

(11)

> $A := \text{Matrix}(2, 2, [0.2161, 0.1441, 1.2969, 0.8648]);$

$$A := \begin{bmatrix} 0.2161 & 0.1441 \\ 1.2969 & 0.8648 \end{bmatrix}$$

(12)

> $\text{Determinante} = \text{Determinant}(A);$

$\text{Kondition} = \text{ConditionNumber}(A);$

$$\text{Determinante} = -1. \times 10^{-8}$$

$$\text{Kondition} = 3.2706520 \times 10^8$$

(13)

> $P, L, R := \text{LUdecomposition}(A);$

$\text{UntereDreiecksmatrix} = L;$

$\text{ObereDreiecksmatrix} = R;$

$Probe := P \cdot A - L \cdot R;$

$$P, L, R := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1.0 & 0. \\ 0.166628113192999 & 1.0 \end{bmatrix},$$

$$\begin{bmatrix} 1.296900000000000 & 0.864800000000000 \\ 0. & 7.71069473247357 \times 10^{-9} \end{bmatrix}$$

$$UntereDreiecksmatrix = \begin{bmatrix} 1.0 & 0. \\ 0.166628113192999 & 1.0 \end{bmatrix}$$

$$ObereDreiecksmatrix = \begin{bmatrix} 1.296900000000000 & 0.864800000000000 \\ 0. & 7.71069473247357 \times 10^{-9} \end{bmatrix}$$

$$Probe := \begin{bmatrix} 0. & 0. \\ 0. & 0. \end{bmatrix}$$

(14)

> # $PAx = LRx = Pb = c$
 # $y = Rx$
 # $Ly = c$, L gut konditioniert
 # $Rx = y$, R schlecht konditioniert
 $c1 := b2;$
 $c2 := b1;$
 $LInvers := MatrixInverse(L);$
 $y := LInvers \cdot Vector([c1, c2]);$
 $RInverse := MatrixInverse(R);$
 $Loesung := RInverse \cdot y;$

$$c1 := b2$$

$$c2 := b1$$

$$LInvers := \begin{bmatrix} 1.0 & 0. \\ -0.166628113192999 & 1.0 \end{bmatrix}$$

$$y := \begin{bmatrix} 1.0 b2 \\ -0.166628113192999 b2 + 1.0 b1 \end{bmatrix}$$

$$RInverse := \begin{bmatrix} 0.771069473359550 & -8.64800000125831 \times 10^7 \\ 0. & 1.29690000018870 \times 10^8 \end{bmatrix}$$

$$Loesung := \begin{bmatrix} 1.4410000 \times 10^7 b2 - 8.6480000 \times 10^7 b1 \\ -2.1610000 \times 10^7 b2 + 1.2969000 \times 10^8 b1 \end{bmatrix}$$

(15)

> # Kahan: Beobachtung: Kleine Störung der rechten Seite führt auf signifikant andere Lösung
 $subs(\{b1 = 0.144, b2 = 0.8642\}, Loesung);$

```
subs({b1 = 0.144 + 10-8, b2 = 0.8642 - 10-8}, Loesung);
```

$$\begin{bmatrix} 2. \\ -2. \end{bmatrix}$$
$$\begin{bmatrix} 1. \\ -1. \end{bmatrix}$$

(16)

```
> # Kahan: Weitere Beobachtung: Residuen zweier signifikant unterschiedlicher Lösungen klein  
    (bzgl. Genauigkeit)
```

```
# Also: Residuum hier schlechter Indikator für Approximationsgüte
```

```
A • Vector([2, -2]) - Vector([0.144, 0.8642]);
```

```
A • Vector([0.9911, -0.4870]) - Vector([0.144, 0.8642]);
```

$$\begin{bmatrix} -2.77555756156289 \times 10^{-17} \\ -1.11022302462516 \times 10^{-16} \end{bmatrix}$$
$$\begin{bmatrix} 9.99999999473644 \times 10^{-9} \\ -1.00000001612699 \times 10^{-8} \end{bmatrix}$$

(17)

```
Matrizen mit großer Kondition
```

```
> restart;
```

```
> with(LinearAlgebra):
```

```
> p := 2;
```

$$p := 2$$

(18)

```
> d := 3;
```

```
v := Vector(d):
```

```
for i from 1 to d do
```

```
v[i] := 1/i;
```

```
od:
```

```
H := HilbertMatrix(d);
```

```
detH := evalf(Determinant(H));
```

```
HInvers := MatrixInverse(H);
```

```
HKondition := evalf(ConditionNumber(H, p));
```

```
V := VandermondeMatrix(v);
```

```
detV := evalf(Determinant(V));
```

```
VInvers := MatrixInverse(V);
```

```
VKondition := evalf(ConditionNumber(V, p));
```

$$d := 3$$

$$H := \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

```

detH := 0.0004629629630
HInvers := [ [ 9   -36   30
               -36  192  -180
               30  -180  180 ] ]
HKondition := 524.0567776
V := [ [ 1   1   1
         1  1/2  1/4
         1  1/3  1/9 ] ]
detV := -0.055555555556
VInvers := [ [ 1/2  -4   9/2
               -5/2  16  -27/2
               3   -12  9 ] ]
VKondition := 59.65332465

```

(19)

```

> for d from 5 by 5 to 20 do
  v := Vector(d) :
  for i from 1 to d do
    v[i] := 1/i ;
  od:
  H := HilbertMatrix(d) :
  detH := evalf(Determinant(H));
  HKondition := evalf(ConditionNumber(H, p));
  V := VandermondeMatrix(v) :
  detV := evalf(Determinant(V));
  VKondition := evalf(ConditionNumber(V, p));
  print(d, HKondition, VKondition);
od:

```

```

5, 476607.2502, 12166.24432
10, 1.602628687 × 1013, 1.592861070 × 1011
15, 6.116565790 × 1020, 2.878423845 × 1019
20, 2.452156586 × 1028, 2.842676299 × 1028

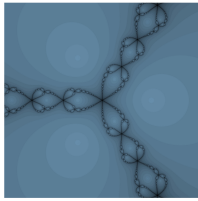
```

(20)

```

>
>

```



Numerische Differentiation

Approximationen mittels Differenzen und Fehleranalyse

Problemstellung. Für eine reguläre reellwertige Funktion

$$f : (a, b) \longrightarrow \mathbb{R}$$

sollen Approximationen an die Werte der ersten und zweiten Ableitung

$$f'(x), \quad f''(x), \quad x \in (a, b),$$

bestimmt werden.

Approximationsverfahren. In Hinblick auf die Definition der ersten Ableitung ist es naheliegend, für ein *kleines* Inkrement $\xi > 0$ die Vorwärtsdifferenz

$$\frac{f(x + \xi) - f(x)}{\xi} \approx f'(x),$$

die Rückwärtsdifferenz

$$\frac{f(x) - f(x - \xi)}{\xi} \approx f'(x),$$

oder die symmetrische Differenz

$$\frac{f(x + \xi) - f(x - \xi)}{2\xi} \approx f'(x)$$

zu verwenden. Eine Kombination dieser Approximationen führt auf die symmetrische Differenz für die zweite Ableitung

$$\frac{f(x + \xi) - 2f(x) + f(x - \xi)}{\xi^2} \approx f''(x).$$

Approximationsfehler. Mittels Taylorreihen-Entwicklungen wie beispielsweise (präzisere Aussage bei Angabe des Restgliedes in Integralform oder mittels Mittelwertsatz)

$$\begin{aligned} f(x + \xi) &= f(x) + \xi f'(x) + \frac{1}{2} \xi^2 f''(x) + \frac{1}{6} \xi^3 f'''(x) + \mathcal{O}(\xi^4), \\ f(x - \xi) &= f(x) - \xi f'(x) + \frac{1}{2} \xi^2 f''(x) - \frac{1}{6} \xi^3 f'''(x) + \mathcal{O}(\xi^4), \end{aligned}$$

erhält man durch Addition bzw. Subtraktion

$$\begin{aligned} f(x + \xi) - f(x) &= \xi f'(x) + \frac{1}{2} \xi^2 f''(x) + \mathcal{O}(\xi^3), \\ f(x - \xi) - f(x) &= -\xi f'(x) + \frac{1}{2} \xi^2 f''(x) + \mathcal{O}(\xi^3), \\ f(x + \xi) - f(x - \xi) &= 2\xi f'(x) + \mathcal{O}(\xi^3), \\ f(x + \xi) - 2f(x) + f(x - \xi) &= \xi^2 f''(x) + \mathcal{O}(\xi^4). \end{aligned}$$

Dies zeigt folgende Identitäten betreffend den Approximationsfehler (Abhängigkeit der jeweiligen Schranke von Funktionswerten der auftretenden Ableitung von f bzw. deren Maximalwert)

$$\begin{aligned} \frac{f(x + \xi) - f(x)}{\xi} - f'(x) &= \mathcal{O}(\xi), \\ \frac{f(x) - f(x - \xi)}{\xi} - f'(x) &= \mathcal{O}(\xi), \\ \frac{f(x + \xi) - f(x - \xi)}{2\xi} - f'(x) &= \mathcal{O}(\xi^2), \\ \frac{f(x + \xi) - 2f(x) + f(x - \xi)}{\xi^2} - f''(x) &= \mathcal{O}(\xi^2). \end{aligned}$$

Rundungsfehler. Um für elementare Funktionen wie beispielsweise Polynomfunktionen den Einfluss von Rundungsfehlern zu erklären, ist folgende Überlegung für die Vorwärtsdifferenz ausreichend (mit Maschinengenauigkeit ε , Störungen $|\varepsilon_1|$ und $|\varepsilon_2|$ in der Größenordnung von ε , die nicht-negative Konstante $C_1 \geq 0$ ist im Wesentlichen durch den Maximalwert $\|f\|_\infty$ gegeben)

$$\begin{aligned} &\left| \text{rd} \left(\frac{f(x + \xi) - f(x)}{\xi} \right) - \frac{f(x + \xi) - f(x)}{\xi} \right| \\ &\approx \left| \frac{f(x + \xi)(1 + \varepsilon_1) - f(x)(1 + \varepsilon_2)}{\xi} - \frac{f(x + \xi) - f(x)}{\xi} \right| \approx C_1 \frac{\varepsilon}{\xi}. \end{aligned}$$

Entsprechend ergeben sich die Abschätzungen

$$\begin{aligned} &\left| \text{rd} \left(\frac{f(x) - f(x - \xi)}{\xi} \right) - \frac{f(x) - f(x - \xi)}{\xi} \right| \approx C_1 \frac{\varepsilon}{\xi}, \\ &\left| \text{rd} \left(\frac{f(x + \xi) - f(x - \xi)}{2\xi} \right) - \frac{f(x + \xi) - f(x - \xi)}{2\xi} \right| \approx C_1 \frac{\varepsilon}{\xi}, \\ &\left| \text{rd} \left(\frac{f(x + \xi) - 2f(x) + f(x - \xi)}{\xi^2} \right) - \frac{f(x + \xi) - 2f(x) + f(x - \xi)}{\xi^2} \right| \approx C_1 \frac{\varepsilon}{\xi^2}. \end{aligned}$$

Gesamtfehler. Insgesamt erhält man damit Abschätzungen für den Gesamtfehler und folglich eine Aussage zum näherungsweise optimalen Inkrement (Minimalwert von Ap-

proximationsfehler und Rundungsfehler, mit $\varepsilon \approx 10^{-16}$ bei doppelter Genauigkeit)

$$\begin{aligned} \left| \text{rd} \left(\frac{f(x+\xi) - f(x)}{\xi} \right) - f'(x) \right| &\approx C_1 \frac{\varepsilon}{\xi} + C_2 \xi, \quad \xi_{\text{optimal}} \approx \sqrt{\varepsilon} \approx 10^{-8}, \\ \left| \text{rd} \left(\frac{f(x) - f(x-\xi)}{\xi} \right) - f'(x) \right| &\approx C_1 \frac{\varepsilon}{\xi} + C_2 \xi, \quad \xi_{\text{optimal}} \approx \sqrt{\varepsilon} \approx 10^{-8}, \\ \left| \text{rd} \left(\frac{f(x+\xi) - f(x-\xi)}{2\xi} \right) - f'(x) \right| &\approx C_1 \frac{\varepsilon}{\xi} + C_2 \xi^2, \quad \xi_{\text{optimal}} \approx \sqrt[3]{\varepsilon} \approx 10^{-5}, \\ \left| \text{rd} \left(\frac{f(x+\xi) - 2f(x) + f(x-\xi)}{\xi^2} \right) - f''(x) \right| &\approx C_1 \frac{\varepsilon}{\xi^2} + C_2 \xi^2, \quad \xi_{\text{optimal}} \approx \sqrt[4]{\varepsilon} \approx 10^{-4}. \end{aligned}$$

Numerische Tests. Numerische Tests bestätigen die theoretische Fehleranalyse.

% Numerische Differentiation

```
clear all
close all
clc
format long

f = inline('sin(x)');
df = inline('cos(x)');
ddf = inline('-sin(x)');
x = pi/4;

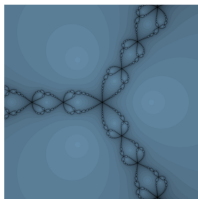
% f = inline('x.^4');
% df = inline('4*x.^3');
% ddf = inline('12*x.^2');
% x = linspace(-2,2,100);

disp('Erste Ableitung')
LoopMax = 16;
for loop = 1:LoopMax
    xi = 1/10^loop;
    Differenz1 = (f(x+xi) - f(x))/xi;
    Fehler1(loop,1) = norm(Differenz1 - df(x),inf);
    Differenz2 = (f(x) - f(x-xi))/xi;
    Fehler2(loop,1) = norm(Differenz2 - df(x),inf);
    Differenz3 = (f(x+xi) - f(x-xi))/(2*xi);
    Fehler3(loop,1) = norm(Differenz3 - df(x),inf);
end
figure
Fehler1
Minimum = min(Fehler1)
InkrementOptimal = 1/10^(find(Fehler1 == Minimum))
semilogy([1:LoopMax]',Fehler1,'o-')
title('Fehler (Vorwärtsdifferenz)')
pause

figure
Fehler2
Minimum = min(Fehler2)
InkrementOptimal = 1/10^(find(Fehler2 == Minimum))
semilogy([1:LoopMax]',Fehler2,'o-')
title('Fehler (Rückwärtsdifferenz)')
pause

figure
Fehler3
Minimum = min(Fehler3)
InkrementOptimal = 1/10^(find(Fehler3 == Minimum))
semilogy([1:LoopMax]',Fehler3,'o-')
title('Fehler (Symmetrische Differenz)')
pause

disp('Zweite Ableitung')
LoopMax = 8;
```

Numerische Integration

Resultat zu symmetrischen Quadraturformeln

Resultat. Die Ordnung einer symmetrischen Quadraturformel ist gerade.

Begründung. Die Knoten und Gewichte der betrachteten Quadraturformel werden mit $(b_i, c_i)_{i=1}^s$ bezeichnet. Man fixiert $m \in \mathbb{N}_{\geq 0}$ und geht von der Annahme aus, dass man für jede Polynomfunktion p_{2m} vom Grad $2m$ mittels Quadraturapproximation den exakten Wert eines bestimmten Integrales

$$\int_a^b p_{2m}(x) dx$$

erhält. Insbesondere gilt (lokale Approximation)

$$\sum_{i=1}^s b_i p_{2m}(c_i) - \int_0^1 p_{2m}(x) dx = 0.$$

Um zu zeigen, dass daraus folgt, dass auch jede Polynomfunktion p_{2m+1} vom Grad $2m+1$ exakt integriert wird, wendet man folgende vereinfachende Überlegungen an. Da man jede Polynomfunktion p_{2m+1} vom Grad $2m+1$ in der Form (Division mit Rest)

$$p_{2m+1} = C q_{2m+1} + p_{2m}, \quad q_{2m+1} : \mathbb{R} \longrightarrow \mathbb{R} : x \longmapsto \left(x - \frac{1}{2}\right)^{2m+1},$$

mit Konstante $C \in \mathbb{R}$ und einer Polynomfunktion p_{2m} vom Grad $2m$ darstellen kann, reicht es aus, die Gleichheit

$$\sum_{i=1}^s b_i q_{2m+1}(c_i) - \int_0^1 q_{2m+1}(x) dx = 0$$

herzuleiten. Aufgrund der getroffenen Annahme erhält man dann das gewünschte Resultat

$$\begin{aligned} & \sum_{i=1}^s b_i p_{2m+1}(c_i) - \int_0^1 p_{2m+1}(x) dx \\ &= C \left(\sum_{i=1}^s b_i q_{2m+1}(c_i) - \int_0^1 q_{2m+1}(x) dx \right) + \sum_{i=1}^s b_i p_{2m}(c_i) - \int_0^1 p_{2m}(x) dx \\ &= 0. \end{aligned}$$

Für die gewählte spezielle Polynomfunktion ist einerseits der Wert des Integrales gleich Null (Symmetrie bzgl. $\frac{1}{2}$, bei einer Transformation des Intervalles $[0, 1]$ auf $[-1, 1]$ ergibt sich eine ungerade Funktion)

$$\int_0^1 q_{2m+1}(x) \, dx = \int_0^1 \left(x - \frac{1}{2}\right)^{2m+1} \, dx = \frac{1}{2m+2} \left(x - \frac{1}{2}\right)^{2m+2} \Big|_0^1 = 0,$$

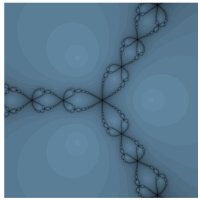
andererseits gilt für jede symmetrische Quadraturformel ($b_s = b_1$, $c_s = 1 - c_1$, $c_s - \frac{1}{2} = \frac{1}{2} - c_1$)

$$b_1 \left(c_1 - \frac{1}{2}\right)^{2m+1} + b_s \left(c_s - \frac{1}{2}\right)^{2m+1} = b_1 \left(c_1 - \frac{1}{2}\right)^{2m+1} (1 + (-1)^{2m+1}) = 0,$$

$$\sum_{i=1}^s b_i q_{2m+1}(c_i) = \sum_{i=1}^s b_i \left(c_i - \frac{1}{2}\right)^{2m+1} = b_1 \left(c_1 - \frac{1}{2}\right)^{2m+1} + b_s \left(c_s - \frac{1}{2}\right)^{2m+1} + \dots = 0.$$

Somit ist die Ordnung einer symmetrischen Quadraturformel gerade. ◇

Illustrationen. Zu symmetrischen Quadraturformeln zählen die Newton–Côtes Quadraturformeln und die Gauß-Quadraturformeln.



Ergänzung zum Thema Numerische Integration Schwach singuläre Integranden und Ordnungsreduktionen

Beobachtung. Die bei schwach singulären Integranden beobachteten Ordnungsreduktionen werden im Folgenden für einen Spezialfall erklärt, die Mittelpunktsregel für eine äquidistante Zerlegung des Einheitsintervalles

$$\begin{aligned}\alpha &\in (0, 1), \quad f: (0, \infty) \longrightarrow \mathbb{R}: x \longmapsto x^{-1+\alpha}, \\ \beta > 0, \quad \int_0^\beta f(x) \, dx &= \frac{1}{\alpha} x^\alpha \Big|_0^\beta = \frac{1}{\alpha} \beta^\alpha < \infty, \\ N \in \mathbb{N}, \quad h > 0, \quad Nh &= \beta = 1, \\ S(h) &= h \sum_{j=0}^{N-1} f(jh + \frac{h}{2}) = h^\alpha \sum_{j=0}^{N-1} (j + \frac{1}{2})^{-1+\alpha}, \\ I(h) &= \int_0^{Nh} f(x) \, dx = \sum_{j=0}^{N-1} \int_{jh}^{(j+1)h} f(x) \, dx = \frac{1}{\alpha} h^\alpha \sum_{j=0}^{N-1} ((j+1)^\alpha - j^\alpha).\end{aligned}$$

Der globale Verfahrensfehler ist in diesem Fall durch den Absolutbetrag der Differenz

$$D(h) = S(h) - E(h) = h^\alpha \sum_{j=0}^{N-1} \left((j + \frac{1}{2})^{-1+\alpha} - \frac{1}{\alpha} ((j+1)^\alpha - j^\alpha) \right)$$

gegeben. Mittels der Potenzreihendarstellung (Binomische Reihe für reelle Exponenten)

$$(a+b)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} a^{\alpha-k} b^k, \quad \binom{\alpha}{k} = \prod_{\ell=1}^k \frac{\alpha+1-\ell}{\ell},$$

kann man beispielsweise folgende Gleichheiten ($\xi \leftrightarrow j + \frac{1}{2} > 0$, $2i+1 \leftrightarrow k$ ungerade)

$$\begin{aligned}(\xi + \frac{1}{2})^\alpha - (\xi - \frac{1}{2})^\alpha &= \sum_{k=0}^{\infty} \binom{\alpha}{k} \xi^{\alpha-k} \left(\frac{1}{2}\right)^k (1 - (-1)^k) = \sum_{i=0}^{\infty} \binom{\alpha}{2i+1} \left(\frac{1}{4}\right)^i \xi^{-2i-1+\alpha} \\ &= \alpha \xi^{-1+\alpha} + \sum_{i=1}^{\infty} \binom{\alpha}{2i+1} \left(\frac{1}{4}\right)^i \xi^{-2i-1+\alpha}, \\ \xi^{-1+\alpha} - \frac{1}{\alpha} \left((\xi + \frac{1}{2})^\alpha - (\xi - \frac{1}{2})^\alpha \right) &= -\frac{1}{\alpha} \sum_{i=1}^{\infty} \binom{\alpha}{2i+1} \left(\frac{1}{4}\right)^i \xi^{-2i-1+\alpha},\end{aligned}$$

ableiten. Daraus ergibt sich eine mit den Beobachtungen konsistente Abschätzung für den globalen Fehler (Beträge der Binomialkoeffizienten beschränkte durch Eins, konvergente geometrische Reihe, konvergente Reihe)

$$\sum_{i=1}^{\infty} \left(\frac{1}{4}\right)^i < \infty, \quad \sum_{j=0}^{\infty} \left(j + \frac{1}{2}\right)^{-2} < \infty,$$
$$|D(h)| \leq C h^\alpha.$$

```
clear all
close all
clc
format long e
```

```
% Anzahl der Stufen
s = 1
```

```
% Wahl des Integranden
Fall = 1;
```

```
if Fall == 1
    % Regulärer Integrand
    Exponent = 2*s + 1
    a = [-1,2];
end
if Fall == 2
    % Integrand mit singulärer Ableitung
    a = [0,2];
    Exponent = 1/2;
end
if Fall == 3
    % Schwach singulärer Integrand
    a = [0,2];
    Exponent = - 1/2;
end
if Fall == 4
    % Schwach singulärer Integrand
    a = [0,2];
    alpha = 1/3
    Exponent = - 1 + alpha;
end
```

```
% Bestimmung der Ordnung
% dx = (b-a)/M
% Fehler(dx) approx C dx^p
% Fehler(dx/2) approx C (dx/2)^p
% ln(Fehler(dx)/Fehler(dx/2)) approx p ln(2)
```

```
IntegralExakt ...
= Stammfunktion(a(2),Exponent) - Stammfunktion(a(1),Exponent)
```

```
CounterMax = 10;
for counter = 1:CounterMax
    M(counter,1) = 2^counter;
    Approximation(counter,1) = GaussQuadratur(Exponent,a,s,M(counter));
    Fehler(counter,1) ...
    = abs(Approximation(counter,1) - IntegralExakt);
end
```

```
M
```

```
Approximation
Fehler
ErwarteteOrdnung = 2*s
Ordnung = log(Fehler(2:end)./Fehler(1:end-1))./log(M(1:end-1)./M(2:end))
```

```
% Integrand, Stammfunktion
```

```
function y = f(x,Exponent)
```

```
y = x.^Exponent;
```

```
end
```

```
function y = Stammfunktion(x,Exponent)
```

```
y = 1/(Exponent+1)*x.^(Exponent+1);
```

```
end
```

```
function Approximation = GaussQuadratur(Exponent,a,s,M)
```

```
x = linspace(a(1),a(2),M).';
dx = x(2)-x(1);
```

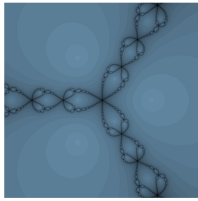
```
if s == 1
    b = 1;
    c = 1/2;
    Knoten = x(1:end-1) + c*dx;
    Approximation = b*dx*sum(f(Knoten,Exponent));
end
```

```
if s == 2
    b = 1/2;
    c = 1/2*(1 - 1/sqrt(3));
    Knoten = x(1:end-1) + c*dx;
    Aux1 = b*dx*sum(f(Knoten,Exponent));
    b = 1/2;
    c = 1/2*(1 + 1/sqrt(3));
    Knoten = x(1:end-1) + c*dx;
    Aux2 = b*dx*sum(f(Knoten,Exponent));
    Approximation = Aux1 + Aux2;
end
```

```
if s == 3
    b = 5/18;
    c = 1/2*(1 - sqrt(3/5));
    Knoten = x(1:end-1) + c*dx;
    Aux1 = b*dx*sum(f(Knoten,Exponent));
    b = 8/18;
    c = 1/2;
    Knoten = x(1:end-1) + c*dx;
    Aux2 = b*dx*sum(f(Knoten,Exponent));
```

```
b = 5/18;  
c = 1/2*(1 + sqrt(3/5));  
Knoten = x(1:end-1) + c*dx;  
Aux3 = b*dx*sum(f(Knoten,Exponent));  
Approximation = Aux1 + Aux2 + Aux3;  
end  
  
end
```





Numerische Mathematik 1 & 2
Mechthild Thalhammer
Illustrationen (Kapitel *Polynominterpolation*)

Illustrationen zum Kapitel *Polynominterpolation*

- **Dividierte Differenzen / Horner-Schema, Divergenz / Konvergenz (MATLAB)**
- **Kondition (MATLAB)**

% Polynominterpolation

```
clear all
close all
clc
```

```
Teil = 1
```

```
if Teil == 1
    % Zugrundeliegende Polynomfunktion
    Grad = 5;
    f = inline('x.^5 + 2*x.^4 + 3*x.^3 + 4*x.^2 + 5*x + 6');
    Grad = 6;
    f = inline('1/2*x.^6 + x.^5 + 2*x.^4 + 3*x.^3 + 4*x.^2 + 5*x + 6');
    d = Grad + 1;
    % Alternative 1: Äquidistante Stützstellen
    a = [-3,2]';
    x = linspace(a(1),a(2),d)';
    format rational
    % Alternative 2: Zufällig gewählte Stützstellen
    a = [-1,1]';
    x = a(1) + (a(2)-a(1)).*rand(d,1)';
    format short e
    % Zugehörige Funktionswerte
    y = f(x);
    % Dividierte Differenzen
    Delta = DividierteDifferenzenMitAusgabe(x,y);
    % Auswertung mittels Horner-Schema
    format long e
    M = 100;
    X = linspace(a(1),a(2),M)';
    Y = f(X);
    P = HornerSchema(x,X,Delta);
    MaximaleDifferenz = max(P - Y)
    plot(X,Y,'b',X,P,'r.')
end
```

```
if Teil == 2
    % Sinusfunktion / Cosinusfunktion
    a = [-3,2]';
    % Alternative 2
    f = inline('cos(5*x)');
    % Alternative 1
    f = inline('sin(x)');
    CounterMax = 5;
    for counter = 1:5
        % Alternative 2
        d = 10*counter;
        % Alternative 1
        d = 2*counter;
```

```

x = linspace(a(1),a(2),d)';
% Alternative 2: Zufällig gewählte Stützstellen
x = a(1) + (a(2)-a(1)).*rand(d,1)';
x(1) = a(1);
x(end) = a(2);
% Alternative 1: Äquidistante Stützstellen
x = linspace(a(1),a(2),d)';
% Zugehörige Funktionswerte
y = f(x);
% Dividierte Differenzen
Delta = DividierteDifferenzen(x,y);
% Auswertung mittels Horner-Schema
M = 1001;
Inkrement = 0;
X = linspace(a(1)-Inkrement,a(2)+Inkrement,M)';
Y = f(X);
P = HornerSchema(x,X,Delta);
MaximaleDifferenz(counter,1) = max(P - Y)
figure
plot(X,Y,'b',X,P,'r.')
axis([a(1)-Inkrement,a(2)+Inkrement,-1,1])
grid on
pause
end
end

```

%%%

```

if Teil == 3
% Rationale Funktion
a = [-1,1]';
f = inline('1./(1 + 25*x.^2)');
CounterMax = 5;
for counter = 1:5
d = 10*counter;
% Äquidistante Stützstellen
x = linspace(a(1),a(2),d)';
% Zugehörige Funktionswerte
y = f(x);
% Dividierte Differenzen
Delta = DividierteDifferenzen(x,y);
% Auswertung mittels Horner-Schema
M = 1001;
Inkrement = 0;
X = linspace(a(1)-Inkrement,a(2)+Inkrement,M)';
Y = f(X);
P = HornerSchema(x,X,Delta);
MaximaleDifferenz(counter,1) = max(P - Y)
figure
plot(X,Y,'b',X,P,'r.')
axis([a(1)-Inkrement,a(2)+Inkrement,-10,10])
axis([a(1)-Inkrement,a(2)+Inkrement,-0.5,1.5])
grid on
pause
end
end

```

```

if Teil == 4
    a = [-1,1]';
    CounterMax = 5;
    for counter = 1:5
        Grad = 10*counter
        M = 1000;
        x = linspace(a(1),a(2),M)';
        % Chebychev-Polynom
        p = cos(Grad*acos(x));
        % Zugehörige Nullstellen
        Nullstellen = cos(([1:Grad]'-1/2)*pi/Grad)';
        figure
        plot(x,p,'r')
        hold on
        plot(Nullstellen,0*Nullstellen,'ko')
        axis([a(1),a(2),-1,1])
        grid on
        pause
    end
end

```

```

if Teil == 5
    % Rationale Funktion
    a = [-1,1]';
    f = inline('1./(1 + 25*x.^2)');
    CounterMax = 5;
    for counter = 1:5
        d = 10*counter;
        % Chebychev-Knoten
        x = cos(([1:d]'-1/2)*pi/d)';
        % Zugehörige Funktionswerte
        y = f(x);
        % Dividierte Differenzen
        Delta = DividierteDifferenzen(x,y);
        % Auswertung mittels Horner-Schema
        M = 1001;
        Inkrement = 0;
        X = linspace(a(1)-Inkrement,a(2)+Inkrement,M)';
        Y = f(X);
        P = HornerSchema(x,X,Delta);
        MaximaleDifferenz(counter,1) = max(P - Y)
        figure
        plot(X,Y,'b',X,P,'r.')
        axis([a(1)-Inkrement,a(2)+Inkrement,-10,10])
        axis([a(1)-Inkrement,a(2)+Inkrement,-0.5,1.5])
        grid on
        pause
    end
end

```

% Kondition der Polynominterpolation (Abschätzung)

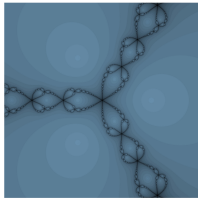
```
clear all
close all
clc
format short e

a = -1;
b = 1;
M = 1000;
for counter = [3,5,10,20,30]
    Grad = counter
    % Alternative 2 (Chebychev)
    Knoten = cos(([1:Grad]'-1/2)*pi/Grad)';
    % Alternative 1
    Knoten = linspace(a,b,Grad);
    X = linspace(a,b,M);
    Kondition = 0;
    figure
    for j = 1:Grad
        L{j} = 1;
        for k = 1:j-1
            L{j} = L{j}.*(X - Knoten(k))/(Knoten(j)-Knoten(k));
        end
        for k = j+1:Grad
            L{j} = L{j}.*(X - Knoten(k))/(Knoten(j)-Knoten(k));
        end
        Wert = 0*Knoten;
        Wert(j) = 1;
        plot(Knoten,Wert,'ro')
        hold on
        plot(X,L{j})
        Kondition = Kondition + abs(L{j});
    end
    pause
    figure
    MaxKondition = max(Kondition)
    plot(X,Kondition)
    pause
end
```

% Kondition (Vandermonde-Matrix)

```
clear all
close all
clc
format short e

CounterMax = 6;
for counter = 2:CounterMax
    Grad = 2^counter
    % Alternative 1
```

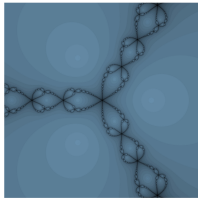
Numerische Mathematik 1 & 2
Mechthild Thalhammer
Einführung in Matlab

Einführung in MATLAB (Peter Arbenz, 2007/2008)

1. Einleitung

1.1. Was ist MATLAB?

1.1. Geschichtliches



Einführung in MATLAB (Peter Arbenz, 2007/2008)

2. Grundlegendes

2.1. Das MATLAB-Fenster

2.2. Hilfe aus dem Befehlsfenster

2.3. Matrizen als grundlegender Datentyp

2.4. Zuweisungen

2.5. Namen

2.6. Eingabe von Matrizen

2.7. Operationen mit Matrizen

2.8. Ausgabeformate

2.9. Komplexe Zahlen

2.10. Speichern von Daten zur späteren Weiterverwendung

2.11. Strukturen (structure arrays)

2.12. Zeichenketten (string arrays)

2.13. Zellen (cell arrays)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einführung in Matlab (Peter Arbenz, 2007/2008)
% people.inf.ethz.ch/arbenz/MatlabKurs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Kapitel 2: Grundlegendes
% 2.1 Das Matlab-Fenster
% 2.2 Hilfe aus dem Befehlsfenster
% 2.3 Matrizen als grundlegender Datentyp
% 2.4 Zuweisungen
% 2.5 Namen
% 2.6 Eingabe von Matrizen
% 2.7 Operationen mit Matrizen
% 2.8 Ausgabeformate
% 2.9 Komplexe Zahlen
% 2.10 Speichern von Daten zur späteren Weiterverwendung
% 2.11 Strukturen (structure arrays)
% 2.12 Zeichenketten (string arrays)
% 2.13 Zellen (cell arrays)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Matlab öffnen
% Datei erstellen (-> New -> Script)
% Speichern (DateiName.m)
% Ausführen (-> Run)
% Alternative: Aufruf einer Datei im Befehlsfenster (DateiName ohne Endung)
% Matlab beenden
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Help center (Befehlsfenster: help matlab -> Matlab Documentation)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erste Zeilen eines Matlab-Skripts
% Sämtliche Variablen und Funktionen im Arbeitsspeicher löschen
clear all
% Sämtliche Bilder schließen
close all
% Sämtliche Eingaben im Befehlsfenster nicht mehr sichtbar
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Grundkonstrukt [Variable =] Ausdruck
% Ausdruck: Verknüpfung von Variablennamen, Operatoren und Funktionen
% Resultat: Matrix (array) mit zugewiesenem Variablennamen oder "ans"
% Zahl ohne / mit Ausgabe
pi;
pi
% Zuweisung ohne / mit Ausgabe
a = 1;
a = 1
% Auflistung der Variablen im Arbeitsspeicher
whos
clear ans a
whos
% Trennung von (längeren) Ausdrücken
a = 1;
b = a + a;
c = a ...
+ b;
whos
clear all

```

```

whos
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Einzelne Punkte zeichnen
plot(1,2,'ro')
hold on
plot(1,2,'b.')
figure
plot(2,1,'k.')
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
% Namen von Variablen und Funktionen beginnen mit einem Buchstaben
% Beliebige Zahl von Buchstaben, Zahlen, Unterstrichen
% Matlab unterscheidet zwischen Gross- und Kleinschreibung
% Nicht definiert ("Unrecognized function or variable 'a'.")
% a
% Variablenname nicht zulässig
% ("Invalid expression. Check for missing multiplication operator, missing
% or unbalanced delimiters, or other syntax error. To construct matrices,
% use brackets instead of parentheses. Did you mean: a = 1")
% 1a = 1
% Variablenname zulässig (Zahl)
a1 = 1
% Einzelne Zahl
a(1) = 1
a = 1
% Spalte/Matrix (Elemente werden mit Null belegt)
B(3,1) = 1
C(3,2) = -1
% Buchstaben/Wörter als Variablenname zulässig (string)
a = 'Buchstaben/Wörter sind auch eine Option'
if a == 'Buchstaben/Wörter sind auch eine Option'
    b = 1
else
    b = -1
end
% Fehlermeldung ("Arrays have incompatible sizes for this operation.")
% if a == 'Länge des String nicht passend'
% end
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Komplexe Zahlen
% VORSICHT! 1i, 2i etc. verwenden (i ist oft Index)
i
for i = 1:2
end
i
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% IEEE Standard für Fließkommazahlen
% +/- Infinity, Not a number

```

```

1/0
-1/0
1/(-0)
-1/(-0)
Inf/0
-Inf/0
Inf/(-0)
-Inf/(-0)
0*Inf
0*(-Inf)
% Erweiterung auf komplexe Zahlen (Konvention)
1i/0
Inf*1i/0
0*Inf*1i
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Eingabe von Matrizen durch explizite Angabe der Elemente
% Eckige Klammern, übliche Anordnung der Elemente
% (Erstes Element (1,1) bzw. bei Vektor (1))
% Zeilen werden durch Strichpunkte getrennt
A = [1,2,3,4;2,3,4,5]
A = [1:4;2:5]
% Nicht empfohlen
A = [1:4;
2:5]
% Empfohlen (Kennzeichnung des Zeilenumbruches)
A = [1:4; ...
2:5]
A(1,1)
A(1,end)
A(end,1)
A(end,end)
A(1,:)
A(:,1)
MagischesQuadrat = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
% Matlab ermöglicht dynamische Speicherplatz-Allokation
C = [A;MagischesQuadrat]
Dimension = size(C)
AnzahlZeilen = size(C,1)
AnzahlSpalten = size(C,2)
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Eingabe von Matrizen mittels generierender Funktionen
% Nullmatrix, Elemente gleich eins, Erweiterung der Einheitsmatrix
Dimension = [2,3]
A = zeros(Dimension)
A = ones(Dimension)
A = eye(Dimension)
% Diagonalmatrix
B = diag([1,2])
% Subtraktion
C = B - 1
A = ones(2);
C = B - A

```

```

Diagonale = diag(C)
% Zufällige Elemente
d = 100;
R = rand(d);
figure
pcolor(R)
colormap(hot)
colorbar
R = randn(d);
figure
pcolor(R)
colormap(hot)
colorbar
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Entsprechende mehrdimensionale Erweiterung
Dimension = [2,3,4]
for i = 1:Dimension(3)
    A(:, :, i) = i*ones(Dimension(1:2));
end
A
% Maximale Datengröße bei Vektoren/Matrizen
% Fehlermeldungen
% ("Requested 1000000000x1 (74.5GB) array exceeds maximum array size
% preference (16.0GB). This might cause MATLAB to become unresponsive.")
% ("Requested 100000x100000 (74.5GB) array exceeds maximum array size
% preference (16.0GB). This might cause MATLAB to become unresponsive.")
clear all
format short
% LoopMax = 10
LoopMax = 9
for loop = 1:LoopMax
    Dimension = 10^loop
    Vektor = ones(Dimension,1);
    clear Vektor
end
clear all
format short
% LoopMax = 5
LoopMax = 4
for loop = 1:LoopMax
    Dimension = 10^loop
    Matrix = ones(Dimension);
    clear Matrix
end
% Zeitmessung
clear all
format long
LoopMax = 4
for loop = 1:LoopMax
    Dimension = 10^loop
    A = rand(Dimension);
    B = rand(Dimension);
    tic
    C = A + B;

```

```

    ZeitAddition(loop,1) = toc
    tic
    C = A*B;
    ZeitMultiplikation(loop,1) = toc
    clear A B C
end
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Eingabe von Matrizen mittels Dateien
% Speichern von bestimmten Daten (mat-Datei)
A = [1,2,3;4,5,6;7,8,9]
save TestDatei1 A
% Speichern aller Daten
B = A.^2
save TestDatei2
% Löschen aller Daten
clear all
% A
% B
% Verwendung der Daten aus TestDatei2.mat
load TestDatei2
C = A + B
% Verwendung der Daten aus externer Datei TestDatei3.c (geeignetes Format)
% Variablenname = Dateiname
load TestDatei3.c
TestDatei3
% VORSICHT! A wird ersetzt
A = TestDatei3
B
C
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Doppelpunktoperator
i = [1:2:10]'
j = i - 1
x = [-5:0.1:5]';
plot(x,exp(-x.^2),'k.')
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Zugriff auf Elemente (Siehe oben)
d = [2,3];
A = round(10*randn(d))
A11 = A(1,1)
A23 = A(end,end)
% Fehlermeldung
% "Index in position 1 exceeds array bounds. Index must not exceed 2."
% A(3,3)
AZeile1 = A(1,:)
ASpalte2 = A(:,2)
B = A(1:2,2:3)
% Elementweiser Betrag
ABetrag = abs(A)
% Summieren

```



```

ASummeSpalten = sum(A,1)
ASummeZeilen = sum(A,2)
sum(sum(A))
sum(sum(abs(A)))
% Zeitlicher Vergleich (Summation aller Elemente)
clear all
format long
d = [10,10];
d = [10^3,10^4];
% d = [10^4,10^5];
A = randn(d);
tic
ASumme = 0;
for i = 1:d(1)
    for j = 1:d(2)
        ASumme = ASumme + A(i,j);
    end
end
Resultat(1,1) = ASumme;
Zeit(1,1) = toc;
tic
Resultat(2,1) = sum(sum(A));
Zeit(2,1) = toc;
Zeit
Resultat
Differenz = Resultat(1,1)-Resultat(2,1)
% Zeitlicher Vergleich (Summation positiver/negativer Elemente)
clear all
format long
d = [10,10];
d = [10^3,10^4];
A = randn(d);
tic
ASummePositiv = 0;
ASummeNegativ = 0;
for i = 1:d(1)
    for j = 1:d(2)
        if A(i,j) > 0
            ASummePositiv = ASummePositiv + A(i,j);
        end
        if A(i,j) < 0
            ASummeNegativ = ASummeNegativ + A(i,j);
        end
    end
end
end
Resultat(1,:) = [ASummePositiv,ASummeNegativ];
Zeit(1) = toc;
tic
B = reshape(A,prod(d),1);
B = sort(B);
IndexPositiv = find(B>0);
IndexNegativ = find(B<0);
ASummePositiv = sum(B(IndexPositiv));
ASummeNegativ = sum(B(IndexNegativ));
Resultat(2,:) = [ASummePositiv,ASummeNegativ];
Zeit(2,1) = toc;

```



```

clear all
% Punktweise Multiplikation/Division von Matrizen derselben Dimension
d = [2,3];
A = round(10*rand(d))
B = round(10*rand(d))
APunktweiseMultipliziertMitB = A.*B
APunktweiseDividiertDurchB = A./B
% VORSICHT! Division nicht wohldefiniert
B(1,1) = 0
APunktweiseDividiertDurchB = A./B
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
format rational
% Alternative Berechnung von euklidischen Skalarprodukten/Normen
d = 3;
v = round(10*rand(d,1)) + 1i*round(10*rand(d,1))
vZeile = v.'
vKonjugiert = conj(v)
vKonjugiert = (v').'
format short
vNorm = sqrt(vZeile*vKonjugiert)
vNorm = sqrt(sum(v.*vKonjugiert))
vNorm = norm(v,2)
vNorm = norm(v)
vNorm1 = norm(v,1)
vNormInf = norm(v,inf)
format rational
w = round(10*rand(d,1)) + 1i*round(10*rand(d,1))
vwSkalarprodukt = sum(v.*conj(w))
% Vorteil kompakter Codierung
vwSkalarprodukt = 0;
for i = 1:d
    vwSkalarprodukt = vwSkalarprodukt + v(i)*conj(w(i));
end
vwSkalarprodukt
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Potenzen
d = 3;
A = round(10*rand(d))
A*A
A^2
% Punktweises Potenzieren der Elemente
A.^2
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Systeme linearer Gleichungen A x = b (Lösung/"Division")
d = 3;
A = round(10*rand(d))
x = ones(d,1);
b = A*x
Loesung = A\b
pause

```

```

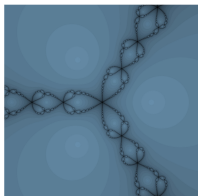
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Vergleichsoperationen
d = 3;
A = 1 + round(10*rand(d))
B = round(10*rand(d))
% Bedingung A(i,j) > B(i,j) erfüllt / nicht erfüllt entspricht 1/0
Elemente_AGroesserB = A > B
% Zugehörige Indizes (Spaltenweise Durchnummerierung der Elemente)
Index = find(A>B)
C = A;
C(Index) = 0
C == 0
C ~= 0
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
% Logische Operatoren
% Veranschaulichung des ersten und dritten Quadranten
Inkrement = 0.1;
xBereich = [-4,1];
yBereich = [-2,3];
x = [xBereich(1):Inkrement:xBereich(2)];
y = [yBereich(1):Inkrement:yBereich(2)];
[X,Y] = ndgrid(x,y);
A = -1*ones(size(X));
for i = 1:length(x)
    for j = 1:length(y)
        if (x(i) > 0 & y(j) > 0) | (x(i) < 0 & y(j) < 0)
            A(i,j) = 1;
        end
        if (x(i) > 0 & y(j) < 0)
            A(i,j) = 1/2;
        end
        if (x(i) < 0 & y(j) > 0)
            A(i,j) = -1/2;
        end
    end
end
end
figure
pcolor(X,Y,A)
colormap(flipud(hot))
colorbar
xlabel('x')
ylabel('y')
title('Der erste und der dritte Quadrant entsprechen dem Maximalwert.')
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ausgabeformate
% Vgl. Hilfe zu format, fprintf, sprintf
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
format rational
% Komplexe Zahlen
% VORSICHT! i,j werden oft als Indizes verwendet

```

```

i
j
% Empfohlen
ImaginaereEinheit = sqrt(-1)
ImaginaereEinheit = 1i
conj(ImaginaereEinheit)
ImaginaereEinheit'
% Realteil, Imaginärteil, Absolutbetrag, Argument
format long
z = 1/sqrt(2)*(1 + 1i)
zRe = real(z)
zIm = imag(z)
zAbs = abs(z)
zArgument = angle(z)
pi/4
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
format rational
% Speichern von Daten
% Vgl. Hilfe zu
% what (Matlab-Dateien im aktuellen Verzeichnis)
% who (Variablen), whos (Variablen und entsprechende Größen)
% save
d = 3
A = 1i*ones(d)
B = A*A'
C = real(A(1:2,1:2))
what
who
whos
save TestDatei
what
pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Weitere Datenstrukturen (structure arrays, string arrays, cell arrays)
% Vgl. Hilfe
A = ones(2);
B = zeros(3);
% Strukturen (structure arrays)
% Beliebige Kombinationen von Datenstrukturen
% Hilfreich für Angabe von Problemdaten beim Aufruf von Funktionen
Struktur.A = A;
Struktur.B = B;
Struktur.C = 'C'
Struktur.A
Studierende.Vorname = 'Anna';
Studierende.Nachname = 'Bauer';
Studierende.Semester = 3;
Studierende(2) ...
= struct('Vorname', 'Peter', 'Nachname', 'Mair', 'Semester', 4);
Studierende
Studierende(1)
Studierende(2)
% Zeichenketten (string arrays)

```

Blatt 1

Newton-Verfahren für komplexe polynomiale Gleichungen und Newton-Fraktale.

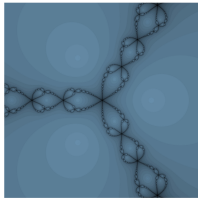
- (1) (i) Betrachte die komplexe Gleichung

$$z^3 = 1.$$

Bestimme alle Lösungen (Einheitswurzeln), und veranschauliche sie.

- (ii) Setze $z = x + iy \in \mathbb{C}$ (Realteil/Imaginärteil), und gib das entsprechende nichtlineare Gleichungssystem für $(x, y) \in \mathbb{R}^2$ an.
- (iii) Formuliere das Newton-Verfahren zur näherungsweise Lösung des Gleichungssystems. Überlege, welche Struktur der zugehörige Algorithmus hat (Eingabedaten, Ausgabedaten, Iteration). Implementiere das Verfahren in einer Programmiersprache Deiner Wahl. Wähle *geeignete* Startwerte, und beobachte quadratische Konvergenz gegen die jeweilige Lösung.
- (iv) Stelle für einen Startwert die zugehörigen Iterationswerte graphisch dar (dynamisch).
- (v) Überlege, wie das Newton-Verfahren für die ursprüngliche komplexe Gleichung lautet. Adaptiere die Implementierung entsprechend.
- (2) Erweitere die Implementierung, indem Du als Startwerte Gitterpunkte in einem Quadrat wählst, in dem alle Lösungen liegen. Weise diesen Punkte bei Konvergenz gegen eine Lösung eine Zahl (Farbe) zu, und stelle das Ergebnis graphisch dar.
- (3) Erweitere die Implementierung auf eine komplexe Gleichung ähnlicher Form.¹ Gib verschiedene Startwerte vor, und speichere im Falle von Konvergenz die jeweiligen Lösungen.

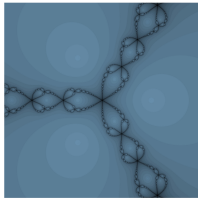
¹Siehe etwa de.wikipedia.org/wiki/Newtonfraktal.



Blatt 2

Binärdarstellung, Horner Schema, Nullstellen von Polynomen / Eigenwerte.

- (1) Gib an, wie man für eine natürliche Zahl die Koeffizienten der zugehörigen Binär- und Dezimaldarstellungen berechnet. Implementiere das Verfahren in einer Programmiersprache Deiner Wahl.
- (2) Gib das herkömmliche Verfahren sowie das Horner Schema zur Auswertung eines Polynomes an. Implementiere beide Verfahren in einer Programmiersprache Deiner Wahl. Bestimme jeweils die Anzahl der Additionen und Multiplikationen. Wende ein Polynom höheren Grades für eine große Anzahl von Argumenten an, und beobachte Unterschiede in den Rechenzeiten.
- (3) Berechne die Nullstellen eines Polynomes höheren Grades mittels einer Programmiersprache Deiner Wahl (in MATLAB mittels des Befehles *roots*). Beobachte bei einer kleinen Änderung der Koeffizienten eine signifikante Änderung der Nullstellen. Übertrage die Überlegungen auf die (reellen) Eigenwerte einer symmetrischen Matrix höherer Dimension.



Bairstow-Verfahren

Näherungsweise Berechnung der Nullstellen eines Polynomes. Vgl. etwa

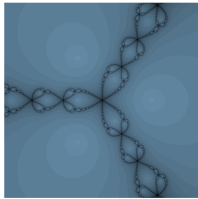
de.wikipedia.org/wiki/Bairstowverfahren.

- (1) Überlege, wie bei Division eines Polynomes p vom Grad $n \geq 3$ (mit reellen Koeffizienten und Leitkoeffizient gleich Eins) durch ein Polynom a vom Grad zwei (mit reellen Koeffizienten und Leitkoeffizient gleich Eins) die Koeffizienten der zugehörigen Polynome

$$p = ab + r, \quad \text{gr}(r) \leq 1,$$

lauten. Implementiere das sich ergebende Verfahren, und teste es anhand von Beispielen mit bekannter Lösung (Vorgabe von Polynomen a, b, r und Bestimmung von $p = ab + r$).

- (2) Skizziere die wesentlichen Schritte des Bairstow-Verfahrens.
- (3) Implementiere das Bairstow-Verfahren, und teste es anhand eines Beispiels mit bekannter Lösung (Vorgabe der Nullstellen bzw. Linearfaktoren bzw. quadratischen Faktoren). Beobachte quadratische Konvergenz (analog zum Newton-Verfahren).



Blatt 3

Einführung in MATLAB, vgl. Aufgaben in ARBENZ (Abschnitte 2.6.5 / 2.7.8).

- (1) (i) Überlege, welche Möglichkeiten MATLAB bietet, um die Matrix (Nullelemente entlang der Diagonale, Einselemente außerhalb)

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

zu generieren. Gib entsprechende Verallgemeinerungen für den Fall von d Dimensionen an.

- (ii) Überlege, welche Möglichkeiten MATLAB bietet, um die Matrix (zeilenweise Anordnung von aufsteigenden Zahlen)

$$A = \begin{pmatrix} a+1 & a+2 & a+3 \\ a+4 & a+5 & a+6 \\ a+7 & a+8 & a+9 \end{pmatrix}$$

mittels einer zweifachen Schleife (über Zeilen und Spalten) zu generieren. Was bewirkt der Befehl $A(d, d+2) = A(d, d) + 1$ mit $d = 3$? Gib entsprechende Verallgemeinerungen für den Fall von d Dimensionen an.

- (iii) Was bewirkt die Anwendung der Befehle `fliplr`, `flipud`, `rot90` auf die zuvor eingegebenen Matrizen?
(iv) Generiere ausgehend von der Einheitsmatrix

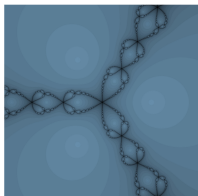
$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

die Matrizen (Einselemente entlang der Richtung Südwest–Nordost, zusätzliche Einselemente entlang der Diagonale)

$$B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

und entsprechende Verallgemeinerung auf d Dimensionen.

- (v) Stelle das Symbol des Roten Kreuzes sowie das Schweizerkreuz mittels des Befehles `spy` dar.
- (2)
 - (i) Generiere die Matrix aus (1)(ii) einerseits mittels einer einfachen Schleife und andererseits mittels des Befehles `reshape`. Erweitere die Überlegungen auf nicht-quadratische Matrizen.
 - (ii) Generiere eine Matrix der Form (1)(ii) sowie eine Matrix der Form (1)(v). Bestimme einerseits das Matrix-Produkt und andererseits das punktweise Produkt der beiden Matrizen.
 - (iii) Generiere eine Matrix der Form (1)(ii), und setze sämtliche Elemente außerhalb eines (geeignet gewählten) Wertebereiches $[a, b]$ auf Null.

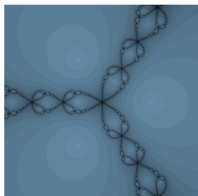


Blatt 4

Grundlegende Fragestellungen in den Bereichen Systeme linearer Gleichungen, Differential- und Integralrechnung.

- (1) Betrachte eine reguläre quadratische Matrix $A \in \mathbb{R}^{2 \times 2}$.
 - (i) Wende den Gauß-Algorithmus zur Lösung eines zugehörigen Systemes linearer Gleichungen an.
 - (ii) Implementiere das entsprechende Verfahren (Pivotwahl, Elimination, Rücksubstitution).
 - (iii) Gib eine Matrix an, für welche die Wahl des Pivotelementes das berechnete Resultat signifikant beeinflusst.
 - (iv) Erkläre die Beobachtungen mittels einer Rundungsfehleranalyse. Folgere daraus die Wahl des betragsmäßig größten Elementes der Matrix als Pivot (vollständige Pivotsuche).
- (2) Betrachte reguläre quadratische Matrizen $A \in \mathbb{C}^{d \times d}$ gegeben in unterer bzw. oberer Dreiecksform.
 - (i) Überlege, wie man zugehörige Systeme linearer Gleichungen löst.
 - (ii) Implementiere und teste die entsprechenden Verfahren (Vorwärtssubstitution / Rückwärtssubstitution).
 - (iii) Wie viele Rechenoperationen (Additionen / Subtraktionen, Multiplikationen / Divisionen) sind auszuführen?
- (3)
 - (i) Überlege, wie man mittels Differenzenquotienten Approximationen an die ersten und zweiten Ableitungen von regulären Funktionen erhält.
 - (ii) Leite mittels Taylorreihenentwicklung ein Resultat für den jeweiligen Approximationsfehler her.
 - (iii) Implementiere die Approximationsverfahren, und teste sie anhand von elementaren Funktionen.
 - (iv) Führe für einfache Beispiele, etwa Polynomfunktionen, eine Rundungsfehleranalyse durch.

- (4)
 - (i) Überlege, wie man mittels Riemann-Summen Approximationen an bestimmte Integrale von (regulären) Funktionen erhält.
 - (ii) Implementiere die Approximationsverfahren, und teste sie anhand von elementaren Funktionen.
 - (iii) Überlege, wie man mittels Taylorreihenentwicklungen bzw. der Betrachtung von Polynomfunktionen auf Aussagen über die Approximationsgüte der Verfahren kommt.
 - (iv) Bestätige die Vermutungen für einfache Testprobleme (Polynomfunktionen, äquidistante Zerlegungen).
- (5) Erweiterung von Aufgabe (4): Approximiert man auf dem jeweiligen Teilintervall den Integranden durch ein interpolierendes kubisches Polynom, so ergibt sich die Simpson-Regel.
 - (i) Gib das interpolierende kubische Polynom und die Simpson-Regel an.
 - (ii) Implementiere und teste das entsprechende Verfahren.
 - (iii) Welchen Approximationsfehler erwartet man? Welchen Approximationsfehler beobachtet man?



Blatt 5

Wahl der Summationsreihenfolge und Approximationen an unendliche Reihenwerte.

- (1) Berechne für $J \in \{10, 10^2, 10^3, 10^4, 10^5\}$ und $k \in \{1, 2, 3\}$ die Werte der endlichen Reihen

$$\sum_{j=1}^J \frac{1}{j^k}, \quad \sum_{j=1}^J \frac{(-1)^j}{j^k}.$$

Beobachte bei unterschiedlicher Wahl der Summationsreihenfolge unterschiedliche Resultate. Überlege, welche Reihenfolge zu empfehlen ist.

- (2) Erweitere die Überlegungen auf bekannte Funktionen wie Sinusfunktion und Exponentialfunktion, welche durch unendliche Reihen gegeben sind. Überlege, welche Strategien auf Approximationen hoher Genauigkeit führen.

Einführung in MATLAB, vgl. Aufgaben in ARBENZ (Abschnitt 4.6).

- (1) Verwende einerseits `for`-Schleifen und andererseits `while`-Schleifen sowie `if`-Verzweigungen, um die Matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 6 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 12 \\ 0 & 0 & 0 & 13 & 14 \\ 0 & 0 & 0 & 0 & 15 \end{pmatrix} \in \mathbb{R}^{5 \times 5}$$

und entsprechend $A \in \mathbb{R}^{d \times d}$ zu generieren.

- (2) Berechne den Wert des Kettenbruches

$$1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + 1}}}}}}}}}}$$

und entsprechender Verallgemeinerungen.

Bemerkung. Im Grenzfall ergibt sich der goldene Schnitt $\frac{1}{2}(1 + \sqrt{5})$.

- (3) Berechne ausgehend von verschiedenen Startwerten $a_0 \in \mathbb{N}$ (viele) Glieder der folgenden Folge

$$a_{n+1} = \begin{cases} 3a_n + 1, & \text{falls } a_n \text{ ungerade,} \\ \frac{1}{2}a_n, & \text{falls } a_n \text{ gerade,} \end{cases} \quad n \in \{1, 2, \dots, N\},$$

und plote sie.

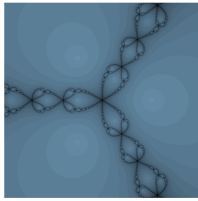
Bemerkung. LOTHAR COLLATZ stellte im Jahr 1937 die bis heute nicht vollständig bewiesene Vermutung auf, dass die Folge unabhängig vom gewählten Startwert gegen Eins konvergiert. Siehe auch

<https://de.wikipedia.org/wiki/Collatz-Problem>.

- (4) Die Chebychev-Polynome $p_n : [-1, 1] \rightarrow \mathbb{R}$ mit $n \in \mathbb{N}_{\geq 0}$ haben unter anderem im Bereich der Interpolation besondere Bedeutung. Bestimme mittels der Zweiterm-Rekursion

$$\begin{cases} p_0(x) = 1, & p_1(x) = x, \\ p_n(x) = 2x p_{n-1}(x) - p_{n-2}(x), & n \in \mathbb{N}_{\geq 2}, \end{cases} \quad x \in [-1, 1],$$

(viele) Funktionswerte der ersten Polynome zur Illustration deren Graphen.



Blatt 6

Gauß-Algorithmus, Cholesky-Algorithmus.

- (1) Betrachte eine reguläre Matrix mit komplexen Koeffizienten

$$A \in \mathbb{C}^{d \times d}, \quad \det(A) \neq 0.$$

Implementiere die LR -Zerlegung mittels Spaltenpivotsuche, d.h. bestimme eine Permutationsmatrix $P \in \mathbb{R}^{d \times d}$, eine untere Dreiecksmatrix $L \in \mathbb{C}^{d \times d}$ (mit Koeffizienten gleich Eins in der Diagonale) und eine obere Dreiecksmatrix $R \in \mathbb{C}^{d \times d}$, sodass

$$PA = LR.$$

- (i) Definiere Permutationsmatrizen und spezielle untere Dreiecksmatrizen, welche die benötigten Zeilenvertauschungen und Eliminationsschritte widerspiegeln. Schätze die benötigten Rechenoperationen, die bei tatsächlicher Ausführung der Matrix-Multiplikationen benötigt würden, ab.
- (ii) Adaptiere die Implementierung derart, dass keine Matrix-Multiplikationen ausgeführt werden.
- (iii) Ersetze zur Reduktion des benötigten Speicherplatzes die Koeffizienten der vorgegebenen Matrix durch die Koeffizienten der sich schrittweise ergebenden unteren und oberen Dreiecksmatrizen, und halte zusätzlich benötigte Zeilenvertauschungen mittels eines Vektors fest.
- (iv) Konstruiere reguläre quadratische Matrizen hoher Dimension und Lösungen zugehöriger Systeme linearer Gleichungen (mit $c = Pb$)

$$Ax = b \iff LRx = c \iff \begin{cases} Ly = c, \\ Rx = y, \end{cases}$$

vgl. Vorwärts- und Rückwärtssubstitution. Verwende diese zur Validierung der Implementierung.

- (v) *Zusatz.* Speichere die Koeffizienten der vorgegebenen Matrix in Form einer Zeile $a \in \mathbb{C}^{d^2}$ und adaptiere die Implementierung entsprechend. Überlege, wie man die Implementierung der LR -Zerlegung zeilenweise (mit minimalem Aufwand) ausführen würde.

(2) Betrachte eine symmetrische, positiv definite reelle Matrix

$$A \in \mathbb{R}^{d \times d}, \quad A^T = A, \quad \forall x \in \mathbb{R}^d \setminus \{0\}: x^T A x > 0.$$

Implementiere die Cholesky-Zerlegung (ohne Spaltenpivotsuche), d.h. bestimme eine untere Dreiecksmatrix $L \in \mathbb{R}^{d \times d}$ (mit positiven Koeffizienten in der Diagonale), sodass

$$A = LL^T.$$

Lineare Regression (Methode der kleinsten Quadrate, Überbestimmte Gleichungssysteme).

(1) Betrachte für eine vorgegebene reelle Matrix $A \in \mathbb{R}^{d \times d}$ und eine Spalte $b \in \mathbb{R}^d$ das Minimierungsproblem (bezüglich der euklidischen Norm)

$$\min_{x \in \mathbb{R}^d} \|Ax - b\|_2^2.$$

Setze dabei voraus, dass eine eindeutige Lösung existiert. Zeige einerseits mittels Resultaten der linearen Algebra und andererseits mittels Differentialrechnung, dass diese Lösung durch die zugehörigen (Gaußschen) Normalengleichungen

$$A^T A x = A^T b$$

bestimmt ist.

Hinweis. Begründe insbesondere, dass die reellwertige Funktion

$$f: \mathbb{R}^d \rightarrow \mathbb{R}: x \mapsto \frac{1}{2} \|Ax - b\|_2^2 = \frac{1}{2} (Ax - b)^T (Ax - b) = \frac{1}{2} (x^T A^T A x - 2b^T A x + b^T b)$$

folgende erste und zweite Ableitungen hat

$$(f'(x))^T = A^T A x - A^T b \in \mathbb{R}^{d \times 1}, \quad (f''(x))^T = A^T A.$$

(2) Für vorgegebene Datenpunkte

$$(\xi_i, \eta_i) \in \mathbb{R}^2, \quad i \in \{1, \dots, n\},$$

sollen die Koeffizienten eines Polynomes vom Grad $m \ll n$

$$p: \mathbb{R} \rightarrow \mathbb{R}: \xi \mapsto c_0 + c_1 \xi + \dots + c_m \xi^m = \begin{pmatrix} 1 & \xi & \dots & \xi^m \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{pmatrix}$$

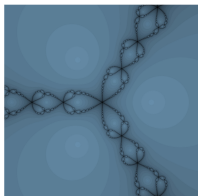
so bestimmt werden, dass die Bedingung

$$\sum_{i=1}^n (p(\xi_i) - \eta_i)^2 \rightarrow \text{minimal}$$

erfüllt ist. Überlege, dass dies dem Minimierungsproblem

$$\min_{x \in \mathbb{R}^d} \|Ax - b\|_2^2,$$
$$A = \begin{pmatrix} 1 & \xi_1 & \dots & \xi_1^m \\ 1 & \xi_2 & \dots & \xi_2^m \\ \vdots & \vdots & & \vdots \\ 1 & \xi_n & \dots & \xi_n^m \end{pmatrix} \in \mathbb{R}^{n \times (m+1)}, \quad x = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{pmatrix} \in \mathbb{R}^{m+1}, \quad b = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{pmatrix} \in \mathbb{R}^n,$$

entspricht. Gib an, wie man dessen (als eindeutig vorausgesetzte) Lösung berechnen kann. Wende die Vorgehensweise auf vorgegebene Polynome vom Grad $m \in \{1, 2, 3\}$ an, und verändere die exakten Funktionswerte mittels zufälliger Störungen.



Blatt 7

Newton-Côtes und Gauß-Quadraturformeln.

- (1) Bestimme die Knoten und Gewichte der zweistufigen Gauß-Quadraturformel mittels folgender Zugänge.
- (i) Lösung der Ordnungsbedingungen für $p = 2$ s.
 - (ii) Verwendung der Symmetrie-Eigenschaft und Lösung der relevanten Ordnungsbedingungen.
 - (iii) Berechne die Nullstellen des Legendre-Polynomes entsprechenden Grades. Die Transformation des Intervalles $[-1, 1]$ auf das Einheitsintervall führt auf die Gauß-Knoten. Die zugehörigen Gewichte erhält man aus den relevanten Ordnungsbedingungen, welche sich auf ein System linearer Gleichungen reduzieren. *Hinweis.* Die Legendre-Polynome $p_n : [-1, 1] \rightarrow \mathbb{R}$ mit $n \in \mathbb{N}_{\geq 0}$ erfüllen die Zweiterm-Rekursion

$$\begin{cases} p_0(x) = 1, & p_1(x) = x, \\ p_n(x) = \frac{1}{n} \left((2n-1)x p_{n-1}(x) - (n-1)p_{n-2}(x) \right), & n \in \mathbb{N}_{\geq 2}, \end{cases} \quad x \in [-1, 1].$$

- (2) Wähle elementare Integranden (Kompositionen von Polynomfunktionen, Trigonometrischen Funktionen, Exponentialfunktion) mit bekannten Stammfunktionen. Vergleiche für eine Folge von äquidistanten Zerlegungen die erzielte Genauigkeit von Trapezregel versus Mittelpunktsregel und Simpson-Regel versus zweistufiger Gauß-Quadraturformel. Verifiziere die Ordnung der Verfahren mittels folgender Überlegungen (Vernachlässigung relativ kleiner Terme höherer Ordnung)

Globaler Verfahrensfehler für regulären Integranden f und Feinheit $h > 0$:

$$\begin{aligned} \text{err}(h) &\approx C h^p \|f^{(p)}\|_{\infty}, \\ \ln\left(\frac{\text{err}(h_1)}{\text{err}(h_2)}\right) &\approx \ln\left(\frac{h_1}{h_2}\right)^p = p \ln\left(\frac{h_1}{h_2}\right), \quad p \approx \frac{\ln\left(\frac{\text{err}(h_1)}{\text{err}(h_2)}\right)}{\ln\left(\frac{h_1}{h_2}\right)}, \end{aligned}$$

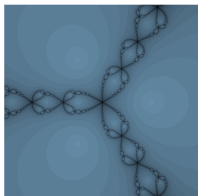
vgl. Skriptum, Abschnitt Elementare Quadraturformeln.

- (3) Adaptiere die bisherige Implementierung von Quadraturformeln (minimale Anzahl an Funktionsauswertungen, User wählt Integrationsintervall / Integrand / Quadraturformel).

- (4) Zeige, dass die Ordnung einer symmetrischen Quadraturformel gerade ist. Betrachte dazu exemplarisch eine Quadraturformel mit sechs Stufen, und folgere aus der Gültigkeit der Ordnungsbedingung

$$\sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}$$

für $q \in \{1, 2, 3, 4, 5\}$ sowie der Symmetrie-Eigenschaft die Gültigkeit der Bedingung für $q = 6$.



Blatt 8

Numerische Integration für singuläre Integranden.

(1) Betrachte die Gaußsche Quadraturformeln mit $s \in \{1, 2, 3\}$ Stufen.

(i) Verifiziere zunächst für Monome höheren Grades und mittels der Kenntnis des bestimmten Integrales

$$\int_a^b x^q dx, \quad q \in \mathbb{N},$$

numerisch deren Ordnung $p = 2s$.

(ii) Welche Ordnungen beobachtet man, wenn man diese Verfahren auf die Integranden

$$f: [0, b] \rightarrow \mathbb{R}: x \mapsto \sqrt{x}, \quad g: (0, b] \rightarrow \mathbb{R}: x \mapsto \frac{1}{\sqrt{x}},$$

anwendet?

(iii) Erweitere die Beobachtungen auf bestimmte Integrale der Form

$$\int_0^b x^{-1+\alpha} dx, \quad \alpha > 0.$$

Polynominterpolation.

(1) Gib das Verfahren der dividierten Differenzen sowie das verallgemeinerte Horner-Schema zur Bestimmung und Auswertung eines Interpolationspolynomes an, und implementiere beide Verfahren. Verifiziere den Code anhand einfacher Beispiele.

(2) Verwende die Verfahren zur Interpolation der rationalen Funktion

$$f: [-1, 1] \rightarrow \mathbb{R}: x \mapsto \frac{1}{1 + 25x^2}.$$

Bestimme Approximationen an Funktionswerte zu Argumenten im Intervall $[-1, 1]$ und etwas außerhalb davon. Wähle äquidistante Stützstellen, erhöhe den Grad des Interpolationspolynomes schrittweise ($\{10, 20, 30, 40, 50\}$), und stelle die Funktion sowie das interpolierende Polynom graphisch dar. Gib eine grobe Abschätzung für den maximalen Approximationsfehler auf $[-1, 1]$ an. Welche Beobachtungen macht man?

(3) Überlege, dass die durch

$$p_d : [-1, 1] \longrightarrow \mathbb{R} : x \longmapsto \cos(d \arccos(x)), \quad d \in \mathbb{N}_{\geq 0},$$

definierte Funktion auf eine Polynomfunktion, das Chebychev-Polynom vom Grad d , führt, die Dreiterm-Rekursion

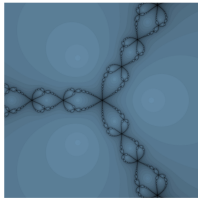
$$p_{d+1}(x) + p_{d-1}(x) = 2x p_d(x), \quad d \in \mathbb{N}_{\geq 1},$$

gilt, und die Nullstellen durch

$$x_j = \cos\left(\frac{(j-\frac{1}{2})\pi}{d}\right), \quad j \in \{1, \dots, d\}.$$

gegeben sind. Veranschauliche die Polynome sowie die zugehörigen Nullstellen für $d \in \{10, 20, 30, 40, 50\}$.

(4) Verwende zur Interpolation der zuvor angegebenen rationalen Funktion anstelle äquidistanter Stützstellen die Nullstellen von Chebychev-Polynomen. Erhöhe den Grad des Interpolationspolynomes schrittweise ($\{10, 20, 30, 40, 50\}$), und stelle die Funktion sowie das interpolierende Polynom graphisch dar. Welche Beobachtungen macht man?



Blatt 9

Vergleich von Zugängen.

- (1) Überlege, in welchen Situationen die numerische Berechnung eines bestimmten Integrals mittels einer Quadraturformel im Vergleich zur analytischen Bestimmung (Angabe und Auswertung einer Stammfunktion) von Vorteil ist, speziell in Hinblick auf den benötigten Rechenaufwand.
- (2) Im Zusammenhang mit dem Thema Polynominterpolation wird oft die Vandermonde-Matrix und ein zugehöriges lineares Gleichungssystem betrachtet (vgl. etwa de.wikipedia.org/wiki/Vandermonde-Matrix, Anwendung). Wähle ein einfaches Testproblem, für welches die Koeffizienten des Polynominterpolanten bekannt sind, erhöhe schrittweise den Grad des interpolierenden Polynomes, und bestimme einerseits die Kondition der auftretenden Vandermonde-Matrix (mittels des Befehles *cond*) und andererseits den Fehler der berechneten Koeffizienten. Vergleiche die Resultate mit jenen, die man mittels der bereits erstellten Implementierung (Dividierte Differenzen) erhält.

Implementierungen zu grundlegenden Problemstellungen.

- (1) Betrachte folgende Themen und effiziente numerische Verfahren.
 - (i) Direkte Verfahren für lineare Gleichungssysteme
(Gauß-Algorithmus mit Spaltenpivotsuche, Substitutionen)
 - (ii) Lineare Ausgleichsrechnung
(Cholesky-Algorithmus, Substitutionen)
 - (iii) Numerische Differentiation
(Symmetrische Differenzenverfahren)
 - (iv) Numerische Integration
(Quadraturformeln hoher Ordnung)
 - (v) Polynominterpolation
(Chebychev-Knoten, Dividierte Differenzen, Horner-Schema)

Adaptiere bestehende Implementierungen derart, dass jeweils ein einfaches Testproblem mit bekanntem Resultat und ein komplexes Problem gelöst wird.

Abgabe unter OLAT: Eine Datei pro Thema.