Role of the dynamical functional particle method for solving linear equations

S. Edvardsson, M. Neuman, P. Edström*
Department of natural sciences, Mid Sweden University, SE-871 88, Härnösand, Sweden

M. Gullikson and H. Olin
Department of natural sciences, Mid Sweden University, SE-851 70, Sundsvall, Sweden

The present work is concerned with exploiting a second order dynamical system approach for solving equations. The particular focus here is for real valued linear equation systems having a wide eigenvalue spectrum, often encountered in applications as discretization becomes dense. The Dynamical Functional Particle Method (DFPM) is developed as an interdisciplinary method between physics and numerical mathematics. The method is this context optimized to take advantage of critically damped oscillators resulting in good performance. Convergence is reached when all eigenvalues are either positive or negative. The case with complex eigenvalues is also studied. The particular structure of the matrix turns out to be unimportant for convergence. Furthermore, DFPM is not limited with respect to a spectral radius as is common for iterative methods. A first order dynamical system is also studied and compared. Its performance is not competitive in comparison with DFPM. The performance of DFPM scales equally to the well known conjugate gradient method, but lacks some of its limitations imposed by matrix symmetry. Several physical test examples are provided and compared with various existent numerical approaches.

I. INTRODUCTION

Particle methods in computational physics

The numerical treatment through the application of particle methods has been a very active research area during many years but these methods have still not been exploited to their full potential [1, 2]. Particle methods are conceptually attractive since they are sprung from the very fundamental core of physics. The basic assumption is that all matter consists of interacting particles that obey certain laws of motion (such as Newton's second law). The first particle methods were developed soon after the birth of fast computers some 40 years ago. A few examples of particle treatments of dynamical objects in celestial mechanics are given in [3, 4]. The long-term evolution of dynamical objects in the Solar system (or a small galaxy) can readily be calculated [5]. The discrete element method (DEM) is among the early methods which appeared in the 1970s [6]. Today DEM is often applied to treat engineering problems in granular and discontinuous materials. Another relatively early example is the molecular dynamics simulation technique (MDS) that was developed during the 1980s. Today, millions of molecules or atoms are more or less non-problematic to simulate due to the simplicity of parallelizing the N-body algorithms [7]. Smoothed-particle hydrodynamics (SPH) is yet another particle technique that deals with simulations of fluids. SPH was originally developed by Lucy [8] and Gingold and Monaghan [9] to deal with 3D problems in astrophysics. SPH has also been adopted to solid mechanics to study impact fractures in solids. The method is in this context abbreviated SPAM (smooth particle applied mechanics) [10]. There are also MDS-like particle methods (Hybrid Lattice Particle Modeling), where Lennard-Jones potentials or similar interaction potentials are utilized for quasi-particles [11]. In the limit when the quasi-particles approach the atomic scale the particle method becomes an ordinary MDS method. The Car-Parrinello method is an extremely well known particle method for dynamical computation of electronic quantum states during molecular dynamics simulation [12]. Also in density functional theory, particle methods have been applied to compute electronic states, see a good review by Payne et al. [13]. Particle methods have often been found attractive for running fast computer graphics of deformable objects [14, 15]. A quite comprehensive review on various particle approaches is provided by Li and Liu [2].

In a recent paper by Edvardsson et al. [16], it was argued that the particle concept is more general than its standard usage in physics or mechanics. Inspired by the connection between particles in physics and their usual translation into differential equations, the opposite possibility to translate mathematical problems into a quite general particle scheme was recognized. The dynamical functional particle method (DFPM) was developed. Edvardsson et al. applied DFPM to eigenvalue problems (Schrödinger equation) and compared its computational speed to standard numerical

*Electronic address: sverker.edvardsson@miun.se

libraries such as ARPACK and LAPACK. For large matrix sizes it was seen that the particle method was much more efficient in the determination of a few eigenvalues and eigenvectors. The conjugate gradient method was tested later dealing with the same Schrödinger problem and found to be some 50% less efficient than DFPM [17]. DFPM was also demonstrated to work well for the non-linear Schrödinger equation [16]. Convergence properties were addressed, and in analogy with many-particle systems in mechanics, the existence of a potential minimum is sufficient to guarantee convergence. However, it was recognized empirically that there exist many cases where convergence still occurs despite the lack of a potential. This fact, and the details related to computational complexity, was left out for further studies. These details will be studied in the present work for the special, but important, case of linear equation systems. As we shall see the DFPM algorithm is attractive due to several reasons. The most interesting points are related to a surprisingly general applicability and robustness, computational complexity, easiness of implementation, interesting Hamiltonian dynamics and stability of symplectic integration.

Related mathematical methods

Although the second order dynamical system approach provided by the present work to solve linear problems is novel, there are related ideas given previously in mathematics. The most common method is the formulation of a first order dynamical system, i.e., the setup: $du/dt = \mathcal{F}(u), u(0) = u_0$. As $du/dt \to 0$ the original problem $\mathcal{F}(u) = 0$ is solved. Thus the idea is to solve a damped time dependent problem to identify the stationary solution. First order dynamical systems have frequently been applied to solve various kinds of equations $\mathcal{F}(u) = 0$, both as a general approach and intended for specific mathematical problems. An example of this is the solution of an elliptic PDE such as the heat equation, see Sincovec and Madsen [18]. Since the stationary state is sought, the evolution of the system is considered to take place in *artificial time*. The concept of artificial time is further discussed and analyzed in [19]. Other works are for example the damped harmonic oscillator in classical mechanics, the damped wave equation, [20] and the heavy ball with friction [21]. These problem settings are specific mathematical examples of physical systems and not developed to solve equations in general. In [22, 23], iterative processes to solve, e.g., eigenvalue problems are considered as (gradient driven) dynamical systems. So called fictitious time is used in [24] where a Dirichlet boundary value problem of quasilinear elliptic equation is solved by using the concept of a fictitious time integration method. The inverse problem of recovering a distributed parameter model is considered in [19] using the first order ODE attained from the necessary optimality conditions of the inverse problem. Another approach is that of continuation, see [25] for an introduction, where $\mathcal{F}(u) = 0$ is embedded in a family of problems depending on a parameter s, i.e., $\mathcal{F}(u; s) = 0$. The solution to $\mathcal{F}(u) = 0$ is found by solving a sequence of problems for values of s decreasing from 1 to 0. Further, see Nocedal and Wright [26] for a discussion in the context of optimization.

Outline of the paper

The paper starts with an introduction to the general ideas behind DFPM and its obvious connections to classical mechanics. The versatility of DFPM is emphasized. The paper then continues to treat specifically the $Ax = b$ problem. The convergence and convergence rate is investigated. This analysis is expected to be useful also later for other related problems (e.g. eigenvalue problems). The connections with the particle scheme in physics is emphasized throughout the article. It turns out that physical arguments and many-particle views provide important shortcuts in the derivations and analyzes. A comparison between DFPM and the related first order dynamical system is also made. The article is concluded with numerical examples and comparisons with existent methods in the numerical linear algebra literature. In order to enhance readability of the paper we have provided an Appendix where several related details are given.

II. THE DYNAMICAL FUNCTIONAL PARTICLE METHOD

In the following we make a brief summary for the reader about DFPM as a quite general approach for solving equations. Let $\mathcal{F}$ be a functional and $u = u(x)$, $u : X \to \mathbb{R}^k$, $k \in \mathbb{N}$ and consider the abstract equation

$$\mathcal{F}(u) = 0 \tag{1}$$

that could be, e.g., a functional, differential, integral or integro-differential equation. Further, a time parameter t is introduced that belongs to some (unbounded) interval T and a dynamical system in $u = u(x, t)$, $u : X \times T \to \mathbb{R}^k$ is formed as

$$\mathcal{F}(u) = \eta\ddot{u} + \mu\dot{u}. \tag{2}$$

where the dots are the standard notation for time derivatives in mechanics. The symbols $\eta = \eta(x,t)$ and $\mu = \mu(x,t)$ are the mass and damping parameters, respectively. The main idea is to solve the original equation (1) by instead solving (2) in such a way that $\dot{u}$, $\ddot{u} \to 0$ when $t \to T$, $T < \infty$, i.e., $u(x,t)$ approach $u(x)$ as $t \to T$. Further, two independent initial conditions are needed for (2) to be well defined. In order to obtain a numerical solution of $u(x,t)$, equation (2) is discretized such that $u_i(t)$ approximates $u(x_i,t)$ and $\mu_i(t) = \mu(x_i,t)$, $\eta_i(t) = \eta(x_i,t)$ for $i = 1,\ldots,n$. The discretization is made here with finite differences, but it is possible to use basis functions, finite elements, or any other method of discretization. After a finite difference discretization we have the following equations

$$\mathcal{F}_i(u_1 \ldots, u_n) = \eta_i \ddot{u}_i + \mu_i \dot{u}_i,\ i = 1,\ldots,n \tag{3}$$

corresponding to equation (2). This discretized second order dynamical system is a system of ordinary differential equations. We call the approach for solving (1) using (3) the *Dynamical Functional Particle Method*, DFPM. We emphasize that the idea behind DFPM is quite versatile. There is for example no linear restriction built into the method so non-linear equations (functionals) are certainly possible to attack. However, in this work we shall investigate only the special, but important, case of systems of linear equations.

A. Connection between DFPM and classical mechanics

It is clear that DFPM is a particle method belonging to the domain of classical mechanics. A natural interpretation of the functional $\mathcal{F}(u)$ is that after discretization it may be viewed as a vector force field. Consider the special case when the discretized version of a linear differential equation reduces to n linear equations, i.e., $Ax = b$. One possibility is then to write the functional as $\mathcal{F}(x) = b - Ax$. This force field is thus n-dimensional. A dynamical particle method can be constructed by viewing this as the problem to determine the positions of n particles where the force is zero, i.e., the equilibrium point of a conservative many-particle system. That is, if $\mathcal{F}(x)$ is conservative we know that there exists a corresponding scalar potential $\Phi(x)$. The many-particle system will tend to equilibrate towards one of its minimum points if dissipation is present [27]. In this particular case the DFPM equation is simply given by

$$b - Ax = \eta\ddot{x} + \mu\dot{x}$$

where the dissipation term is $\mu\dot{x}$. In the case of a symmetric matrix A, a many-particle potential $\Phi(x)$ does exist and is explicitly given by

$$\Phi(x) = \frac{1}{2}\sum_i A_{ii}x_i^2 + \sum_{i<j} A_{ij}x_i x_j - \sum_i b_i x_i = \frac{1}{2}x^T Ax - b^T x \tag{4}$$

because each component k of $\mathcal{F}(x)$ is given by

$$\mathcal{F}_k = -\frac{\partial\Phi}{\partial x_k} = b_k - A_{kk}x_k - \sum_{i\neq k} A_{ki}x_i$$

which is completely consistent with each component of $\mathcal{F}(x) = b - Ax$. At the solution x, all the component forces fulfill $\mathcal{F}_k = 0$ and thus also the gradient of $\Phi(x)$ is zero. This critical point corresponds to a minimum if A is positive definite and a maximum if A is negative definite. Obviously, if A is negative definite one can instead use the ansatz $\mathcal{F}(x) = Ax - b$. However, if A is nonsymmetric, a potential function does *not* exist [17], so a possible convergence of the dynamical system needs to be analyzed in further detail. Also, the convergence rate is not available by the above potential analysis. These important issues will be addressed in the following.

III. DFPM FOR THE $Ax = b$ PROBLEM

Consider the following discretized interaction functional $\mathcal{F} = b - Ax$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. In this case $\mathcal{F}$ correspond to the ordinary residual in numerical linear algebra. To simplify the presentation we make the additional assumption that the eigenvalues $\lambda(A)$ are all real and positive. Other cases will be dealt with as we go on. To proceed, consider the second order dynamical system:

$$\mathcal{F}(x_1, x_2, \dots, x_n) = b - Ax = \eta\ddot{x} + \mu\dot{x} \tag{5}$$

where $\mu\dot{x}$ is the dissipation term. The real scalar parameters $\eta > 0$ (mass) and $\mu > 0$ (damping) are here assumed to be constants. We thus suggest a particle method where the positions $x_1, x_2, \dots, x_n$ are optimized in order to approach the unique equilibrium point, i.e., $\mathcal{F}(x_1, x_2, \dots, x_n) \to 0$ as $t \to T$. In practice, one may start with a random ansatz for the vector $x\left(0\right)$ and let $\dot{x}\left(0\right) = 0$. The particular numerical time integration, using a real valued time step $\Delta t > 0$ to move the vectors $x\left(t\right)$ and $\dot{x}\left(t\right)$, is here made by the cost effective and stable symplectic Euler [28–30]:

$$\left\{ \begin{array}{rcl} \dot{x}\left(t+\Delta t\right) & = & \dot{x}\left(t\right) + \left(\frac{1}{\eta}b - \frac{1}{\eta}Ax\left(t\right) - \frac{\mu}{\eta}\dot{x}\left(t\right)\right)\Delta t \\ x\left(t+\Delta t\right) & = & x\left(t\right) + \dot{x}\left(t+\Delta t\right)\Delta t. \end{array} \right. \tag{6}$$

Firstly, it is important to establish that a convergent result solves the original problem $Ax = b$. If $x\left(t+\Delta t\right) \approx x\left(t\right)$, the second equation of (6) gives that $\dot{x}\left(t+\Delta t\right) \approx 0$. If $\dot{x}\left(t+\Delta t\right) \approx \dot{x}\left(t\right)$, the first equation of (6) gives that $\mu\dot{x}\left(t\right) \approx b - Ax\left(t\right)$. Since $\dot{x}\left(t\right) \approx \dot{x}\left(t+\Delta t\right)$ and $\dot{x}\left(t+\Delta t\right) \approx 0$, we indeed have that $Ax\left(t\right) \approx b$.

Admittedly, the algorithm (6) is not particularly accurate but its stability and accuracy is still superior to the Euler method [29]. Further, it should be noted that high numerical accuracy of the many-particle system during its evolution towards the stationary state is *not* necessary. The only desired property is that the approach towards the equilibrium point is made as fast as possible.

The features of interest are thus: 1) good numerical stability allowing a large time step Δt, and 2) application of a cheap algorithm in order to spend little CPU time per iteration. These features are all provided by the symplectic integration method [28, 30, 31]. Other symplectic algorithms may still be of interest, among which the Verlet-Störmer method is one of the most common [32]. However, although this method is more accurate, it was found that no gain could be obtained. The number of iterations towards convergence is similar (if not the same) but the cost per iteration is higher.

In order to be flexible in the analysis below, both parameters η and μ are kept throughout the derivations, i.e., we shall not let e.g. $\eta = 1$ as suggested by a continuum view. The reason for this is that, possibly, a discrete numerical algorithm that is nonlinear (as for example higher order symplectic algorithms or Runge-Kutta methods) can result in performance properties depending on both η and μ. Such an example is provided later (A.58). Below we make the assumption that the matrix A is diagonalizable since it simplifies the analysis tremendously.

A. Optimal parameters for a single oscillator

In order to identify optimal parameters μ, η and Δt in (6) let us rewrite (5) by making a change of basis into A's eigenvectors. That is, $b \to c$, $x \to u$, $A \to \Lambda_A = \mathrm{diag}\left(\lambda_1, \lambda_2, ..., \lambda_n\right)$. We then obtain the following decoupled system of equations:

$$c - \Lambda_A u = \eta\ddot{u} + \mu\dot{u}. \tag{7}$$

After a change of variables $w_i = u_i - c_i/\lambda_i$ we have

$$\begin{array}{rcl} -\lambda_1 w_1 & = & \eta\ddot{w}_1 + \mu\dot{w}_1 \\ -\lambda_2 w_2 & = & \eta\ddot{w}_2 + \mu\dot{w}_2 \\ \vdots & \vdots & \vdots \\ -\lambda_n w_n & = & \eta\ddot{w}_n + \mu\dot{w}_n. \end{array} \tag{8}$$

The reader should note that (8) is only used for the analysis of optimal parameters, i.e., (8) should not be confused with the actual computational algorithm (6). The great advantage of (8) is that the whole problem has been reduced to the analysis of n damped oscillators in classical mechanics. Now consider the symplectic time integration of one such oscillator, $-\lambda w = \eta\ddot{w} + \mu\dot{w}$:

$$\left\{ \begin{array}{rcl} \dot{w}\left(t+\Delta t\right) & = & \dot{w}\left(t\right) + \left(-\frac{\lambda}{\eta}w\left(t\right) - \frac{\mu}{\eta}\dot{w}\left(t\right)\right)\Delta t \\ w\left(t+\Delta t\right) & = & w\left(t\right) + \dot{w}\left(t+\Delta t\right)\Delta t. \end{array} \right. \tag{9}$$

The iterations can conveniently be summarized on the form $z_{n+1} = Bz_n$:

$$z_{n+1} = \begin{bmatrix} w_{n+1} \\ \dot{w}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 - \frac{\lambda}{\eta}\Delta t^2 & \left(1 - \frac{\mu}{\eta}\Delta t\right)\Delta t \\ -\frac{\lambda}{\eta}\Delta t & 1 - \frac{\mu}{\eta}\Delta t \end{bmatrix} \begin{bmatrix} w_n \\ \dot{w}_n \end{bmatrix}. \tag{10}$$

One can also write $z_{n+1} = B^n z_0$. If we apply the similarity transformation such that $B = C^{-1}\Lambda_B C$ where $\Lambda_B = \mathrm{diag}(\alpha_1, \alpha_2)$, we note that convergence is achieved if $|\alpha_{1,2}| < 1$. A minimization of the maximum eigenvalue gives the optimal convergence rate. The two eigenvalues are explicitly given by:

$$\alpha_{1,2} = 1 - \frac{\lambda}{2\eta}\Delta t^2 - \frac{\mu}{2\eta}\Delta t \pm \frac{\Delta t}{2\eta}\sqrt{\zeta} \tag{11}$$

where $\zeta = (\mu + \lambda\Delta t)^2 - 4\eta\lambda$. There are three possible cases, but only two are of interest. The overdamped case ($\zeta > 0$) is well known in mechanics [33] to only slowly restore the system to the equilibrium point and is therefore discarded. Ideally, we have the critically damped case ($\zeta = 0$), but we cannot expect all the oscillators in (8) to be critically damped so the underdamped case ($\zeta < 0$) will also be of interest. The problem is then to minimize ζ for all of them. The best solution for a single oscillator is given by $(\mu + \lambda\Delta t)^2 = 4\eta\lambda$. Since $\lambda > 0$ we find that

$$\mu + \lambda\Delta t = 2\sqrt{\eta\lambda} \tag{12}$$

In this case, both eigenvalues are simply given by $\alpha_{1,2} = 1 - \sqrt{\lambda/\eta}\Delta t$ so $\Delta t_{\mathrm{opt}} = \sqrt{\eta/\lambda}$ and $\Delta t_{\max} = 2\sqrt{\eta/\lambda}$. The optimal choice of time step, Δt_{opt}, makes the algorithm to finish in just one iteration. This is great for this single oscillator. However, the problem (5) corresponds to many oscillators as given by (8), so most oscillators will need much more than one single iteration to converge. In the case of underdamping one can similarly as above show that $|\alpha_{1,2}|^2 = 1 - (\mu/\eta)\Delta t$, so an underdamped oscillator converges if $0 < (\mu/\eta)\Delta t < 1$. We are now prepared to understand the general behavior of n oscillators. Eq. (12) cannot be fulfilled for all the oscillators but we shall attempt to make (12) to be nearly fulfilled for them.

B. Optimal parameters for n oscillators

As mentioned above, it is important for optimal convergence that none of the oscillators is overdamped, because such a situation would give a very slow progress towards the equilibrium point. The convergence properties of the whole system would then suffer since all the other oscillators would have to wait. We shall therefore proceed by studying the cases where only critical and light damping are allowed.

First consider only two oscillators. As seen in (6) the parameters μ and Δt are global for all oscillators. Eq. (12) thus gives that

$$\begin{bmatrix} 1 & \lambda_1 \\ 1 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mu \\ \Delta t \end{bmatrix} = 2\sqrt{\eta} \begin{bmatrix} \sqrt{\lambda_1} \\ \sqrt{\lambda_2} \end{bmatrix}. \tag{13}$$

If $\lambda_1 \neq \lambda_2$, there is an unique solution for μ and Δt so that both oscillators evolve according to the critically damped case. This solution is given by $\mu = 2\sqrt{\eta}\sqrt{\lambda_1\lambda_2}/\left(\sqrt{\lambda_1} + \sqrt{\lambda_2}\right)$ and $\Delta t = 2\sqrt{\eta}/\left(\sqrt{\lambda_1} + \sqrt{\lambda_2}\right)$. In the general case of n oscillators the condition $\mu + \lambda_i\Delta t \leq 2\sqrt{\eta\lambda_i}$ must be fulfilled for all λ_i. This condition only holds if

$$\mu_{\mathrm{opt}} = 2\sqrt{\eta}\sqrt{\lambda_{\min}\lambda_{\max}}/\left(\sqrt{\lambda_{\min}} + \sqrt{\lambda_{\max}}\right) \tag{14}$$

$$\Delta t_{\mathrm{opt}} = 2\sqrt{\eta}/\left(\sqrt{\lambda_{\min}} + \sqrt{\lambda_{\max}}\right) \tag{15}$$

This can be seen by inserting μ_{opt} and Δt_{opt} into the condition $\mu + \lambda_i\Delta t \leq 2\sqrt{\eta\lambda_i}$. One then finds that

$$\left(2\sqrt{\lambda_i} - \sqrt{\lambda_{\min}} - \sqrt{\lambda_{\max}}\right)^2 \leq \left(\sqrt{\lambda_{\min}} - \sqrt{\lambda_{\max}}\right)^2 \tag{16}$$

which indeed is fulfilled for all $\lambda_i \in [\lambda_{\min}, \lambda_{\max}]$. We thus have the desired situation that the oscillators with $\lambda_{\min}$ and $\lambda_{\max}$ are both critically damped and the rest are lightly damped. Convergence of the two critically damped oscillators is ensured because $\Delta t_{\text{opt}} < \Delta t_{\max} = 2\sqrt{\eta/\lambda_i}$, $i = \max, \min$ (see previous Section). The optimized results given by (14) and (15) also ensures convergence for all the lightly damped oscillators. Their convergence is guaranteed if $0 < (\mu/\eta)\,\Delta t < 1$ (see previous Section), i.e., (14) and (15) then implies that

$$\frac{\mu_{\text{opt}}}{\eta}\Delta t_{\text{opt}} = 4\frac{\sqrt{\lambda_{\min}\lambda_{\max}}}{\left(\sqrt{\lambda_{\min}}+\sqrt{\lambda_{\max}}\right)^2} < 1 \tag{17}$$

A well known fact is that an arithmetic mean is always greater than its corresponding geometric mean, so $\left(\sqrt{\lambda_{\min}}+\sqrt{\lambda_{\max}}\right)/2 \geq \sqrt{\sqrt{\lambda_{\min}}\sqrt{\lambda_{\max}}} \Rightarrow \left(\sqrt{\lambda_{\min}}+\sqrt{\lambda_{\max}}\right)^2 \geq 4\sqrt{\lambda_{\min}\lambda_{\max}}$ and thus the inequality in (17) is proved, meaning that all the underdamped oscillators also converge. The convergence of the particle method is thus always fulfilled for the optimized parameters. The only way to get a divergent result is to apply considerably larger values than those suggested here.

Simulations suggest that precise estimates of the eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ are not crucial so we suggest that crude estimates can be derived in just a few iterations by any of the standard numerical procedures available, see e.g. [34]. In the Appendix I we also provide estimates of μ and Δt directly without knowledge of eigenvalues.

C. Convergence rate of the wide eigenvalue spectrum

In order to estimate the number of iterations required for DFPM to converge we shall consider the important case of a wide eigenvalue spectrum (i.e., $\lambda_{\max} \gg \lambda_{\min}$). In physics this case is very common as the grid is made more and more dense or the number of basis functions is increased. In the previous Section we note that two oscillators are critically damped ($\lambda_{\min}$ and $\lambda_{\max}$) and the rest are underdamped. The critically damped oscillators will converge first. Among the underdamped oscillators, the one furthest away from critical damping will be the last one to converge and thereby determine the number of iterations of the method.

Consider the condition for the damping: $\mu + \lambda_i\Delta t \leq 2\sqrt{\eta\lambda_i}$. Let $f(\lambda) = \mu + \lambda\Delta t - 2\sqrt{\eta\lambda}$ where $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ is a continuous real variable. The critically damped cases ($\lambda_{\min}$ and $\lambda_{\max}$) correspond to $f(\lambda) = 0$. This is also the maximum of $f(\lambda)$. The oscillator furthest away from critical damping is given by $\min(f)$. The first derivative is $f'(\lambda) = \Delta t - (\eta\lambda)^{-1/2}\eta$, which is zero when

$$\lambda = \frac{\eta}{\Delta t^2} = \frac{\left(\sqrt{\lambda_{\min}}+\sqrt{\lambda_{\max}}\right)^2}{4}, \tag{18}$$

where we have used Eq. (15) in the second equality. This corresponds to a minimum because $f''(\lambda) = \frac{1}{2}(\eta\lambda)^{-3/2}\eta^2 > 0$. Thus (18) gives the worst case scenario. In the discrete problem the eigenvalue λ_i closest to this λ will be associated with the weakest oscillator. Let us denote this eigenvalue λ_b. When this oscillator has converged the whole system of oscillators have converged. Thus, to analyze the convergence time of the method we need to study this oscillator. In order to simplify calculations, we assume that the symplectic integrator is able to approximately follow the analytical continuum solution (a common feature of symplectic integrators, see [28, 30, 31]).

Consider the evolution of the method (6). After a time T the distance between the evolution of (6) and the exact solution is given by $\|x_T - x\| = d$. In the corresponding system in (8) we have that $\|w_T - 0\| = d$. However, after a long enough time T we have that $\|w_T\| = \sqrt{\sum_i w_i^2} = \sqrt{w_b^2} = |w_b(T)|$ because then all the other oscillators will already have converged to nearly zero.

The most convenient way to study the convergence of the oscillator associated with λ_b is to use the exponential decay of the mechanical energy E, see e.g. [33]. Such a decay formula is valid provided that the oscillator is underdamped, i.e., it needs to fulfill $\mu \ll 2\sqrt{\lambda_b\eta}$. It does indeed, because if we insert (14) and assume that λ_b is approximately given by (18) we find that the condition simply becomes $2\sqrt{\lambda_{\min}\lambda_{\max}} \ll \lambda_{\min} + \lambda_{\max} + 2\sqrt{\lambda_{\min}\lambda_{\max}}$. The exponential decay formula can thus be applied

$$E(t) \approx E(0)\, e^{-\frac{\mu}{\eta}t} \tag{19}$$

and the mechanical energy is defined by

$$E(t) = E_k(t) + V(t) = \frac{1}{2}\eta\dot{w}_b^2 + \frac{1}{2}\lambda_b w_b^2$$

At $t = 0$ we have that $E(0) = \frac{1}{2}\lambda_b w_b(0)^2$ (since $\dot{w}_b(0) = 0$) and at $t = T$ we have that $E(T) = \frac{1}{2}\eta \dot{w}_b(T)^2 + \frac{1}{2}\lambda_b w_b(T)^2$. From earlier we know that $\|w_T\| = \|w_b(T)\| = d$ so $w_b(T)^2 = d^2$, but the kinetic energy is unknown. A reasonable estimate here is to use the mean kinetic energy. According to the virial theorem for a harmonic oscillator it is known that $\langle E_k \rangle = \langle V \rangle$ (this remains approximately true also for the underdamped harmonic oscillator). Therefore a reasonable estimate is that $E(T) \approx 2V(T) = \lambda_b w(T)^2 = \lambda_b d^2$. Using (19) we then conclude that at time T we have

$$\lambda_b d^2 \approx \frac{1}{2}\lambda_b w_b(0)^2 e^{-\frac{\mu}{\eta}T}$$

so the convergence time is

$$T \approx \frac{\eta}{\mu} \log\left(\frac{w_b(0)^2}{2d^2}\right). \tag{20}$$

The number of iterations n_{it} can now be estimated by evaluating $n_{it} \approx T/\Delta t$. The optimal parameters μ and Δt are given by Eqs. (14) and (15) respectively, leading to the final result

$$n_{it} \approx \frac{\left(\sqrt{\lambda_{\min}} + \sqrt{\lambda_{\max}}\right)^2}{\sqrt{\lambda_{\min}\lambda_{\max}}} \frac{1}{4} \log\left(\frac{w_b(0)^2}{2d^2}\right). \tag{21}$$

We see that (21) is independent of the mass parameter so in this case one may set $\eta = 1$ in (6). If we apply the convergence criterion $d = 10^{-10}$ we find that $(1/4)\log\left(w_b(0)^2/2d^2\right) \sim 10$, almost independently of $w_b(0)^2$. In applications the matrix A is often very large. It is then often the case that $\lambda_{\max} \gg \lambda_{\min}$. Then, as a rule of thumb, one would expect the number of iterations to be approximately given by

$$n_{it} \approx 10\sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}. \tag{22}$$

so even though the number of iterations suffers when the $\lambda_{\max}/\lambda_{\min}$ ratio grows, it only does so according to the square root dependence.

D. Convergence rate of the narrow eigenvalue spectrum

Here we consider the case of a narrow eigenvalue spectrum (i.e., $\lambda_{\max} \sim \lambda_{\min}$). According to (14) and (15) the optimal time step and damping then are given by $\Delta t = \sqrt{\eta/\lambda}$ and $\mu = \sqrt{\eta\lambda}$, (where one can take $\lambda = (\lambda_{\min} + \lambda_{\max})/2$) In this case all oscillators are closely critically damped so Eq. (19) is no longer applicable. However, now the convergence can be investigated by considering the evolution of a single critically damped oscillator [33]:

$$w(t) = w(0)\left(1 + \sqrt{\frac{\lambda}{\eta}}t\right)e^{-\sqrt{\frac{\lambda}{\eta}}t}.$$

This expression satisfies the initial condition $\dot{w}(0) = 0$. We are interested in the convergence time T for which $\|w(T)\| \leq d$, i.e. where, $\left(1 + \sqrt{\frac{\lambda}{\eta}}T\right)e^{-\sqrt{\frac{\lambda}{\eta}}T} \leq d/\|w(0)\|$. For simplicity, we introduce the variables $x = \sqrt{\frac{\lambda}{\eta}}T$ and $\delta = d/\|w(0)\|$, meaning that we need to find x for which $(1+x)e^{-x} = \delta$. This is rewritten into $\ln(1+x) - x - \ln\delta = 0$ and by applying one symbolic Newton-Raphson iteration with the initial guess $x = -\ln\delta$, we find

$$x \approx -\frac{(\ln\delta)^2 + (1-\ln\delta)\ln(1-\ln\delta)}{\ln\delta} \equiv \xi(\delta),$$

which is a sufficiently good approximation for our purposes here (typical error is less than 1 percent). The convergence time T then becomes

$$T = \sqrt{\frac{\eta}{\lambda}}\xi\left(\frac{d}{\|w(0)\|}\right) \tag{23}$$

Given that the optimal time step is $\Delta t = \sqrt{\eta/\lambda}$ and $T \approx n_{it}\Delta t$, the number of iterations is expected to be approximately given by

$$n_{it} = \xi\left(\frac{d}{\|w(0)\|}\right) \tag{24}$$

The number of iterations is thus no longer dependent on the eigenvalues. Besides the special interest of a narrow eigenvalue spectrum, the full usefulness of this result cannot be underestimated. If a method could be introduced that is capable of running all oscillators as critically damped, its convergence would be given by (24). The computational complexity would in such a case be exceptionally good. This would yield a $\mathcal{O}\left(n^2\right)$ method for a dense matrix A and a $\mathcal{O}(n)$ method for a sparse matrix A, thus by far outperforming all existing methods. In standard numerical linear algebra, attempts to densify the eigenvalue spectrum is in fact made through preconditioning techniques [34]. These techniques are, however, not always successful which is why a fully critically damped dynamical particle method would be of extremely high interest.

IV. COMPARISON WITH THE FIRST ORDER DYNAMICAL SYSTEM

There is one relevant algorithm that is closely related to DFPM, namely the first order dynamical system. We shall therefore study this system in some detail. In fact, there are many papers in the literature where first order equations have been applied in order to obtain various numerical solution methods (see the Introduction). The first order equation has the interesting mathematical property that it, just as DFPM, yields exponential evolutions in time. We shall therefore investigate how such a method compares in performance with DFPM.

Thus, consider again the interaction functional $\mathcal{F} = b - Ax$ where A, as before, is a real matrix and the vectors are real valued. The various eigenvalues are still assumed to be real and positive. Let us attempt to solve for $\mathcal{F} = 0$ through the first order dynamical system:

$$\mathcal{F}(x_1, x_2, \ldots, x_n) = b - Ax = \mu\dot{x} \tag{25}$$

where $\mu\dot{x}$ is the dissipation term and $\mu > 0$ is a constant. We see that the derivative takes the simple form $f(x) \equiv \dot{x} = (b - Ax)/\mu$. The idea is that the vector components $x_1, x_2, \ldots, x_n$ evolve exponentially in time in such a way that $\mathcal{F}(x_1, x_2, \ldots, x_n) \to 0$ as $t \to T$. One may start with a random ansatz for the vector $x(0)$. The numerical time integration using a time step $\Delta t > 0$ for the vector x can be made by any of the standard first order methods available. In particular, an explicit method can generally be written as

$$x(t + \Delta t) = x(t) + \Delta t g(x(t)). \tag{26}$$

In the case that $g(x(t)) = f(x(t))$ this becomes the first order Runge-Kutta method (i.e., the Euler method). If $g(x(t)) = f(x(t) + (\Delta t/2) f(x(t)))$, we have the second order Runge-Kutta method, etc. For simplicity, we here provide the derivations for the first order Runge-Kutta integration. The second order integration is provided in the Appendix II. The goal is to derive the computational complexity and compare it with DFPM.

In order to identify the optimal parameters Δt and μ let us as previously apply the change of basis into A's eigenvectors. I.e., $b \to c$, $x \to u$, $A \to \Lambda_A = \operatorname{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$. We then get the following decoupled system of equations:

$$c - \Lambda_A u = \mu\dot{u} \tag{27}$$

After a change of variables $w_i = u_i - c_i/\lambda_i$ we arrive at

$$\begin{array}{rcl} -\lambda_1 w_1 & = & \mu\dot{w}_1 \\ -\lambda_2 w_2 & = & \mu\dot{w}_2 \\ \vdots & \vdots & \vdots \\ -\lambda_n w_n & = & \mu\dot{w}_n \end{array} \tag{28}$$

The reader should once again note that (28) is used only for the analysis, i.e., this should not be confused with the actual numerical solution of (25) using (26). Now consider the time integration of one such equation: $-\lambda w = \mu \dot{w}$:

$$w\left(t+\Delta t\right) = w\left(t\right) + \Delta t\left(-\frac{\lambda}{\mu}w\left(t\right)\right). \tag{29}$$

We see that at time $t = n\Delta t$ we have

$$w\left(t\right) = \left(1 - \frac{\lambda}{\mu}\Delta t\right)^{n} w\left(0\right) \tag{30}$$

The corresponding continuum result ($\Delta t = t/n$; $n \to \infty$) is given by $w\left(t\right) = \exp\left(-\frac{\lambda}{\mu}t\right)w\left(0\right)$. We shall therefore call the equations in (28) “exponentials”. We get convergence in (30) if $\left|1 - \frac{\lambda}{\mu}\Delta t\right| < 1$, leading to $\Delta t_{\text{max}} = 2\mu/\lambda$. The optimal time step is given by $\Delta t_{\text{opt}} = \mu/\lambda$. These results are helpful in order to understand the general behavior of equation (28). Below we provide the optimization of Δt and μ for the whole system of exponentials.

A. Optimal parameters for n exponentials

We shall now proceed, as before, by attempting to optimize two exponentials in (28). The optimal situation for a particular exponential i is if $\alpha_i \equiv 1 - \left(\lambda_i/\mu\right)\Delta t$ equals zero, see (30). For two exponentials, the optimal situation translates into

$$\begin{bmatrix} 1 & -\lambda_1 \\ 1 & -\lambda_2 \end{bmatrix}\begin{bmatrix} \mu \\ \Delta t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{31}$$

However, if $\lambda_1 \neq \lambda_2$ the only solution is $\Delta t = \mu = 0$. Accordingly, a fully optimal situation is not possible the two exponentials. If one instead let $\alpha_1 = \alpha_2$, only $\Delta t = 0$ is possible. It turns out that the only possibility is if $\alpha_1 = -\alpha_2$, in which case we have that

$$\Delta t = \frac{2\mu}{\lambda_1 + \lambda_2}.$$

Note the restriction on the time step: $0 < \Delta t < \Delta t_{\text{max}} = 2\mu/\lambda$ for each exponential (previous Section). It is realized that the exponential with the largest eigenvalue will limit the overall stability of the method, i.e., $\Delta t < 2\mu/\lambda_{\text{max}}$ is absolutely required. The best time step for this exponential is $\Delta t = \mu/\lambda_{\text{max}}$. On the other hand, the exponential with the smallest eigenvalue leads to a larger time step $\Delta t = \mu/\lambda_{\text{min}}$ possibly outside the boundary Δt_{max}. For the group of exponentials it is therefore the extreme cases that determines the optimal time step:

$$\Delta t_{opt} = \frac{2\mu}{\lambda_{\text{min}} + \lambda_{\text{max}}}. \tag{32}$$

As we compare (32) with the stability condition $\Delta t < 2\mu/\lambda_{\text{max}}$ we see that it is satisfied so convergence is guaranteed.

B. Convergence rate of the first order system

Among the exponentials in (28) one will show the slowest convergence and thereby determine the convergence properties of the whole system. This exponential is the one corresponding to the smallest eigenvalue. The convergence properties of the whole system of exponentials can therefore be investigated by considering this single exponential. The continuum solution for this exponential is given by:

$$w\left(t\right) = w\left(0\right)e^{-\frac{\lambda_{\text{min}}}{\mu}t} \tag{33}$$

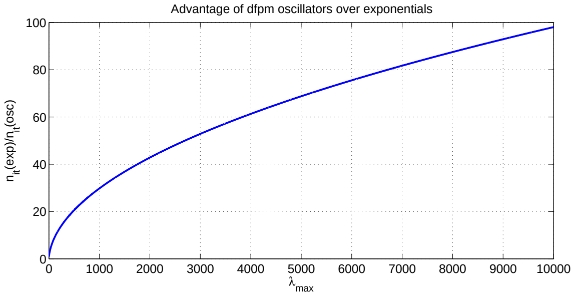

Figure 1: The ratio between the number of iterations for exponentials and oscillators (ratio of the expressions (A.58) and (21)), i.e. $n_{it}(exp)/n_{it}(osc)$. The example is made for $\mu = 1$, $\lambda_{\min} = 1$ and various $\lambda_{\max} = 2, 3, ..., 10000$. The logarithmic function in (A.58) and (21) is taken to be 20 and 40, respectively (see text). It is seen that the second order method (DFPM) is expected to be orders of magnitudes more efficient.

For the purposes here, it is sufficient to assume that the evolution in discrete time approximately follows (33). We are interested in the time T for which $\|w(T)\| \leq d$, i.e., $e^{-\frac{\lambda_{\min}}{\mu}T} \leq d/\|w(0)\|$. The convergence time T thus becomes

$$T = \frac{\mu}{\lambda_{\min}} \log\left(\frac{\|w(0)\|}{d}\right). \tag{34}$$

Given the optimal time step in (32) and $T \approx n_{it}\Delta t_{\text{opt}}$, the number of iterations is approximately given by

$$n_{it} \approx \frac{\lambda_{\min} + \lambda_{\max}}{2\lambda_{\min}} \log\left(\frac{\|w(0)\|}{d}\right). \tag{35}$$

If we apply the convergence criterion $d = 10^{-10}$ we find that $\log(\|w(0)\|/d) \sim 20$, almost independently of $\|w(0)\|$. If we also have that $\lambda_{\max} \gg \lambda_{\min}$ then, as a rule of thumb, we expect the number of iterations to be given by

$$n_{it} \approx 10\frac{\lambda_{\max}}{\lambda_{\min}}. \tag{36}$$

This result is inferior to the result of DFPM given in (22). It can therefore be concluded that the second order dynamical system proposed in the present work is a good idea, though it is not at all obvious from a continuum point of view.

In order to illustrate the difference between the first order dynamical system and DFPM we provide an example in Fig. 1. The ratio between the number of iterations for exponentials and oscillators (ratio of the expressions (A.58) and (21)) is plotted for various $\lambda_{\max}$. It is clear that (21) gives a significant advantage in many applications as the ratio $\lambda_{\max}/\lambda_{\min}$ increases. However, a possible objection could be that the poor result in (36) is due to the simple first order time integration algorithm applied. We have therefore provided results in the Appendix II where an analogous derivation is made, but this time for a second order integration algorithm. This analysis shows that the resulting convergence rate is not improved.

V. COMPARISON WITH OTHER METHODS

To the best of our knowledge, there exists no iterative method that can handle matrices having an arbitrary structure. Instead, there are a huge number of specialized iterative methods in the literature that solve $Ax = b$ for various structured matrices A (positive definite, symmetric, tri-diagonal, block diagonal, etc.). A complete survey of all algorithms is not possible here. Even a detailed comparison with just a single method would require lots of space and far too many details. This is not justified considering the present context. Instead we shall only provide some arguments that give the reader at least a practical view of some of the differences. These details are provided in the Appendix III. The methods considered are: Jacobi iterations, Gauss-Seidel iterations, SSOR iterations, the conjugate gradient method, the steepest descent method and the Krylov subspace method GMRES. Among these methods, there are only two that stand out for the case $\lambda_{\max} \gg \lambda_{\min}$. Firstly, we have the conjugate gradient method, and secondly, SSOR iterations. Both of them are expected to perform similarly as the DFPM method.

The conjugate gradient method is originally designed only for symmetric positive definite A, but can be rewritten to work also for a symmetric negative definite matrix. SSOR is more versatile, but suffers by the limitation that its spectral radius must be less than one (see Appendix III). It is guaranteed to converge for a diagonally dominant symmetric positive definite A. DFPM has no limitation regarding spectral radius. DFPM is guaranteed to converge for any real valued matrix A if all its eigenvalues are larger than zero. In the Appendix IV, we show that DFPM converges also if all the eigenvalues are negative, and in many cases for complex eigenvalues (as long as $Re\,(\lambda) > 0$ or $Re\,(\lambda) < 0$). An example with complex eigenvalues will be provided below. Needless to say, the particular structure of the matrix is unimportant for DFPM, making it a very robust method.

Most of the below results will focus on the number of iterations needed until completion. However, one should remember that the number of iterations is not sufficient to rank methods. The number of floating point operations per iteration must also be analyzed. It can be seen that the conjugate gradient method needs to go through more operations than the DFPM algorithm (6), c.f. pp. 200 [35]. The same is even more true for SSOR (c.f. (6) with [36]). Numerical examples showing the number of iterations are nevertheless common in the literature and considered interesting to get a practical view of the differences between various methods.

VI. NUMERICAL EXAMPLES

Here we provide a range of examples demonstrating the robustness of DFPM and that its performance is expected to approach that of the conjugate gradient method (when applicable). This is in accordance with the above assessments and those in Appendix III. We start with a few relatively simple examples.

A. A nonsymmetric introductory example

Consider the nonsymmetric matrix

$$A = \begin{pmatrix} 3 & 1 & 4.2 \\ 1 & 4 & 2 \\ 3 & 2 & 7 \end{pmatrix}$$

where its eigenvalues are given by λ =0.9271, 3.153 and 9.919, i.e., all positive. The spectral radius $\rho\,(T_J) = 1$ so the Jacobi method diverges (and also SSOR). The conjugate gradient method also fails since A is nonsymmetric. One could in principle rewrite $Ax = b$ into $A^T Ax = A^T b$, but the drawback is that the condition number increases according to $\kappa_2\left(A^T A\right) = \kappa_2\,(A)^2$, making the number of iterations to grow, see (A.66). Also, if A is very large it may simply be intractable to compute $A^T A$. The DFPM iterations was started with $x\,(0) = (0.8,\,0.2,\,0.1)$ and completes in 43 iterations for which its error $d = ||x^{(k)} - x|| < 10^{-10}$. According to (21),

$$n_{it} \approx 10\frac{\left(\sqrt{\lambda_{\min}} + \sqrt{\lambda_{\max}}\right)^2}{\sqrt{\lambda_{\min}\lambda_{\max}}},$$

the estimated number of iterations is 56, which is thus slightly overestimated. Among several approximations, the factor 10 is just an estimate from that Section. Gauss-Seidel works for this matrix because $\rho\,(T_{GS}) = 0.609$ and it completes in 49 iterations.

B. A matrix with a zero on its diagonal

Consider a matrix with one of its diagonal elements equal to zero:

$$A = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 6 & 1 \\ 1 & 1 & 7 \end{pmatrix}$$

Its eigenvalues are λ =0.0246, 5.3006 and 7.6749. The matrix is nonsymmetric so the conjugate gradient method is not applicable. Both Jacobi and Gauss-Seidel fail since they require the diagonal elements of A to be non-zero. SSOR fails for the same reason. DFPM is fully functional because it only requires that the eigenvalues are positive.

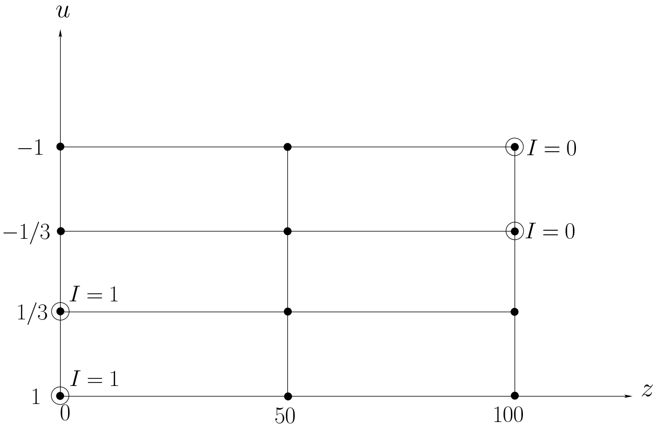

Figure 2: Example discretization grid for the RT equation showing the boundary conditions: $I\,(0,\,-1 \leq u < 0) = 1$ and $I\,(100,\,0 < u \leq 1) = 0$. $N_u = 4$ and $N_z = 3$. Size of matrix is $N_uN_z - N_u = 8$.

C. The RT equation in 2-D

The RT equation in 2-D is an example where several iterative methods encounter problems while it is straightforward to apply DFPM. Consider a plane-parallel laterally homogeneous medium in which light scatters and absorbs. If the single scattering is isotropic and the light intensity is symmetric w.r.t. the azimuthal angle ϕ, the problem is described by the following simplified radiative transfer equation:

$$u\frac{\partial I\,(z,u)}{\partial z} = -\sigma_e I\,(z,u) + \frac{\sigma_s}{2}\int_{-1}^{1} I\,(z,u')\,du', \tag{37}$$

where I is light intensity, z is depth, $u = \cos\theta$ (with θ denoting polar angle) and σ_s and σ_a are scattering and absorption coefficients respectively. The extinction coefficient σ_e is given by $\sigma_e = \sigma_a + \sigma_s$. We assign numerical values to the model parameter as $\sigma_s = 0.1$ and $\sigma_a = 0.001$, which are relevant values from an application point of view [37, 38]. The light incident from above the medium at $z = 0$ is isotropic and there is no light incident from below at $z = 100$. This gives the following boundary conditions: $I\,(0,\,-1 \leq u < 0) = 1$ and $I\,(100,\,0 < u \leq 1) = 0$. See also Fig. 2.

We approximate the derivative with standard forward and backward differences at the boundaries and otherwise central differences. The integral is approximated by the trapezoidal formula. The resulting discretized equations then take the form $AI = c$, where I now being the discretized version of the light intensity. Improved results can conveniently be obtained through Richardson extrapolations. An example discretization grid is shown in Fig. 2. The corresponding matrix A and vector c are shown below where we have used the abbreviations $a = -h_zh_u\sigma_s/2$, $b = 2h_z\sigma_E - h_zh_u\sigma_s/2$ and $u_k = 1 - (k-1)\,h_u$ with $h_{z,u}$ being the step length in depth and polar angle, respectively. Due to the integral term in (37), the matrix A is rather dense for any discretization. It is also nonsymmetric. The eigenvalues of A are in general complex valued. DFPM performs well also in this case. (An analysis of the case with complex eigenvalues is provided in Appendix IV.)

$$AI = \begin{pmatrix} b & u_1 & 2a & 0 & 0 & 2a & 0 & a \\ -u_1 & u_1 + b/2 & 0 & a & 0 & 0 & 0 & 0 \\ a & 0 & a+b & u_2 & 0 & 2a & 0 & a \\ 0 & a/2 & -u_2 & u_2 + (a+b)/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -u_3 + (a+b)/2 & u_3 & a/2 & 0 \\ a & 0 & 2a & 0 & -u_3 & a+b & 0 & a \\ 0 & 0 & 0 & 0 & a & 0 & -u_4 + b/2 & u_4 \\ a & 0 & 2a & 0 & 0 & 2a & -u_4 & b \end{pmatrix} \begin{pmatrix} I_{21} \\ I_{31} \\ I_{22} \\ I_{32} \\ I_{13} \\ I_{23} \\ I_{14} \\ I_{24} \end{pmatrix} = \begin{pmatrix} u_1 \\ 0 \\ u_2 \\ 0 \\ -3/2a \\ 0 \\ -3/2a \\ 0 \end{pmatrix} = c$$

In Table I we show results for various discretizations N_u and N_z in order to test that consistent continuum results can be obtained. Extrapolations w.r.t. h_z are made by assuming a truncation series $k_1h_z + k_2h_z^2$ which is consistent with the truncation errors of the first derivatives. Extrapolation to $\hat{I}(0,1)$ is made by assuming that the angular truncation error is $\mathcal{O}\left(h_u^2\right)$, as is expected for the trapezoidal formula. We see in Table I that the final

Table I: Richardson extrapolations for the discretized radiative transfer equation. A certain point $(z, u) = (0, 1)$ is here evaluated for various discretizations and compared with DORT [39]. Sizes of the matrices range from 320 to 5120.

N_u	N_z	$I(0, 1; N_z; N_u)$	$\hat{I}(0, 1; N_u)$
20	17	0.744070859223879	-
20	33	0.742813581335503	-
20	65	0.742493226091208	0.742378393313508
20	129	0.742415747675200	0.742393402063285
N_u	N_z	$I(0, 1; N_z; N_u)$	$\hat{I}(0, 1; N_u)$
40	17	0.743926268731036	-
40	33	0.742670673372421	-
40	65	0.742352291858317	0.742240187787682
40	129	0.742276182465662	0.742255460649272
N_u	-	$\hat{I}(0, 1; N_u)$	$\hat{I}(0, 1)$
20	-	0.742393402063285	-
40	-	0.742255460649272	0.74221(2)
DORT	-	-	0.742212

extrapolated value converges to continuum and agrees well with a computer code based on the DORT method [39]. Depending on the desired accuracy, a satisfactory result can be obtained without extrapolations. For example, already at $N_u = 20$ and $N_z = 33$ the relative error is less than 0.1 percent.

Table II: Number of iterations to convergence ($d = 10^{-10}$).

N_u	N_z	matrix size	DFPM	Jacobi	G-S
16	9	128	604	751	379
32	17	512	1220	∞	∞
64	33	2048	2475	∞	∞
128	65	8192	5058	∞	∞

In Table II we show the convergence behavior of some iterative methods that at least in principle could work for this problem (the conjugate gradient method does not work due to the nonsymmetry of the matrix). It is seen that DFPM converges for all matrix sizes while the other methods do not. The reason is that both of them have a spectral radius exceeding one as the matrix size n increases. In DFPM, there is no limitation due to spectral radius. It is evident from Table II that the number of iterations in DFPM is $\mathcal{O}\left(n^{1/2}\right)$. If the sparsity of A is disregarded the cost is $\mathcal{O}\left(n^2\right)$ for each iteration, thus giving the total time complexity $\mathcal{O}\left(n^{5/2}\right)$. As a comparison, a similar non-sparse treatment using Gaussian elimination gives the standard result $(2/3)\, n^3$. Gaussian elimination can be improved by taking advantage of the sparsity of A. In such a case a result for the time complexity for an $n \times n$ matrix having m non-zero elements is $\mathcal{O}\left(n^3/\log n\right)$ [40, 41]. This is not sufficient to overcome DFPM's $\mathcal{O}\left(n^{5/2}\right)$ as n increases. Moreover, DFPM's complexity $\mathcal{O}\left(n^{5/2}\right)$ would improve into $\mathcal{O}\left(n^{3/2}\right)$ if we also in this method take advantage of the sparsity of A. We conclude that DFPM is a very efficient choice for solving the RT equation.

D. The Poisson equation in 3-D

Consider the three-dimensional Poisson equation

$$\frac{\partial^2 U\left(x,y,z\right)}{\partial x^2} + \frac{\partial^2 U\left(x,y,z\right)}{\partial y^2} + \frac{\partial^2 U\left(x,y,z\right)}{\partial z^2} = -\frac{\rho\left(x,y,z\right)}{\varepsilon_0} \tag{38}$$

where $(x, y, z) \in [0, 1] \times [0, 1] \times [0, 1]$ and $U = 0$ on the boundaries. We consider the special case that $\rho\left(x,y,z\right) = -\varepsilon_0 \sin\left(\pi x\right) \sin\left(\pi y\right) \sin\left(\pi z\right)$. We discretize the derivatives using central differences, and write the resulting equation system as $AU = b$. Using Kronecker products the matrix A can be computed from

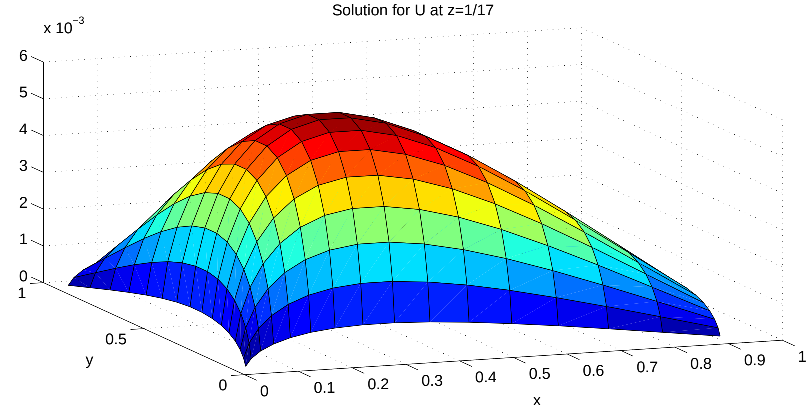

Figure 3: Example solution for U at z=1/17. Mesh is 16 × 16 × 16. Boundary points are not included in the plotted solution surface. Size of matrix A is 4096.

$$A = \frac{1}{h^2}\left(T \otimes I \otimes I + I \otimes T \otimes I + I \otimes I \otimes T\right), \tag{39}$$

where I is the identity matrix and

$$T = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}.$$

The components of the vector b are given by $b_{ijk} = h^2\left(-\rho\left(x_i, y_j, z_k\right)/\varepsilon_0\right)$. The meshes of the unit cubes are $2 \times 2 \times 2$, $4 \times 4 \times 4$, $8 \times 8 \times 8$ and $16 \times 16 \times 16$ (not including the boundary points). In Fig. 3 the solution $U\left(x, y, 1/17\right)$ is plotted for the mesh $16 \times 16 \times 16$.

In Table III we show the number of iterations needed to solve the Poisson equation for some relevant methods. It is immediately seen that neither Jacobi nor Gauss-Seidel is competitive. The number of iterations needed for the conjugate gradient method is taken from (A.67). In the case of an exact arithmetic this number will be lower. However, it is known that in practice with limited arithmetic and for very large matrices the inequality (A.66) may fail. The number of flops per iteration is lower for DFPM so a CPU time comparison could have been more favorable. Nevertheless, in line with the previous analysis, we conclude that DFPM performs in parity with the conjugate gradient method. It can also be noted in Table III that the convergence of the various methods depends approximately as expected on the ratios $\lambda_{\max}/\lambda_{\min}$ and $\sqrt{\lambda_{\max}/\lambda_{\min}}$. Given a matrix size n, the number of iterations for Jacobi and Gauss-Seidel scale as $\mathcal{O}\left(n^{2/3}\right)$, while DFPM and the conjugate gradient method scale as $\mathcal{O}\left(n^{1/3}\right)$. The total computational complexity for the 3-D problem becomes $\mathcal{O}\left(n \times n^{2/3}\right)$ and $\mathcal{O}\left(n \times n^{1/3}\right)$ respectively, since the matrix A is sparse. This is consistent with known complexity results of general dimensionality D, i.e., complexity $\mathcal{O}\left(n \times n^{1/D}\right)$ for the conjugate gradient method [42].

Table III: Number of iterations to convergence ($d = 10^{-10}$).

matrix size	$\lambda_{\max}/\lambda_{\min}$	Jacobi	G-S	DFPM	Conj. grad. (A.67)
8	3	36	19	21	15
64	9.47	122	62	43	30
512	32.16	429	216	84	56
4096	116.46	1610	806	167	108

Table IV: Iterations of the conjugate gradient method versus DFPM. The eigenvalues were computed using the DFPM method in [16]. A large part of this computation can be avoided, see text.

k	C-G	DFPM	$5.1\sqrt{\lambda_{max}/\lambda_{min}}$	λ_{min}	λ_{max}
10	244	249	249	1.126	2695
12	298	302	302	1.124	3943
14	364	365	365	1.123	5772
16	444	442	442	1.123	8449
18	541	534	535	1.122	12369
20	659	647	647	1.122	18107
22	804	782	784	1.121	26509
24	980	946	948	1.121	38811
26	1194	1145	1148	1.121	56820
28	1454	1386	1389	1.121	83189
30	1772	1677	1681	1.121	121795

E. A case for which $\lambda_{\max} \gg \lambda_{\min}$

Here we consider a symmetric matrix as an example where $\lambda_{\max} \gg \lambda_{\min}$. The matrix is from a s-limit three-particle Hamiltonian. Its eigenvalues are all the so called S-states, i.e., ${}^1S$, ${}^3S$ etc. so the lowest eigenvalue is the groundstate ${}^1S$ and the largest is determined by the discretization, i.e., $\lambda_{\max} = \infty$ in the continuum limit. We shall not focus on the eigenvalue problem here, but instead use this matrix to setup an appropriate $Au = b$ problem in order to study the properties of DFPM for the case $\lambda_{\max} \gg \lambda_{\min}$. Since the eigenvalues of the Hamiltonian matrix represent both bound and free states there are both negative and positive eigenvalues. We therefore introduce a constant shift δ in the diagonal elements, thus making all eigenvalues positive. Accordingly, the eigenvalue problem for the shifted Hamiltonian for the s-limit case is [43]:

$$\hat{A}v\left(r_1, r_2\right) = \left[-\frac{1}{2}\frac{\partial^2}{\partial r_1^2} - \frac{1}{2}\frac{\partial^2}{\partial r_2^2} - \frac{2}{r_1} - \frac{2}{r_2} + \frac{1}{max\left(r_1, r_2\right)} + \delta\right] v\left(r_1, r_2\right) = \lambda v\left(r_1, r_2\right)$$

The boundary conditions are given by $v\left(r_1, 0\right) = v\left(0, r_2\right) = v\left(R, r_2\right) = v\left(r_1, R\right) = 0$ $(R = 15)$. The discretization is made by using central differences with equidistant mesh sizes $h = 0.1/1.1^k$ (for both r_1 and r_2), where k is an integer selected to get different problem sizes (see Table IV). Let us now setup a $\hat{A}u = b$ problem and discretize as follows

$$\hat{A}u_{ij} = -\frac{1}{2}\frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij}}{h^2} - \frac{2u_{ij}}{r_{1i}} - \frac{2u_{ij}}{r_{2j}} + \frac{u_{ij}}{max\left(r_{1i}, r_{2j}\right)} + \delta u_{ij} = b_{ij}$$

The "force" acting on a particle p_{ij} at the position (r_{1i}, r_{2j}) is given by $\mathcal{F}_{ij} = b_{ij} - \hat{A}u_{ij}$. Note that the matrix A is not explicitly derived. In fact, it is often convenient to avoid an explicit matrix formulation. Also note that this formulation is automatically sparse. We apply a constant mass $\eta = 1$ and the shift was selected as $\delta = 4$. The damping μ and time step Δt are given by (14, 15). A single C-code was written where the only difference was whether a function call was made to DFPM or to the conjugate gradient method [42]. All tests were performed on a Linux PC with 3 GB primary memory and the CPU was a Intel core2duo 2.4GHz. The compiler used were gcc version 3.4.4. The C-code were compiled by using the optimization: '-O'. This benchmark is shown in Table IV. The first column shows k which determines the discretization h as mentioned before. In the next two columns we show the number of iterations needed to achieve the desired accuracy (here $d = 10^{-6}$). As can be seen, the performance of DFPM is improved for the larger problems. This trend is expected to continue. Secondly, the cost per iteration is smaller for DFPM so its advantage is actually larger than that indicated by the Table IV. As can be seen in the fourth column the number of DFPM iterations is closely proportional to $\sqrt{\lambda_{max}/\lambda_{min}}$ as was derived theoretically in (22). The exact proportionality coefficient depends on the required accuracy d (here 10^{-6}) and how far from the solution one starts. The last two columns show the eigenvalues λ_{min} and λ_{max}. Note that the lowest eigenvalue is hardly affected by the discretization ($\lambda_{min}\left(h\right) = \lambda_{min}\left(0\right) + \mathcal{O}\left(h^2\right)$). It is easy to show that the maximum eigenvalue depends on the mesh size h according to: $\lambda_{max} = \mathcal{O}\left(1/h^2\right)$ which is due to the central difference formula applied (free state). The proportionality constant is given by 4. It can therefore be concluded that it is non-problematic to get very good estimates for both λ_{min} and λ_{max}. Thus it is straight forward to compute optimal DFPM parameters given by (14, 15). The overall computational complexity for DFPM is $\mathcal{O}\left(n^{\frac{3}{2}}\right)$ in this 2-D example. It can be concluded that DFPM is an excellent method for these type of problems as they grow large.

VII. CONCLUSION

DFPM is clearly a convenient and robust method to solve equations. The approach is strongly interdisciplinary since it applies ideas from physics but operates in the field of numerical mathematics. The basic idea of DFPM as a dynamical system may also be attractive from a user's perspective since the algorithm is physically intuitive and very pedagogical. The convergence of the DFPM oscillators in time is closely exponential since they are optimized to be nearly critically damped. The method is not sensitive with respect to a spectral radius or the structure of the given matrix. Its computational complexity is as good as that of the conjugate gradient method. DFPM as concept, is much more general than the particular study performed here. The DFPM algorithm remains essentially the same also for e.g. nonlinear problems. The idea must therefore be considered quite versatile. The efficiency of DFPM can of course, just as for all other methods, be further optimized by applying preconditioning of the matrix. Optimization w.r.t. variation of the DFPM parameters during the iterations has not been studied here. We believe that this is probably more advantageous in the case of non-linear problems. A future parallelization requirement (e.g. using MPI) is expected to be as straight forward as it is for e.g. molecular dynamics. The possibility to improve the method further, i.e., run all oscillators with critically damped parameters requires that the inflexibility imposed by a global time step and damping parameter is lifted. This is left for a future study.

Acknowledgment

Financial support from Vinnova (The Swedish Governmental Agency for Innovation Systems) and the Swedish Knowledge foundation is greatly acknowledged.

[1] S. Li, W. K. Liu, Meshfree particle methods, Springer-Verlag, 2004.
[2] S. Li, W. K. Liu, Meshfree and particle methods and their applications, Appl. Mech. Rev. 55 (2002) 1.
[3] T. R. Quinn, S. Tremaine, M. Duncan, A three million year integration of the Earth's orbit, Astron. J. 101 (1991) 2287.
[4] S. Edvardsson, K. G. Karlsson, Astronomy & Astrophysics 384 (2002) 689.
[5] S. Edvardsson, K. Karlsson, Astron. J. 135 (2008) 1151.
[6] P. A. Cundall, O. D. L. Strack, Discrete numerical-model for granular assemblies, Geotechnique 29 (1979) 47.
[7] K. Kadau, `http://www.thp.uni-duisburg.de/~kai/index_1.html`.
[8] L. B. Lucy, Astronomical Journal 82 (1977) 1013.
[9] R. Gingold, J. J. Monaghan, MNRAS 181 (1977) 375.
[10] W. G. Hoover, Smooth particle applied mechanics, Advanced Series in Nonlinear Dynamics, Vol. 25, World Scientific Publishing Co., 2006.
[11] G. Wang, et al., Hybrid lattice particle modelling approach for polymeric materials subject to high strain rate loads, Polymers 2 (2010) 3.
[12] R. Car, M. Parrinello, Phys. Rev. Lett. 55 (1985) 2471.
[13] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, J. D. Joannopoulos, Rev. Mod. Phys. 64 (1992) 1045.
[14] O. Etzmuss, J. Gross, W. Strasser, IEEE Trans. Visualization and Computer Graphics 9 (2003) 538.
[15] B. A. Lloyd, G. Szekely, M. Harders, IEEE Trans. Visualization and Computer Graphics 13 (2007) 1081.
[16] S. Edvardsson, M. Gulliksson, J. Persson, The dynamical functional particle method: An approach for boundary value problems, J. Appl. Mech. 79 (2012) 021012.
[17] M. Gulliksson, S. Edvardsson, A. Lind, The dynamical functional particle method, `http://arxiv.org/pdf/1303.5317v2.pdf` (2012).
[18] R. Sincovec, N. Madsen, Software for nonlinear partial differential equations, ACM Trans. Math. Softw. 1 (1975) 232–260.
[19] U. Ascher, H. Huang, K. van den Doel, Artificial time integration, BIT 47 (2007) 3–25.
[20] V. Pata, M. Squassina, On the strongly damped wave equation, Commun. Math. Phys. 253 (2005) 511–533.
[21] F. Alvarez, On the minimization property of a second order dissipative system in Hilbert spaces, SIAM J. Control Optim. 38 (2000) 1102–1119.
[22] M. T. Chu, On the continuous realization of iterative processes, SIAM Review 30 (3) (1988) 375–387.
[23] M. T. Chu, Numerical linear algebra algorithms as dynamical systems, Acta Numerica (2008) 1–86.
[24] C.-C. Tsai, C.-S. Liu, W.-C. Yeih, Fictious time integration method of fundamental solutions with chebyshev polynomials for solving Poisson-type nonlinear pdes, CMES 56 (2) (2010) 131–151.
[25] E. L. Allgower, K. Georg, Numerical Continuation Methods: An Introduction, Springer, New York, 1990.
[26] J. Nocedal, S. Wright, Numerical Optimization, Springer, New York, 1999.
[27] H. Goldstein, Classical Mechanics, 2nd ed., Addison-Wesley Publishing Company, 1980.
[28] B. Leimkuhler, S. Reich, Simulating hamiltonian dynamics, Cambridge university press, 2004.
[29] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration, 2nd ed., Springer, 2006.

[30] A. Cromer, Stable solutions using the Euler approximation, Am. J. Phys. 49 (1981) 455.
[31] J. M. Sanz-Serna, M. P. Calvo, Numerical Hamiltonian Problems, Chapman & Hall, 1994.
[32] http://en.wikipedia.org/wiki/Verlet_integration.
[33] D. Kleppner, R. Kolenkow, An introduction to mechanics, 7th ed., McGraw-Hill Book Co., 1986.
[34] G. H. Golub, C. F. V. Loan, Matrix computations, JHU Press, 1996.
[35] Y. Saad, Iterative methods for sparse linear systems, 2nd ed., Society for Industrial and applied mathematics, 2003.
[36] G. Moore, Computational linear algebra, pp. 46, http://www2.imperial.ac.uk/~gmoore/M3N4/iter09.pdf.
[37] M. Neuman, L. G. Coppel, P. Edström, Opt. Express 19 (2011) 1915.
[38] M. Neuman, L. G. Coppel, P. Edström, Nord. Pulp Pap. Res. J. 27 (2012) 426.
[39] P. Edström, SIAM Rev. 47 (2005) 447.
[40] M. Bomhoff, Computer Science - Theory and Applications, Springer Verlag, Vol. 6651, pp. 443, 2011.
[41] J. P. Spinrad, Discrete Appl. Math. 138 (2004) 203.
[42] J. R. Shewchuk, http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf.
[43] S. Edvardsson, D. Aberg, P. Uddholm, Comp. Phys. Commun. 165 (2005) 260.
[44] R. L. Burden, J. D. Faires, Numerical Analysis, fourth edition, PWS-KENT Publishing company, pp. 402-405, 1989.
[45] P. G. Ciarlet, Introduction to numerical linear algebra and optimization, Press syndicate of the university of Cambridge, pp. 174-175, 1989.
[46] S. Brakken-Thal, http://buzzard.ups.edu/courses/2007spring/projects/brakkenthal-paper.pdf.

Appendix

I. Estimation of DFPM parameters

According to Section B the optimal parameters are related to the eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$. These are in general not known in advance. The Section B suggests a strategy where these values are determined approximately before DFPM starts. Through iterative techniques the min/max eigenvalues can be approximated by using just a few iterations, see e.g. [34] for various possibilities. Precise eigenvalues are not needed to get good performance from DFPM. However, if determination of eigenvalues are considered to be too awkward to compute for a particular problem, we provide below some reasonable damping and time step parameters through approximations. These suggested DFPM parameters are all very simple to calculate.

The case that $\lambda_{\min} \sim \lambda_{\max}$

We shall here estimate the DFPM parameters by using least squares optimization. This treatment cannot guarantee that all oscillators fulfill $\mu + \lambda_i \Delta t \leq 2\sqrt{\eta\lambda_i}$ so some oscillators may become slightly overdamped. To proceed, let us optimize the parameters by minimizing the square distances

$$\sum_{i=1}^{n} \left(\mu + \lambda_i \Delta t - 2\sqrt{\eta\lambda_i}\right)^2 \tag{40}$$

Differentiating w.r.t. Δt and μ gives that

$$\Delta t = 2\sqrt{\eta}\frac{\left\langle\lambda^{3/2}\right\rangle - \left\langle\lambda^{1/2}\right\rangle\left\langle\lambda\right\rangle}{\left\langle\lambda^{2}\right\rangle - \left\langle\lambda\right\rangle^{2}} \tag{41}$$

$$\mu = 2\sqrt{\eta}\left\langle\lambda^{1/2}\right\rangle - \Delta t\left\langle\lambda\right\rangle \tag{42}$$

where the symbol $\langle . \rangle$ is the arithmetic mean. The result (A.41) is not useful here while (A.42) is. The mean of the eigenvalues is directly given by $\langle\lambda\rangle = tr\left(A\right)/n$, i.e., $tr\left(A\right)$ is the trace of the matrix. An estimate for $\left\langle\lambda^{1/2}\right\rangle$ can be derived either by means of a Taylor expansion of $\left(\langle\lambda\rangle + (\lambda - \langle\lambda\rangle)\right)^{1/2}$ or through Hölder's inequality:

$$\sum_{i=1}^{n} \|a_i b_i\| \leq \left(\sum_{i=1}^{n} \|a_i\|^p\right)^{1/p} \left(\sum_{i=1}^{n} \|b_i\|^q\right)^{1/q}$$

By taking $p = q = 2$ and $a_i = \lambda_i^{1/2}$ and $b_i = 1$ we find that

$$\left\langle \lambda^{1/2} \right\rangle \lesssim \langle \lambda \rangle^{1/2} = \sqrt{tr\,(A)\,/n} \tag{43}$$

A comparison with (15) suggests the following estimate for the time step

$$\Delta t \approx \sqrt{\eta} / \left\langle \lambda^{1/2} \right\rangle \approx \sqrt{\eta} / \langle \lambda \rangle^{1/2} = \sqrt{\eta} / \sqrt{tr\,(A)\,/n} \tag{44}$$

so according to (A.42) the damping parameter becomes

$$\mu \approx \sqrt{\eta} \sqrt{tr\,(A)\,/n} \tag{45}$$

A differential analysis of (A.42) together with the inequality (A.43) shows that the damping estimate (A.45) is overestimated. One can thus expect an improved convergence rate if one actually selects a slightly lower value. According to (14) and (15) the best ratio is given by $\mu_{\text{opt}}/\Delta t_{\text{opt}} = \sqrt{\lambda_{\min}\lambda_{\max}}$, while (A.44) and (A.45) gives that $\mu/\Delta t = \langle \lambda \rangle$, i.e., the geometric mean of $\lambda_{\min}$ and $\lambda_{\max}$ has been replaced by the arithmetic mean of all the eigenvalues. The approximation is appropriate as long as $\lambda_{\min}$ is similar in size as $\lambda_{\max}$.

The case that $\lambda_{\min}$ is small relative to $\lambda_{\max}$

If we have the interesting case for which $\lambda_{\min} \ll \lambda_{\max}$, the approximations in the previous Section will not be good. One possibility is then to recognize that (15) is approximated into $\Delta t = 2\sqrt{\eta}/\left(\sqrt{\lambda_{\max}}\right)$. Then one could use that for any norm, $\lambda_{\max} \lesssim \|A\|$ [34]. The most convenient norm is the ∞-norm so we get

$$\Delta t \approx \frac{2\sqrt{\eta}}{\sqrt{\|A\|_\infty}} \tag{46}$$

This time step is somewhat underestimated than the optimal. The damping parameter becomes according to (A.42, A.43)

$$\mu \approx 2\sqrt{\eta} \langle \lambda \rangle^{1/2} - \Delta t \langle \lambda \rangle \tag{47}$$

where $\langle \lambda \rangle = tr\,(A)\,/n$.

Matrix A is close to the identity matrix

Let us apply the following properties of norms [34]

$$\begin{cases} \lambda_{\max} \leq \|A\| \\ \lambda_{\min} \geq \frac{1}{\|A^{-1}\|} \end{cases} \tag{48}$$

According to (14) and (15) we have that $\mu_{\text{opt}}/\Delta t_{\text{opt}} = \sqrt{\lambda_{\min}\lambda_{\max}}$. By taking the simplest norm, i.e., the ∞-norm, a reasonable estimation of this ratio becomes

$$\frac{\mu_{\text{opt}}}{\Delta t_{\text{opt}}} \approx \sqrt{\frac{\|A\|_\infty}{\|A^{-1}\|_\infty}} \tag{49}$$

Unfortunately A^{-1} is unknown. Let us therefore introduce a new matrix B given by $A = I - B$, where I is the identity matrix. Then we write the following geometric series

$$A^{-1} = (I - B)^{-1} = \sum_{k=0}^{\infty} B^k \tag{50}$$

which is allowed if $\|B\| < 1$. Thus, if A in a certain application is such that $\|B\| < 1$, this becomes useful. If $\|B\|$ is small it might be sufficient with:

$$A^{-1} \approx I + B$$

so that

$$\frac{\mu_{\text{opt}}}{\Delta t_{\text{opt}}} \approx \sqrt{\frac{\|I - B\|_{\infty}}{\|I + B\|_{\infty}}} \tag{51}$$

Then the suggested time step becomes according to (15)

$$\Delta t_{\text{opt}} \approx \frac{2\sqrt{\eta}}{1/\sqrt{\|I + B\|_{\infty}} + \sqrt{\|I - B\|_{\infty}}} \tag{52}$$

and μ_{opt} is obtained from (A.51). One might consider to replace $I + B$ above with e.g. $I + B + B^2$ in order to get a better estimate but the computational cost may be too high for a large matrix B.

II. Integrating the first order system with a second order algorithm

Here we consider the second order Runge-Kutta method. A greater time step could thus be possible and the approximation that the evolution in discrete time follows (33) should be improved. In this case we have that $g(w(t)) = f(w(t) + (\Delta t/2) f(w(t)))$, where for a exponential $f(w(t)) = -(\lambda/\mu) w$, see (26) and (28). We find that

$$g(w(t)) = -\frac{\lambda}{\mu} w(t) - \frac{\lambda}{2\mu} w(t) \Delta t$$

so the numerical algorithm according to (26) reads

$$w(t + \Delta t) = w(t) + \Delta t \left(-\frac{\lambda}{\mu} w(t) \right) + \Delta t \left(-\frac{\lambda}{2\mu} w(t) \Delta t \right) \tag{53}$$

Again at $t = n\Delta t$, this is simplified into

$$w(t) = \left(1 - \frac{\lambda}{\mu} \Delta t - \frac{\lambda}{2\mu} \Delta t^2 \right)^n w(0) \tag{54}$$

The optimal time step is in this case given by

$$\Delta t_{\text{opt}} = \sqrt{1 + \frac{2\mu}{\lambda}} - 1 \tag{55}$$

and the maximum time step for a given exponential is

$$\Delta t_{\max} = \sqrt{1 + \frac{4\mu}{\lambda}} - 1 \tag{56}$$

Just as before among the group of exponentials, $\lambda = \lambda_{\max}$ determines the maximum time step. Also as before ($\alpha_{\min} = -\alpha_{\max}$), one finds that

$$\Delta t_{\text{opt}} = \sqrt{1 + \frac{4\mu}{\lambda_{\min} + \lambda_{\max}}} - 1 \tag{57}$$

The inequality $\Delta t_{\text{opt}} < \Delta t_{\max} = \sqrt{1 + (4\mu/\lambda_{\max})} - 1$ is clearly fulfilled so convergence is guaranteed. The analysis can now proceed in much the same way as above. Apply the relation $T \approx n_{it} \Delta t_{\text{opt}}$ and equation (34) which then gives that

$$n_{it} \approx \frac{\mu}{\lambda_{\min}\left(\sqrt{1 + \frac{4\mu}{\lambda_{\min}+\lambda_{\max}}} - 1\right)} \log\left(\frac{\|w(0)\|}{d}\right) \tag{58}$$

Note that the damping parameter μ can not be eliminated in this expression as it could before. Consider as before the case that $\lambda_{\max} \gg \lambda_{\min}$ and apply a Taylor expansion. We then get

$$\sqrt{1 + \frac{4\mu}{\lambda_{\min} + \lambda_{\max}}} \approx \sqrt{1 + \frac{4\mu}{\lambda_{\max}}} \approx 1 + \frac{2\mu}{\lambda_{\max}}$$

In such a case, (A.57) now gives that $\Delta t_{\text{opt}} \approx 2\mu/\lambda_{\max}$ and we find that

$$n_{it} \approx 10 \frac{\lambda_{\max}}{\lambda_{\min}} \tag{59}$$

which is the same result as (36). The higher order integration algorithm thus gives no apparent advantage and it has a higher cost/iteration. The dimensionality dependence between n_{it} and the eigenvalues remains unchanged.

III. Comparisons with other iterative methods

Stationary iterative methods - Jacobi iterations

The iteration formula of these methods are given by

$$x^{(k)} = Tx^{(k-1)} + c$$

, where c is a constant vector. The convergence can be studied through

$$d = ||x^{(k)} - x|| \approx \rho(T)^k \, ||x^{(0)} - x|| \tag{60}$$

, where $\rho(T)$ is the spectral radius of the iteration matrix T, i.e., $max\,|\lambda(T)|$. As seen in [44], the Jacobi and Gauss-Seidel iterations matrices are $T_J = D^{-1}(L+U)$ and $T_{GS} = (D-L)^{-1}U$, respectively. We have that $A = D - L - U$, where the matrix D is the diagonal of A, L and U are the strictly lower and upper triangular matrices of A, see [44]. From (A.60) the number of iterations n_{it} becomes

$$n_{it} \approx -\frac{\log\left(\|x^{(0)} - x\| / d\right)}{\log(\rho(T))} \tag{61}$$

where $\rho(T) < 1$ is required, see (A.60). We need to rewrite (A.61) in order to be able compare with the convergence results of the present work. We first consider the Jacobi iterative method. Its iteration matrix can be expressed as $T_J = I - D^{-1}A$, where I is the identity matrix, so $|\lambda(T_J)| = \left|\lambda\left(I - D^{-1}A\right)\right| = \left|\lambda(I) - \lambda\left(D^{-1}A\right)\right| = \left|1 - \lambda\left(D^{-1}A\right)\right| \approx \left|1 - \langle D\rangle^{-1}\lambda(A)\right|$. The last approximation is reasonable if the diagonal elements in D^{-1} are similar in size (it is exact if $D^{-1} = \mu I$, where μ is a constant scalar). Finally we set $\left|1 - \langle D\rangle^{-1} \lambda(A)\right| = |1 - \lambda(A) / \langle\lambda(A)\rangle|$ because $\langle D\rangle^{-1} \equiv (tr(A)/n)^{-1} = 1/\langle\lambda(A)\rangle$. Since $\rho(T_J) = max|\lambda(T_J)| \approx max|1 - \lambda(A)/\langle\lambda(A)\rangle| < 1$, one possibility corresponds to $1 - \lambda_{\min}(A)/\langle\lambda(A)\rangle$. The following approximate formula can then be used

$$n_{it} \approx -\frac{\log\left(\left\|x^{(0)} - x\right\|/d\right)}{\log\left(1 - \lambda_{\min}(A)/\langle\lambda(A)\rangle\right)} \approx \frac{\log\left(\left\|x^{(0)} - x\right\|/d\right)}{\lambda_{\min}(A)/\langle\lambda(A)\rangle} \tag{62}$$

The last approximation is through Taylor expansion. Consider a reasonably homogenous eigenvalue spectrum, then $\langle\lambda(A)\rangle \approx (\lambda_{\min}(A) + \lambda_{\max}(A))/2$. As before, by taking the convergence criterion $d = 10^{-10}$, one finds that $\log\left(\left\|x^{(0)} - x\right\|/d\right) \approx 20$, almost independent of $\left\|x^{(0)} - x\right\|$. We arrive at the result

$$n_{it} \approx 10\left(1 + \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}\right) \sim 10\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \tag{63}$$

The explicit use of (A.61) gives of course a much more accurate result, but as stated above, a comparison with our other convergence results is valuable. The Jacobi result (A.63) is seen to be just similar in performance as the first order system's result (36) which is not sufficiently competitive.

It should be pointed out that there is a second possibility for $max|\lambda(T_J)|$ than that studied above, namely the possibility that $\lambda(A)/\langle\lambda(A)\rangle \sim 2$. In order to be brief, this case is not given in detail. That possibility just leads to the same result as already given by (A.63).

Stationary iterative methods - Gauss-Seidel iterations

For the case of the Gauss-Seidel method there is no simple way to make a similar assessment as above. No general result exists telling which of the methods will be more efficient for an arbitrary system $Ax = b$. In many practical cases, however, it is known that $\rho(T_{GS}) < \rho(T_J)$ rendering the Gauss-Seidel method more efficient [44]. For example, in the case that A is positive definite and tri-diagonal, block diagonal or similar it has been shown that $\rho(T_{GS}) = \rho(T_J)^2 < 1$ [45]. Eq. (A.61) then tells us that only half of the number of iterations is needed for the Gauss-Seidel method. Thus also the Gauss-Seidel method performs similarly as the first order dynamical system (36).

Stationary iterative methods - SSOR iterations

Symmetric successive over relaxation (SSOR) is a popular iterative method that combines a forward and backward iteration of SOR [35]. The spectral radius of SSOR is given by [36]

$$\rho_{SSOR} \lesssim \frac{1 - \sqrt{(1 - \rho(T_J))/2}}{1 + \sqrt{(1 - \rho(T_J))/2}} \tag{64}$$

which is valid provided that $\rho(T_J) \leq 1$. According to the previous Jacobi Section, $\rho(T_J) \approx 1 - \lambda_{\min}(A)/\langle\lambda(A)\rangle \approx 1 - 2\lambda_{\min}(A)/\lambda_{\max}(A)$ so

$$\rho_{SSOR} \lesssim \frac{1 - \sqrt{\lambda_{\min}(A)/\lambda_{\max}(A)}}{1 + \sqrt{\lambda_{\min}(A)/\lambda_{\max}(A)}} \approx 1 - 2\sqrt{\frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}} \tag{65}$$

and thus according to (A.61)

$$n_{it} \lesssim -\frac{\log\left(\left\|x^{(0)} - x\right\|/d\right)}{\log\left(1 - 2\sqrt{\frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}}\right)} \approx \frac{\log\left(\left\|x^{(0)} - x\right\|/d\right)}{2\sqrt{\frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}}} \approx 10\sqrt{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}}$$

where it has been assumed as before that $\log\left(\left\|x^{(0)} - x\right\|/d\right) \approx 20$. This approximate result is competitive because it is the same as DFPM, see (22).

Krylov subspace methods - the conjugate gradient method

The conjugate gradient method is a popular iterative method commonly referred as very efficient for solving sparse linear systems whose matrix is positive definite, symmetric and diagonal dominant [42]. In this method the convergence can be studied through (pp. 215 in [35])

$$d = ||x^{(k)} - x||_A \lesssim 2\left(\frac{\sqrt{\kappa_2}-1}{\sqrt{\kappa_2}+1}\right)^k ||x^{(0)} - x||_A \tag{66}$$

where $\kappa_2 = \lambda_{\max}(A)/\lambda_{\min}(A)$ is the euclidean condition number and $||x||_A = (Ax, x)^{1/2}$. From (A.61) the number of iterations n_{it} becomes

$$n_{it} \lesssim -\frac{\log\left(2\left\|x^{(0)} - x\right\|_A/d\right)}{\log\left(\frac{\sqrt{\kappa_2}-1}{\sqrt{\kappa_2}+1}\right)} = -\frac{\log\left(2\left\|x^{(0)} - x\right\|_A/d\right)}{\log\left(1-\frac{2}{\sqrt{\kappa_2}+1}\right)} \tag{67}$$

If $\lambda_{\max} \gg \lambda_{\min}$, Taylor expansion gives that $\log\left(1-\frac{2}{\sqrt{\kappa_2}+1}\right) \approx -\frac{2}{\sqrt{\kappa_2}+1} \approx -\frac{2}{\sqrt{\kappa_2}} = -2/\left(\sqrt{\lambda_{\max}(A)/\lambda_{\min}(A)}\right)$. By taking the convergence criterion $d = 10^{-10}$, one finds that $\log\left(2\left\|x^{(0)} - x\right\|_A/d\right) \approx 20$. The final estimate is the familiar

$$n_{it} \lesssim 10\sqrt{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}} \tag{68}$$

This is again as good as the DFPM result (22).

The steepest gradient method

The steepest gradient method is another method which is similar to the conjugate gradient method except that the $\sqrt{\kappa_2}$ in (A.66) is replaced by κ_2, see pp. 215 [35]. In the same way as above, this then lead to the less good result

$$n_{it} \lesssim 10\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \tag{69}$$

Krylov subspace methods - GMRES

There are other Krylov subspace methods that can handle more general matrices than the conjugate gradient method. One such example is the generalized minimal residual method (GMRES) which is capable of handling non-symmetric problems. According to pp. 216 [35] the convergence is given by

$$d \lesssim \left(1 - \frac{\lambda_{\min}\left(A^T + A\right)}{2\lambda_{\max}\left(A^T + A\right)}\right)^{n/2} ||x^{(0)} - x|| \tag{70}$$

For simplicity to get an estimate, consider a symmetric matrix A: $\lambda\left(A^T + A\right) = \lambda(2A) = 2\lambda(A)$, so

$$d \lesssim \left(1 - \frac{\lambda_{\min}(A)}{2\lambda_{\max}(A)}\right)^{n/2} ||x^{(0)} - x|| \tag{71}$$

Similarly as above after Taylor expansion one find

$$n_{it} \lesssim 4\log\left(2\left\|x^{(0)} - x\right\|/d\right)\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \approx 80\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \tag{72}$$

which indicates that GMRES does not appear to be competitive for the case of interest here: $\lambda_{\max} \gg \lambda_{\min}$.

IV. Generalization of DFPM

Negative eigenvalues

If the matrix A only has negative eigenvalues the same DFPM algorithm can in principle be reused. By inspecting (8), it is seen that if $\lambda < 0$ all that is needed is to multiply each equation with -1, so $\eta \to -\eta$ and $\mu \to -\mu$. The easiest update of the algorithm (6) to accomplish this is to change the sign of the functional $\mathcal{F}$. Thus let $b - Ax \to Ax - b$ in (6) and the optimal parameters are still given by (14, 15) except the change $\lambda \to |\lambda|$.

The sign of the eigenvalues can often be mathematically motivated by inspecting the structure of the original operator. Another way to find out for a given matrix whether all $\lambda > 0$ or $\lambda < 0$ is to apply Gershgorin's Theorem: $|\lambda - A_{ii}| \leq \sum |A_{ij}|$, see the very nice examples in [46]. If a matrix is strictly diagonally dominant and all its diagonal elements are positive, then the real parts of its eigenvalues are positive; if all its diagonal elements are negative, then the real parts of its eigenvalues are negative. These results follow from the Gershgorin circle theorem, see e.g. [34].

Complex eigenvalues

Throughout this work we have assumed that A is real valued. However, eigenvalues may still be complex valued. If so, it is well known in linear algebra that they appear in pairs, i.e., $\lambda_1, \lambda_1^*, \lambda_2, \lambda_2^*, \lambda_3, \lambda_3^*, .., \lambda_{n/2}, \lambda_{n/2}^*$, see (8). Equation (11) is still valid but a new analysis is needed. Let us therefore consider (11) again

$$\alpha_{1,2} = 1 - \frac{\lambda}{2\eta}\Delta t^2 - \frac{\mu}{2\eta}\Delta t \pm \frac{\Delta t}{2\eta}\sqrt{\zeta} = U \pm W \tag{73}$$

where $\zeta = (\mu + \lambda \Delta t)^2 - 4\eta\lambda$. Since $\lambda \in \mathbb{C}$, both U and W are complex in general. A critical condition for convergence is that $|\alpha_{1,2}| < 1$. These two eigenvalues are given by $|\alpha_{1,2}| = |U \pm W|$. However, this expression is too intractable to analyze in detail. A much simpler way is to instead look at $|U \pm W| \leq |U| + |W|$. If $|U| + |W| \leq 1$ convergence is ensured. To get an efficient method we need to identify $min\left(|U| + |W|\right)$. The problem becomes very much simplified if we recognize that, in any case, a good thing is to minimize $|U|$. We note that there is no real valued Δt and μ that renders $U = 0$. Nevertheless let us find its minimum by considering

$$|U|^2 = UU^* = \left(1 - \frac{\lambda}{2\eta}\Delta t^2 - \frac{\mu}{2\eta}\Delta t\right)\left(1 - \frac{\lambda^*}{2\eta}\Delta t^2 - \frac{\mu}{2\eta}\Delta t\right)$$

$$= \left(\lambda \Delta t^2 + \mu \Delta t - 2\eta\right)\left(\lambda^* \Delta t^2 + \mu \Delta t - 2\eta\right)/4\eta$$

By differentiation w.r.t. μ one finds the condition that

$$Re\left(\lambda\right)\Delta t^2 + \mu \Delta t - 2\eta = 0$$

This is all good but it is only optimum for one oscillator. It cannot be fulfilled for all the oscillators. In order to find the global optimum we therefore suggest minimization through the following least square expression:

$$\sum_{i=1}^{n}\left(Re\left(\lambda_i\right)\Delta t^2 + \mu \Delta t - 2\eta\right)^2 \tag{74}$$

Differentiation w.r.t. μ then gives that

$$\Delta t^2 \frac{1}{n}\sum_{i=1}^{n} Re\left(\lambda_i\right) + \mu \Delta t - 2\eta = 0$$

Since the eigenvalues are real valued or comes in pairs with complex conjugates we have that $\sum_{i=1}^{n} Im\left(\lambda_i\right) = 0$. We thus conclude that $\sum_{i=1}^{n} Re\left(\lambda_i\right) = tr\left(A\right)$. We find that a good damping of the system is given by

$$\mu = \frac{2\eta - \left(tr\left(A\right)/n\right)\Delta t^2}{\Delta t} \tag{75}$$

The second part $|W|^2 = WW^* = \left(\Delta t/2\eta\right)|\zeta|$ is minimized by minimizing $\zeta = \left(\mu + \lambda\Delta t\right)^2 - 4\eta\lambda$. Let us consider the possibility that $\zeta = 0$, i.e., $\mu = \pm 2\sqrt{\eta\,|\lambda|}e^{i\varphi/2} - |\lambda|\,e^{i\varphi}\Delta t$. Since μ is real this is only possible if the imaginary part of this equation becomes zero. This gives that $0 = \pm 2\sqrt{\eta\,|\lambda|}\sin\left(\varphi/2\right) - |\lambda|\sin\left(\varphi\right)\Delta t$. For a single oscillator we find that the time step is

$$\Delta t = 2\sqrt{\frac{\eta}{|\lambda|}}\left|\frac{\sin\left(\varphi/2\right)}{\sin\left(\varphi\right)}\right| \tag{76}$$

For a pair of oscillators with λ and $\lambda^*$the same time step is obtained for both. The question is what happens when we have the full system of oscillators? From (A.73) and (A.75) we have that

$$U = \left(\Delta t^2/2\eta\right)\left(k - \lambda\right) \tag{77}$$

, where $k = \left(tr\left(A\right)/n\right)$. From (A.77) we see that $|U|$ is zero only when $\Delta t = 0$ and otherwise increasing so we have to turn to $|W|$ to identify an optimal time step. Equations (A.73), (A.75) and (A.77) gives that

$$W = \left(1 + U^2 - \frac{2k}{k-\lambda}U\right)^{1/2} = \left(\left(U - \frac{k}{k-\lambda}\right)^2 - \left(\frac{k}{k-\lambda}\right)^2 + 1\right)^{1/2} \tag{78}$$

Unfortunately there is no explicit formula for Δt that minimizes $|U \pm W|$. However, one can plot $|U \pm W|$ versus Δt in order to find the optimum Δt or alternatively use an appropriate numerical optimization method to identify the optimal Δt.

In order to at least get some educated guess we ask the question: for which $\lambda_j \in \{\lambda\}$ is $|U|$ largest? Consider $UU^*/\left(\Delta t^2/2\eta\right)^2 = \left(k - |\lambda_j|\,e^{i\varphi}\right)\left(k - |\lambda_j|\,e^{-i\varphi}\right) = k^2 - 2k\,|\lambda_j|\cos\varphi + |\lambda_j|^2 \leq \left(k + |\lambda_j|\right)^2$. Thus we have

$$|U| = \frac{\Delta t^2}{2\eta}\sqrt{k^2 - 2k\,|\lambda_j|\cos\varphi + |\lambda_j|^2} \leq \frac{\Delta t^2}{2\eta}\left(k + |\lambda_j|\right) \tag{79}$$

which need to be fulfilled for all oscillators $j = 1,\, 2, .., \, n$. We therefore have that

$$\frac{\Delta t^2}{2\eta}\left(k + \max\left(|\lambda|\right)\right) < 1 \tag{80}$$