

PAPER

An efficient implicit time integration method for discrete dislocation dynamics

To cite this article: Gábor Péterffy and Péter Dusán Ispánovity 2020 *Modelling Simul. Mater. Sci. Eng.* **28** 035013

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

An efficient implicit time integration method for discrete dislocation dynamics

Gábor Péterffy  and Péter Dusán Ispánovity 

Department of Materials Physics, Eötvös Loránd University, Pázmány Péter sétány 1/A, H-1117, Budapest, Hungary

E-mail: pgabor@caesar.elte.hu

Received 27 September 2019, revised 15 January 2020

Accepted for publication 14 February 2020

Published 5 March 2020



CrossMark

Abstract

Plastic deformation of most crystalline materials is due to the motion of lattice dislocations. Therefore, the simulation of the interaction and dynamics of these defects has become state-of-the-art method to study work hardening, size effects, creep and many other mechanical properties of metallic specimens. Lot of efforts have been made to make the simulations realistic by including specific dislocation mechanisms and the effect of free surfaces. However, less attention has been devoted to the numerical scheme that is used to solve the equations of motion. In this paper we propose a scheme that speeds up simulations by several orders of magnitude. The scheme is implicit because this type is the most efficient one for solving stiff equations that arise due to the long-range nature of dislocation interactions. The numerical results show that the method is not only faster than other approaches at the same numerical precision, but it can also be efficiently applied even without dislocation annihilation. The suggested method significantly increases the achievable volume and/or duration of discrete dislocation dynamics simulations and can be generalized for complex 2D and 3D simulations as well.

Keywords: dislocation dynamics, numerical simulation, implicit, efficient, numerical method, integration, discrete dislocation dynamics

(Some figures may appear in colour only in the online journal)

1. Introduction

Although plastic deformation of crystalline materials is seemingly a smooth process on macroscopic scales, on the microscopic level it is characterized by intermittent local strain bursts [1, 2]. The reason for this behavior is that plastic deformation is the result of the motion of individual dislocations. Due to their long-range stress fields and complex short-range

interactions these line-like defects can entangle forming a rigid network of dislocation lines. Upon external stress dislocations may locally disentangle and become mobile resulting in the accumulation of plastic strain. This and many other processes involving collective dislocation motion can be efficiently modeled by discrete dislocation dynamics (DDD) simulations [3]. They were successfully applied to various problems of materials science, such as describing the role of multi-junctions [4], delivering a general picture of size effects [5] or to understand the statistics of strain burst sizes [6], just to mention a few.

Several different kind of simulators are in use. The most important difference among them is the dimension of the simulation cell. Three-dimensional (3D) approaches aim at modeling the full 3D dislocation microstructure and its dynamics. Due to the topological difficulties these simulations are numerically rather complex. Examples of such 3D DDD simulations include ParaDiS [4], algorithm of Weygand [7], microMegas [8], and Parametric Dislocation Dynamics [9]. These simulations are used when the role of specific 3D mechanisms are investigated [10–12] or when one aims at a quantitative comparison with experimental results [13, 14]. A much simpler representation of the dislocation systems is delivered by two-dimensional (2D) simulations which only consider parallel straight dislocations. These are primarily used in cases when the physical consequences of long range dislocation interactions are investigated and when large simulation volumes, large ensembles for statistical averaging, longer timescales and/or higher numerical precision is required. Due to the simplification mentioned above a quantitative agreement with experiments cannot be expected, yet, these tools have been successfully applied to investigate, e.g. creep [15, 16], dislocation avalanches [17–19] and patterning [20, 21] of dislocations. An intermediate class is represented by 2.5D simulations, which are essentially 2D but with the inclusion of some 3D mechanisms, such as dislocation multiplication or dislocation pinning [22–28].

As it will be discussed in this paper the arising differential equation system that describes the dynamics of dislocations or dislocation segments is stiff. In such cases explicit methods are inefficient, because the achievable time-step is limited by the shortest timescale in the system, which is determined by the shortest dislocation dipole. This typically leads to a very slow propagation of the simulations even when the system is stationary. Nonetheless, most of the simulators mentioned above employ explicit methods [23, 29–31]. This issue can be solved by implicit schemes [30]. Sills *et al* showed that with the help of the simple implicit trapezoidal method and the Newton–Raphson nonlinear solver, in certain scenarios a speedup is achievable compared to the default Heun-method of ParaDiS if a sparse matrix was used. This was achieved by taking into account only short range elastic interactions in the Jacobian, defined with a fixed limit on the distance [30]. Gardner *et al* applied diagonally implicit Runge–Kutta methods and concluded that with high number of dislocation segments the gain in the stepsize compared to explicit methods was not large enough to compensate for the increase in the time needed to calculate one step [32]. Both of these studies were done using ParaDiS or DDLab (which is the serial version of ParaDiS written in MATLAB).

In this paper we present an efficient implicit integration scheme that can speed up simulations with several orders of magnitude while the numerical precision is the same or even higher than in case of explicit methods. The proposed scheme is capable of handling short dislocation dipoles efficiently that are responsible for the slowing down when using explicit methods. Another important property of the proposed scheme is that it can be tuned by a scale parameter which makes it possible to increase the efficiency of the simulator in different regimes of activity.

The method is tested on 2D edge dislocation systems because it is conceptually one of the simplest dislocation models. This simplicity makes it easier to compare the performance

of different methods more accurately and precisely. However, the proposed method can be applied in more complicated systems or in simulations with higher dimension as well.

The structure of the paper is as follows: first, the 2D dislocation model and its background is introduced in section 2. Then, in section 3 the numerical difficulties arising from the properties of dislocation stress fields are discussed in detail followed by demonstrating the difference in performance between an explicit and an implicit method on the case of a single dislocation dipole. In section 4, the new numerical method is presented and it is also explained how the previously highlighted features (better precision, larger stepsize, etc) can be achieved. Then, a short insight is given into the actual implementation in section 5 followed by the presentation of the numerical results on the efficiency in section 6. Section 7 presents a summary of the results and section 8 concludes the paper with an outlook on possible applications of the new scheme.

2. Discrete dislocation dynamics

2.1. Model description

In the model considered, only straight and parallel edge dislocations are present with parallel slip planes. This system is effectively 2D since it is enough to track the dislocation positions on a plane perpendicular to the dislocation lines. Let the x axis of the 2D coordinate system be parallel with the Burgers vector \mathbf{b} of the dislocations. The mechanical shear stress field of an individual dislocation in this case can be described with [33]

$$\tau_{\text{ind}}(x, y) = \frac{\mu b}{2\pi(1 - \nu)} \frac{x(x^2 - y^2)}{(x^2 + y^2)^2}, \quad (1)$$

where μ and ν are material dependent elastic parameters whereas x and y are the distances from the dislocation along the corresponding axes.

For the motion of dislocations usually a linear velocity-force relationship is assumed. This corresponds to overdamped dynamics and is argued to be valid due to strong phonon drag acting on dislocations [33]. The equations of motion in this case are

$$\begin{aligned} \dot{x}_i(t) &= f_i(x_1(t), \dots, x_N(t), y_1, \dots, y_N) \\ &= B^{-1}s_i b \left(\sum_{j=1, j \neq i}^N s_j \tau_{\text{ind}}(x_i(t) - x_j(t), y_i - y_j) + \tau_{\text{ext}} \right), \end{aligned} \quad (2)$$

$$\dot{y}_i(t) = 0, \quad (3)$$

where B is the dislocation drag coefficient and the indices i and j refer to the N different dislocations. Symbol s is equal to 1 or -1 depending on the direction of the Burgers vector. The Burgers vector's length b is the same for all dislocations. τ_{ext} represents external loading or any internal static stress field. For simplicity, in the current paper the focus is on the integration scheme, so, $\tau_{\text{ext}} = 0$ is considered, but the generalization to $\tau_{\text{ext}} \neq 0$ is rather straightforward. A square-shaped simulation area with periodic boundary conditions is employed in order to mimic an infinite medium. It is noted, that periodic dislocation images can be taken into account by using a modified stress field instead of τ_{ind} [34]. This model is not affected by the problem of conditional convergence, according to [35] neither constant nor linear terms arise in such a model.

This model is one of the simplest representations of a complex dislocation network. Due to its assumptions, it is incapable for investigating many deformation phenomena related to, e.g. dislocation curvature or multiple slip. Its advantage is, on the other side, that it is

conceptually much simpler than its 3D counterparts which makes numerical investigations faster and more reliable leading to the possibility to study larger systems and larger statistical ensembles at once with preserving high numerical precision. Another advantage is, opposed to 3D simulators where the same simulation can lead to different results based on loading and other factors [30, 32], that the results in this case are reproducible and consistent. Consequently, several fundamental properties related to the long-range internal stresses of dislocations systems have already been successfully investigated with this model [36–40].

2.2. Dimensionless variables

In the rest of this paper, in accordance with the $1/r$ -type scale-free interaction τ_{ind} , distance, stress and time inside the simulation are measured in units of $\rho^{-1/2}$, $\frac{\mu b \rho^{1/2}}{2\pi(1-\nu)}$ and $\frac{2\pi(1-\nu)B}{\mu b^2 \rho}$, respectively, where $\rho = N/L^2$ is the total dislocation density and L is the linear size of the simulation area. This is possible because, due to the absence of annihilation or other dislocation reactions with an associated length parameter, there is no length scale in the system other than the system size and average dislocation spacing. With this choice one does not have to specify the system size and/or material-dependent parameters making the results more general.

2.3. Relaxation simulations

In the simulations we consider N dislocations with zero net Burgers vector, i.e. $\sum_{i=1}^N s_i = 0$. The size of the simulation cell in the dimensionless units is then $\sqrt{N} \times \sqrt{N}$. Both x and y components of the initial coordinates of the dislocations are independent uniformly distributed random variables in the interval $[0, \sqrt{N}]$ (figure 1(a)). Then the equations of motion (2, 3) are solved with zero external stress ($\tau_{\text{ext}} = 0$). This leads to an initially rapid motion of dislocations which later, due to the strong dissipation introduced by the overdamped dynamics, slows down as the system gradually approaches an equilibrium state. During this process strong spatial correlations in the dislocation coordinates build up, corresponding to the build-up of low energy dislocation structures like dislocation dipoles and dislocation walls (figures 1(b) and (c)) [36]. The relaxation itself is a slow, scale-free process, and the relaxation time strongly increases with the number of dislocations considered [41], and so does time needed to run the simulations.

3. Numerical difficulties

3.1. Long-range interactions

Due to the $1/r$ -type long-range stress field τ_{ind} one cannot apply a cut-off in the mutual interactions. So, all the terms in the sum of equation (2) have to be taken into account irrespective of the relative distance of the dislocations. Introduction of a cut-off is known to lead to the appearance of artificial dislocation patterns with a length scale proportional with the cut-off radius [42].

The time complexity of every timestep is, therefore, $\mathcal{O}(N^2)$. In practice this means that if the linear extension of a simulation cell size is doubled (i.e. the number of dislocations is increased by a factor of 4) than computing a single timestep lasts 16 times longer. This extreme increase in computational cost makes investigation of large configurations rather difficult.

One possible solution to overcome this difficulty is the fast multipole method first introduced in 3D DDD simulations [43–45], then also used successfully in the present 2D set-up [34, 46]. This method reduces the time complexity to approximately $\mathcal{O}(N \ln(N))$. In this

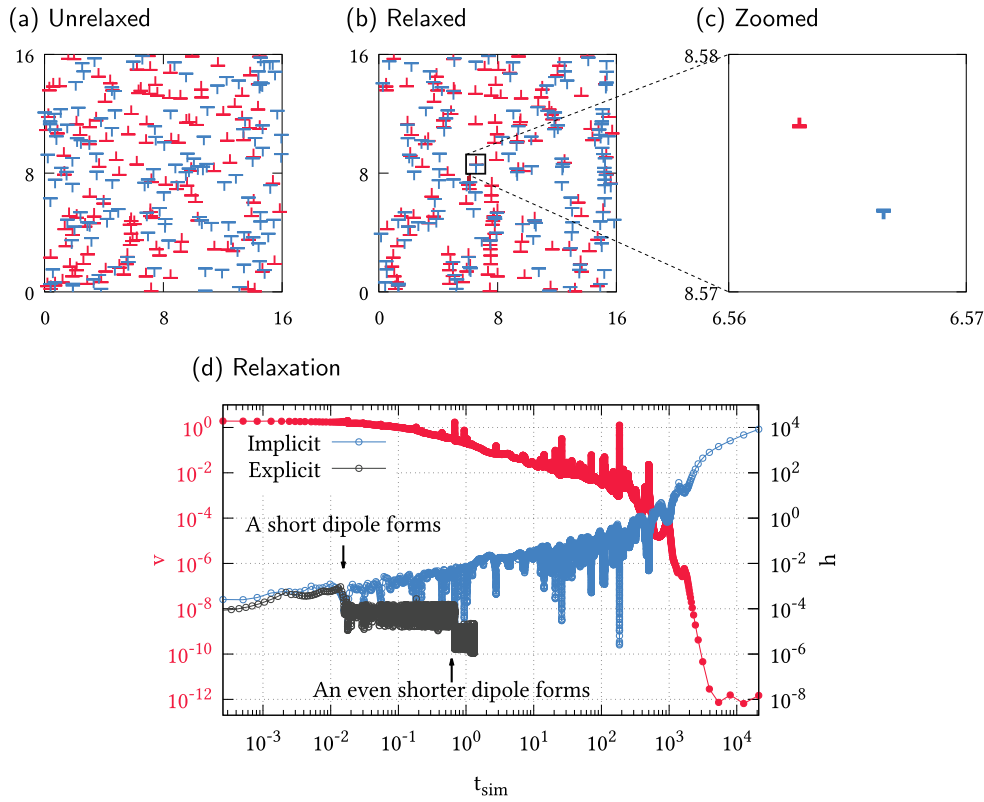


Figure 1. Relaxation of an initially random dislocation configuration with $N = 256$ dislocations. (a) The initial random configuration. (b) The final relaxed configuration. (c) The relaxed configuration are characterized by spatial correlations. For instance, if opposite sign dislocations get close to each other they form a dipole as illustrated in the zoomed figure. (d) Average dislocation speed v (left scale, red color) is shown as a function of simulation time t_{sim} . The relaxation is characterized by strong velocity fluctuations which eventually cease and the average velocity drops with 12 orders of magnitude. The numerical stepsizes h (right scale) in case of an explicit 4.5th order Runge–Kutta (RK45) and the weighted implicit trapezoid scheme (WITS) method are also plotted. Whereas the timestep gradually increases during the relaxation for the implicit scheme, for the explicit case it is influenced only by the shortest dislocation dipole in the system. (It is noted, that the datapoints for the explicit method are restricted to small t_{sim} values because of the significantly increased costs of computation.)

paper we do not address this issue any further and will use the exact summation of equation (2) in the simulations.

3.2. Stiff equations: limitations on time step

The previous section was about how to evaluate the sum in equation (2). Now we continue with the numerical scheme that is needed to perform the integration of equation (2). The average velocity of the dislocations can be used to monitor the activity of a system during a simulation. In figure 1(d) the average velocity

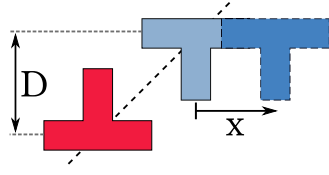


Figure 2. A dislocation dipole that is used for the stability analysis of the different numerical methods. D represents the size of the dipole while x denotes the distance from the equilibrium position.

$$v(t) = \frac{1}{N} \sum_{i=1}^N |v_i(t)| \quad (4)$$

is seen during the course of a representative simulation run with the full implicit weighted trapezoid method (abbreviated as WITS, described later in section 4.3). As it is seen, dislocation activity strongly fluctuates but gradually ceases as the system approaches equilibrium. Although the average velocity has dropped in this case around 12 orders of magnitude which is close to the precision of the used double data type, it does never reach absolute zero due to some numerical noise always present in the system (the level of which can be controlled by the tolerance parameter of the adaptive stepsize routine down to the used architecture's limit).

One of the most common methods for dislocation simulations is the 4.5th order explicit Runge–Kutta scheme with adaptive stepsize control (abbreviated as RK45). This method is an explicit scheme (thus, relatively easy to implement) and its popularity in solving a system of ordinary differential equations is due to its high accuracy and relatively low computational cost of performing a single timestep [47].

In figure 1(d) time step sizes as a function of the simulation time for both mentioned methods are also shown. In the beginning of the simulation the actual time step sizes for the RK45 method vary due to the adaptive stepsize control (being smaller in active periods and larger in quiescent regions) but with the formation of a short dipole it quite quickly settles at a quite low approximately constant value. At $t_{\text{sim}} \approx 0.6$ an even shorter dipole forms, so the timestep drops further and remains at the same value for the rest of the simulation. On the other hand, the timestep of the implicit method increases throughout of the simulation run with a total increase of more than 7 orders of magnitude. While it cannot be seen in the figure, the timestep is strongly affected by the tolerance parameter: better precision (lower tolerance) decreases the timestep and leads to longer simulation runs in real time.

The peculiar behavior of the timestep shown above is due to the *stiffness* of the governing equations. This term applies to systems of ordinary differential equations where the timescales of the parallel processes are on a broad scale. The timestep of explicit methods follows the smallest timescale in the system, so for stiff systems the usage of implicit methods is advised. In the case of dislocations the stiffness is due to the $1/r$ -type long-range interactions, since relaxation is very fast for nearby dislocations and slow down with increasing mutual distance. In the next section this will be quantified for a single dislocation dipole.

3.2.1. Demonstration on a dislocation dipole. To understand the basic reason for the stepsize behavior observed above and the difference between the explicit and the implicit numerical methods applied to the problem, a small dipole and its dynamics is considered. Such dipoles form in a large number during the simulations, and ones with small distances have the highest

probability [48]. The considered setup can be seen in figure 2: the distance of the glide planes is D , and one of the dislocations is slightly shifted from the equilibrium 45° position.

After the Taylor-expansion of equation (2) for this dipole, in the $|x| \ll D$ limit one concludes for the top dislocation that

$$\dot{x} = -\frac{1}{D^2}x. \quad (5)$$

Here the dimensionless units of section 2.2 are used and it is noted that similar equation holds for the bottom dislocation as well. The solution of this equation is an exponential relaxation to $x = 0$ with time constant $\tau = D^2$, that is, the relaxation timescale tends to zero quadratically for small dislocation dipoles¹.

The different numerical integration schemes that can be used to solve equation (5) can be categorized as explicit or implicit methods. They can be sorted by inspecting the equation

$$\frac{x^{k+1} - x^k}{h} = F(x^{k+1}, x^k, x^{k-1}, \dots, t^k), \quad (6)$$

where x^k approximates the solution $x(t^k)$ at $t^k = kh$, and h is the timestep considered. If F does not depend on x^{k+1} then the method is an explicit method. Typical examples are the 4th order Runge–Kutta method or the simplest, forward Euler-method. If F is dependent upon x^{k+1} it is an implicit method and usually a system of nonlinear equations has to be solved at every time step. In the following we investigate how the simplest explicit (Forward Euler) and implicit (Backward Euler) methods handle the above introduced perturbed dipole.

Forward Euler method The right-hand side of equation (5) is here evaluated at timestep k , i.e. $F(x^{k+1}, x^k, x^{k-1}, \dots, t^k) = -x^k/D^2$. By rearranging equation (6) one obtains

$$x^{k+1} = x^k \left(1 - \frac{h}{D^2} \right). \quad (7)$$

It is seen, that if the time step h is larger than $h_{\text{crit}} = 2D^2 = 2\tau$ then instead of converging to $x = 0$ the dislocation will diverge in an oscillatory way (that is, $|x^{k+1}| > |x^k|$ for $h > h_{\text{crit}}$). This happens irrespective of how close the dislocation is to the equilibrium position.

Backward Euler method In contrast, $F(x^{k+1}, x^k, x^{k-1}, \dots, t^k) = -x^{k+1}/D^2$ is assumed when the backward Euler formula is used, and one gets

$$x^{k+1} = \frac{x_i^k}{1 + \frac{h}{D^2}}. \quad (8)$$

This means there is no limitation for h in terms of convergence, dislocations will approach the equilibrium position $x = 0$ independently how large h is (that is, $|x^{k+1}| < |x^k|$ for every h). This means this scheme is stable for the given problem.

Comparison of the two methods The forward Euler method is thus seriously limited in stepsize h when a dipole forms, no matter what else happens in the system. The maximum allowed stepsize scales with the D_{min}^2 , where D_{min} is the characteristic distance of the shortest dipole in the system. This is also true, when this particular dipole is already in equilibrium. This can be observed in figure 1 for a more complicated explicit scheme. Hence, the common solution in case of explicit methods is that under a certain threshold these dipoles are annihilated. Although this process also happens in real crystals, in simulations in favor of simulation speed often unrealistically large annihilation distances are applied that may lead to

¹ Note, that throughout the paper the dimensionless units introduced in section 2.2 are used. For instance, equation $\tau = D^2$ in normal units would read as $\tau \frac{\mu b^2}{2\pi(1-\nu)B} = D^2$.

physical side effects. In addition, even with annihilation the stepsize is still limited by the smallest surviving dipole.

On the other hand, these kind of explicit methods have the advantage that there is no need to solve systems of nonlinear equations which is usually necessary for implicit methods. This is a serious drawback for the backward Euler formula, but near equilibrium, theoretically, infinite stepsizes are possible which can compensate for the extra time but it is not as efficient when the system is far from it. Consequently, an *in between* method would be preferred.

4. The applied numerical scheme

4.1. Weighted implicit trapezoidal scheme (WITS)

We start by introducing the so-called WITS which is often referred to as θ -method [49]. When solving the $\dot{x}(t) = f(x(t), t)$ ordinary differential equation this scheme with the notation of equation (6) reads as

$$F(x^{k+1}, x^k, x^{k-1}, \dots, t^k) = \frac{(1-d)f(x^k, t^k) + (1+d)f(x^{k+1}, t^{k+1})}{2}. \quad (9)$$

In the equation d is a weight factor that tunes the system between a backward Euler ($d = 1$) and a forward Euler ($d = -1$) method.

The advantage of the introduction of the weight factor d is that if it is chosen properly it can improve the efficiency of the integration. As it will be seen later, the symmetric trapezoidal rule ($d = 0$) is not optimal in situations where dislocations can *jump over* their equilibrium positions, which may lead to spurious (though non-diverging) oscillations. This nonphysical behavior can be removed by appropriate choice of d (see section 4.3) which ensures energy dissipation. It is noted that a similar approach was applied in the parametric dislocation dynamics simulator [50].

When applying the method to a dislocation system equation (6) with (9) has to be solved for every dislocation. From now on, the lower index of a symbol will identify the particular dislocation and upper index will mark the given time step. After introducing the function

$$g_i(x_1, \dots, x_N) := x_i - \frac{(1+d_i)h}{2} f_i(x_1, \dots, x_N, y_1, \dots, y_N) - x_i^k - \frac{(1-d_i)h}{2} f_i^k, \quad (10)$$

reorganizing the formula and inserting equation (2) one obtains

$$\begin{aligned} g_i^{k+1} &= g_i(x_1^{k+1}, \dots, x_N^{k+1}) = x_i^{k+1} - \frac{(1+d_i)h}{2} f_i^{k+1} - x_i^k - \frac{(1-d_i)h}{2} f_i^k \\ &= x_i^{k+1} - \frac{(1+d_i)h}{2} s_i \sum_{j \neq i}^N s_j \tau_{\text{ind}}(x_i^{k+1} - x_j^{k+1}, y_i^{k+1} - y_j^{k+1}) \\ &\quad - x_i^k - \frac{(1-d_i)h}{2} f_i^k = 0, \end{aligned} \quad (11)$$

which is a system of nonlinear equations that needs to be solved to get the coordinates x_i^{k+1} of the dislocations at the new timestep (the x_i^k , and, thus, f_i^k , values are known from the previous timestep).

4.2. Solving the nonlinear equation system

To solve equation (11) Newton–Raphson iteration method was chosen where we assume that the solution is close to the actual position. Hence a few iterations (in our case 2) are expected to deliver a close enough solution. If this was not the case the adaptive stepsize protocol will correct the error by switching to smaller stepsize. The initial guess for the solution is the coordinate vector from the previous time step

$$x_i^{k+1,(0)} = x_i^k, \quad (12)$$

and an iteration step takes the form

$$\sum_{j=1}^N (x_j^{k+1,(n)} - x_j^{k+1,(n+1)}) J_{ij}^k = g_i^{k+1,(n)}, \quad (13)$$

where $x_i^{k+1,(n)}$ approximates x_i^{k+1} as n increases and J_{ij}^k is the Jacobian matrix and is evaluated at timestep k . The latter reads as

$$J_{ij}^k = \frac{\partial g_i}{\partial x_j}(x_1^k, \dots, x_N^k) = \begin{cases} 1 - \frac{1+d_i}{2} h s_i \sum_{l=1, l \neq i}^N s_j \partial_x \tau_{\text{ind}}(x_i^k - x_l^k, y_i^k - y_l^k), & \text{if } i = j, \\ \frac{1+d_i}{2} h s_i s_j \partial_x \tau_{\text{ind}}(x_i^k - x_j^k, y_i^k - y_j^k), & \text{else.} \end{cases} \quad (14)$$

4.3. Selection of the weight factors

As it was pointed out previously, it is very important to select the right d_i values to avoid oscillations of dislocations around equilibrium positions because that would violate conservation of energy in the system. Firstly, let us consider the case of a dipole of section 3.2.1. Close to equilibrium the equation of motion is approximately

$$\dot{x} \approx -\frac{1}{\tau} x, \quad (15)$$

where $\tau = D^2$. By applying the weighted implicit trapezoid rule on that equation and requiring $x(t) > 0$ for every t if $x(0) > 0$, the ideal weight factor d can be obtained (for details see the [appendix](#)) and reads as

$$d = \frac{1}{1 + \frac{2}{h/\tau}}. \quad (16)$$

According to this formula if the timestep $h \ll \tau$ then $d \approx 0$, so one may use the second order symmetric trapezoidal scheme. If, however, $h \gg \tau$ then $d \approx 1$ is obtained meaning that the lower order backward Euler method must be used to avoid oscillations. This modification to the scheme is not only advantageous numerically, but also motivated by physical arguments. In the equation of motion equation (2) overdamped dynamics are assumed, so, the total energy of the system may only decrease due to the strong dissipation. Any numerical oscillation around an equilibrium position would, in fact, introduce energy to the system, which could lead to unphysical phenomena. With the weight factors introduced this possibility can be avoided in the case of a dipole (for details see the [appendix](#)).

In order to generalize this approach to the system of N dislocations it is useful to define matrix \hat{A}^k as

$$A_{ij}^k = \begin{cases} -hs_i \sum_{l \neq i}^N s_j \partial_x \tau_{\text{ind}}(x_i^k - x_l^k, y_i^k - y_l^k), & \text{if } i = j, \\ hs_i s_j \partial_x \tau_{\text{ind}}(x_i^k - x_j^k, y_i^k - y_j^k), & \text{else.} \end{cases} \quad (17)$$

The d_i^k weights for each dislocation can be then determined as:

$$d_i^k = \begin{cases} \frac{1}{1 + \frac{1}{A_{ii}^k}}, & \text{if } A_{ii}^k > 0, \\ 0, & \text{else.} \end{cases} \quad (18)$$

For a short dipole this exactly reproduces equation (16). Finally, at a given timestep the Jacobian matrix \hat{J}^k (according to equation (14)) can be computed as

$$J_{ij}^k = \delta_{ij} + \frac{1 + d_i^k}{2} A_{ij}^k, \quad (19)$$

where δ_{ij} is the Kronecker delta, which means that complexity of the method is not increased with the introduction of the weight factors d_i^k .

It is noted, that for the weight factors d_i only the diagonal elements of the Jacobian are used motivated by our result obtained for a dislocation dipole. In this case, however, due to the existence of collective modes, oscillations are not fully avoided. As it will be shown in section 6.3 this method still considerably reduces the level of oscillations.

4.4. Reducing the complexity of the implicit method

The time complexity of a single timestep for the explicit integration schemes in DDD simulations is $\mathcal{O}(N^2)$, since all the pair interactions have to be taken into account at every timestep. Implicit methods, as explained above in detail, are expected to perform better due to larger possible stepsizes that can be made with the same numerical precision. Yet, there is a fundamental issue with such schemes that prevent them from being superior to explicit schemes for large system sizes. Namely, the linear system of equation of equation (13) has to be solved several times at every timestep. The complexity of such solution is $\mathcal{O}(N^3)$ because the Jacobian matrix is dense (its components, representing a dislocation pair, decay as r^{-2} with the mutual distance r between the dislocations) which makes this complexity also apply for the whole scheme. This means that there is always a critical system size where the larger obtainable stepsize compared to the explicit one cannot compensate for the longer runtime of a single timestep.

In this section this problem is addressed by the introduction of a special cut-off function during the calculation of the Jacobian matrix. The basic idea is that the stepsize constraint in explicit systems are posed by the shortest dipoles in the system. So, by considering only these short dipoles in an implicit manner would already increase significantly the possible stepsize, similarly to the annihilation procedure applied for explicit schemes. We, therefore, define the cut-off function as

$$c(x, y) = \begin{cases} 1, & \text{if } x^2 + y^2 < r_c^2, \\ \exp\left(-\frac{(\sqrt{x^2 + y^2} - r_c)^2}{r_c^2}\right), & \text{else,} \end{cases} \quad (20)$$

where r_c is a cut-off parameter setting a lengthscale. With this the Jacobian reads as

$$J_{ij}(x_i, \dots, x_N) = \begin{cases} 1 - \frac{1+d_i}{2} h s_i \sum_{l \neq i}^N s_j c(x_i - x_l, y_i - y_l) \partial_x \tau_{\text{ind}}(x_i - x_l, y_i - y_l), & \text{if } i = j, \\ \frac{1+d_i}{2} h s_i s_j c(x_i - x_j, y_i - y_j) \partial_x \tau_{\text{ind}}(x_i - x_j, y_i - y_j), & \text{else.} \end{cases} \quad (21)$$

The exponentially decaying function c makes the values of the Jacobian at a certain distance smaller than the floating-point precision. In the present case if the result of the function c is less than 10^{-16} the corresponding element is replaced by zero. Consequently, the Jacobian becomes a sparse matrix. It can be easily seen that number of non-zero elements is approximately Na , where a does only depend on r_c as long as $r_c \ll L = \sqrt{N}$ holds. To solve a system of equations which is described by a sparse matrix is significantly faster than that of a dense matrix with complexity under $\mathcal{O}(N^2)$ [51], so the complexity of the whole method remains $\mathcal{O}(N^2)$ (since the computation of pair interactions is still necessary at every timestep) just like in case of the explicit methods. As such, with the introduction of the cut-off r_c , the performance ratio between the WITS and explicit methods is not expected to depend on the number of simulated dislocations.

A similar method was used in [30], where elastic interactions between dislocation segments were only considered if they were closer than a specified length parameter (a hard limit), but the lack of continuity in the derivative can cause performance drop because the Newton–Raphson iteration method may not converge if the root is close to or at the discontinuity.

It is clear, that by introducing the cut-off function an artificial analytical error appears in the calculation. Since only the Jacobian is influenced by this modification the guesses of the Newton–Raphson method are affected during the solution and the results of the nonlinear equations become less precise. This will not result in systematic error since the adaptive stepsize control (see below) monitors the introduced error and recomputes a step with decreased timestep if necessary.

To elaborate further on this point it is noted that if the cut-off parameter r_c is 0 the method becomes exactly an explicit method (in particular, with two iteration cycles inside the Newton–Raphson method it becomes the explicit predictor-corrector method), while if r_c approaches infinity it becomes a fully implicit method. So this r_c can be used to *tune* the scheme between explicit and implicit. Typically those dipoles (or dislocation pairs in general) are considered implicitly where dislocations are closer than r_c .

To summarize the above statements, two control parameters are present in the scheme: the weight factors d_i which are fixed by the actual configuration of a dislocation system at a given time and the cut-off distance r_c which can be selected at any given timepoint differently and is not restricted physically.

4.5. Adaptive stepsize control

A simple algorithm was used to control the precision of the solution described by the following procedure: With a given h stepsize a step was calculated, after that the same timepoint from the original timepoint was reached by two $\frac{h}{2}$ sized steps. The difference between the coordinates of the dislocations at $t + h$ was calculated between the two, the largest absolute value of these differences is called the error ε . This is required to be under a given limit ε_{max} (tolerance). If this condition is not fulfilled the results are marked as failed and need to be recalculated. Otherwise, the result from the two smaller steps is stored at the new $t + h$

Table 1. Number of different dislocation configurations for each dislocation system size for the density measurement of the Jacobian.

Number of dislocations N	Number of configurations
64	10 000
256	5000
1024	2000
4096	1000
16 384	100

timepoint. The new stepsize is determined by the following equation in both cases:

$$h_{\text{new}} = \begin{cases} 0.9h \cdot \min(2, \frac{\varepsilon_{\text{max}}}{\varepsilon}), & \text{if } \varepsilon \neq 0 \\ 2h, & \text{else.} \end{cases} \quad (22)$$

5. Implementation

The method described above was implemented in c++ in a strictly non-parallel fashion, but the scheme can be parallelized in an efficient manner and ported to GPU as well. To solve the arising, previously described linear equation systems, the UMFPACK library [52] was used which is used in MATLAB as well to solve systems of linear equations described by a sparse matrix.

During the implementation special care was taken to store the Jacobian directly in sparse format to avoid unnecessary storage operations. This is very efficient if the matrix is indeed sparse, but if it is not the case (e.g. a cut-off is not applied or it is comparable to the system size) it comes with an overhead during element lookup. Hence, with the fully implicit method ($r_c = \infty$) faster results are possible with a storage scheme optimized for dense matrices.

The shear stress field of a dislocation was precalculated with periodic boundary conditions with good resolution on a grid [34]. Near the origin the exact value of equation (1) was calculated but for greater distances linear interpolation between the precalculated values was performed. The $\partial_x \tau_{\text{ind}}$ field needed for the Jacobian was evaluated in a similar manner. This approach outperforms the image based field calculation in case of single thread computing on our available hardware.

A version of the implementation is available on github under GNU General Public License v3.0: <https://github.com/pgabor/sdddst>.

6. Numerical results

6.1. Effect of the cut-off parameter on the denseness of the Jacobian

As it was described in section 4.4 the cut-off parameter reduces the complexity of the problem by calculating the elements of the Jacobian only where the distance between two dislocations is under a certain threshold, thus, the Jacobian matrix becomes a sparse one. In order to quantify its sparseness, we generated random dislocation configurations and the Jacobians were calculated for each one with different cut-off parameters. The actual number of the different configurations are summarized in table 1.

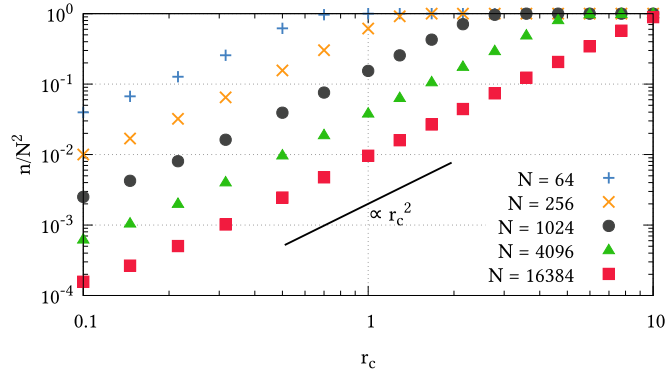


Figure 3. The ratio of the non-zero elements in the Jacobian matrix for different system sizes as a function off the cut-off distance r_c . n stands for the number of the non-zero elements and N is the number of dislocations in the system.

The results can be seen in figure 3 where n denotes the number of non-zero elements in the $N \times N$ sized Jacobian. As explained above in section 4.4 the number of non-zero elements is approximately $n = Na$, where a is the average number of non-zero elements in a row. The latter is obviously proportional with the number of dislocations where the cut-off function c returns a non-zero value, so, $a \propto r_c^2$. This leads to the result $n \propto Nr_c^2$, which can be clearly observed in figure 3. There are two limits to this behavior: (i) for small r_c values n tends to N , since the diagonal elements do not vanish even for $r_c = 0$ (and one arrives at an explicit method as explained previously) and (ii) for large r_c values all matrix elements become non-zero, so n saturates at N^2 .

These results clearly show that with the right choice of r_c a sparse matrix can be obtained that with increasing N becomes even sparser. Hence, the complexity may go below $\mathcal{O}(N^2)$ to solve the linear equation system. In the next section we compare the runtime and the achieved stepsize for different cut-off parameters for dislocation systems of various sizes.

6.2. Effect of the cut-off parameter on the runtime of the simulations

The relaxation of different initially random dislocation systems was performed with 1000 realizations for $N = 256$ dislocations where the tolerance was 1.6×10^{-5} and with 500 realizations for $N = 1024$ dislocations with 3.2×10^{-5} as maximal tolerance. Different r_c parameters were used but r_c was kept fixed during every simulation run. In each case the spent wallclock time (Δt_{real}) and the achieved stepsize (h) was measured between every successful subsequent simulation step and the actual total simulation time (t_{sim}) was also recorded². The resulting data was sampled on an equidistant interval on logarithmic scale and the corresponding data were averaged for the different realizations (same dislocation number N and parameter r_c). If no exact datapoint has been found, linear interpolation was used between the two closest ones. The results are shown in figures 4(a) and (b) for the $N = 256$ and $N = 1024$ case, respectively.

According to the figures, in line with the expectations, Δt_{real} increases significantly with increasing cut-off r_c . In addition, it hardly changes during the course of the relaxation. This means that the sparseness of the Jacobian is nearly constant so it is not much affected by the

² It is noted, that r_c , h and t_{sim} are dimensionless units as introduced in section 2.2, whereas wallclock time Δt_{real} is measured in seconds.

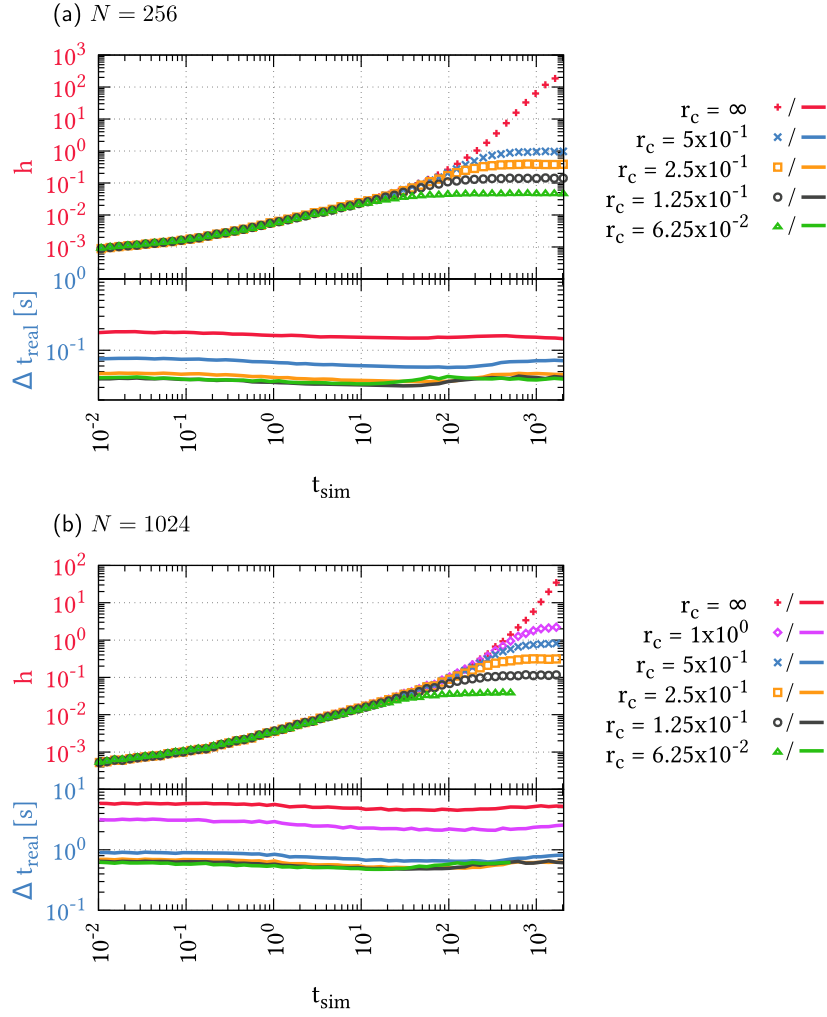


Figure 4. Dependence of the stepsize (h) and the spent wallclock time (Δt_{real}) between to successful timesteps as a function of the simulation time (t_{sim}) for different r_c values and (a) $N = 256$ dislocations and (b) $N = 1024$ dislocations.

formation of local correlated configurations seen in figures 1(b) and (c). This also implies that the results of the previous section conducted for random dislocation ensembles remain approximately valid even after relaxation. On the other hand, the achieved average timestep h increases significantly as the final equilibrium configuration is approached. More specifically, in the first part of the relaxation, where the system exhibits the largest activity, the achieved stepsize does not depend on r_c for the values considered. However, as the activity ceases, the timesteps saturate at values that increase significantly with r_c and even seemingly diverge for $r_c = \infty$.

Based on the previous observations the efficiency, defined as the simulation time advance under unit wallclock time, i.e. $h/\Delta t_{\text{real}}$, was also calculated for both system sizes. According to the plots of figure 5 at the beginning in the active regime, due to the smaller computational cost, the efficiency is better for small r_c values. However, as the system gets closer to

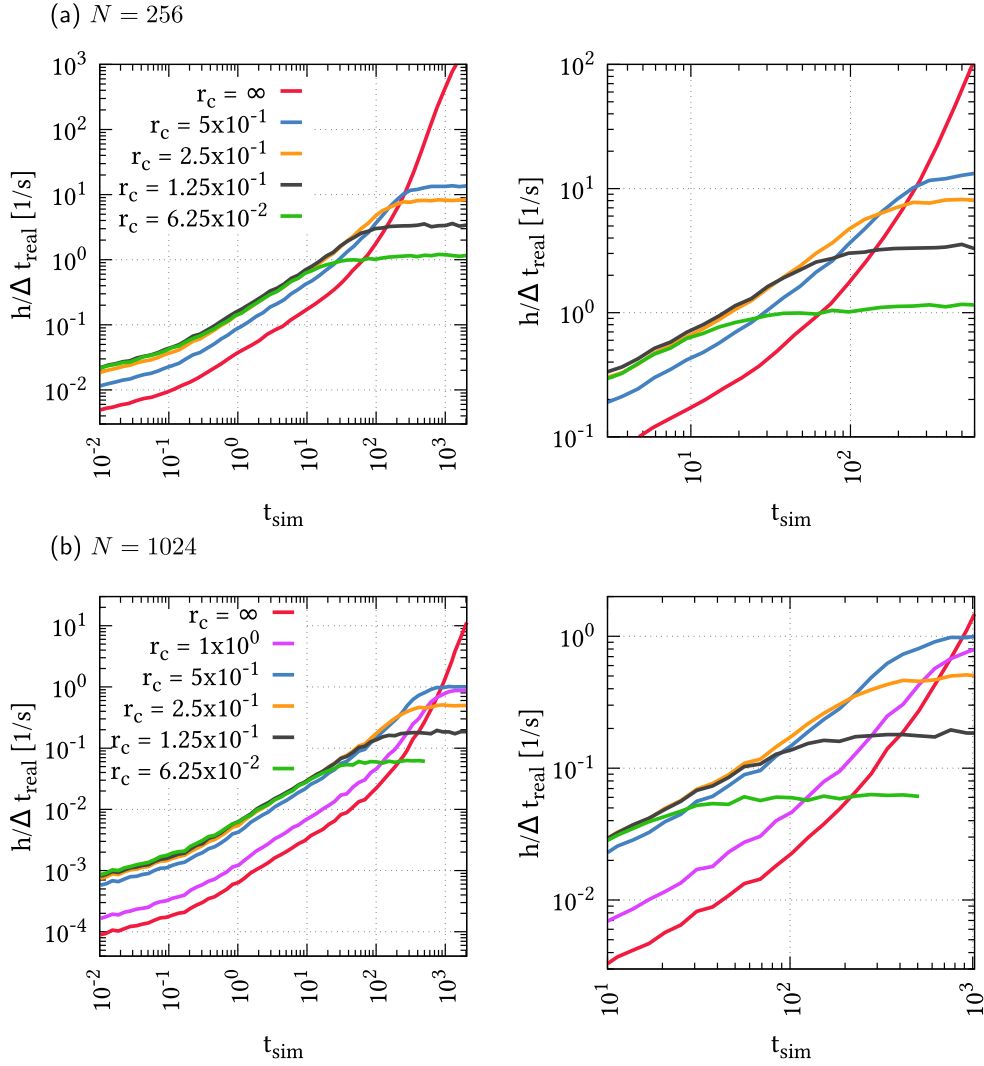


Figure 5. The efficiency $h/\Delta t_{\text{real}}$ of the new method for different r_c values as a function of the simulation time t_{sim} for (a) $N = 256$ and (b) $N = 1024$ dislocations. The left panel plots the whole simulation time interval while the right one magnifies the part where the curves cross each other.

equilibrium a saturation in efficiency is seen and here the efficiency gets better for larger r_c due to the increased timestep. The efficiency without a cut-off ($r_c = \infty$) seems to diverge in equilibrium.

Every considered choice of r_c leads to a significant decrease in total runtime compared to explicit methods. As seen in figure 5, smaller r_c values are more favorable in active (small t_{sim}) regions and larger ones gradually become more efficient as activity ceases (t_{sim} increases). It is interesting to note, that for $N = 1024$ dislocations the choice of $r_c = 1$ is never the most efficient. The reason behind this behavior turned out to be that at this sparsity

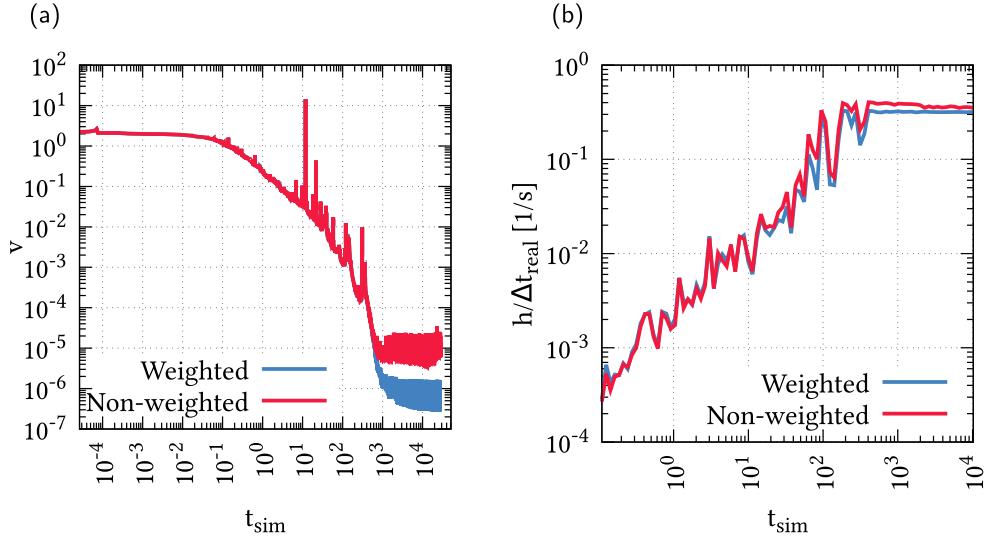


Figure 6. (a) The average dislocation speed as a function of simulation time with the implicit trapezoidal scheme with and without weighting with $r_c = 0.25$ and $N = 1024$. (b) The corresponding efficiency.

level (approx. 20% according to figure 3) the used linear solver (UMFPack) started to use a different kind of strategy than for sparser cases. So, such details of the applied algorithms may also need to be taken into account for choosing the appropriate value for parameter r_c .

6.3. The importance of weighting

In order to decrease numerical oscillations around the equilibrium position of dislocations, an effect that violates conservation of energy in this system, the WITS was introduced in section 4.1. To demonstrate the advantage of this scheme relaxation simulations were performed with the same original dislocation configuration of $N = 1024$ dislocations with the weighted and with the symmetric (i.e. $d_i = 0$ for all dislocations) trapezoidal scheme. The cut-off was set to $r_c = 0.25$ in both cases. According to the average velocity-time profiles seen in figure 6(a) there is no noticeable difference in the active regime, so, the motion of the dislocations is identical in the two cases. However, when motion stops and only numerical noise is seen in the final regime the level of noise is more than an order of magnitude lower in the case of the weighted scheme. This better performance does not require higher computational cost as seen in figure 6(b), where the efficiency of the two methods, defined above, are compared.

It is noted that further decrease of the numerical noise is possible with increasing the cut-off parameter r_c . As mentioned before, with the fully implicit method ($r_c = \infty$) the numerical noise can be decreased to the floating point precision but with the cost of significantly higher computational time of a single timestep. With a finite value of r_c the simulation stepsize h will converge to a finite value in equilibrium (see figure 4), signaling the presence of some numerical noise. The fact that the WITS scheme is able to decrease significantly this noise with preserving the same r_c value (and, at the same time, not increasing the computational

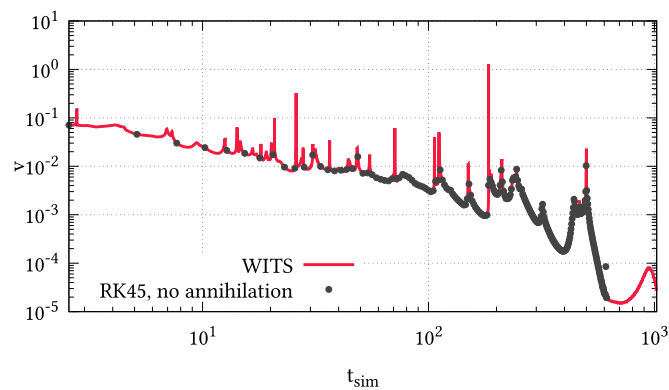


Figure 7. Comparison of average speed v obtained by relaxing the same dislocation configuration containing $N = 256$ dislocations with the 4.5th order explicit Runge–Kutta (RK45) method without dislocation annihilation and 1.6×10^{-9} as tolerance and the WITS implicit method introduced in this paper with 1.6×10^{-5} tolerance and cut-off parameter $r_c = 0.5$. There are fewer datapoints in the figure for the RK45 method because it recorded the data at fixed intervals of approx. $\Delta t_{\text{sim}} \approx 2.5$, whereas WITS recorded data after every successful simulation step.

cost of a single timestep) proves its advantageous properties compared to the traditional symmetric trapezoidal scheme.

6.4. Comparison with an explicit method

In this section the performance of the WITS method described above and an efficient 4.5th order Runge–Kutta (RK45) explicit scheme are compared. As it was mentioned above, annihilation is usually introduced in RK45 with a prescribed threshold distance to increase the efficiency. To be able to analytically compare the two methods annihilation was not performed with RK45, instead, 1000 random dislocation configurations with 256 dislocations were created and the one with the highest minimal possible dipole size was chosen as the starting configuration. During the relaxation the smallest dipole formed had a distance of 1.9×10^{-3} in dimensionless coordinates, that is, approx. 0.2% of the average dislocation spacing. Assuming, for instance, 10^{12} m^{-2} as statistically stored dislocation density, this distance would be 2 nm, which is certainly above the annihilation distance at small temperatures. Therefore, in a such a scenario, introducing annihilation with realistic parameters would not speed up the simulation at all.

The average velocity versus time plots are plotted in figure 7 demonstrating that the motion of the dislocations is identical with the different methods. For the WITS method $r_c = 0.5$ was chosen. It is noted that to achieve this overlap a significantly smaller tolerance parameter had to be set for explicit method than for implicit (1.6×10^{-9} and 1.6×10^{-5} , respectively). Further decrease of the tolerance parameter did not affect the relaxation curves of figure 7.

The simulation runtime is remarkably different for the two methods, as shown in figure 8. The total runtime needed to reach $t_{\text{sim}} = 614.4$ (at the time of the termination of the RK45 simulation, this was its latest recorded entry) is plotted for the RK45 and the WITS method. Whereas to finish the simulations the RK45 needed more than 60 d it took only between 5 and 40 min for the WITS depending on the parameter r_c .

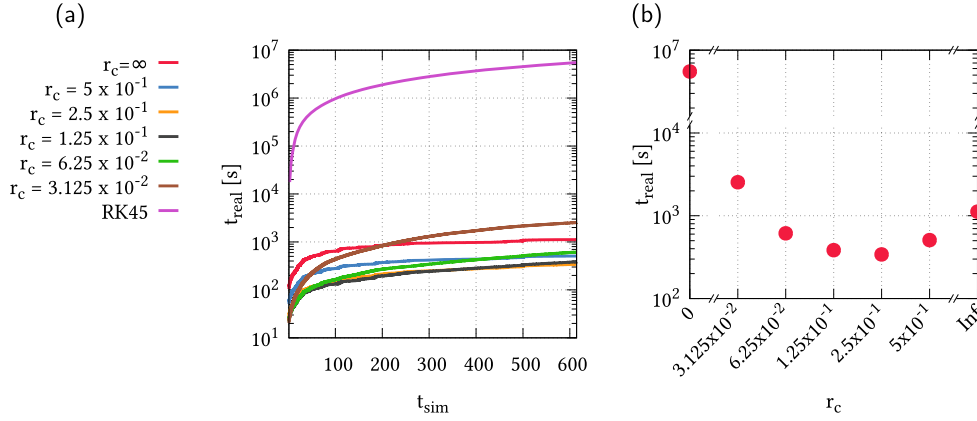


Figure 8. (a) The overall spent wallclock time (t_{real}) needed to reach a simulation timepoint (t_{sim}) for different methods (RK45 and WITS) and different cut-off parameters r_c in the case of WITS. At every choice of r_c WITS performs significantly better than the explicit RK45. (b) Overall spent wallclock time as a function of the cut-off parameter r_c to reach $t_{\text{sim}} = 614.4$ where we terminated the RK45 simulation after approximately two months of runtime. $r_c = 0$ corresponds to the RK45 data. In case of $r_c = \infty$ no cut-off function was used for the calculation of the Jacobian.

7. Summary

DDD simulations play a central role in today's computational materials research as they can be used to model plastic deformation in crystalline materials without major approximations or assumptions in most cases. Unfortunately its application is strongly limited in achievable sample size (typically few microns) and duration (typically few microseconds) because of the long-range nature of stress fields individual dislocations create. These stress fields do not only decay slowly at far distances but also diverge at the core of the dislocation. As described in this paper these features make the equations of motion of the dislocations a *stiff* set of differential equations. So far mostly explicit schemes were employed in DDD simulations where strong upper limit applies to the achievable timestep even in equilibrium. In particular, the maximum possible timestep is basically determined by the smallest dislocation dipole in the system even if the dipole is in its equilibrium, stationary state. To weaken this constraint dislocation annihilation is usually introduced, where the smallest dipoles are removed from the system thus increasing the stepsize.

Since these type of stiff equations can generally be more efficiently solved with implicit schemes, in this paper a novel implicit method was introduced with adaptive stepsize control to solve the differential equations governing the motion of dislocations. A weighted trapezoidal scheme was employed with specific weight factors in order to decrease oscillations around equilibrium and resulting numerical noise. A cut-off parameter r_c was also introduced in the Jacobian matrix with the following properties.

- At $r_c = \infty$ (that is, without using the cut-off functions) the numerical noise can be decreased down to floating point precision and the simulation timestep at equilibrium can practically diverge. These optimal properties are somewhat flawed by an increased computational cost of a single timestep because the Jacobian matrix is dense due to the long-range nature of interactions. In fact, the complexity of computing a single timestep is larger than that of the explicit methods (being $\mathcal{O}(N^3)$ instead of $\mathcal{O}(N^2)$), so for large

systems the implicit method with $r_c = \infty$ only close to equilibrium may be more efficient than traditional explicit methods, because here the gain in the timestep can be large enough to compensate for the slower calculation of a single timestep.

- At finite cut-off r_c an in between method is obtained: on the one hand, it makes the Jacobian sparse thus decreasing the complexity of a timestep to $\mathcal{O}(N^2)$, which is the same complexity of explicit methods, thus, it becomes efficient for large system sizes, too. On the other hand, it still allows the stepsize to be significantly larger than that of explicit schemes. It, in fact, acts similarly to the annihilation distance of explicit methods: dipoles with distance smaller than r_c are treated implicitly (and thus not limiting the timestep) whereas those with distance larger than r_c are computed explicitly and do limit the timestep even at equilibrium. The advantage is, that the small dipoles here do not have to be annihilated, and their dynamics are still solved with high precision. It was found that the runtime of the new method was approximately 4 orders of magnitude lower than that of the explicit method with a realistic annihilation distance.

The results obtained indicate that when activity was high in the system a smaller r_c value was the most efficient and as activity ceased an increasing value of r_c showed better performance. Future work will aim at developing an algorithm that dynamically changes the value of r_c based on the dynamic properties of the system. Such a method could further improve the efficiency of this scheme. In addition, significant increase in computational speed can be expected from porting the source code to GPU, which is also relegated to future work.

In this paper the performance of the WITS was demonstrated on relaxation simulations where dislocations start from initially random configurations. After an initial high activity period dislocations gradually slow down as they approach equilibrium. The motivation behind studying such simulations was that dislocation dynamics is usually an intermittent process characterized by short high activity periods (strain bursts or dislocation avalanches) and long quiescent periods in between [1, 2]. The relaxation process contains both limits: a very active initial phase together with an absolutely stationary state at large simulation times. An optimal numerical method should be efficient in both regimes, and according to the results presented so far, the implicit scheme described here fulfills this criterion.

8. Outlook

The proposed scheme, thus, can be successfully applied to 2D DDD systems, where it leads to a significant speed-up compared to explicit methods. In addition, the newly developed scheme, due to its relative simplicity, is not constrained to the specific 2D model used in this paper but, as we show below, can be easily generalized for more complex 2D or 3D DDD simulations.

The main assumption we made about dislocation systems is that their equations of motion are stiff. This means that if x_i is a degree of freedom (DOF) of the dislocation network (position of a dislocation node, segment, etc) then in equilibrium $f_i(x_i) = 0$ with some function f_i that stands for the generalized force acting on the dislocation node, segment, etc. The slope of this function $m = f_i'(x_i)$ (similarly to a spring constant) characterizes how quickly the DOF returns into its equilibrium position after some perturbation. The system is stiff if the m values for the different DOFs have a huge scatter. In 2D it was shown (equation (5)) that for close dislocation pairs m increases as $1/D^2$, where D is the distance to the closest dislocation, and this leads to the stiffness. The reason for the divergent behavior of

m at small distances is that dislocation interactions diverge for small distances. In more complicated setups, like multiple slip geometries or curved dislocations in 3D it is still true that dislocation interactions diverge for small distances, so the dynamic equations in these cases must also be stiff [30]. The problem is usually addressed by either introducing a minimal distance between dislocation slip planes or nodes or by using core regulated stress fields that do not diverge at small distances. Unfortunately, these approaches only maximize the possible value of m but the equations still remain stiff.

To tackle the problem of stiffness in more complicated set-ups the method proposed in this paper have to be generalized. The generalized force $f_i(x_i)$ introduced above may depend on the other DOFs, externally applied tractions/displacements or internal objects (precipitates, vacancies, etc) and the geometry of the system (such as the configuration of the slip planes), but in every case the analytical form of f_i is available. Therefore, one can compute the Jacobian as $J_{ij} = \frac{\partial f_i}{\partial x_j}$, although, it may be complicated and, thus, numerically demanding. In order to make the Jacobian sparse and reduce the complexity of its evaluation to $\mathcal{O}(N)$, the introduction of the cut-off radius r_c can be done in a straightforward way, by neglecting those J_{ij} elements where the distance of the i th and j th DOF is larger than r_c and one may also introduce the cut-off function of equation (20) after replacing $\sqrt{x^2 + y^2}$ with the distance of the corresponding DOFs. The details of the determination of the Jacobian in a 3D implementation can be found in [30, 32]. Although a somewhat different method was used there to obtain a sparse Jacobian, it was found that the numerical evaluation of the sparse matrix was rather efficient and did not slow down the simulation [30, 32].

Our method is only concerned with the scheme of the numerical integration. All other parts of the algorithm that take place between timesteps are unaffected by the new algorithm. As such, it can be easily combined with dislocation reactions/collisions, dislocation sources, re-meshing of curved dislocation segments, etc. It can be also combined with the fast multipole method mentioned in section 3.1 for the calculation of interactions in order to reduce the complexity below $\mathcal{O}(N^2)$. We, therefore, believe that our method will prove useful in many implementations of DDD.

Acknowledgments

Fruitful discussions with István Groma, Zoltán Boldizsár and Dániel Tüzes are gratefully acknowledged. This work was completed in the ELTE Institutional Excellence Program (1783-3/2018/FEKUTSRAT) supported by the Hungarian Ministry of Human Capacities. The present work was supported by the National Research, Development and Innovation Fund of Hungary (contract numbers: NVKP_16-1-2016-0014, NKFIH-K-119561, NKFIH-KH-125380). GP is also supported by the ÚNKP-19-3 New National Excellence Program of the Hungarian Ministry for Innovation and Technology. PDI is also supported by the ÚNKP-18-4 New National Excellence Program of the Hungarian Ministry of Human Capacities and by the János Bolyai Scholarship of the Hungarian Academy of Sciences.

Appendix. Selection of the ideal weight factor

In section 4.3 we concluded that close to equilibrium the equation of motion of a dislocation dipole can be simplified to

$$\dot{x} \approx -\frac{1}{\tau}x, \quad (23)$$

where $x = 0$ is the equilibrium position. In this appendix we aim at finding the best weight factors for the WITS method in order to solve this equation. This method for equation (23) reads as

$$\frac{x^{k+1} - x^k}{h} = \frac{-(1-d)x^k/\tau - (1+d)x^{k+1}/\tau}{2}. \quad (24)$$

If the weight factor is zero ($d=0$), then one arrives at the trapezoidal scheme, which has an $\mathcal{O}(h^2)$ convergence, superior to backward Euler's $\mathcal{O}(h)$ convergence at $d=1$. So, $d=0$ is preferred but in this case after rearranging equation (24) one obtains

$$x^{k+1} = \frac{1 - \frac{h}{2\tau}}{1 + \frac{h}{2\tau}} x^k. \quad (25)$$

If $h \ll \tau$ holds for the timestep, then $x^{k+1} \approx x^k - (h/\tau) x^k$, so, if x^k is positive then so is x^{k+1} . However, if $h \gg \tau$ then $x^{k+1} \approx -x^k$, so the solution oscillates around the equilibrium position $x = 0$. It is easy to see that if $d=1$ (backward Euler) then x^k and x^{k+1} have the same sign for every choice of h . So, our intention is to select the weight factor d such that

- at small timesteps $d \rightarrow 0$ to exploit the higher precision and
- $d \rightarrow 1$ for large timesteps to avoid oscillations.

To fulfill these criteria we first rearrange equation (24):

$$x^{k+1} = \frac{1 - (1-d)\frac{h}{2\tau}}{1 + (1+d)\frac{h}{2\tau}} x^k. \quad (26)$$

To avoid oscillations the prefactor of x^k on the right-hand side must be positive, so

$$d > 1 - \frac{2\tau}{h} \quad (27)$$

must hold.

The choice of

$$d = \frac{1}{1 + \frac{2\tau}{h}} \quad (28)$$

introduced in section 4.3

- tends to zero as $h \rightarrow 0$,
- fulfills the criterion of equation (27), that is, suppresses oscillations around the equilibrium position and
- tends to one as $h \rightarrow \infty$.

To sum up, the choice of (28) for the weight factor d fulfills all the prescribed criteria to be efficient and avoid oscillations at the same time.

ORCID iDs

Gábor Péterffy  <https://orcid.org/0000-0001-9543-5231>

Péter Dusán Ispánovity  <https://orcid.org/0000-0002-9956-0061>

References

- [1] Miguel M C, Vespignani A, Zapperi S, Weiss J and Grasso J-R 2001 Intermittent dislocation flow in viscoplastic deformation *Nature* **410** 667
- [2] Dimiduk D M, Woodward C, LeSar R and Uchic M D 2006 Scale-free intermittent flow in crystal plasticity *Science* **312** 1188–90
- [3] Bulatov V and Cai W 2006 Computer simulations of dislocations *Oxford Series on Materials Modelling* (Oxford: Oxford University Press)
- [4] Bulatov V V *et al* 2006 Dislocation multi-junctions and strain hardening *Nature* **440** 1174–8
- [5] El-Awady J A 2015 Unravelling the physics of size-dependent dislocation-mediated plasticity *Nat. Commun.* **6** 5926
- [6] Csikor F F, Motz C, Weygand D, Zaiser M and Zapperi S 2007 Dislocation avalanches, strain bursts, and the problem of plastic forming at the micrometer scale *Science* **318** 251–4
- [7] Weygand D, Friedman L H, van der Giessen E and Needleman A 2001 Discrete dislocation modeling in three-dimensional confined volumes *Dislocations 2000: An Int. Conf. on the Fundamentals of Plastic Deformation; Mater. Sci. Eng. A* **309-310** 420–4
- [8] Devincere B, Hoc T and Kubin L 2008 Dislocation mean free paths and strain hardening of crystals *Science* **320** 1745–8
- [9] El-Awady J A, Biner S B and Ghoniem N M 2008 A self-consistent boundary element, parametric dislocation dynamics formulation of plastic flow in finite volumes *J. Mech. Phys. Solids* **56** 2019–35
- [10] Hussein A M, Rao S I, Uchic M D, Dimiduk D M and El-Awady J A 2015 Microstructurally based cross-slip mechanisms and their effects on dislocation microstructure evolution in fcc crystals *Acta Mater.* **85** 180–90
- [11] Cui Y, Po G and Ghoniem N 2018 Size-tuned plastic flow localization in irradiated materials at the submicron scale *Phys. Rev. Lett.* **120** 215501
- [12] Lehtinen A, Granberg F, Laurson L, Nordlund K and Alava M J 2016 Multiscale modeling of dislocation-precipitate interactions in Fe: from molecular dynamics to discrete dislocations *Phys. Rev. E* **93** 013309
- [13] Chang H-J, Fivel M, Rodney D and Verdier M 2010 Multiscale modelling of indentation in FCC metals: from atomic to continuum *C.R. Phys.* **11** 285–92
- [14] Greer J R, Weinberger C R and Cai W 2008 Comparing the strength of f.c.c. and b.c.c. sub-micrometer pillars: compression experiments and dislocation dynamics simulations *Mater. Sci. Eng. A* **493** 21–5
- [15] Miguel M-C, Vespignani A, Zaiser M and Zapperi S 2002 Dislocation jamming and andrade creep *Phys. Rev. Lett.* **89** 165501
- [16] Ovaska M, Paananen T, Laurson L and Alava M J 2016 Collective dynamics of dislocations interacting with mobile solute atoms *J. Stat. Mech* **043204**
- [17] Ispánovity P D, Laurson L, Zaiser M, Groma I, Zapperi S and Alava M J 2014 Avalanches in 2d dislocation systems: plastic yielding is not depinning *Phys. Rev. Lett.* **112** 235501
- [18] Ovaska M, Laurson L and Alava M J 2015 Quenched pinning and collective dislocation dynamics *Sci. Rep.* **5** 10580
- [19] Tsekenis G, Uhl J T, Goldenfeld N and Dahmen K A 2013 Determination of the universality class of crystal plasticity *Europhys. Lett.* **101** 36003
- [20] Zhou C, Reichhardt C, Reichhardt C J O and Beyerlein I J 2015 Dynamic phases, pinning, and pattern formation for driven dislocation assemblies *Sci. Rep.* **5** 8000
- [21] Ispánovity P D, Papanikolaou S and Groma I 2020 Emergence and role of dipolar dislocation patterns in discrete and continuum formulations of plasticity *Phys. Rev. B* **101** 024105
- [22] Benzerga A A, Bréchet Y, Needleman A and Van der Giessen E 2004 Incorporating three-dimensional mechanisms into two-dimensional dislocation dynamics *Modelling Simul. Mater. Sci. Eng.* **12** 159–96
- [23] van der Giessen E and Needleman A 1995 Discrete dislocation plasticity—a simple planar model: a simple planar model *Modelling Simul. Mater. Sci. Eng.* **3** 689–735 Journal SEP RW456 MODEL SIMUL MATER SCI ENG
- [24] Cleveringa H H M, Van Der Giessen E and Needleman A 1997 Comparison of discrete dislocation and continuum plasticity predictions for a composite material *Acta Mater.* **45** 3163–79

- [25] Yefimov S, Groma I and van der Giessen E 2004 A comparison of a statistical-mechanics based plasticity model with discrete dislocation plasticity calculations *J. Mech. Phys. Solids* **52** 279–300
- [26] Kondori B, Needleman A and Benzerga A A 2017 Discrete dislocation simulations of compression of tapered micropillars *J. Mech. Phys. Solids* **101** 223–34
- [27] Song H, Yavas H, Van der Giessen E and Papanikolaou S 2019 Discrete dislocation dynamics simulations of nanoindentation with pre-stress: hardness and statistics of abrupt plastic events *J. Mech. Phys. Solids* **123** 332–47
- [28] Lodh A, Tak T N, Prakash A, Guruprasad P J, Keralavarma S M, Benzerga A A, Hutchinson C and Samajdar I 2019 Microstructural origin of residual stress relief in aluminum *Metall. Mater. Trans. A* **50** 5038–55
- [29] Papanikolaou S, Song H and Van der Giessen E 2017 Obstacles and sources in dislocation dynamics: strengthening and statistics of abrupt plastic events in nanopillar compression *J. Mech. Phys. Solids* **102** 17–29
- [30] Sills R B and Cai W 2014 Efficient time integration in dislocation dynamics *Modelling Simul. Mater. Sci. Eng.* **22** 025003
- [31] Devincere B, Madec R, Monnet G, Queyreau S, Gatti R and Kubin L 2011 Modeling crystal plasticity with dislocation dynamics simulations: the ‘micromegas’ code *Mechanics of Nano-Objects* ed O Thomas, A Ponchet and S Forest (Paris: Pressesdes Mines) pp 81–100
- [32] Gardner D J, Woodward C S, Reynolds D R, Hommes G, Aubry S and Arsenlis A 2015 Implicit integration methods for dislocation dynamics *Modelling Simul. Mater. Sci. Eng.* **23** 025006
- [33] Hirth J P and Lothe J 1982 *Theory of Dislocations* (Malabar, FL: Krieger)
- [34] Bakó B, Groma I, Györgyi G and Zimányi G 2006 Dislocation patterning: the role of climb in meso-scale simulations *Comput. Mater. Sci.* **38** 22–8
- [35] Kuykendall W P and Cai W 2013 Conditional convergence in two-dimensional dislocation dynamics *Modelling Simul. Mater. Sci. Eng.* **21** 055003
- [36] Zaiser M, Miguel M-C and Groma I 2001 Statistical dynamics of dislocation systems: the influence of dislocation-dislocation correlations *Phys. Rev. B* **64** 224102
- [37] Laurson L, Miguel M-C and Alava M J 2010 Dynamical correlations near dislocation jamming *Phys. Rev. Lett.* **105** 015501
- [38] Ispánovity P D, Groma I, Györgyi G, Csikor F F and Weygand D 2010 Submicron plasticity: yield stress, dislocation avalanches, and velocity distribution *Phys. Rev. Lett.* **105** 085503
- [39] Tsekenis G, Goldenfeld N and Dahmen K A 2011 Dislocations jam at any density *Phys. Rev. Lett.* **106** 105501
- [40] Salmenjoki H, Alava M J and Laurson L 2018 Machine learning plastic deformation of crystals *Nat. Commun.* **9** 1–7
- [41] Ispánovity P D, Groma I, Györgyi G, Szabó P and Hoffelner W 2011 Criticality of relaxation in dislocation systems *Phys. Rev. Lett.* **107** 085506
- [42] Gulluoglu A N, Srolovitz D J, LeSar R and Lomdahl P S 1989 Dislocation distributions in two dimensions *Scr. Metall.* **23** 1347–52
- [43] Arsenlis A, Cai W, Tang M, Rhee M, Oppelstrup T, Hommes G, Pierce T G and Bulatov V V 2007 Enabling strain hardening simulations with dislocation dynamics *Modelling Simul. Mater. Sci. Eng.* **15** 553–95
- [44] Zhao D, Huang J and Xiang Y 2010 A new version fast multipole method for evaluating the stress field of dislocation ensembles *Modelling Simul. Mater. Sci. Eng.* **18** 045006
- [45] Yin J, Barnett D M, Fitzgerald S P and Cai W 2012 Computing dislocation stress fields in anisotropic elastic media using fast multipole expansions *Modelling Simul. Mater. Sci. Eng.* **20** 045015
- [46] Ispánovity P D, Groma I, Hoffelner W and Samaras M 2011 Abnormal subgrain growth in a dislocation-based model of recovery *Modelling Simul. Mater. Sci. Eng.* **19** 045008
- [47] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (Cambridge: Cambridge University Press)
- [48] Csikor F F and Groma I 2004 Probability distribution of internal stress in relaxed dislocation systems *Phys. Rev. B* **70** 8
- [49] Berzins M and Fuzeland R M 1992 An adaptive theta method for the solution of stiff and nonstiff differential equations *Appl. Numer. Math.* **9** 1–19
- [50] Ghoniem N M, Tong S-H and Sun L Z 2000 Parametric dislocation dynamics: a thermodynamics-based approach to investigations of mesoscopic plastic deformation *Phys. Rev. B* **61** 913–27

- [51] Liu H and Jiao D 2010 A H-LU based direct finite element solver accelerated by nested dissection for large-scale modeling of ICs and packages *PIERS Online* **6** 679–83
- [52] Davis T A 2004 Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method *ACM Trans. Math. Softw.* **30** 196–9