Community integration algorithms (CIAs): A novel computational approach for large-scale dynamical systems and extensions to networks

Tobias Böhle[*], Christian Kuehn[†], Mechthild Thalhammer[‡]

February 16, 2022

The algorithm applied for the evaluation of the defining function constitutes a crucial factor in the numerical simulation of a large-scale dynamical system. This fundamental tool is essential for the significance of the obtained results and the overall performance. A novel modus operandi based on well-adapted global approximations and precomputations of reappearing quantities is proposed. It is demonstrated that this approach is expedient for a wide range of dynamical systems and permits substantial reductions from polynomial to linear computational complexity. Combining such concepts with state-of-the-art algorithms for the detection of communities in networks leads to community integration algorithms (CIAs) for dynamical systems on graphs. Numerical comparisons for relevant test problems with applications in various disciplines including higher-order Kuramoto–Daido-type systems arising in the description of synchronisation, Cucker–Smale systems exhibiting flocking behaviour, and extended systems on real world graphs for animal networks, confirm the robustness and efficiency of the novel computational approach.

Fascinating phenomena. The collective behaviour of groups of interacting individuals has inspired generations of scientists from various disciplines and encouraged them for thorough investigations. Most of us will remember impressive visualisations of hundreds of migratory birds gathering in one place and setting out for their new destination. Likewise, in our everyday life, we have the opportunity to observe the flexible movement of flocks of domestic birds around blocks of buildings.

Numerical simulations. It suggests itself to strive towards a deeper understanding of these kinds of fascinating phenomena and to use numerical simulations as valuable means for the study of the decisive underlying mechanisms. The design of appropriate models and efficient implementation methods are thus fundamental issues in applied as well as computational mathematics and have fruitful impacts on findings in other fields.

As established models for flocking behaviour and synchronisation, we highlight Cucker–Smale and Kuramoto–Daido-type systems, whose applications range from biology, chemistry, ecology, economy, neuroscience, and physics to sociology.

Over the last years, an essential part of research interest has focused on large-scale dynamical systems and the incorporation of network theory. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]

Computational issues. A characteristics of complex systems describing the dynamics of interacting components is that the evolution of a single individual is affected by the evolution of other individuals. Typically, when pairwise interaction mechanisms are taken into account, the equation for each component of the dynamical system involves a sum over all other components, which results in quadratic computational complexity with respect to the total number of components. More generally, multiple sums arising in higher-order dynamical systems imply polynomial complexity.

These matters of fact lead to exceeding computation times for single evaluations of the defining functions and hence considerably limit the total number of components that can be treated in practice. Further obstacles in connection with large-scale applications on networks are that convenient representations based on matrices and tensors instantly rise above usual memory capacities.

[*] Technical University of Munich, School of Computation Information and Technology, Department of Mathematics, Boltzmannstraße 3, 85748 Garching, Germany.

[†] Technical University of Munich, School of Computation Information and Technology, Department of Mathematics, Boltzmannstraße 3, 85748 Garching, Germany. Complexity Science Hub Vienna, Josefstädterstraße 39, 1080 Vienna, Austria.

[‡] University of Innsbruck, Department of Mathematics, Technikerstraße 13/7, 6020 Innsbruck, Austria.

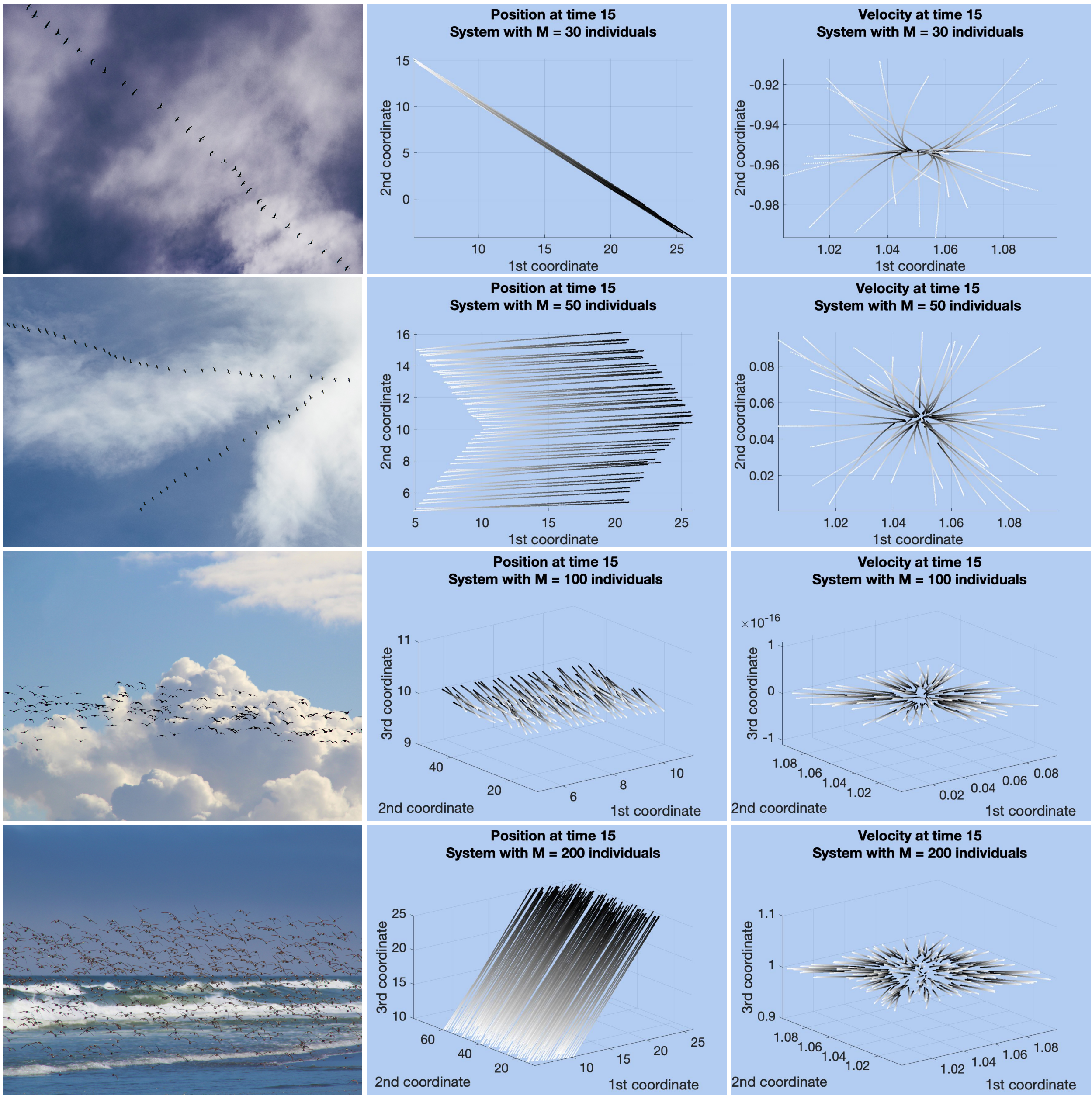

Figure 1: First column: Visualisations of flocks of birds (Kranich17 and Minka2507 at pixabay.com). Second and third columns: Numerical simulations of Cucker–Smale systems modelling the collective behaviour of groups of individuals with pairwise interactions. The numbers of individuals and the positions 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.

Results

Novel algorithms. In the present work, we propose community integration algorithms (CIAs) as fundamental tools in the numerical simulation of large-scale dynamical systems on networks. The principal purpose is to ensure reliable and efficient evaluations of the defining functions. We primarily attach importance to broad applicability and robustness with regard to real data sets.[18]

Fundamental ingredients are global approximations to attain well-adapted representations of the underlying classical systems that reveal precomputations of reappearing quantities. Regarding well-established theoretical results and eminent observations in benchmark tests for classical

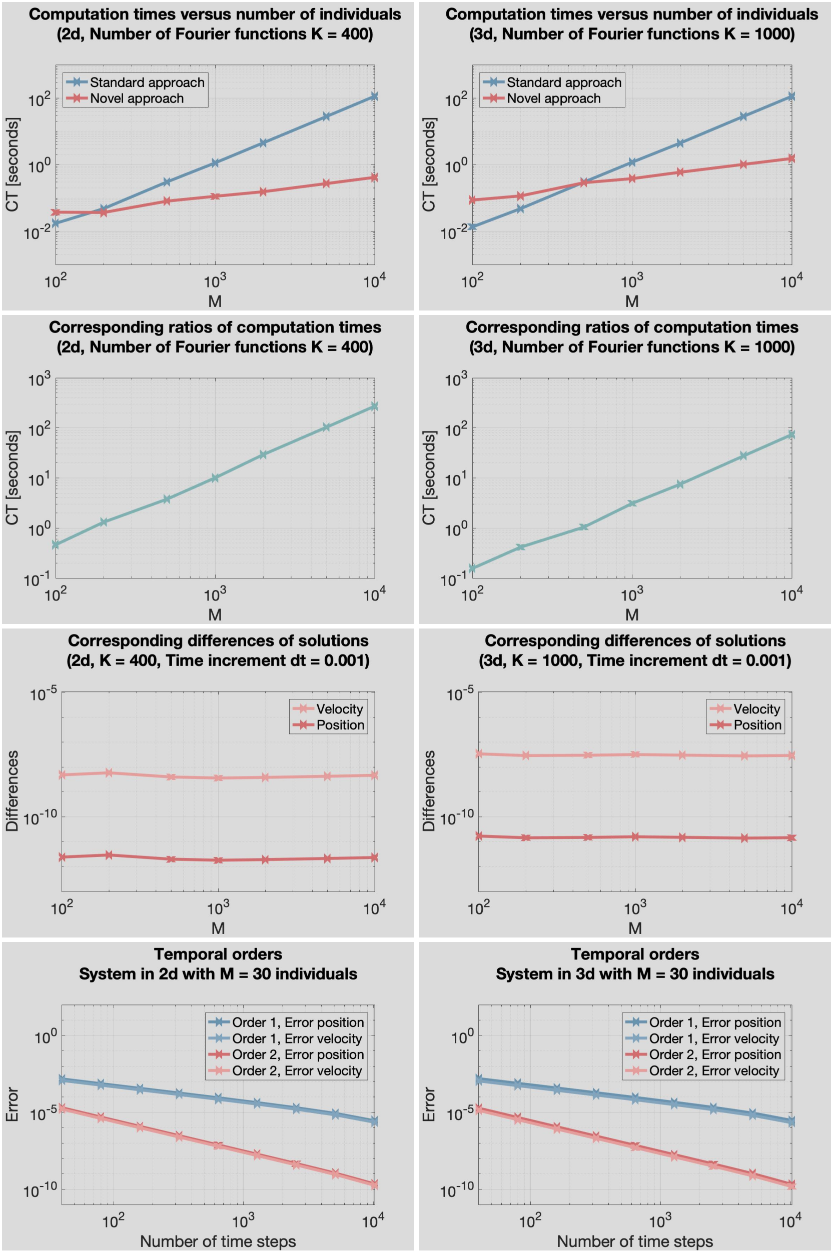

Figure 2: Numerical comparisons for Cucker–Smale systems in two and three space dimensions. First row: The standard approach for the evaluation of the decisive sums results in quadratic computational complexity, which is reflected in the computation times. A novel approach based on a global approximation of a part of the defining function by a Fourier series expansion reveals the precomputation of reappearing quantities and permits the significant reduction to linear computational complexity. Second row: For systems that involve a number of individuals M that is beyond the total number of Fourier basis functions, $K = 20 \times 20$ in two space dimensions and $K = 10 \times 10 \times 10$ in three space dimensions, respectively, the effective gains become evident when calculating the ratios of the computation times. Third and fourth rows: Complementary tests, which confirm that high accuracy is retained during numerical integration. The differences between the solutions values, computed on the one hand by the standard approach and on the other hand by the novel approach, are not effected by the sizes of the systems. For explicit Runge–Kutta methods of orders one and two, respectively, the global errors at final time $T = 1$ versus the total numbers of time steps are displayed. In a logarithmic scaling, the slopes of the lines correctly reproduce the consistency orders.

Cucker–Smale systems, we favour approximations based on Fourier series expansions. For classical higher-order

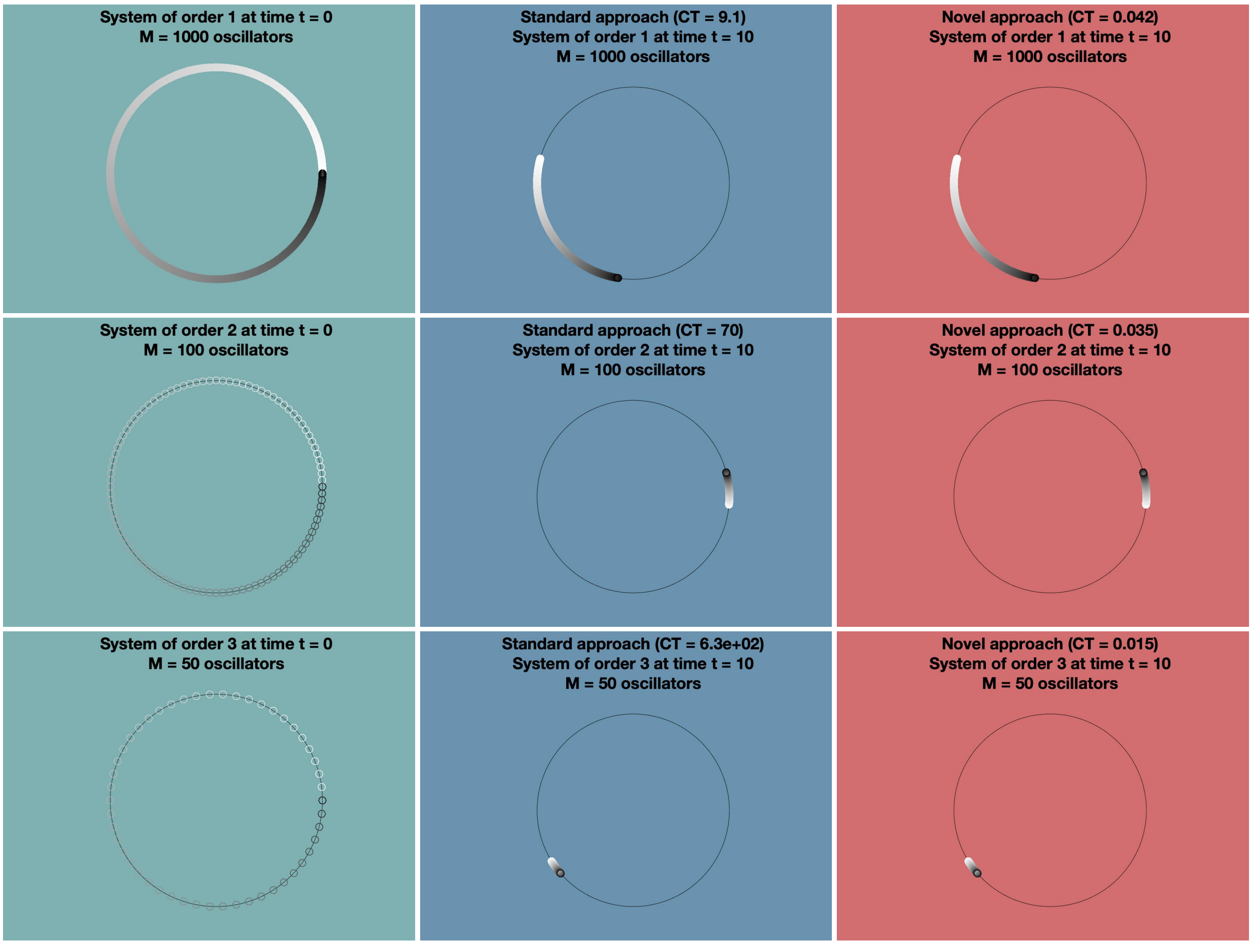

Figure 3: Numerical simulations of higher-order Kuramoto–Daido-type systems modelling (partial) synchronisation of the phases of coupled oscillators. First column: For systems of orders one, two, and three involving 1000, 100, and 50 components, respectively, uniformly distributed initial states are considered. For each phase $x \in [0, 2\pi)$, a point on the unit circle represents the associated pair $(\cos(x), \sin(x))$. Second column: Standard approaches for the evaluation of the arising multiple sums result in polynomial computational complexity. This is reflected in the strongly increasing computation times from about 10 seconds to 11 minutes. Third column: Novel approaches based on equivalent representations yield coinciding solutions and permit striking reductions of the computation times to fractions of a second.

Kuramoto–Daido-type systems, we even succeed in deducing equivalent reformulations, which make reductions from polynomial to linear computational complexity with respect to the dimensions of the systems evident. [9, 11] (Figures 1–3, Table 1)

For extended systems on networks, we combine these concepts with state-of-the-art algorithms for the detection of communities in the underlying graphs. In case major parts of interactions between components of the considered systems occur within certain communities, the associated adjacency matrices comprise relatively dense submatrices along the diagonal and sparse submatrices otherwise. In such situations, our novel computational approach effects significant reductions of the overall costs measured by the total numbers of function evaluations and the required memory capacities.[19, 20, 21, 22, 23, 24, 25, 26, 27] (Figures 4–5)

Perspectives. Altogether, this makes numerical simulations of discrete as well as continuous large-scale dynamical systems on graphs by advanced integration methods practicable and thus enables realisations of numerically challenging issues in the context of long-term integration and optimisation.

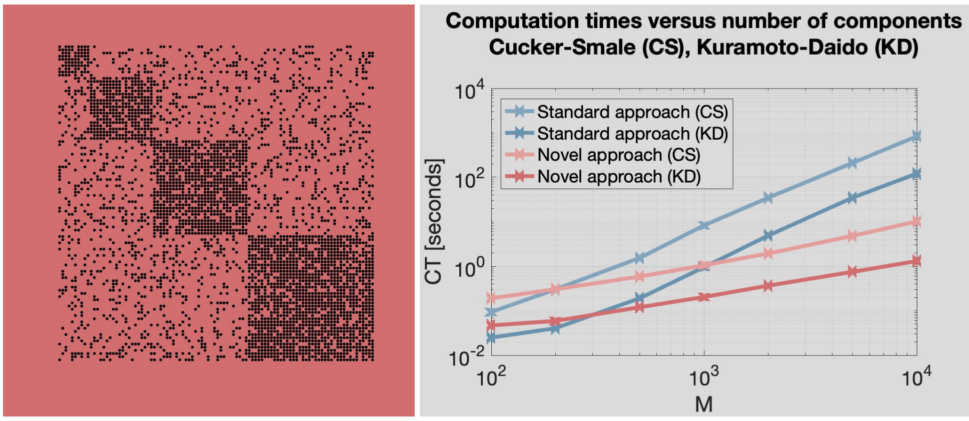

Figure 4: Numerical comparisons for Cucker–Smale and Kuramoto-Daido systems on networks. Left: The considered adjacency matrices comprise relatively dense submatrices along the diagonal, which reflect numerous pairwise interactions within four communities of different sizes. Right: Standard approaches for the evaluation of the decisive sums result in quadratic computational complexity, which is reflected in the computation times. Novel approaches that observe the community structure and use well-adapted global approximations to identify reappearing quantities permit reductions to linear complexity.

Computational methods

Unifying formulation of dynamical systems on networks. In sight of evident analogies between time-continuous and time-discrete dynamical systems and due to the fact that the numerical solution of boundary and optimisation problems often relies on the numerical integration of a series of initial value problems, we identify the initial value problem for a system of coupled nonlinear autonomous differential equations

$$\begin{cases} x'(t) = H\big(x(t)\big), & t \in (0,T), \\ x(0) \text{ given}, \end{cases} \tag{1a}$$

as fundamental test problem. Here, we represent by

$$\begin{aligned} &x = (x_1, \dots, x_M) : [0,T] \to \mathscr{X}^M, \\ &H = (H_1, \dots, H_M) : \mathscr{X}^M \to \mathscr{Y}^M, \end{aligned} \tag{1b}$$

the unknown time-dependent solution and the defining function. A wide range of relevant dynamical systems on networks that take into account pairwise interactions of all components can be cast into the unifying formulation

$$\begin{gathered} H_m(x_1, \dots, x_M) = f_m(x_m) + \sum_{\ell=1}^{M} A_{m\ell}\, g(x_m, x_\ell), \\ m \in \{1, \dots, M\}, \end{gathered} \tag{1c}$$

with symmetric adjacency matrix capturing the structure of the underlying graph

$$\begin{gathered} A = \big(A_{m\ell}\big)_{m,\ell=1}^{M}, \quad A_{m\ell} \in \{0,1\}, \quad A_{\ell m} = A_{m\ell}, \\ \ell, m \in \{1, \dots, M\}. \end{gathered} \tag{1d}$$

The incorporation of more general interactions naturally results in multiple sums

$$\begin{gathered} \sum_{\ell_1, \dots, \ell_L = 1}^{M} A^{(L)}_{m\ell_1 \dots \ell_L}\, g^{(L)}(x_m, x_{\ell_1}, \dots, x_{\ell_L}), \\ A^{(L)}_{m\ell_1 \dots \ell_L} \in \{0,1\}, \quad \ell_1, \dots, \ell_L, m \in \{1, \dots, M\}. \end{gathered} \tag{2}$$

Accordingly, we henceforth refer to (1) as a *first-order dynamical system* and to a generalisation comprising a term of the form (2) as a *dynamical system of order L*.

Curse of dimensionality. For a large-scale dynamical system on a network, a crucial factor in the successful numerical integration of the initial value problem (1) is the algorithm that is applied for the evaluation of the defining function. This basic ingredient in the end determines reliability and efficiency, the significance of the obtained results and the attainable performance.

Indeed, whenever the total number of components is high, i.e. $M \gg 1$, and the adjacency matrix comprises relatively few zero entries, the issues in the computation of the decisive sums

$$\left(\sum_{\ell=1}^{M} A_{m\ell}\, g(x_m, x_\ell) \right)_{m=1}^{M}$$

become apparent. On the one hand, quadratic computational complexity with respect to the number of components rapidly converts to impractical computation times. On the other hand, for the purpose of simultaneous function evaluations, it is convenient to store the values

$$(x_m, x_\ell), \quad \ell, m \in \{1, \dots, M\}.$$

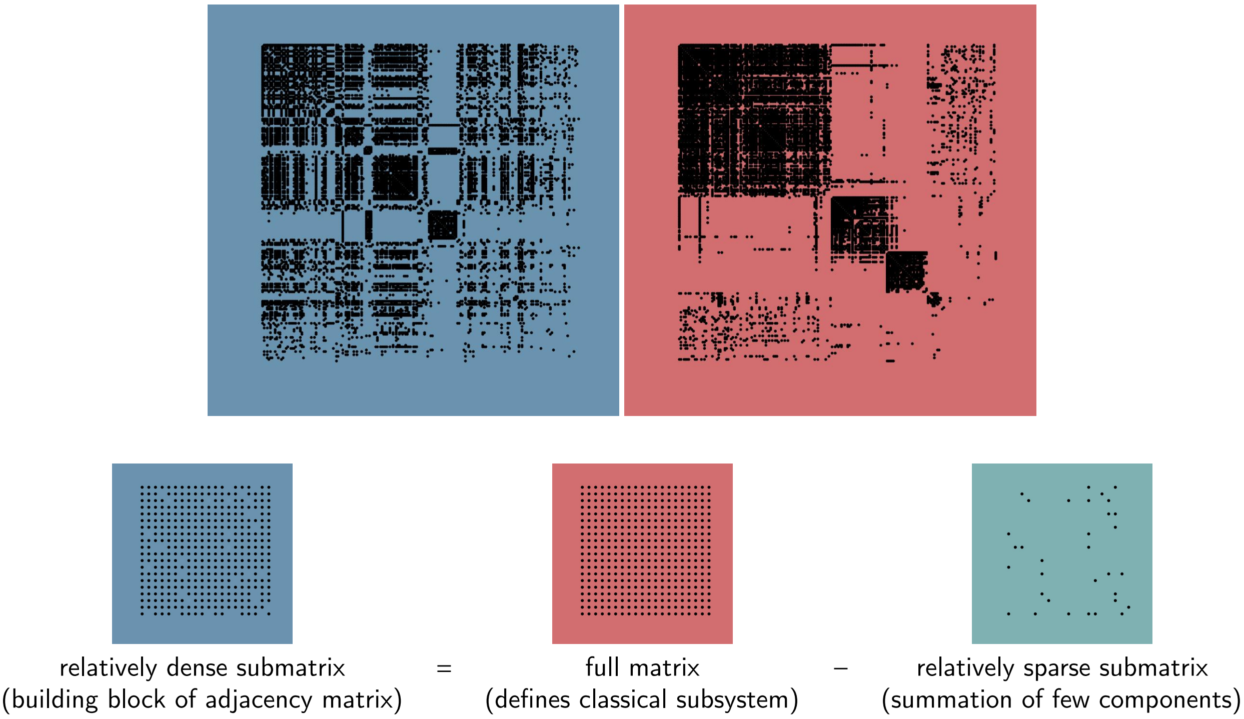
relatively dense submatrix
(building block of adjacency matrix)
=
full matrix
(defines classical subsystem)
−
relatively sparse submatrix
(summation of few components)

Figure 5: First row: The graph of a real data network (https://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. Second row: 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.

However, regarding the memory capacity of a common data processor, this procedure severely limits the treatable system size. For multiple sums

$$\left(\sum_{\ell_1,\dots,\ell_L=1}^{M} A^{(L)}_{m\ell_1\dots\ell_L}\, g^{(L)}(x_m, x_{\ell_1}, \dots, x_{\ell_L}) \right)_{m=1}^{M}$$

implying polynomial computational complexity, the numerical challenges become even more striking and relate to the *curse of dimensionality*.

Classical dynamical systems. For both reasons, as important issues on their own and as first steps towards the numerical simulation of dynamical systems on networks, it is of major interest to study inherent classical dynamical systems. Formally, they are retained from (1)-(2) for the particular choices of complete graphs

$$A^{(L)}_{m\ell_1\dots\ell_L} = 1, \quad \ell_1,\dots,\ell_L, m \in \{1,\dots,M\},$$

and hence the decisive sums read as

$$\left(\sum_{\ell_1,\dots,\ell_L=1}^{M} g^{(L)}(x_m, x_{\ell_1}, \dots, x_{\ell_L}) \right)_{m=1}^{M}.$$

Guideline. In the following, we specify the main items of community integration algorithms (CIAs) for the simultaneous computation of the decisive sums arising in first-order dynamical systems on networks. Furthermore, we comment on extensions to higher-order systems.

Two pre-simulation steps (P1) and (P2) are carried out in advance. From them, two evaluation steps (E1) and (E2), applied several times in the course of the numerical integration, infer naturally.

(P1) Identification of a well-adapted global approximation of a part of the function defining the underlying classical system.

(P2) Application of an effective community detection algorithm and transformation of the adjacency matrix to block form.

(E1) Treatment of classical subsystems associated with complete subgraphs by precomputations of reappearing quantities.

(E2) Treatment of remaining contributions associated with relatively sparse submatrices by straightforward summations.

Pre-simulation step (P1). For the decisive sums defining the underlying classical dynamical system, we calculate a global approximation, e.g. by polynomial interpolation, Taylor series expansion, or Fourier series expansion, respectively, such that

$$\left(\sum_{\ell=1}^{M} g(x_m, x_\ell)\right)_{m=1}^{M} \approx \sum_{j\in\mathscr{J}} \psi_j(x) \left(\Psi_j(x_m)\right)_{m=1}^{M}.$$

Computational complexity. We point out that the selected structure of the employed global approximation permits the precomputation of the reappearing quantities $(\psi_j(x))_{j\in\mathscr{J}}$. As a consequence, quadratic complexity $\mathscr{O}(M^2)$ with respect to the total number of components is reduced to linear complexity $\mathscr{O}(M)$, but with the addition that the number of elements in the index set $\mathscr{J}$ has an influence on the actual cost. We indicate this with the symbolic notation $\mathscr{O}(|\mathscr{J}|M)$. For a particular dynamical system, it is advisable to balance the computational effort and the requested accuracy, both related to the choice of $\mathscr{J}$.

Pre-simulation step (P2). Numerous algorithms facilitate the detection of communities in networks.[1] Accordingly, we transform the adjacency matrix associated with the underlying graph by means of a permutation matrix to block structure

$$PAP^T = \begin{pmatrix} A_1 & & A_1^{(+)} \\ A_2^{(-)} & A_2 & A_2^{(+)} \\ \vdots & \vdots & \vdots \\ A_{I-1}^{(-)} & A_{I-1} & A_{I-1}^{(+)} \\ & A_I^{(-)} & A_I \end{pmatrix} \in \mathbb{R}^{M\times M}.$$

For concreteness and in view of visualisations given in Figure 5, we suppose that the square matrices along the diagonal

$$A_i \in \mathbb{R}^{d_i\times d_i}, \quad i\in\{1,\dots,I\}, \quad \sum_{i=1}^{I} d_i = M,$$

are relatively dense and that the remaining matrices to the right-hand and left-hand sides of the diagonal blocks

$$A_1^{(+)},\dots,A_{I-1}^{(+)}, \quad A_2^{(-)},\dots,A_I^{(-)},$$

are relatively sparse. The components of the problem data and the solution to the initial value problem (1) are reordered correspondingly.

For the sake of simplicity, we employ the same notation for the original and the transformed system.

Evaluation steps. Based on the two pre-simulation steps, we trace the evaluation of the full system back to the evaluation of subsystems and thereby distinguish two complementary cases.

For the purpose of exposition, we consider the first block of the system, which comprises a relatively dense square matrix and a relatively sparse matrix

$$S^{(\text{block})} = \left(\sum_{\ell=1}^{M} A_{m\ell}^{(\text{block})}\, g(x_m, x_\ell)\right)_{m=1}^{d_1},$$
$$A^{(\text{block})} = \left(A^{(\text{dense})}\ A^{(\text{sparse})}\right) \in \mathbb{R}^{d_1\times M}.$$

The remaining blocks of the form

$$\left(A^{(\text{sparse})}\ A^{(\text{dense})}\ A^{(\text{sparse})}\right) \in \mathbb{R}^{d_i\times M}, \quad i\in\{2,\dots,I-1\},$$
$$\left(A^{(\text{sparse})}\ A^{(\text{dense})}\right) \in \mathbb{R}^{d_I\times M},$$

are treated in an analogous manner. Our basic concept is to optimise the required number of functions evaluations and the memory capacity. That is, in order to accelerate the computation of sums and to avoid the storage of submatrices, we use the decomposition

$$S^{(\text{block})} = S^{(\text{dense})} + S^{(\text{sparse})},$$
$$S^{(\text{dense})} = \left(\sum_{\substack{\ell=1\\ A_{m\ell}^{(\text{dense})}=1}}^{d_1} g(x_m, x_\ell)\right)_{m=1}^{d_1},$$
$$S^{(\text{sparse})} = \left(\sum_{\substack{\ell=d_1+1\\ A_{m\ell}^{(\text{sparse})}=1}}^{d_1} g(x_m, x_\ell)\right)_{m=1}^{d_1}.$$

Due to the fact that for relatively dense submatrices the numbers of non-zero coefficients are relatively high, we store the complementary contributions, i.e., all indices corresponding to coefficients equal to zero. In contrast, regarding that for relatively sparse matrices the numbers of coefficients equal to zero are relatively high, we store all indices that correspond to non-zero coefficients.

Our key idea for the subsystem that is defined by the relatively dense submatrix is visualised in Figure 5. The global approximation provided for the underlying classical system applies to any reduced number of components. As a consequence, we obtain the relation

$$S^{(\text{dense})} \approx \sum_{j\in\mathscr{J}} \psi_j(x_1,\dots,x_{d_1}) \left(\Psi_j(x_m)\right)_{m=1}^{d_1}$$
$$- \left(\sum_{\substack{\ell=1\\ A_{m\ell}^{(\text{dense})}=0}}^{d_1} g(x_m, x_\ell)\right)_{m=1}^{d_1}.$$

[1] Amongst a variety of choices, we mention the community detection algorithm RBER_POTS, available through the PYTHON package CDLIB.[26]

Evaluation step (E1). The reasoning for classical dynamical systems conforms to the arising classical subsystems. Again, for sufficiently large communities, it is to the best advantage to compute reappearing quantities such as $(\psi_j(x_1,\dots,x_{d_1}))_{j\in\mathscr{J}}$ in advance.

Besides, it is worth mentioning that for dynamical systems of Kuramoto–Daido-type, e.g., these quantities also bear further meaning as localisations of order parameters.

Evaluation step (E2). For compensating and remaining contributions such as

$$\Bigg(-\sum_{\substack{\ell=1\\ A_{m\ell}^{(\mathsf{dense})}=0}}^{d_1} g(x_m,x_\ell)+\sum_{\substack{\ell=d_1+1\\ A_{m\ell}^{(\mathsf{sparse})}=1}}^{d_1} g(x_m,x_\ell)\Bigg)_{m=1}^{d_1},$$

we use straightforward summation.

Numerical results

In order to substantiate our basic concepts for the key item (P1), we next include detailed calculations for classical Cucker–Smale and Kuramoto–Daido-type systems, where the defining functions have the form

$$g(x_m,x_\ell)=\tfrac{C}{M}\,h(x_\ell-x_m)\,,\quad \ell,m\in\{1,\dots,M\}\,,\qquad (3)$$

with constant $C>0$.

Cucker–Smale models. Classical Cucker–Smale systems describe collective motions of groups of individuals with flocking behaviour. (Figure 1)

In three space dimensions, the positions (ξ_1,ξ_2,ξ_3) and the associated velocities (ξ_4,ξ_5,ξ_6) are captured by the relations

$$\begin{gathered}\mathscr{X}=\mathbb{R}^6\,,\quad \mathscr{Y}=\mathbb{R}^6\,,\quad \xi=(\xi_1,\dots,\xi_6)\in\mathscr{X}\,,\\ f_m(\xi)=(\xi_4,\xi_5,\xi_6,0,0,0)^T\,,\quad m\in\{1,\dots,M\}\,,\\ \alpha,\beta>0\,,\quad \varphi(\xi_1,\xi_2,\xi_3)=\big(\alpha^2+\xi_1^2+\xi_2^2+\xi_3^2\big)^{-\beta}\,,\\ h(\xi)=\varphi(\xi_1,\xi_2,\xi_3)\,(0,0,0,\xi_4,\xi_5,\xi_6)^T\,.\end{gathered}$$

Regarding (1) and (3), summation applies to

$$\begin{aligned}&G_i(x)=G_i(x_{11},\dots,x_{16},\dots,x_{M1},\dots,x_{M6})\\ &=\Bigg(\sum_{\ell=1}^{M}\varphi(x_{m1}-x_{\ell1},x_{m2}-x_{\ell2},x_{m3}-x_{\ell3})\,(x_{\ell i}-x_{mi})\Bigg)_{m=1}^{M}\end{aligned}$$

for each index $i\in\{4,5,6\}$. From benchmark tests, we favour Fourier series expansions of the power function

$$\begin{gathered}\varphi(\xi_1,\xi_2,\xi_3)\approx\sum_{k\in\mathscr{K}}\varphi_k^{(s)}\,F_k(\xi_1,\xi_2,\xi_3)\,,\quad (\xi_1,\xi_2,\xi_3)\in D\,,\\ \varphi_k^{(s)}\approx\int_D\varphi(\xi_1,\xi_2,\xi_3)\,F_{-k}(\xi_1,\xi_2,\xi_3)\,\mathrm{d}(\xi_1,\xi_2,\xi_3)\,,\quad k\in\mathscr{K}\,.\end{gathered}$$

For suitably chosen cartesian products of intervals and sufficiently large numbers of basis functions

$$\begin{gathered}F_k(\xi_1,\xi_2,\xi_3)=e_k\,E_k(\xi_1,\xi_2,\xi_3)\,,\\ e_k=\prod_{i=1}^{3}\big(\sqrt{2\,a_i}\,\mathrm{e}^{\mathrm{i}\pi\,k_i}\big)^{-1}\,,\quad E_k(\xi_1,\xi_2,\xi_3)=\prod_{i=1}^{3}\mathrm{e}^{\mathrm{i}\pi\,k_i\,\xi_i/a_i}\,,\\ k=(k_1,k_2,k_3)\in\mathscr{K}=\mathscr{K}_1\times\mathscr{K}_3\times\mathscr{K}_5\,,\\ (\xi_1,\xi_2,\xi_3)\in D=[-a_1,a_1]\times[-a_2,a_2]\times[-a_3,a_3]\,,\end{gathered}$$

highly accurate approximations result. Straightforward calculations yield the representation

$$\begin{gathered}G_i(x)\approx\sum_{\kappa=1}^{2}\sum_{k\in\mathscr{K}}\widetilde{\psi}_{ki}^{(\kappa)}(x)\Big(\widetilde{\Psi}_{ki}^{(\kappa)}(x_m)\Big)_{m=1}^{M}\,,\\ \widetilde{\psi}_{ki}^{(1)}(x)=\sum_{\ell=1}^{M}E_{-k}(x_{\ell1},x_{\ell2},x_{\ell3})\,x_{\ell i}\,,\\ \widetilde{\Psi}_{ki}^{(1)}(x_m)=\varphi_k^{(s)}\,F_k(x_{m1},x_{m2},x_{m3})\,,\\ \widetilde{\psi}_{ki}^{(2)}(x)=\sum_{\ell=1}^{M}E_{-k}(x_{\ell1},x_{\ell2},x_{\ell3})\,,\\ \widetilde{\Psi}_{ki}^{(2)}(x_m)=-\,\varphi_k^{(s)}\,F_k(x_{m1},x_{m2},x_{m3})\,x_{mi}\,,\end{gathered}$$

which can be cast into the desired form by systematically numbering all elements of the index set $\{1,2\}\times\mathscr{K}$.

Simplified relations follow for flocking motions in two dimensions.

Higher-order Kuramoto–Daido-type models. Classical Kuramoto–Daido systems describe the phases of sets of coupled oscillators and reveal synchronisation phenomena. (Figure 3)

The initial first-order formulation accordingly to (1) and (3) relies on the relations

$$\begin{gathered}\mathscr{X}=\mathbb{R}/(2\pi\mathbb{Z})\,,\quad \mathscr{Y}=\mathbb{R}\,,\quad \xi\in\mathscr{X}\,,\\ f_m(\xi)=\omega_m\in\mathbb{R}\,,\quad m\in\{1,\dots,M\}\,,\quad h(\xi)=\sin(\xi)\,.\end{gathered}$$

An application of the elementary addition theorem for the sine function yields the equivalent reformulation

$$\begin{gathered}\Bigg(\sum_{\ell=1}^{M}\sin(x_\ell-x_m)\Bigg)_{m=1}^{M}=\sum_{j=1}^{2}\psi_j(x)\,\big(\Psi_j(x_m)\big)_{m=1}^{M}\,,\\ \psi_1(x)=\sum_{\ell=1}^{M}\sin(x_\ell)\,,\quad \Psi_1(x_m)=\cos(x_m)\,,\\ \psi_2(x)=-\sum_{\ell=1}^{M}\cos(x_\ell)\,,\quad \Psi_2(x_m)=\sin(x_m)\,,\end{gathered}$$

which obviously permits the reduction from quadratic to linear computational complexity.

At first glance, it is unexpected that this procedure extends to higher-order systems involving terms of the form

$$\begin{gathered}\Bigg(\sum_{\ell_1,\dots,\ell_L=1}^{M}\sin(\sigma_1\,x_{\ell_1}+\dots+\sigma_L\,x_{\ell_L}-x_m)\Bigg)_{m=1}^{M}\,,\\ \sigma_1,\dots,\sigma_L\in\{-1,1\}\,,\end{gathered}$$

	First-order Cucker–Smale systems	Kuramoto–Daido-type systems of order L
Novel approach	$\mathscr{O}(\lvert\mathscr{J}\rvert M)$	$\mathscr{O}(M)$
Standard approach	$\mathscr{O}(M^2)$	$\mathscr{O}(M^{L+1})$

Table 1: Computational complexity for the evaluation of the decisive sums arising in classical first-order Cucker–Smale and higher-order Kuramoto–Daido-type systems. Standard approaches result in polynomial complexity. Well-adapted global approximations or reformulations, respectively, yield representations that reveal precomputations of reappearing quantities and permit reductions to linear complexity.

but straightforward calculations confirm the striking reductions from polynomial to linear computational complexity likewise in such cases.

Computational complexity and numerical evidence. Numerical results for first-order Cucker–Smale and higher-order Kuramoto–Daido-type systems are given in Figures 1-4.

When providing simulations for large-scale dynamical systems, it is reasonable to assume that the impact of computational side effects is diminished and computation times become significant quantities. In the provided numerical comparisons of standard and novel approaches, we thus may consider the ratios of computation times as reliable indicators for the effective gains.

The numerical results presented in Figures 2 and 3 are in accordance with the general statements of Table 1. As confirmed by Figure 4, the conclusions also hold for systems on networks, where numerous pairwise interactions take place within communities of larger sizes.

Code and data availability. MATLAB scripts to reproduce numerical results as well as additional simulations for Cucker–Smale systems (Figure 1)

Movie_CuckerSmale_Flock1.m4v
Movie_CuckerSmale_Flock2.m4v
Movie_CuckerSmale_Flock3.m4v
Movie_CuckerSmale_Flock4.m4v

and for higher-order Kuramoto–Daido-type systems (Figure 3)

Movie_KuramotoDaido_Order1.m4v
Movie_KuramotoDaido_Order2.m4v
Movie_KuramotoDaido_Order3.m4v

are found at the provided link.[2] A more comprehensive code for the simulation of various dynamical systems on networks will be accessible in the near future. Visualisations of flocks of birds and real data networks are freely available.[3]

Discussion

In this work, we have introduced community integration algorithms (CIAs) for the numerical simulation of large-scale dynamical systems on networks.

It has been our intention to state on the one hand the fundamental concepts to such an extent that technical details are avoided and to demonstrate on the other hand the capability of our novel modus operandi on the basis of representative test problems.

As scopes with significant benefits, we have identified correspondent classical dynamical systems involving a high number of components with pairwise or higher-order interactions. Accordingly, when combined with effective community detection algorithms, the advantages over standard approaches are transferred to dynamical systems on networks in case that they have the characteristics of numerous interactions within large communities.

Evidently, for the numerical simulations of demanding large-scale applications, it is advisable to optimise free parameters in the global approximation and to amend CIAs with state-of-the-art methods ranging from the automatic recognition of blocks in adjacency matrices to parallelisation.

Concluding, we hope that this work inspires current research in various fields as it complements the conventional assessment that numerical simulations of large-scale systems involving high numbers of interactions are out of reach. We hope to open up novel perspectives according to the principle *revealing intrinsic structures is our secret of success*.

Acknowledgements. TB acknowledges support by the TUM Institute for Advanced Study (TUM-IAS) through a Hans Fischer Fellowship awarded to Christian Bick and by the TUM TopMath elite study program. CK acknowledges support by the VolkswagenStiftung through a Lichtenberg Professorship.

[2] See https://techmath.uibk.ac.at/mecht/MyHomepage/Research/Public_CuckerSmale_KuramotoDaido_2022

[3] See pixabay.com and https://networkrepository.com/aves-wildbird-network.php.

References

[1] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[2] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[3] A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical Processes on Complex Networks*. CUP, 2008.

[4] M. Porter and J. Gleeson, *Dynamical Systems on Networks: A Tutorial*. Frontiers in Applied Dynamical Systems: Reviews and Tutorials, Springer, 2016.

[5] I. Kiss, J. Miller, and P. Simon, *Mathematics of Epidemics on Networks: From Exact to Approximate Models*. Springer, 2017.

[6] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization*. CUP, 2001.

[7] T. Vicsek and A. Zafiris, "Collective motion," *Phys. Rep.*, pp. 71–140, 2012.

[8] J. S. Adelman, S. C. Moyers, D. R. Farine, and D. M. Hawley, "Feeder use predicts both acquisition and transmission of a contagious pathogen in a North American songbird," *Proceedings of the Royal Society B: Biological Sciences*, vol. 282, sep 2015.

[9] F. Cucker and S. Smale, "Emergent behavior in flocks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 852–862, 2007.

[10] H. Daido, "Onset of cooperative entrainment in limit-cycle oscillators with uniform all-to-all interactions: Bifurcation of the order function," *Physica D: Nonlinear Phenomena*, vol. 91, pp. 24–66, 1996.

[11] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*. Springer, 2012.

[12] T. Böhle, C. Kuehn, and M. Thalhammer, "On the reliable and efficient numerical integration of the Kuramoto model and related dynamical systems on graphs," *International Journal of Computer Mathematics*, pp. 1–27, jul 2021.

[13] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003.

[14] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, "Networks beyond pairwise interactions: structure and dynamics," *Phys. Rep.*, vol. 874, pp. 1–92, 2020.

[15] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Phys. Rep.*, vol. 544, no. 1, pp. 1–122, 2014.

[16] T. Gross and H. Sayama, eds., *Adaptive Networks: Theory, Models and Applications*. Springer, 2009.

[17] P. Holme and J. Saramäki, "Temporal networks," *Phys. Rep.*, vol. 519, no. 3, pp. 97–125, 2011.

[18] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015.

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P10008, oct 2008.

[20] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, p. 066111, dec 2004.

[21] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," *7th Python in Science Conference (SciPy 2008)*, no. February, pp. 11–15, 2008.

[22] E. A. Leicht and M. E. J. Newman, "Community Structure in Directed Networks," *Physical Review Letters*, vol. 100, p. 118703, mar 2008.

[23] P. Pons and M. Latapy, "Computing Communities in Large Networks Using Random Walks," in *Computer and Information Sciences - ISCIS 2005* (P. Yolum, T. Güngör, F. Gürgen, and C. Özturan, eds.), (Berlin, Heidelberg), pp. 284–293, Springer Berlin Heidelberg, 2005.

[24] J. Reichardt and S. Bornholdt, "Detecting fuzzy community structures in complex networks with a potts model," *Physical Review Letters*, vol. 93, no. 21, pp. 19–22, 2004.

[25] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical Review E*, vol. 74, p. 016110, jul 2006.

[26] G. Rossetti, L. Milli, and R. Cazabet, "CDLIB: a python library to extract, compare and evaluate communities from complex networks," *Applied Network Science*, vol. 4, no. 1, 2019.

[27] V. A. Traag, G. Krings, and P. Van Dooren, "Significant Scales in Community Structure," *Scientific Reports*, vol. 3, p. 2930, dec 2013.