

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Higher-order Kuramoto-Daido-type models
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First-order model
% theta'(t)
% = Omega + K (SM(theta(t)) cos(theta(t)) - CM(theta(t)) sin(theta(t)))
% Precomputation of
% SM(theta(t)) = 1/M sum_{m=1}^M sin(theta_m(t))
% CM(theta(t)) = 1/M sum_{m=1}^M cos(theta_m(t))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Higher-order generalisations, see Kuramoto_HigherOrder_RHS
% sin(sigma1 thetal1) = sigma1 sin(thetal1)
% cos(sigma1 thetal1) = |sigma1| cos(thetal1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Maximal time stepsize for ode45 essential!
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
format long
clc
pause off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Large computation times for standard approach (-1,-2,-3) - Slow!
% Novel approach (1,2,3)
% Movies (11,12,13) - Slow!
% MyChoiceCases = [-1,-2,-3]
% MyChoiceCases = [11,12,13]
MyChoiceCases = [1,2,3]

for Case = MyChoiceCases
    Case
    clear Param
    [Param,ChoiceM,ChoiceOrderRKM,ChoiceK] = SetParameters(Case);
    for OrderRKM = ChoiceOrderRKM
        Param.OrderRKM = OrderRKM;
        disp(['OrderRKM = ',num2str(Param.OrderRKM)])
        for M = ChoiceM
            Param.M = M;
            disp(['M = ',num2str(Param.M)])
            Radius = 0.8*ones(Param.M,1);
            for K = ChoiceK
                Param.K = K;
                disp(['K = ',num2str(Param.K)])
                % Omega, Theta0
                Param.Omega = 1 + Param.omega*[-1:2/(Param.M-1):1]';
                Param.OmegaPotential = Param.Omega;
                Param.Theta0 = 2*pi/Param.M*[1:Param.M]';

                % Time integration
                t = Param.t0;
                Theta = Param.Theta0;
                if Param.OrderRKM == 1 | Param.OrderRKM == 2 |
                    Param.OrderRKM == 4

```

```

dt = (Param.TFinal-Param.t0)/Param.N;
All.t = t;
All.Theta = Theta;
tic
for n = 1:Param.N
    Theta = ExplicitRKM(Param.OrderRKM,t,Theta,dt, ...
                        @Kuramoto_HigherOrder_RHS
                        ,Param);
    t = Param.t0 + n*dt;
    All.t(n+1) = t;
    All.Theta(:,n+1) = Theta;
end
All.ComputationTime = toc;
end
if Param.OrderRKM == 45
    TimeGrid = linspace(Param.t0,Param.TFinal,Param.N+1);
    dtMax = TimeGrid(2)-TimeGrid(1);
    options = odeset('maxstep',dtMax);
    tic
    if Param.OrderRKM == 45
        [All.t,All.Theta] ...
        = ode45(@(t,Theta)
                Kuramoto_HigherOrder_RHS(t,Theta,Param), ...
                TimeGrid,Theta,options);
    end
    All.ComputationTime = toc;
    All.t = All.t';
    All.Theta = All.Theta';
end
% Graphics
if Param.GraphicsInitialFinalStateYes == 1
    Param.ColorBackground = Param.ColorInitial;

    Param = PlotState(1,All,Param);
    if Case < 0
        Param.ColorBackground = Param.ColorStandard;
    end
    if Case > 0
        Param.ColorBackground = Param.ColorNovel;
    end
    Param = PlotState(Param.N+1,All,Param);
    pause
end
% Movie
if Param.MovieEvolutionYes == 1
    if Case < 0
        Param.ColorBackground = Param.ColorStandard;
    end
    if Case > 0
        Param.ColorBackground = Param.ColorNovel;
    end
    MovieEvolution(All,Param)
    pause
end
end
end

```

```
        end
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [Param,ChoiceM,ChoiceOrderRKM,ChoiceK] = SetParameters(Case)
```

```
Param.MyCase = Case;
disp('Higher-order Kuramoto-Daido-type model')
Param.omega = 2;
Param.t0 = 0;
ChoiceOrderRKM = 45;
```

```
if abs(Case) < 10
    Param.OrderKuramoto = abs(Case);
    OrderKuramoto = Param.OrderKuramoto
    if Case < 0
        Param.SolutionWithPrecomputationYes = 0;
        disp(['Computation of solution without precomputation'])
    end
    if Case > 0
        Param.SolutionWithPrecomputationYes = 1;
        disp(['Computation of solution based on precomputation'])
    end
    % sigma1 in {1,-1}
    if Param.OrderKuramoto == 2
        Param.sigma1 = 1;
        Param.sigma2 = -1;
    end
    if Param.OrderKuramoto == 3
        Param.sigma1 = 1;
        Param.sigma2 = -1;
        Param.sigma3 = 1;
    end
    end
    if abs(Case) == 1
        ChoiceM = 1000;
        ChoiceK = 3;
    end
    if abs(Case) == 2
        ChoiceM = 100;
        ChoiceK = 12;
    end
    if abs(Case) == 3
        ChoiceM = 50;
        ChoiceK = 25;
    end
    end
    % Param.TFinal = 10;
    Param.TFinal = 1;
    Param.N = (Param.TFinal - Param.t0)*10;
    Param.PrintPlotYes = 1;
    Param.GraphicsFinalStateYes = 1;
    Param.GraphicsInitialFinalStateYes = 1;
    Param.MovieEvolutionYes = 1;
    Param.MovieN0 = 10;
```

```

    Param.FontSize = 20;
end

if Case > 10
    Param.OrderKuramoto = Case - 10;
    OrderKuramoto = Param.OrderKuramoto
    Param.SolutionWithPrecomputationYes = 1;
    disp(['Computation of solution based on precomputation'])
    % sigma1 in {1,-1}
    if Param.OrderKuramoto == 2
        Param.sigma1 = 1;
        Param.sigma2 = -1;
    end
    if Param.OrderKuramoto == 3
        Param.sigma1 = 1;
        Param.sigma2 = -1;
        Param.sigma3 = 1;
    end
    end
    ChoiceM = 1000;
    if Param.OrderKuramoto == 1
        ChoiceK = 3;
    end
    if Param.OrderKuramoto == 2
        ChoiceK = 13;
    end
    if Param.OrderKuramoto == 3
        ChoiceK = 25;
    end
    end
    % Param.TFinal = 50;
    Param.TFinal = 1;
    Param.N = (Param.TFinal - Param.t0)*10;
    ChoiceOrderRKM = 45;
    Param.PrintPlotYes = 0;
    Param.GraphicsFinalStateYes = 0;
    Param.GraphicsInitialFinalStateYes = 0;
    Param.MovieEvolutionYes = 1;
    Param.MovieN0 = 1;
    Param.FontSize = 14;
end

Param.ColorWhite = [1,1,1];
Param.ColorBlack = [0,0,0];
% d36e70
Param.ColorMyRed = [82.75,43.14,43.92]/100;
% 6a93b0
Param.ColorMyBlue = [41.57,57.65,69.02]/100;
% 7fb0b2
Param.ColorMyGreen = [49.8,69.02,69.8]/100;

Param.MarkerSize = 10;
Param.LineWidth = 1.5;
Param.ColorInitial = Param.ColorMyGreen;
Param.ColorStandard = Param.ColorMyBlue;
Param.ColorNovel = Param.ColorMyRed;

```

end

%%%

function RHS = Kuramoto_HigherOrder_RHS(t,Theta,Param)

if Param.SolutionWithPrecomputationYes == 0

RHS = zeros(size(Theta));

for m = 1:Param.M

if Param.OrderKuramoto == 1

for l = 1:Param.M

RHS(m) = RHS(m) + sin(Theta(l) - Theta(m));

end

end

if Param.OrderKuramoto == 2

for l1 = 1:Param.M

for l2 = 1:Param.M

RHS(m) = RHS(m) ...

+ sin(Param.sigma1*Theta(l1) ...

+ Param.sigma2*Theta(l2) - Theta(m));

end

end

end

if Param.OrderKuramoto == 3

for l1 = 1:Param.M

for l2 = 1:Param.M

for l3 = 1:Param.M

RHS(m) = RHS(m) ...

+ sin(Param.sigma1*Theta(l1) ...

+ Param.sigma2*Theta(l2) + Param.sigma3*Theta(l3) ...

- Theta(m));

end

end

end

end

end

RHS = Param.Omega + Param.K/Param.M^Param.OrderKuramoto*RHS;

end

if Param.SolutionWithPrecomputationYes == 1

S = sin(Theta);

C = cos(Theta);

SM = sum(S)/Param.M;

CM = sum(C)/Param.M;

if Param.OrderKuramoto == 1

Term1 = SM;

Term2 = CM;

end

if Param.OrderKuramoto == 2

Term1 = (Param.sigma1*abs(Param.sigma2) +

abs(Param.sigma1)*Param.sigma2)*SM*CM;

Term2 = (abs(Param.sigma1)*abs(Param.sigma2)*CM^2 -

Param.sigma1*Param.sigma2*SM^2);

end

if Param.OrderKuramoto == 3

```

    TermS = (Param.sigma2*abs(Param.sigma3) +
    abs(Param.sigma2)*Param.sigma3)*SM*CM;
    TermC = abs(Param.sigma2)*abs(Param.sigma3)*CM^2 -
    Param.sigma2*Param.sigma3*SM^2;
    Term1 = Param.sigma1*SM*TermC + abs(Param.sigma1)*CM*TermS;
    Term2 = abs(Param.sigma1)*CM*TermC - Param.sigma1*SM*TermS;
end
RHS = Param.Omega + Param.K*(Term1*C - Term2*S);
end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function All = ComputePotentialOrderParameterPreservation(All,Param)

SM_CM = Kuramoto_SM_CM(All.Theta,Param);
Param.OrderParameter = inline('SM_CM(2,:) + 1i*SM_CM(1,:)', 'SM_CM');
All.OrderParameterTheta = Param.OrderParameter(SM_CM);
All.OrderParameterModulusTheta = abs(All.OrderParameterTheta);
All.OrderParameterAngleTheta = angle(All.OrderParameterTheta);

if Param.KuramotoClassicalYes == 1
    All.PotentialTheta = - Param.Omega'*All.Theta ...
        + Param.M*Param.K/2 ...
        *(1 - All.OrderParameterModulusTheta.^2);
end

if Param.KuramotoClassicalYes == 0
    for n = 1:size(All.Theta,2)
        C = cos(All.Theta(:,n));
        AC = Param.APotential*C;
        S = sin(All.Theta(:,n));
        AS = Param.APotential*S;
        All.PotentialTheta(n) = C'*AC + S'*AS;
    end
    All.PotentialTheta = - Param.OmegaPotential'*All.Theta ...
        + Param.K/(2*Param.M) ...
        *(sum(Param.MmOne) - All.PotentialTheta);
end

% Preservation
% sum(theta') = sum(Omega)
% sum(theta(T)) - sum(theta(t0)) = (T - t0) sum(Omega)
All.Preservation = 1/Param.M*(sum(All.Theta) - sum(All.Theta(:,1)) ...
    - (All.t - All.t(1))*sum(Param.Omega));

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function MovieEvolution(All,Param)

close all

```

```

vidObj = VideoWriter('Kuramoto_Movie.avi');
if Param.OrderKuramoto == 1
    vidObj = VideoWriter('Kuramoto1_Movie.avi');
end
if Param.OrderKuramoto == 2
    vidObj = VideoWriter('Kuramoto2_Movie.avi');
end
if Param.OrderKuramoto == 3
    vidObj = VideoWriter('Kuramoto3_Movie.avi');
end
open(vidObj);

for n = 1:length(All.t)
    if n == 1 | mod(n,Param.MovieN0) == 0
        t = All.t(n)
        PlotState(n,All,Param);
        currFrame = getframe(gcf);
        writeVideo(vidObj,currFrame);
    end
end

n = length(All.t);
for loop = 1:4
    PlotState(n,All,Param);
    currFrame = getframe(gcf);
    writeVideo(vidObj,currFrame);
end

close(vidObj);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Param = PlotState(nPlusOne,All,Param)

close all

n = nPlusOne;
set(gca, 'Color',Param.ColorBackground);
set(gcf, 'Color',Param.ColorBackground);
set(gcf, 'InvertHardCopy', 'off');
Radius = 1;
Param.Radius = Radius*ones(Param.M,1);
phi = [0:0.01:2*pi]';
plot(Radius*cos(phi),Radius*sin(phi),'k')
axis square
axis off
hold on
Param.Points = zeros(Param.M,1);
for m = 1:Param.M
    s = (Param.M-m)/Param.M;
    Color = sqrt(s)*Param.ColorWhite + (1-sqrt(s))*Param.ColorBlack;
    Param.Points(m) ...
        = plot(...

```

```

        Param.Radius(m)*cos(All.Theta(m,n)), ...
        Param.Radius(m)*sin(All.Theta(m,n)), ...
        'Marker','o','MarkerSize',Param.MarkerSize,...
        'Color',Color);
    hold on
    axis square
end

if nPlusOne == 1
    Title1a = ['System of order ',num2str(Param.OrderKuramoto), ...
    ' at time t = ',num2str(All.t(n))];
    Title1b = ['M = ',num2str(Param.M),' oscillators'];
    Title2 = ['Point (cos / sin of single phase)'];
    title({Title1a;Title1b;[]})
end
if nPlusOne > 1
    Title2a = ['System of order ',num2str(Param.OrderKuramoto), ...
    ' at time t = ',num2str(All.t(n))];
    Title2b = ['M = ',num2str(Param.M),' oscillators'];
    CT = All.ComputationTime;
    if Param.SolutionWithPrecomputationYes == 0
        Title1 = ['Standard approach (CT = ',num2str(CT,2),')'];
    end
    if Param.SolutionWithPrecomputationYes == 1
        Title1 = ['Novel approach (CT = ',num2str(CT,2),')'];
    end
    title({Title1;Title2a;Title2b;[]})
end
set(gca,'FontSize',Param.FontSize);
if nPlusOne == 1 | nPlusOne == Param.N+1
    SavePlot(n,Param);
end

drawnow
pause

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function SavePlot(n,Param)

if n == 1
    if Param.MyCase == -1
        if Param.M == 50
            saveas(gcf,'Kuramoto1_IC_WithoutPrecomputation_M50.jpg');
        end
        if Param.M == 100
            saveas(gcf,'Kuramoto1_IC_WithoutPrecomputation_M100.jpg');
        end
        if Param.M == 1000
            saveas(gcf,'Kuramoto1_IC_WithoutPrecomputation_M1000.jpg');
        end
    end
end
if Param.MyCase == -2

```



```

    if Param.M == 50
        saveas(gcf, 'Kuramoto2_IC_WithoutPrecomputation_M50.jpg');
    end
    if Param.M == 100
        saveas(gcf, 'Kuramoto2_IC_WithoutPrecomputation_M100.jpg');
    end
    if Param.M == 1000
        saveas(gcf, 'Kuramoto2_IC_WithoutPrecomputation_M1000.jpg');
    end
end
if Param.MyCase == -3
    if Param.M == 50
        saveas(gcf, 'Kuramoto3_IC_WithoutPrecomputation_M50.jpg');
    end
    if Param.M == 100
        saveas(gcf, 'Kuramoto3_IC_WithoutPrecomputation_M100.jpg');
    end
    if Param.M == 1000
        saveas(gcf, 'Kuramoto3_IC_WithoutPrecomputation_M1000.jpg');
    end
end
end

if Param.MyCase == 1
    if Param.M == 50
        saveas(gcf, 'Kuramoto1_IC_WithPrecomputation_M50.jpg');
    end
    if Param.M == 100
        saveas(gcf, 'Kuramoto1_IC_WithPrecomputation_M100.jpg');
    end
    if Param.M == 1000
        saveas(gcf, 'Kuramoto1_IC_WithPrecomputation_M1000.jpg');
    end
end
end

if Param.MyCase == 2
    if Param.M == 50
        saveas(gcf, 'Kuramoto2_IC_WithPrecomputation_M50.jpg');
    end
    if Param.M == 100
        saveas(gcf, 'Kuramoto2_IC_WithPrecomputation_M100.jpg');
    end
    if Param.M == 1000
        saveas(gcf, 'Kuramoto2_IC_WithPrecomputation_M1000.jpg');
    end
end
end

if Param.MyCase == 3
    if Param.M == 50
        saveas(gcf, 'Kuramoto3_IC_WithPrecomputation_M50.jpg');
    end
    if Param.M == 100
        saveas(gcf, 'Kuramoto3_IC_WithPrecomputation_M100.jpg');
    end
    if Param.M == 1000
        saveas(gcf, 'Kuramoto3_IC_WithPrecomputation_M1000.jpg');
    end
end
end
end

```

end

```
if n == Param.N+1
  if Param.MyCase == -1
    if Param.M == 50
      saveas(gcf, 'Kuramoto1_WithoutPrecomputation_M50.jpg');
    end
    if Param.M == 100
      saveas(gcf, 'Kuramoto1_WithoutPrecomputation_M100.jpg');
    end
    if Param.M == 1000
      saveas(gcf, 'Kuramoto1_WithoutPrecomputation_M1000.jpg');
    end
  end
  if Param.MyCase == -2
    if Param.M == 50
      saveas(gcf, 'Kuramoto2_WithoutPrecomputation_M50.jpg');
    end
    if Param.M == 100
      saveas(gcf, 'Kuramoto2_WithoutPrecomputation_M100.jpg');
    end
    if Param.M == 1000
      saveas(gcf, 'Kuramoto2_WithoutPrecomputation_M1000.jpg');
    end
  end
  if Param.MyCase == -3
    if Param.M == 50
      saveas(gcf, 'Kuramoto3_WithoutPrecomputation_M50.jpg');
    end
    if Param.M == 100
      saveas(gcf, 'Kuramoto3_WithoutPrecomputation_M100.jpg');
    end
    if Param.M == 1000
      saveas(gcf, 'Kuramoto3_WithoutPrecomputation_M1000.jpg');
    end
  end
end

if Param.MyCase == 1
  if Param.M == 50
    saveas(gcf, 'Kuramoto1_WithPrecomputation_M50.jpg');
  end
  if Param.M == 100
    saveas(gcf, 'Kuramoto1_WithPrecomputation_M100.jpg');
  end
  if Param.M == 1000
    saveas(gcf, 'Kuramoto1_WithPrecomputation_M1000.jpg');
  end
end
if Param.MyCase == 2
  if Param.M == 50
    saveas(gcf, 'Kuramoto2_WithPrecomputation_M50.jpg');
  end
  if Param.M == 100
    saveas(gcf, 'Kuramoto2_WithPrecomputation_M100.jpg');
  end
end
```

