IX International Conference on Particle-Based Methods
PARTICLES 2025
J. Bonet, M. Cremonesi, A. Franci, A. Gil (Eds)

ADAPTIVE TIME STEP AND SOLVER FOR DISCRETE ELEMENT METHOD - BEAM BOND MODEL

RADEK VARGA, MARTIN ČERMÁK

Department of Mathematics, Faculty of Civil Engineering
VŠB-TUO,
Ludvíka Podéště 1875/17, 708 00 Ostrava–Poruba, Czech Republic
e-mail: radek.varga@vsb.cz

Key words: DEM, BBM, Time Step, Adaptive Solver

Abstract. This contribution introduces an adaptive strategy for time-stepping and solver selection in simulations based on the Discrete Element Method with beam bond models (DEM-BBM). In this approach, particles are connected by virtual beams capable of transmitting axial, shear, and bending forces, enabling the simulation of both discrete and continuum mechanical behaviour. Such capabilities are particularly relevant in crack propagation problems, where an initially continuous structure is progressively disrupted by fracture. While DEM is traditionally solved using explicit or semi-implicit integration schemes, certain classes of DEM-BBM problems, such as those involving progressive fracture, can benefit from implicit methods. These allow for longer time steps without compromising numerical stability. However, the time step size can affect the accuracy of the solution, particularly under rapidly changing conditions. For this reason, it must be continuously adapted based on the current state of the system, with respect to velocities, stress distribution, and oscillatory response. Depending on the state of the system, different solver types can be employed. When the system remains stable and the time step is constant, direct solvers offer high efficiency. In contrast, when frequent structural changes occur, such as during fracture development, iterative solvers are more suitable. The proposed strategy enables dynamic transitions between solver types and time-stepping adaptions, which improves robustness and computational performance.

1 INTRODUCTION

The Discrete Element Method (DEM) [1] combined with the Beam Bond Model (BBM) [2] provides a framework for simulating materials at the interface between discrete and continuum behaviour. The material is represented by rigid particles interconnected through fictitious beams capable of transmitting axial, shear, and bending forces. DEM-BBM is particularly suited for problems with frequent changes in contact topology, for example bulk solids [3,4], where such conditions typically require explicit or semi-implicit integration schemes due to their strongly nonlinear and unpredictable nature.

This paper addresses a different class of problems, namely crack propagation in structural components, especially in brittle materials such as concrete. In such cases, changes in the interaction structure occur only gradually. This opens opportunities for the effective application of implicit methods.

The focus of this contribution is placed on two aspects that are critical for efficient implicit DEM-BBM simulations: the choice of the solver for the system of equations and the selection of a suitable time-step size. The implicit integration is carried out using the Newmark scheme [5], which provides a robust framework for dynamic structural analysis. Different solver types, such as direct and iterative approaches with or without preconditioning, are considered, since their relative efficiency depends on the current state of the system. In parallel, the time-step size is continuously adapted to balance accuracy with computational cost, especially in situations where sudden structural changes occur during fracture development.

The aim of this study is to demonstrate how combining adaptive solver switching with adaptive time integration can significantly improve both robustness and computational performance. The proposed strategy is verified on benchmark problems, and its efficiency is compared with conventional approaches using fixed solvers and constant time-steps.

2 THEORETICAL BACKGROUND OF DEM-BBM

In the DEM, individual elements interact with each other through contacts. The motion of the system is governed by a set of second-order differential equations,

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}(t)\dot{\mathbf{u}}(t) + \mathbf{f}^{int}(\mathbf{u}(t)) = \mathbf{f}^{ext}(t), \qquad \text{for} \quad t \geq 0, \tag{1}$$

defined by the initial velocity $\mathbf{v}_0$ and displacement $\mathbf{u}_0$ as:

$$\dot{\mathbf{u}}(0) = \mathbf{v}_0, \quad \mathbf{u}(0) = \mathbf{u}_0 \tag{2}$$

and by the boundary conditions

$$\mathbf{Bu}(t) = \mathbf{o}. \tag{3}$$

Where $\mathbf{M} \in \mathbb{R}^{n_d \times n_d}$ is mass matrix of the system with n_d degrees of freedom (DOFs), $\mathbf{C}(t) \in \mathbb{R}^{n_d \times n_d}$ is global dumping matrix dependent on time t. The vector $\mathbf{u}(t) \in \mathbb{R}^{n_d}$ is the displacement vector, with its time derivatives $\dot{\mathbf{u}}(t)$ and $\ddot{\mathbf{u}}(t)$ representing velocities and accelerations, respectively. Vector $\mathbf{f}^{int}(\mathbf{u}(t))) \in \mathbb{R}^{n_d}$ denotes the internal forces between individual contacts in the system. The term $\mathbf{f}^{ext}(t) \in \mathbb{R}^{n_d}$ denotes the vector of external forces. Matrix $\mathbf{B} \in \mathbb{R}^{n_f \times n_d}$ represents the boundary conditions, corresponding to n_f fixed DOFs and $\mathbf{o} \in \mathbb{R}^{n_f}$ is zero vector.

2.1 Mass, damping and stiffness matrix

The global mass matrix $\mathbf{M}$ is constructed as a block diagonal matrix consisting of individual mass matrices for each element e:

$$\mathbf{M} = \operatorname{diag}\left(\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_{n_e}\right), \tag{4}$$

where n_e is the number of elements. Each element mass matrix $\mathbf{M}_e \in \mathbb{R}^{6\times 6}$ is diagonal and contains the element mass m_e and the moment of inertia J_e as:

$$\mathbf{M} = \operatorname{diag}(m_e, m_e, m_e, J_e, J_e, J_e). \tag{5}$$

The global damping matrix $\mathbf{C}(t)$ is constructed according to the Rayleigh model [6] as

$$\mathbf{C}(t) = \alpha\mathbf{M} + \eta\mathbf{K}(t), \tag{6}$$

where α and η are the mass- and stiffness-proportional damping coefficients. The matrix $\mathbf{K}(t) \in \mathbb{R}^{n_d \times n_d}$ is the global stiffness matrix at time t. It is obtained as the sum of the stiffness matrices for bonded contacts $\mathbf{K}_b$ and unbounded contacts $\mathbf{K}_u$. These submatrices are assembled as

$$\mathbf{K}_x = \sum_{k=1}^{n_x} \mathbf{L}_k^{\dagger}\mathbf{T}_k^{\dagger}\mathbf{K}_{x,k}\mathbf{T}_k\mathbf{L}_k, \tag{7}$$

where the index x stands for either bonded b or unbonded u contacts and n_x is the number of corresponding contacts. The matrix $\mathbf{L}_k \in \mathbb{R}^{12\times n_d}$ is the localisation operator that transforms the local stiffness matrix $\mathbf{K}_{x,k} \in \mathbb{R}^{12\times 12}$ into the global system, $\mathbf{T}_k \in \mathbb{R}^{12\times 12}$ is the transformation matrix from local to global coordinates, and $(\cdot)^{\dagger}$ stands for the transpose operator.

The local stiffness matrix for a bonded contact $\mathbf{K}_{b,k}$ is obtained from Timoshenko beam theory [7], where a fictitious beam with a circular cross-section is introduced between the connected elements. The radius of this beam is chosen according to the size of the connected particles and it enables the evaluation of bond failure at the contact, which is essential for simulating crack propagation. For unbonded contacts, $\mathbf{K}_{u,k}$ a partially linearised formulation based on experimental tests is adopted. The detailed construction of these matrices is described in [8] and [9].

2.2 Integration schemes

In this paper, the Newmark integration scheme is employed for the time discretisation of the governing equations of motion. The method provides a unified framework for solving second-order differential equations and is widely used in structural dynamics. Its formulation is based on incremental relations that update displacements and velocities at each time step. The displacement increment $\Delta\mathbf{u}$ is defined as the difference between the current and the previous step as:

$$\mathbf{u}_{\langle n+1\rangle} = \Delta\mathbf{u}_{\langle \mathrm{n}\rangle} + \mathbf{u}_{\langle \mathrm{n}\rangle}. \tag{8}$$

From this definition, the Newmark method yields the following set of equations:

$$\begin{gathered}\dot{\mathbf{u}}_{\langle n+1\rangle} - \gamma\Delta t_{\langle n\rangle}\ddot{\mathbf{u}}_{\langle n+1\rangle} = \dot{\mathbf{u}}_{\langle n\rangle} + (1-\gamma)\Delta t_{\langle n\rangle}\ddot{\mathbf{u}}_{\langle n\rangle}\\ \Delta\mathbf{u}_{\langle n\rangle} - \beta\Delta t_{\langle n\rangle}^2\ddot{\mathbf{u}}_{\langle n+1\rangle} = \Delta t\dot{\mathbf{u}}_{\langle n\rangle} + \frac{\Delta t_{\langle n\rangle}^2}{2}(1-2\beta)\ddot{\mathbf{u}}_{\langle n\rangle}\\ \mathbf{M}\ddot{\mathbf{u}}_{\langle n+1\rangle} + \mathbf{C}_{\langle n\rangle}\dot{\mathbf{u}}_{\langle n+1\rangle} + \mathbf{K}_{\langle n\rangle}\Delta\mathbf{u}_{\langle n\rangle} = \mathbf{f}_{\langle n+1\rangle}^{ext} - \mathbf{f}_{\langle n\rangle}^{int}\end{gathered} \tag{9}$$

where $\langle\cdot\rangle$ denotes the discretised time step, and Δt is the size of the time step. The vector $\mathbf{f}^{int} \in \mathbb{R}^{n_d}$ represents the internal forces within the system. The parameters $\gamma \in (0,1)$ and $\beta \in (0,0.5)$ are coefficients of the Newmark method and govern its numerical stability. The most common combinations are $\gamma = 0.5, \beta = 0$, which correspond to the explicit central difference method (CDM), and $\gamma = 0.5, \beta = 0.25$, which correspond to the implicit average constant acceleration scheme (ACAS).

3 TIME-STEP LIMITS AND CONVERGENCE CRITERIA

The choice of the time step is critical [10] to maintain a balance between computational stability, physical accuracy, and efficiency. In the case of CDM, the stability condition for the integration step is analytically defined as

$$\Delta t_{crit} = \frac{2}{\omega_{max}}, \tag{10}$$

where ω_{max} is the highest natural frequency of the system, obtained as the square root of the largest eigenvalue of the matrix $\mathbf{M}^{-1}\mathbf{K}$**.** Since the direct computation of eigenvalues can be computationally expensive, a simplified diagonal estimate of the critical time step is introduced as

$$\Delta t_{crit} \approx 2 \min_i \sqrt{\frac{\mathbf{M}_{\{\mathrm{i,i}\}}}{\mathbf{K}_{\{i,i\}}}}, \tag{11}$$

where $i \in \{1,2 \dots n_d\}$. The accuracy of this estimate is evaluated in Section 5.

In the case of ACAS, the time step is not restricted by stability, but excessively large steps may suppress high-frequency responses, which can significantly reduce the accuracy of crack propagation. Additional restrictions come from the physical nature of the system. Since the system matrices $\mathbf{K}$ and $\mathbf{C}$ vary during the simulation, the time step must be limited to correctly capture these changes. The first restriction is related to the maximum allowable penetration of particles during a single step. This condition prevents particles from passing through each other without detection and avoids sudden jumps in internal forces. It is expressed as

$$\frac{v_{max}}{r_{min}}\Delta t < \alpha_r, \tag{12}$$

where v_{max} is the maximum velocity in the system, r_{min} is the smallest particle radius, and $\alpha_r \in (0{,}0.5)$ denotes the maximum admissible penetration. A smaller value of α_r is required for stiffer or more nonlinear materials.

Convergence criteria impose further restrictions on the time step during fracture events. In particular, changes in the stiffness matrix **K** during crack propagation must be addressed by reducing the step size. A line-search-like method is employed to evaluate the limiting time step when a crack occurs,

$$\Delta t_{lim} = \frac{-\dot{\mathrm{u}}_{\langle n \rangle,cr} \pm \sqrt{\dot{\mathrm{u}}^2_{\langle n \rangle,cr} - 2\ddot{u}^2_{\langle n+1 \rangle,cr}\,(u_{\langle n \rangle,cr} - u_{\langle n \rangle,lim})}}{\ddot{u}_{\langle n+1 \rangle,cr}}, \tag{13}$$

where index cr denotes the degree of freedom where cracking occurs. The limiting displacement $u_{n,lim}$ calculated from the stress σ interpolation as

$$u_{\langle n \rangle,lim} = u_{\langle n \rangle,cr} + \frac{\sigma_{lim} - \sigma_{\langle n \rangle,cr}}{\sigma_{\langle n+1 \rangle,cr} - \sigma_{\langle n \rangle,cr}} \Delta u_{\langle n \rangle,cr}, \tag{14}$$

where σ_{lim} is material stress limit.

The convergence criteria of the ODE system are evaluated by comparing energies. To ensure physical accuracy, the internal energy E^{int} and the external energy E^{ext} must remain in balance at each time step. The external energy is calculated as

$$E^{ext}_{\langle n \rangle} = \int_0^t \mathbf{f}^{\dagger}(t)^{ext}\dot{\mathbf{u}}(t)\,\mathrm{d}t \;\approx\; \frac{1}{4}\sum_{k=1}^{n_t}\left(\mathbf{f}^{ext}_{\langle k \rangle} + \mathbf{f}^{ext}_{\langle k+1 \rangle}\right)^{\dagger}\left(\dot{\mathbf{u}}_{\langle k \rangle} + \dot{\mathbf{u}}_{\langle k+1 \rangle}\right)\Delta t_{\langle k \rangle}, \tag{15}$$

where n_t is the number of evaluated time steps.

The internal energy is defined as the sum of potential energy E^p, kinetic energy $E^k$and dissipated energy E^d. The individual components are calculated as

$$\begin{gathered} E^p_{\langle n \rangle} = \int_0^t \mathbf{f}(t)^{int}\,\mathrm{d}\mathbf{u}(\mathrm{t}) \;\approx \frac{1}{2}\sum_{k=1}^{n_t}\left(\mathbf{f}^{int}_{\langle k \rangle} + \mathbf{f}^{int}_{\langle k+1 \rangle}\right)^{\dagger}\Delta\mathbf{u}_{\langle k \rangle} \\ E^k_{\langle n \rangle} = \frac{1}{2}\dot{\mathbf{u}}^{\dagger}_{\langle n \rangle}\mathbf{M}\dot{\mathbf{u}}_{\langle n \rangle}, \\ E^d_{\langle n \rangle} = \int_0^t \dot{\mathbf{u}}(t)\mathbf{C}(t)\dot{\mathbf{u}}(t)\,\mathrm{d}t \;\approx \frac{1}{2}\sum_{k=1}^{n_t}\left(\dot{\mathbf{u}}^{\dagger}_{\langle k \rangle}\mathbf{C}_{\langle k \rangle}\dot{\mathbf{u}}_{\langle k \rangle} + \dot{\mathbf{u}}^{\dagger}_{\langle k+1 \rangle}\mathbf{C}_{\langle k \rangle}\dot{\mathbf{u}}_{\langle k+1 \rangle}\right)\Delta t_{\langle k \rangle}. \end{gathered} \tag{16}$$

4 ADAPTIVE DEM-BBM ALGORITHM

This section describes the global algorithm, see Algorithm 1, for solving DEM-BBM with emphasis on adaptive procedures. The adaptive time-stepping follows the rules introduced in Section 3, while the choice of solvers is guided by the numerical experiments presented in Section 5.2.

Algorithm 1: Adaptive DEM-BBM with time-step and solver control.

(Bold style indicates adaptive solver dependencies, details are described afterwards.)

1.	Initialize (Geometry, Material, Boundary conditions, Primary solver settings, etc)
2.	Discretize geometry and identify initial contacts
3.	Construct initial matrices, **evaluate** Δt_{crit} , **Set** $n_{linear} \rightarrow inf$
4.	while $t < t_{max}$
4.1.	while convergence = true, set convergence = false
4.1.1.	**Choose solver option and time-step**
4.1.2.	if large deformation or contact changes detected
4.1.2.1.	Update matrices, **Set** $n_{linear} \rightarrow 0$
4.1.2.2.	else
4.1.2.3.	**Set** $n_{linear} \rightarrow n_{linear} + 1$
4.1.2.4.	end if
4.1.3.	perform numerical integration
4.1.4.	evaluate E^{ext}, E^{int}
4.1.5.	**if** $E^{ext}/E^{int} > 1 + E_{tol}^{max}$ **or** $E^{ext}/E^{int} < 1 - E_{tol}^{min}$
4.1.5.1.	**Set** $\Delta t \rightarrow \Delta t/2$, **break to 4.1**
4.1.5.2.	**end if**
4.1.6.	evaluate stress and detect crack
4.1.7.	**if** $\sigma_{\langle n \rangle,cr}/\sigma_{lim} > c_{tol}$
4.1.7.1.	**Set** $\Delta t \rightarrow \Delta t_{lim}$, **break to 4.1**
4.1.7.2.	**end if**
4.1.8.	**if** crack detected
4.1.8.1.	**Set** $\Delta t \rightarrow m\Delta t_{crit}$, Set $n_{\Delta t} \rightarrow c_{step}$
4.1.8.2.	**end if**
4.1.9.	set convergence = true
4.1.10.	end while
4.2.	update $t \rightarrow t + \Delta t$, step-count, semi-results (acceleration, forces, etc.)
4.3.	end while
5.	Post processing (Export results, visualisation)

In **step 3**, Δt_{crit} is evaluated according to Eq. (11), Although the stiffness matrix **K** changes during the simulation, its re-evaluation is in typical crack-propagation problems not necessary, and Δt_{crit} remains nearly constant throughout the whole simulation. In the same step, the parameter n_{linear} is set to infinity. This parameter indicates the last change in the system.

Step 4.1.1 represents the core of the adaptive solver. The choice is made between explicit and implicit integration. The explicit CDM solver is selected if Δt is smaller than $m\ c_{i,e}\Delta t_{crit}$, where $c_{i,e}$ denotes the ratio of computational cost between explicit and implicit steps, and $m \in (0,1)$ defines the maximum fraction of Δt_{crit} allowed for the explicit solver. If the explicit solver is chosen, the time-step is set to $\Delta t \rightarrow m\Delta t_{crit}$.

For the implicit option, the internal solver is selected between direct Cholesky factorisation and iterative methods based on the Conjugate Gradient (CG). The choice depends on both the number of degrees of freedom (DOFs) and the parameter n_{linear} parameter. If nearly linear behaviour is expected (large n_{linear}), either Cholesky factorization or CG with incomplete Cholesky factorization are prefered, due to possibility of recycling factorizations for multiple steps. When frequent nonlinearity is present, standard CG without preconditioning, or with a simpler preconditioner, is applied.

Step 4.1.1 also controls the parameter $n_{\Delta t}$, which specifies the number of steps performed with a reduced Δt, Once these steps are completed, the time-step is reset to its original value.

In **step 4.1.5** the energy balance of the system is checked. The admissible range of energy imbalance E_{tol}^{min} , E_{tol}^{max} is typically set to 1%. If balance is not achieved within several iterations, the tolerance can be increased with a corresponding warning.

In **Step 4.1.7**, if the stress in a bonded contact exceeds the tolerance c_{tol}, (usually set to 1%), the time-step is adjusted according to Eq. (13), and the step is re-evaluated.

Finally, in **Step 4.1.8**, if a crack is detected, the step size is set as $\Delta t \to m\Delta t_{crit}$ for the following c_{step} steps. The parameter c_{step} representing the estimated number of steps required for high-frequency modes to dissipate, can be evaluated as

$$c_{step} = \frac{\Delta t_{crit}}{m\left(\frac{\alpha}{2}\Delta t_{crit}^2 + 2\eta\right)} . \tag{17}$$

5 NUMERICAL BENCHMARKS

The following benchmarks are employed to compare the performance of the above-described methods. All cases are based on a regular discretisation according to Figure 1, defined by the radius of the primary elements r_1. The material is assumed to be homogeneous and is characterised by the Young's modulus E, Poisson's ratio ν, tensile strength σ_{max}^{+}, compression strength σ_{max}^{-}, shear strength τ_{max} and material density ρ. Specific parameter values are provided for each benchmark in the corresponding figure schemes.

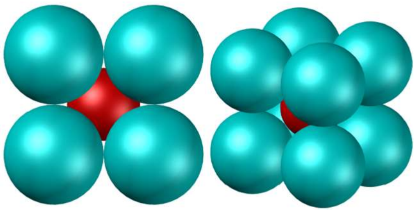

Figure 1: Discretization scheme

5.1 Cantilever beam: Critical time-step estimation

The first benchmark serves to compare the critical time step Δt_{crit} obtained from Eq. (10) with its estimation by Eq. (11). The comparison is based on scaling of the material, geometry, and discretisation properties, see Figure 2. In this benchmark, only the original system matrices are considered, and crack propagation is not included.

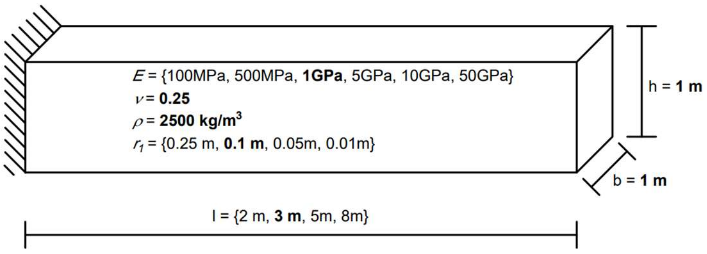
E = {100MPa, 500MPa, 1GPa, 5GPa, 10GPa, 50GPa}
ν = 0.25
ρ = 2500 kg/m³
r₁ = {0.25 m, 0.1 m, 0.05m, 0.01m}
h = 1 m
b = 1 m
l = {2 m, 3 m, 5m, 8m}

Figure 2: Cantilever beam geometry and material parameters. Bold values indicate the fixed parameters used in simulations, while the remaining parameters are varied.

Tables 1–3 show the comparison between the analytical and estimated values of the critical time step for variations in geometry, material parameters, and discretisation density, respectively. In all cases, the proposed diagonal estimate slightly overpredicts the analytical value, with a relative error of approximately 2–3%. This behaviour is consistent across the tested parameter ranges, demonstrating that the estimate is conservative while maintaining good accuracy.

Table 1: Critical time step under geometry variation

l [m]	Δt_{crit} 10^{-4} [s]	Δt_{est} 10^{-4} [s]	rel [%]	abs 10^{-4} [s]
2	2.465	2.530	102.6	0.066
3	2.464	2.530	102.7	0.066
5	2.463	2.530	102.7	0.067
8	2.464	2.530	102.7	0.067

Table 2: Critical time step under material variation

E [MPa]	Δt_{crit} 10^{-4} [s]	Δt_{est} 10^{-4} [s]	rel [%]	abs 10^{-4} [s]
100	7.790	8.001	102.7	0.210
500	3.484	3.578	102.7	0.094
1000	2.464	2.530	102.7	0.066
5000	1.102	1.131	102.7	0.029
10000	0.779	0.800	102.7	0.021
50000	0.348	0.357	102.7	0.009

Table 3: Critical time step under discretisation variation

r_1 [m]	Δt_{crit} 10^{-4} [s]	Δt_{est} 10^{-4} [s]	rel [%]	abs 10^{-4} [s]
0.25	6.226	6.325	101.6	0.099
0.1	2.464	2.530	102.7	0.066
0.05	1.226	1.265	103.2	0.039
0.01	0.246	0.255	103.2	0.077

5.2 Constrained beam: Solver performance

This numerical benchmark is used to compare the performance of different numerical solvers on the constrained beam model, see Figure 3. The comparison evaluates the duration of a single computational step as a function of the time-step size and the system size. Both the solver execution time and the required preprocessing (such as factorisation or preconditioner setup, due to the possibility of reuse between steps) are considered. The tested values of time-step Δt are set as $m\Delta t_{crit}$ where $m \epsilon \{0.1, 0.2, \mathbf{0.5}, 1, 100, 10000\}$. The benchmark includes the explicit CDM, and the implicit ACAS, tested with different internal solvers including direct factorisation, the CG method, and the Preconditioned Conjugate Gradient (PCG) method with an incomplete Cholesky (Ichol) preconditioner. In case of iterrative solver convergece tolerances are set to be 10^{-3} and 10^{-6}. Note that the multipliers larger then 1 are tested only for implicit solver.

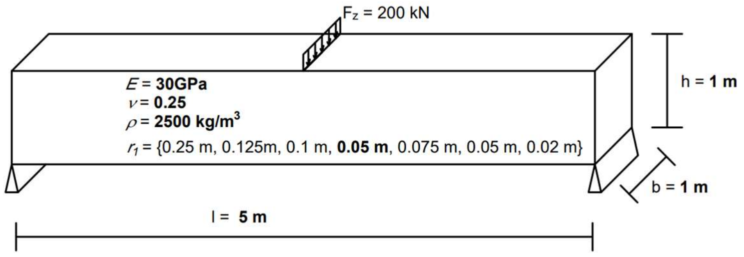

Figure 3: Constrained beam geometry and material parameters. Bold values indicate the fixed parameters used in simulations, while the remaining parameters are varied.

The results considering the scaling parameter m are shown in Figures 4 and 5. As expected, the explicit CDM solver provides the lowest computational time per step, but it is restricted by the critical time step and cannot be applied with larger Δt. From the comparison, the ratio of computational cost between implicit and explicit solvers $c_{i,e}$ can be estimated to be on the order of 10^2.

In Figure 4, the performance per single step is reported, including the cost of the LL decomposition for Cholesky and the preconditioner setup for PCG. In this case, iterative solvers are generally more efficient than the direct solver, since the preprocessing cost dominates. Figure 5 presents the results for 100 steps, where recycling of the factorisation and preconditioner is possible. In this scenario, the direct solver clearly benefits from factor reuse and may even outperform iterative methods.

With increasing Δt, the computational time of iterative solvers grows due to the worsening conditioning of the system. This effect does not influence the direct solver or the explicit method, whose performance remains unaffected by conditioning.

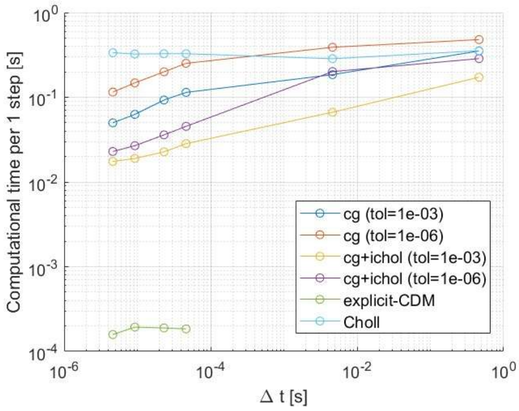

Figure 4: Relation between computational time and step size Δt considering 1 step

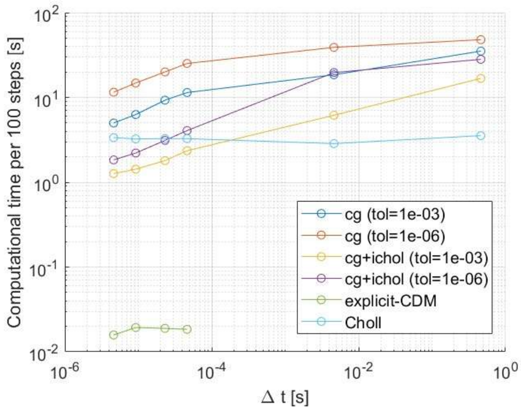

Figure 5: Relation between computational time and step size Δt considering 100 steps with recycling

Figures 6 and 7 show the dependence of computational cost on the number of DOFs. The explicit CDM solver is still the fastest across all tested system sizes, and the results demonstrate that the computational cost ratio between implicit and explicit solvers remains close to $c_{i,e} \approx 10^2$.

Among implicit methods, the results for a single step (Figure 6) show that PCG with incomplete Cholesky significantly outperforms plain CG, while the direct solver becomes increasingly expensive with growing problem size. For 100 steps with recycling (Figure 7), the reuse of the factorisation makes the direct solver efficient for small and medium-sized systems, although it gradually loses competitiveness as the system size increases. PCG, on the other hand, exhibits better scalability and therefore becomes the preferable option for large-scale problems. The influence of solver tolerance appears only as a vertical shift of the curves, without changing their overall trends.

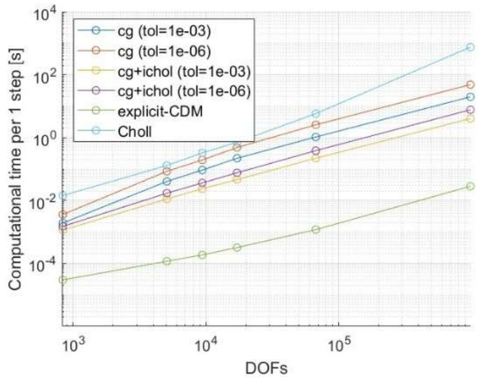

Figure 6: Relation between computational time and problem size (DOFs) considering 1 step

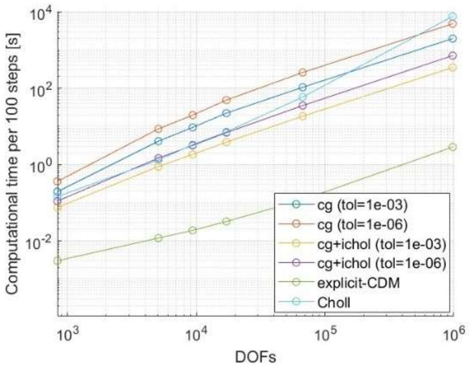

Figure 7: Relation between computational time and problem size (DOFs) considering 100 steps with recycling

5.3 Tension test: Adaptive solver comparison

In this benchmark, a tensile test with scaled material parameters is performed in order to allow comparison of explicit and implicit solvers under the same setup. The reduction of stiffness and strength ensures that the critical time step is sufficiently large for the explicit CDM solver to remain stable. The material properties are defined as follows: Young's modulus $E = 30MPa$, Poisson's ratio $\nu = 0.3$, tensile strength $\sigma_{max}^{+} = 2.2kPa$, compression strength $\sigma_{max}^{-} = 40kPa$, shear strength $\tau_{max} = 4kPa$, and density $\rho = 2500\,\frac{kg}{m^3}$. The geometry of the specimen is shown in Figure 8. The force of total value $F_x = 1.5kN$ is spread uniformly over one face. The damping parameters are set as $\alpha = 10^{-4}$ and $\eta = 10^{-5}$.

For the non-adaptive solver setup, the CDM is tested with a time step of $\Delta t = 0.1\Delta t_{crit}$, in case of ACAS several time step are tested $\Delta t = \{10^{-5}s, 5 \times 10^{-5}\ s, 10^{-4}\ s, 5 \times 10^{-4}\ s\}$ total simulation time $t_{max} = 0.05\ s$ and force is continuously applied during first 0.01 s.

The fracture evolution is illustrated in Figure 9, where debonded contacts are highlighted in red. The failure does not initiate gradually at the loaded boundary but rather localises in the middle section of the specimen. While minor debonding also appears near the loaded edge, the dominant crack develops almost simultaneously across the neck due to the strongly dynamic loading applied in a short time interval. This results in a sudden splitting behaviour instead of a progressive crack growth.

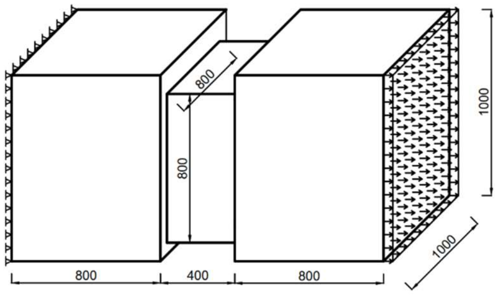

Figure 8: Tension problem geometry [mm].

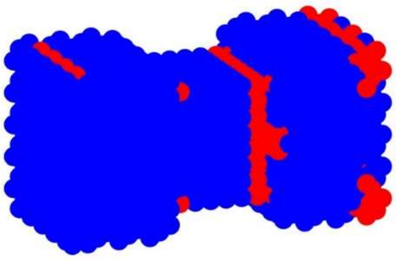

Figure 9: Debonded contacts (highlighted in red)

All simulations maintained the energy balance within the admissible tolerance of 1%. In the explicit CDM method, a slight delay in the growth of internal energy can be observed compared to the external work. This behaviour is natural for explicit integration, where kinetic energy temporarily stores part of the input before being fully converted into internal energy.

Both iterative (PCG with incomplete Cholesky preconditioning) and direct (Cholesky factorisation) solvers preserve the same level of accuracy in terms of energy response. Importantly, no cumulative error due to the iterative nature of PCG was detected, as the energy curves remain aligned with those of the direct solver. However, for a larger time step of $\Delta t = 5 \times 10^{-4}\ s$, a noticeable deviation appears, demonstrating the loss of accuracy caused by over-suppression of high-frequency modes.

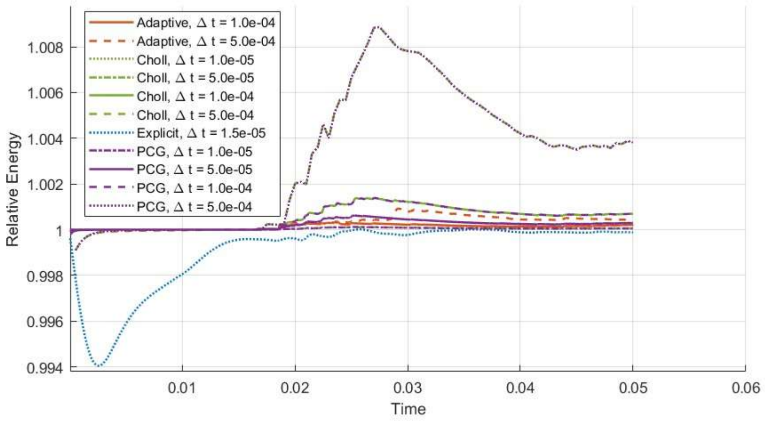

Figure 10: Relative energy during simulation for variety of solvers

In Figure 11, the displacement results obtained by the implicit solvers are compared with the reference explicit scheme. The curves confirm that there is no significant difference between the direct and iterative solvers. The loss of accuracy observed for $\Delta t = 5 \times 10^{-4} s$ is again consistent with the energy analysis, while a gradually increasing deviation is also visible for the smaller step $\Delta t = 1 \times 10^{-4} s$. In these cases, partially different fracture patterns were obtained. The adaptive solver, on the other hand, closely reproduces both the explicit solution and the finest implicit reference, demonstrating its robustness with respect to displacement prediction.

Finally, Figure 12 summarises the total computational cost. The variants that failed to provide representative results due to excessive damping suppression are indicated in red. The adaptive solver achieves the shortest runtime across all tested configurations, showing that the combined time-step and solver adaptation strategy leads to the most efficient solution of the benchmark problem.

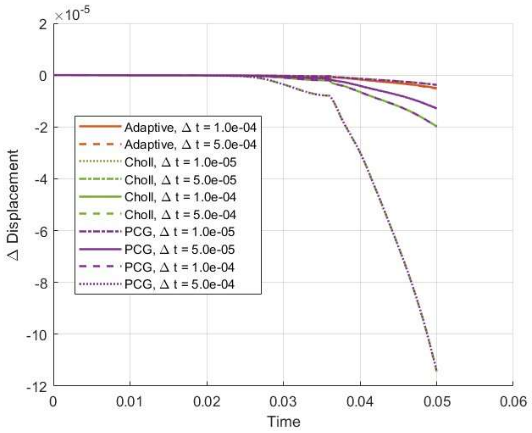

Figure 11: Comparison of maximum displacement between main explicit and implicit solvers

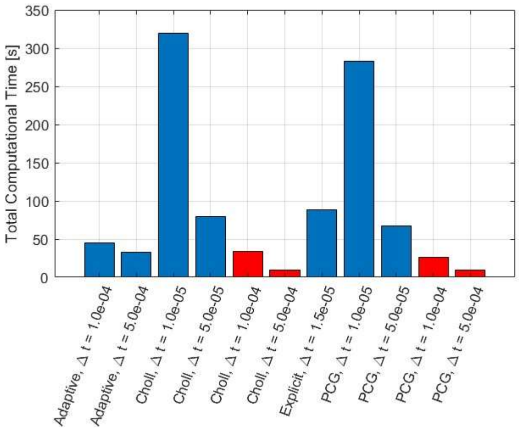

Figure 12: Total computation time over whole simulation

6 CONCLUSIONS

This contribution introduced an adaptive strategy for DEM-BBM simulations that combines solver selection with time-step control. The presented benchmarks confirmed that the adaptive approach achieves the fastest solution times while maintaining physical accuracy and stability. Unlike fixed explicit or implicit schemes, the adaptive solver consistently reproduced the correct fracture behaviour and energy response, yet required significantly less computational effort.

Although the diagonal estimate of the critical time step proved to be sufficiently accurate in the tested cases, its reliability benefits from the regular discretisation and should be examined further for irregular assemblies. Future work will also focus on extending the adaptive framework to plastic materials and heterogeneous material combinations, which are crucial for applications in civil engineering.

ACKNOWLEDGMENTS

The work was supported by the Student Grant Contest of VŠB-TUO. The project registration number is SP2025/089.

REFERENCES

[1] Cundall, P.A. *A computer model for simulating progressive large scale movements in blocky rock systems*. In *Proc. Int. Symp. Rock Fracture*, (1971), pp. 2–8.

[2] Obermayr, M., Dressler, K., Vrettos, C. and Eberhard, P. *A bonded-particle model for cemented sand*. *Computers and Geotechnics*, Vol. 49, (2013), pp. 299–313. Elsevier. doi:10.1016/j.compgeo.2012.09.001.

[3] Li, H., McDowell, G. and Lowndes, I. *Discrete-element modelling of rock comminution in a cone crusher using a bonded particle model*. *Géotechnique Letters*, Vol. 4, No. 2, (2014), pp. 79–82. doi:10.1680/geolett.14.00006.

[4] Chen, X., Wang, L.G., Morrissey, J.P. and Ooi, J.Y. *DEM simulations of agglomerates impact breakage using Timoshenko beam bond model*. *Granular Matter*, Vol. 24, No. 3, (2022), p. 74. Springer. doi:10.1007/s10035-022-01231-9.

[5] Newmark, N.M. *A method of computation for structural dynamics*. *Journal of the Engineering Mechanics Division*, Vol. 85, No. 3, (1959), pp. 67–94. American Society of Civil Engineers. doi:10.1061/JMCEA3.0000098.

[6] Liu, M. and Gorman, D.G. *Formulation of Rayleigh damping and its extensions*. *Computers & Structures*, Vol. 57, No. 2, (1995), pp. 277–285. doi:10.1016/0045-7949(94)00611-6.

[7] Brown, N.J., Morrissey, J.P., Ooi, J. and Chen, J.-F. *EDEM Contact Model: Timoshenko Beam Bond Model*. University of Edinburgh, Technical Report, (2015).

[8] Varga, R. and Čermák, M. *Calibration of beam bond model for the discrete element method*. In: Iványi, P., Kruis, J. and Topping, B.H.V. (eds.), *Proceedings of the Seventeenth International Conference on Civil, Structural and Environmental Engineering Computing*, Civil-Comp Press, Edinburgh, UK, Vol. CCC 6, (2023), Paper 7.2. doi:10.4203/ccc.6.7.2.

[9] Varga, R. and Čermák, M. *Implicit and explicit Newmark method for discrete element method – beam bond model*. In: Iványi, P., Kruis, J. and Topping, B.H.V. (eds.), *Proceedings of the Fifteenth International Conference on Computational Structures Technology*, Civil-Comp Press, Edinburgh, UK, Vol. CCC 9, (2024), Paper 10.2. doi:10.4203/ccc.9.10.2.

[10] O'Sullivan, C. and Bray, J.D. *Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme*. *Engineering Computations*, Vol. 21, No. 2–4, (2004), pp. 278–303. doi:10.1108/02644400410519794.