EFFECTS OF TIME-STEP SIZE ON THE EFFICIENCY OF DISCONTINIOUS DEFORMATION ANALYSIS

M.S. Khan[2], A. Riahi[1,] J.H. Curran[1, 3]
1. Rocscience Inc., Toronto
2. Schlumberger Canada Ltd.
3. Civil Engineering, University of Toronto

Abstract: This paper investigates the expected benefits of the implicit time integration scheme in the solution of jointed rock problems. Discontinuous Deformation Analyses (DDA) exploits the unconditional stability of the implicit time integration scheme, and allows the time step size of the solution to be much larger than the critical time step size dictated by the stability requirement of the explicit time integration schemes. This paper discusses the effects of large time step size on the DDA solution of engineering scale jointed rock problems. The findings are numerically verified by comparing the DDA solution to that of the Distinct Element Method (DEM).

1. Introduction

Heterogeneities in materials take various forms such as fractures, joints, bedding planes, voids, and material boundaries. Such features pose great challenges to understanding and predicting the complex range of behaviour of geomaterials. As a result, in recent years, much attention has been focused on efficient numerical techniques for analyzing jointed rock problems.

Three approaches are commonly used to model jointed rock masses. These are:

1. Discrete element techniques, and their combined discrete-continuum derivatives,
2. Combined continuum-interface methods, which are continuum methods with special joint/interface elements that model discontinuous displacement behaviour, and
3. Cosserat continuum.

In this paper we focus on the first category of methods. We will compare the computational efficiency of two widely-used discrete-based methods, the Discontinuous Deformation Analysis (DDA) and the Distinct Element Method (DEM), when applied to the analysis of jointed rock problems.

The fundamental difference between the two methods arises from their time discretization method. DDA uses an implicit time-integration scheme to solve the governing equations of a discrete problem. DEM is defined as a general term referring to discrete element methods that apply explicit time integration. Depending on the space discretization used, various DEM-based techniques have been developed.

Due to its implicit time integration, DDA is unconditionally stable and can accommodate considerably larger time steps than those of DEM. Explicit time integration, on the other hand, requires the time step to be smaller than a critical value to maintain stability of the solution. It has been argued in the literature that due to its larger time steps, DDA can be expected to solve jointed rock problems more efficiently than DEM. This paper intends to investigate the expected benefits of DDA. It will study the effects of time-step size on the solution times of DDA and DEM when applied to the analysis of engineering scale jointed rock problems. The examples examine the stability of slopes in jointed rock masses.

To make it possible to directly compare the influence of time step size on the solution time of DDA and DEM, other factors that affect solution time were measured and factored out. This paper uses UDEC numerical package [1], as the DEM representative application. Similar to DDA, UDEC discretizes the space domain into a number of polygonal blocks. Also, Shi's implementation of DDA [2] (which will be referred to as DDA-Shi throughout this text) is used as representative application of DDA.

Following this introduction, the governing equations of DDA and DEM will be discussed in Section 2. Section 3 focuses on the space discretization, while Section 4 explains the time discretization used in each method. Section 5 is focused on contact enforcement techniques. Section 6 investigates the effect of time step size on the DDA solution through a slope stability example. Finally, Section 7 provides a brief conclusion.

Throughout this paper, bold-faced fonts refer to vector or matrix quantities, while components of these quantities are described with non-bolded, italicized fonts with sub-indices. Indices denote components in a Cartesian coordinate frame.

2. Governing Equations of Motion in Discrete Systems

The equations governing the behaviour of a system of discrete blocks are the conservation of mass, conservation of linear and angular momentum, and the material constitutive equations. The motion and deformation of each individual block in a discrete system follows from the conservation of linear and angular momentum equations. The conservation of linear momentum equation is expressed by

$$-\frac{\partial u_j^2}{\partial t^2}\rho + \sigma_{ij,i} + b_j = 0\,, \tag{1}$$

where $\mathbf{u}$ is the displacement, $\boldsymbol{\sigma}$, the Cauchy stress tensor, $\mathbf{b}$, the body force, and ρ, the density of material, respectively.

Conservation of angular momentum, in the classical theory of elasticity, requires symmetry of the Cauchy stress tensor and minor symmetry of the elasticity tensor. Thus, angular momentum is implicitly satisfied on a characteristic volume when the components of the Cauchy stress tensor are reduced to six.

3. Space Discretization

Numerical techniques use computational schemes to solve spatially discretized forms of the governing equations through time. There are two equivalent forms of expressing the linear momentum equation. The first form, which comprises Equation (1) and traction boundary conditions, is called the strong form. The second form, known as the weak or variational form, is formulated through the principle of virtual work.

Similar to the Finite Difference Method (FDM), UDEC implementation of DEM discretizes the strong form of Equations (1). DDA, similar to Finite Element Method (FEM), discretizes the weak form. In the analysis of discrete problems, simulation of the deformability of a block requires application of continuum theory using FEM, FDM, or alternative methods.

The original DEM assumes that the blocks are rigid. The displacement field over each rigid block can then be represented by the rigid body displacement of a reference point on the block and the rotation of the discrete body about this point. To determine block rotations, the equation of angular momentum needs to be explicitly solved. DDA-Shi uses a first-order interpolation function to approximate the deformation field of a block from block vertex displacements. The displacement field over a block is then represented by displacement of a reference point on the block, rotation of the block about the axes passing through the reference point and a constant strain field. However, similar to rigid block DEM, since the motion of a volume is expressed by motion of a point, equilibrium of angular momentum needs to be explicitly satisfied to determine the rotation of the block. Modified versions of DDA and DEM for deformable blocks discretize each block into a finite element and finite difference mesh, respectively. In this case, the discrete forms of equations become identical to those of FEM and FDM.

4. Time Discretization

The semi-discretized form of Equation (1) for an assembly of blocks forms a general system of equations of the following form:

$$\mathbf{M\ddot{u}+C\dot{u}+ku=F}\,, \tag{2}$$

with the appropriate boundary conditions. In the above **M**, **C, K** are the global mass, damping and stiffness matrices, **F** is the time dependent applied force vector, and $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$, and **u** denote acceleration, velocity and displacement vectors, respectively.

4.1 DEM and Explicit Time Integration

DEM uses an explicit central difference time-marching scheme to solve Equation (2) through time. In the central difference scheme, the equilibrium of the system at time *t* is considered to calculate the

displacement at time $t+\Delta t$. The solution for the nodal point displacements at time $t+\Delta t$ is obtained using the central difference approximation for the accelerations, $\ddot{\mathbf{u}}_i^t$.

In general, the mass matrix on the left-hand side can be represented as a diagonal matrix. Equation (2) therefore can be rearranged as

$$m_i \ddot{\mathbf{u}}_i^t = (\mathbf{F}_i^t)^{out-of-balance}, \tag{3}$$

where *i* represents number of the degree of freedom. For the particular case of rigid block DEM, m_i becomes the mass of each block. The uncoupling of the equations of motion, which is one of the major advantages of explicit integration schemes, eliminates the need for assembly of global mass or stiffness matrices and inversion of the global matrices.

In rigid block DEM, at each time step, the kinematic variables, i.e., accelerations, velocities and displacements, are first calculated using a central difference scheme, and the dynamic quantities (contact forces or stresses, as well as internal stresses of the elements) are then obtained by invoking the constitutive relations for the contacts and the block materials.

$$\dot{u}^t_{(t+\Delta t/2)} = \dot{u}^i_{(t-\Delta t/2)} + \left(\sum F_t^i / m + g^i\right)\Delta t \text{ and } \dot{\theta}^i_{(t+\Delta t/2)} = \dot{\theta}^i_{(t-\Delta t/2)} + (\sum M_t / I)\Delta t . \tag{4}$$

where $\sum F_t^i$ is the sum of all the forces acting on the block including the damping force, $\dot{\theta}$ is the angular velocity of block about its centroid, $\sum M$ is the total moment acting on the block, $\dot{u}^i$ is the velocity components of the block centroid, and g^i is components of gravitational accelerations (body forces).
Explicit time integration is conditionally stable, which requires that the time-step size must be smaller than a certain critical value, Δt_c, for numerical errors not to grow unbounded. The critical time step is related to the time it takes for stress waves to travel across a block.

$$\Delta t_c \le 2/\omega_{\max}, \tag{5}$$

where ω_{max} is the highest eigenfrequency of the system.

4.2 DDA and Implicit Time Integration

DDA uses an implicit time marching scheme. Different variations of implicit time integration schemes have been developed. In implicit time integration schemes, acceleration and velocity components are expressed in terms of displacement components. In general terms, the displacement at time $t+\Delta t$ is obtained by rearranging the Equation (2) expressed for time $t+\Delta t$ in the following form:

$$\hat{\mathbf{K}}\Delta\mathbf{u}^{t+\Delta t} = \Delta\mathbf{F}^{t+\Delta t} \text{ or } \left[\frac{2\mathbf{M}}{\Delta\mathbf{t}^2} + \frac{2\mathbf{C}}{\Delta\mathbf{t}} + \mathbf{K}\right]\mathbf{u}_{\mathbf{n}+1} = \mathbf{F}_{\mathbf{n}+1} + \frac{2\mathbf{M}}{\Delta\mathbf{t}}\dot{\mathbf{u}}_{\mathbf{n}} + \mathbf{C}\dot{\mathbf{u}}_{\mathbf{n}}, \tag{6}$$

The solution of Equation (6) requires assembling the global mass and stiffness matrices and solving the coupled system of equations using a direct matrix inverse operation or an iterative solver. The global stiffness matrix, $\mathbf{K}$, includes the sub-matrix representing deformability of blocks and contacts, with contact matrices as off-diagonal terms.
DDA uses the Newmark-β method [3], one of the generalized Newmark integration schemes, which provides algorithmic damping (numerical dissipation). Therefore, the explicit damping term, **C**, in DDA-Shi is assumed to be zero.

5. Contact energy, force, and stiffness

Mathematically, contact is treated as a constraint on displacements at the interface between two objects. A normal contact constraint prevents interpenetration of objects, while a tangential constraint enforces sticking/slipping. Various numerical techniques have been developed to satisfy contact constraint conditions [4, 5]. One of the most common approaches is the penalty method that satisfies these constraints approximately. The approximate enforcement of a constraint by the penalty method is achieved through a

proportionality law or penalty function that relates the degree of constraint violation to the size of the corrective measure. Any surface penetration violates the impenetrability constraint and invokes contact forces that tend to return the surfaces to a state of compliance with the imposed constraints. Similarly, tangential penalty forces are developed as a result of relative tangential displacements at the contacting surfaces. From the approximate enforcement of the normal impenetrability and tangential sticking constraints, the following potential energy terms arise:

$$\Pi^{contact} = \frac{1}{2} d_n^2 \alpha_n + \frac{1}{2} d_t^2 \alpha_t \,, \tag{7}$$

where d_n and d_t are the normal and tangential displacements of a contact point on the boundary, measured with respect to the target surface with normal $\mathbf{n}$, α_n and α_t are the normal and tangential penalty coefficients, respectively, in force/length dimensions.

Another technique for simulating contact is the soft (compliant) contact approach. In this approach, penetration displacements are not constrained. This approach assumes that springs exist at the contacts, and permits infinitesimal penetrations. Associated forces are then calculated using the constitutive laws of the springs. Assuming a linear constitutive spring relationship, $F = k\Delta l$, the potential function for each contact point becomes

$$\Pi^{contact} = \frac{1}{2}(k_n d_n^2 + k_t d_t^2) \,, \tag{8}$$

were k_n and k_s are the normal and shear stiffnesses at the contact.

DDA enforces the no-penetration constraint approximately through a penalty function. It requires a number of iterations to adjust the penetration distance to a specified value. In addition, both the force vector and stiffness matrix need to be calculated. Minimization of potential energy with respect to displacements gives the stiffness and force terms arising from contact. Details on the mathematics and coefficients of these matrices can be found in [2, 6, 7]. UDEC is based on a soft contact approach. Since the method uses an explicit time integration, calculation of contact stiffness is not required, and forces can be directly calculated by evaluating the penetration and relative tangential distances using $F_n = k_n d_n$ and $F_s = k_t d_t$.

Given that the penalty coefficients and the spring stiffnesses have the same dimensions, Equations (7) and (8) are identical. Also, both DDA-Shi and UDEC apply similar technique for contact resolution by classifying contact modes into node-to-node, node-to-edge, and edge-to-edge and evaluating the penetration distance. The differences between the contact formulations of the two methods arise from (i) calculation of contact stiffness matrix in DDA-Shi, which is an inherent requirement for implicit time integration, and (ii) enforcement of a threshold for normal penetration in DDA, which results in an iterative procedure often referred to as “open-close” iterations.

6. Numerical example on the comparison of DDA and DEM

In the previous sections, three fundamental aspects of the solution of discrete problems, governing equations, space and time discretization, and contact force and stiffness calculation were discussed. In this section, we investigate the effect of time step size on the solution of DDA and DEM using an example of slope stability analysis in jointed rock.

6.1 Slope stability problem

In this example, a slope with a height of 260 *m* and a slope face angle of 55° is considered. It has two sets of joints, a horizontal set with 40 *m* spacing and another set parallel to the slope face with 10 *m* spacing as shown in Figure 1. The block material and joint properties are given in Table 1 and the input parameters used in DDA and DEM are provided in Table 2.

DEM’s representative program (UDEC version 4.0) and Shi’s DDA program version 2002 (DDA-Shi) are used to analyze this example. In UDEC, the “solve” command is used to automatically detect the steady state, while in DDA, steady-state is assumed to have been achieved when the maximum relative

displacement becomes less than $1x10^{-8}$ [8]. Both analyses were run on the same computer with the system configuration: processor speed = 3.0 GHz and RAM = 1.0 GB.

To reach the equilibrium state, DEM executed 3050 time steps with a time step size Δt of 0.00062. This took less than a minute of CPU time. Vertical displacement of the crest block at steady state is shown in Figure 2. DDA-Shi, on the other hand, ran through 10750 time steps with a specified time step size Δt of 0.001. This took 179 minutes of CPU time (Figure 3). Although the DDA time step size was almost double that of DEM, DDA took significantly longer time to reach equilibrium state.

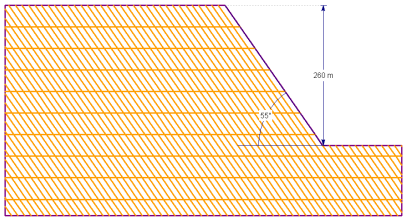


Figure 1: Geometry of the slope.

Table 1. Block material and joint properties for slope stability example

Block material properties
Unit weight: 26.1 KN/m^3 Friction angle: 43° Cohesion: 675 KPa Tensile strength, σ_t : 0
Joint properties
Friction angle = 40°, Cohesion = 100 KPa

Table 2. Input parameters used in DEM and DDA for slope stability example

DEM input parameters
k_n : 10 GPa/m and k_s : 2.5 GPa/m Time step, Δt: 0.00062 sec.

DDA input parameters
Penalty value:$1x10^{12}$, Δt^{max} : 0.001 sec. Max. allowable incremental displacement: 0.001m, Dynamic coefficient: 1.0

In order to investigate whether the observed slowness is due to the inherent characteristics of implicit time integration, the effects of several technical and implementation differences must be factored out. They are:

1- Contact search algorithm implemented in each method,

2- Type of solver used in DDA,

3- Open-close iterations on DDA, which enforces a threshold on the value of block penetration. As a result of open-close iterations for this example, DDA solves a system of 2892 (6×482) coupled equations multiple times during each time step,

4- The three additional degrees of freedom in DDA to represent deformability of blocks through a constant strain field. As a result, for this example, with 482 blocks, DEM solves 1446 (3×482) uncoupled equations in each time step, while DDA solve 2892 (6×482) coupled equations.

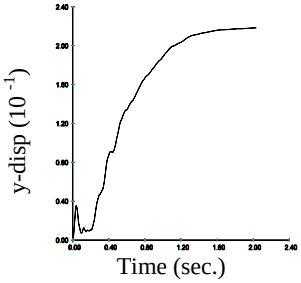


Figure 2. Slope stability example steady state DEM solution, Δt = 0.00062 sec., Time steps = 3050.

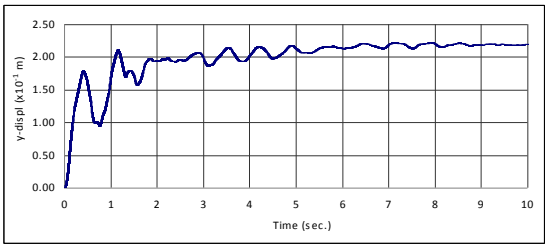


Figure 3. Slope stability example steady state DDA-Shi solution Avg. Δt = 0.001 sec., Time steps = 10750.

Contact Search Algorithm

In order to investigate the effect of the contact search algorithm on solution time, for the geometry of this example (Figure 1), over a dozen different cases with the number of blocks ranging from 60 to 39,070 have been conducted using UDEC and DDA. Contact detection time as a function of the total number of blocks comprising the problem is shown in Figure 4.

The results show that the contact detection implementations of UDEC and DDA take similar CPU time when the number of blocks is about 1000. The contact search algorithm based on orientation theory used in DDA is more efficient when the number of blocks is about 1000 but for larger systems, the domain network contact search algorithm used in UDEC is more efficient.

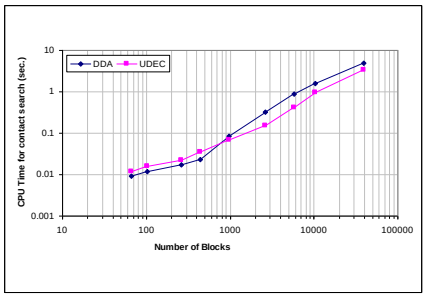


Figure 4. CPU time for contact search versus number of blocks.

Effect of solver on DDA computational efficiency

There are various direct and iterative solvers available for matrix solution [9]. The appropriate choice of solver depends on the characteristics of the algebraic system of equations and the size of the system. In this work, we investigated the efficiency of both direct and iterative solvers for DDA's system of algebraic equations for problems involving a large number of blocks.

Two direct solvers: (i) Shi's Graph theory based solver (Graph) [10] available in DDA-Shi, and (ii) Gaussian Elimination (Gauss) solver as well as three iterative solvers: (iii) the Successive Over-Relaxation solver (SOR) available in DDA-Shi (iv) Conjugate Gradient (CG), which exploits the symmetry and positive definite characteristics of the global stiffness matrix formed in DDA [11], and (v) the commercially available Conjugate Gradient solver from Intel® Math Kernel Library 10.0 (Intel MKL)) [12] were examined.

To assess the efficiency of the aforementioned solvers, a number of physically stable problems have been analyzed with various numbers of blocks. To estimate average CPU time taken for each iteration, the DDA analysis was run for 100 time steps and the total CPU time was divided by the total number of iterations performed in 100 time steps. Figure 5 compares solver speed versus the number of degrees of freedom (NDOF). It suggests three different ranges in terms NDOF. It is clear from this figure that choosing an efficient solver depends on the size of the problem. SOR/CG was identified as the most efficient solver considering the number of blocks in this example, and therefore was used in the rest of this investigation.

Strain degrees of freedom and modified DDA-Rigid Block

In order to investigate the fundamental cause for the observed differences in the solution time of DDA and DEM, it is essential that the problem being solved in both applications be identical. To achieve this DDA-Shi was modified to represent blocks as rigid objects with two translational and one rotational degree of freedom [11, 13]. As a result, DDA-RB solves a system of equations of the same size compared to rigid block DEM.

Vertical displacement of the crest block at steady state is shown in Figure 6. This figure shows that although DDA-RB took approximately less than 15% of CPU time compared to DDA-Shi., its solution time is still 25 times more than that of DEM.

Considering other identified sources of difference between the two applications, i.e., contact detection algorithms and differences in the number of degree of freedom are factored out, it can be concluded that the observed discrepancy can only be attributed to inherent characteristics of the implicit time integration scheme and the penalty-constraint contact approach used in DDA. It must be emphasized that in implicit time integration schemes, the type of solver plays an important role in time and memory efficiency. This investigation was therefore performed using the most efficient solver identified for the problem under consideration.

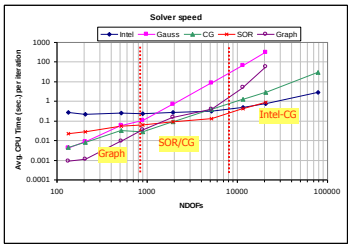


Figure 5. Comparison of solver speed in DDA.

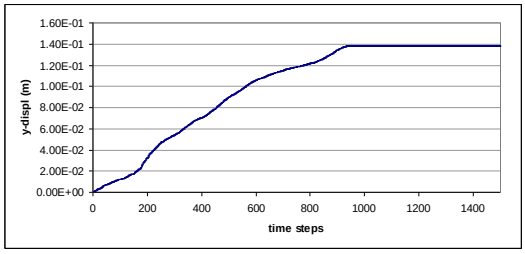


Figure 6. Steady state for DDA-rigid block avg. Δt = 0.0032 sec., Time steps = 985.

Damping characteristics of DDA solution

In this example, DEM used a time-step size of 0.00062 and reached steady state within less than a minute while DDA used Δt=0.0032 and achieved steady state in 16.5 minutes (with CG solver). Although DDA used a time step almost 5 times larger than DEM it could not match the speed of DEM.

In order to study the effect of time-step size on the computational speed of DDA, this example was solved using 5 different values for time step (from 0.001 to 0.1 sec.). For each time-step size the DDA analysis ran until steady state was reached. The relation between the CPU time taken to reach steady state for each Δt (Figure 7) indicates that there is range of Δt time-step sizes for which DDA CPU time is minimized. Values of Δt greater than this range result in considerably more CPU time.

To better investigate this phenomenon a Single Degree of Freedom (SDOF) example is considered, in which a gravity force is applied to a block, and, as expected, it is observed that the mass starts to oscillate indefinitely around its equilibrium position. However, if an appropriate damper is used in the model, the mass comes to rest at its equilibrium position. Figure 8 shows the variation of CPU time versus time step size for the SDOF example. Figure 9 shows the oscillating velocity of the mass as a function of time with respect to four different time-step sizes.

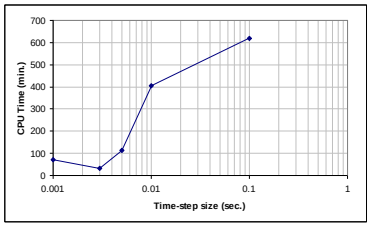


Figure 7. CPU time verses time-step size in DDA to reach steady state.

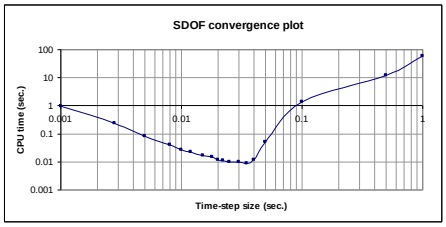
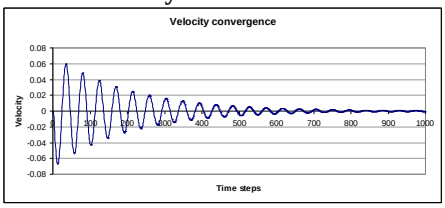
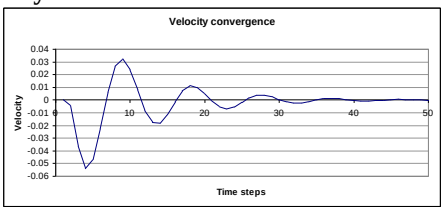


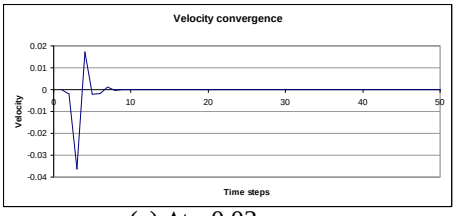
Figure 8. CPU time verses time-step size in DDA to reach steady state in SDOF.



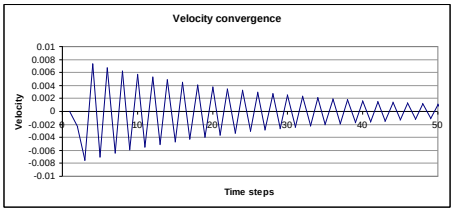
(a) Δt==0.001 sec.



(b) Δt =0.005 sec.



(c) Δt =0.03 sec.



(d) Δt =0.1 sec.

Figure 9. SDOF velocity convergence for various time-step size.

This figure indicates that initially the convergence time decreases as the time step increases, i.e., the larger the time step the faster the convergence. But after reaching a certain time-step size any further increase in the time step introduces oscillations. For $\Delta t=0.001$ to approximately 0.03 sec. convergence time decreases, while for $\Delta t>0.03$, convergence time increases (the full range of variation of CPU time with respect to time step size is shown in Figure 8).

Comparisons of results in Figure 9 shows that although the DDA solution is stable independent of the time step size, its oscillatory response is greatly affected by the time step size. It is shown [6] that the algorithmic damping characteristic of DDA solution can be divided into four distinct regions: *no damping*, *under damping*, *over damping*, and once again *no damping* regions, depending on the time step size. Very large values of Δt lead to little damping and high oscillation of the solution.

Open-close iterations

Open-close iteration (OCI) in DDA refers to the iterative procedure for adjusting contact penetration distance to a specified threshold value. To illustrate the effects of open-close iterations, the slope stability analysis was conducted with different Δt values; each analysis was stopped at the same physical time.

Figure 10 shows the total number of OCIs required to reach convergence for various specified Δt. This figure indicates that DDA converges quickly for small Δt values. For example, for $\Delta t=0.001$, an average of 3 OCIs lead to convergence while for $\Delta t>0.005$, 20-30 OCIs were required to reach convergence.

The observed pattern was further established by investigating the OCI/Δt relationship in the slope stability example under consideration with eight different cases of joint orientations and joint spacing. Each problem was run until steady state was attained; the total number of OCIs and time steps taken to attain steady state were recorded. The average number OCIs per time step as a function of Δt for each problem is plotted in Figure 11. This figure shows that in general OCIs increase as Δt increases. The increase in OCIs becomes more pronounced as the number of blocks increases. In this example, it seems that for problems with less than 200 blocks, OCIs did not increase much with a larger time step, for problems with 500 blocks or more, the increase in OCIs is dramatic.

It must be noted that DDA-Shi considers the specified time step size merely as an upper limit. The actual time step size is determined considering the contact convergence and the maximum allowable displacement in each iteration. Contact convergence requires that the contact state be constant in two consecutive iterations. In DDA-Shi, if contact convergence is not achieved typically within six iterations, the time step is reduced by one third and the analysis is repeated with the reduced time step. For large Δt, block penetrations at the contact points will be large, which will increase the reaction forces maintaining the no-penetration conditions at the contact points. In each open-close iteration, the stiffness matrix is adjusted/corrected to represent the new contact state, then the next iteration begins with the corrected stiffness matrix and the process continues until convergence is attained.

In addition, large Δt results in large displacement and thus violation of small strain theory. The formulation of DDA is based on small strain theory such that large displacements and large deformations are the result of accumulation of small displacements in numerous time steps. DDA-Shi restricts the incremental displacement with a user-specified displacement limit to enforce infinitesimal displacements. If the incremental displacement is greater than the threshold, Δt is divided by three and the analysis is repeated.

Because of the frequent time step cuts, the actual time-step size used in the analysis varies. The average actual time-step used for different specified (upper limit) values of Δt in the slope stability problem is shown in Table 3 and Figure 12. These results indicate that although a larger time step is specified in DDA, the average time-step size is much smaller than that of DEM (0.00062 sec). Frequent time step cuts not only decrease the Δt but also increase cumulative OCIs.

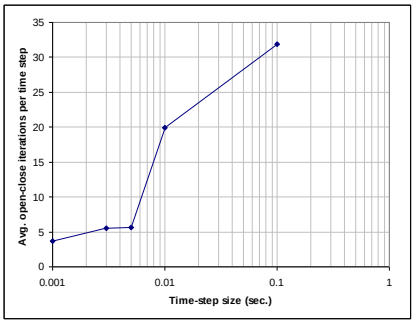


Figure 10. Δt-OCI relationship.

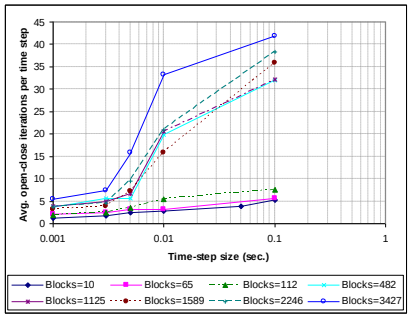


Figure 11. Average number of open-close iterations per time step.

Table 3. Specified time-step size and actual (average) time-step size

Specified time step size (Δt)	Avg. Actual time step used
0.001	0.000424
0.003	0.000483
0.005	0.000405
0.01	0.000376
0.1	0.000129

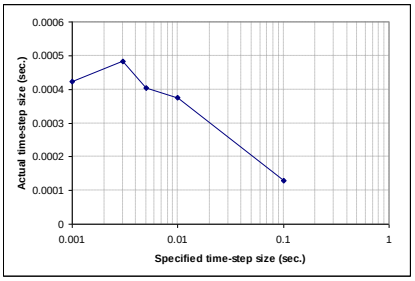


Figure 12. Specified time-step size and actual time-step size used.

5. Conclusion

We investigated the expected benefits of implicit time integration in the DDA solution of jointed rock problems. Although the Newmark implicit time integration scheme adopted by DDA ensures numerical stability unconditionally for any time step size, from a practical point of view a number of factors constrain time step size:

1. In DDA-Shi, the only source of damping is algorithmic damping, which is controlled by the time step size. Frequent cuts in time step size vary algorithmic damping which may result to an under-damped or over-damped solution. Also, large values of Δt can lead to minimal algorithmic damping resulting in an oscillatory response.

2. Large values of Δt may cause large penetrations at contact points; which results in more iterations to satisfy the penetration threshold. Also, large penetrations result in large contact matrices which can reduce the diagonal dominance of the global stiffness matrix leading to poorly conditioned system of equations.

3. Large values of Δt result in violation of small strain theory adopted by DDA.

To satisfy the small strain assumption and also to enforce contact convergence, DDA uses a recursive procedure to reduce the time step size. The recursive process increases the number of iterations required for each time step and results in actual time step sizes that are much smaller than the specified values. Building global matrices and carrying out matrix inversion at each iteration greatly affects the solution time, an aspect that becomes more pronounced especially in large scale problems.

This paper, through a systematic investigation and by factoring out other technical and implementation aspects, showed that in large scale jointed rock problems, DDA solution is slower than the solution time for methods that utilize an explicit time integration scheme. This is caused by the combination of the highly nonlinear behavior of contact problems and the size of the global system of equations for real-scale geomechanics problems. The time step chosen must balance the desirable algorithmic damping and the undesirable consequences mentioned above. The efficiency of the matrix solvers, however, plays a key role in the solution time of implicit time integration methods, and therefore considerable improvements may occur in the future.

References

[1] Itasca Consulting Group, UDEC - Universal Distinct Element Code, Version 4.0 - User's Manual, Minneapolis, MN, USA, 2004.
[2] Shi, G.H., Discontinuous deformation Analysis, Ph.D. thesis, University of California, Berkeley, USA, 1988.
[3] Wang, C.Y., Chuang, C.C., Sheng, J. Time integration theories for the DDA method with finite element meshes. In *Proceedings of 1st Int. Forum on Discontinuous Deformation Analysis (DDA) and Simulations of Discontinuous Media*, Berkeley, CA, M. R. Salami and D. Banks, eds., TSI, Albuquerque, N.M., 263–287, 1996.
[4] Mohammadi, S., Discontinuum mechanics, Using Finite and Discrete Elements, WIT Press, Southhampton, UK, 2003.
[5] Wriggers, P., Computational contact mechanics, Hoboken, N.J., John Wiley & Sons, 2002.
[6] Khan, M.S., Investigation of discontinuous deformation analysis for application in jointed rock masses, Ph.D. Dissertation, University of Toronto, 2010.
[7] Riahi A. Curran, J.H., Hamma, R., Limits of applicability of the finite element explicit joint model in the analysis of jointed rock problems, in the *Proceedings of the 44th ARMA Conference*, Salt Lake City, 2010.
[8] Shi, G.H. Application of Discontinuous Deformation Analysis (DDA) to rock stability analysis. In *Proceedings of 8th international conference on analysis of discontinuous deformation – Fundamentals & Application to mining and civil engineering* (*ICADD-8*), Beijing, China, Aug. 14-19, 1-13. Ju, Y., Fang, X. and Bian, H (Eds.), 2007.
[9] Saad, Y. Iterative Methods for Sparse Linear Systems. PWS publishing company, Boston, USA, 1996.
[10] Shi, G.H. Block system modeling by discontinuous deformation analysis, *Computation Mechanics*, Southampton, U.K, 1993.
[11] Koo, C.Y., Chern, J.C., Modification of the DDA method for rigid block problems, *International Journal of Rock Mechanics and Mining Sciences*, 35(6):684–693, 1998.
[12] Intel http://www.intel.com/cd/software/products/asmo-na/eng/266853.htm, 2006.
[13] Jing, L., Stephanson, O. Fundamentals of discrete element methods for rock engineering - Theory and Applications. *Developments in Geotechnical Engineering*, 85. Elsevier, 2007.

Biography

Mohammad Safdar Khan obtained his Ph.D. from the Civil Engineering Department of the University of Toronto, in 2010. His Ph.D. research was focused on the application of Discontinuous Deformation Analysis to the engineering scale jointed rock problems. He is currently working as a Senior Geomechanics Specialist with Schlumberger Canada. He is engaged on a number of unconventional reservoir geomechanics projects dealing with anisotropic stress profling, wellbore stability and stimulation design for shale gas and tight sand; caprock integrity analysis, coupled thermal reservoir geomechanical modeling, casing and well integrity, reservoir compaction and subsidence in heavy oil reservoirs.

Azadeh Riahi obtained her Ph.D. from the Department of Civil Engineering, University of Toronto in 2008. Since then, she has held an NSERC Industrial Research and Development Fellowship (IRDF) at Rocscience Inc., a leading geotechnical software and applied geomechanics research company. Her research is focused on Computational Geomechanics – the application of Computational Mechanics tools to the solution of geotechnical and mining engineering problems. Her interests include Cosserat continuum theory and its application to the analysis of particulate, layered and blocky materials. Discrete element and combined continuum-discontinuum methods, and their application to the analysis of rock masses.

John H. Curran is R. M. Smith Professor Emeritus in Civil Engineering, University of Toronto where he taught for 32 years. He received his Ph.D. in Mechanical Engineering from the University of California, Berkeley in 1976. In 1996 he started Rocscience Inc., a spinoff company from the University of Toronto, which develops practical software tools for rock and soil, is currently being used in over 110 countries. His current research is focused on numerical modeling of engineering structures in jointed rock masses, FEM-DEM, and rapid simulation of rockfalls based on impact mechanics.