

AN EFFICIENT FORMULATION FOR THE SIMULATION OF ELASTIC WAVE PROPAGATION IN 1-DIMENSIONAL COLLIDING BODIES

Alexandre Depouhon*¹, Emmanuel Detournay², Vincent Denoël³

¹Université de Liège, University of Minnesota
alexandre.depouhon@ulg.ac.be

²University of Minnesota, CSIRO Earth Science and Resource Engineering
detou001@umn.edu

³Université de Liège
v.denoel@ulg.ac.be

Keywords: Elastic impact, Wave propagation, Contact algorithm, Penalty method, Percussive drilling, Event-driven scheme.

Abstract. *This paper proposes an efficient formulation for the simulation of waves propagating in the down-the-hole percussive drilling systems typically used in mining applications. In particular, it treats the problem of elastic wave propagation in 1-dimensional colliding bodies.*

The presented algorithm performs the time integration of the equations of motion arising from the space semi-discretisation of the continuous problem by the Galerkin method, using a dissipative midpoint rule as to the time dimension, and builds its efficiency on the innovating handling of the contact opening/closure events. Specific to our approach is the use of an event-driven scheme to perform the time integration of the discrete equations of motion, combined with a penalty-based definition of the contact forces. This formulation, most adapted to systems presenting a low number of contact interfaces, takes its efficacy in the exploitation of the linearity of the governing equations between events, and is based on recasting the time integration procedure into an explicit scheme between contact status changes.

*The proposed algorithm, also declined in its conservative form, is tested on the elastic bouncing bar benchmark problem proposed by Doyen *et al.* [1]. Further application is made to the system of down-the-hole percussion drilling that consists of a piston impacting a drill bit in contact with the rock, with the bit/rock interaction modeled by a dissipative bilinear contact law.*

1 INTRODUCTION

In this paper, we propose an efficient framework for the simulation of the elastic waves propagating in a down-the-hole (DTH) hammer, which is a rotary percussion device used to drill boreholes in hard rock. Here, the DTH hammer is modeled as a 2-body system with state-dependent piecewise linear contact laws. More generally, the developed framework is applicable to any 1-dimensional mechanical structure resulting in linear equations (but for the contact forces) after space semi-discretisation by the finite element method and enforcement of the unilateral constraints by a quadratic contact potential, i.e. the following class of equations

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{u} = \mathbf{f}^{ext} + \sum_{i \in \mathcal{C}} \mathbf{f}_i^{con}, \quad (1)$$

where the stiffness, damping and mass matrices (\mathbf{K} , \mathbf{C} , \mathbf{M}) are constant. The displacement, velocity and acceleration vectors are referred to by \mathbf{u} , \mathbf{v} , $\dot{\mathbf{v}}$, and the unilateral constraints can be expressed by the forms

$$\mathbf{f}_i^{con} = -k_i^{con} \mathbf{s}_i [{}^0g_i + \mathbf{s}_i^T \mathbf{u}]_-, \quad i \in \mathcal{C}, \quad (2)$$

with $[x]_- = \min(x, 0)$ denoting the Macaulay brackets, 0g_i a scalar number that represents the initial gap related to contact interface i , and \mathbf{s}_i a column vector selecting the active degrees of freedom at contact interface i . The index set \mathcal{C} spans the whole of the contact interfaces.

The paper is organised as follows. First, we introduce the computational algorithm and detail its specificities. Second, we apply it to the benchmark problem of an elastic bar bouncing in the field of gravity. Accuracy and efficiency are assessed by comparison with the analytical solution as well as state-of-the-art integration procedures. Then, we apply the developed framework to the simulation of wave propagation in DTH percussion drilling systems and study the influence of impact conditions on the post-impact velocities of the drill bit and the piston. Finally, we summarize our findings in the last section of the paper.

2 EVENT-DRIVEN INTEGRATION SCHEME

Let us assume the motion of an elastic system is described by equations of motion (1). The common procedure of structural dynamics to march these equations in time is to apply an integration scheme dedicated to the integration of second-order ODEs, e.g. HHT or Chung-Hulbert. However, in the absence of sufficient numerical and/or structural damping, the straightforward combination of these schemes with a penalty-based relaxation of the unilateral constraints conducts to instabilities driven by the non-vanishing work of the contact forces at the closures and openings of the contact interfaces [1]. To overcome this source of instability, Armero and Petócz [2] have proposed a conservative definition of the penalty-based contact force to be coupled with a conservative midpoint time discretisation scheme, i.e. with this integration procedure, the work of the contact force vanishes over a closure/opening cycle of the contact interface while the scheme dissipates no energy numerically. Further work on the topic was conducted by Armero and Romero [3] who rendered the scheme numerically dissipative by introducing terms related to fictitious state variables in the equations of motion.

Algorithm 1 Event-detection integration scheme.

Given active constraints, \mathcal{C}_n , state vectors, $\mathbf{u}_n, \mathbf{v}_n$, iteration matrices, $\mathbf{Z}_i, \mathbf{H}_i$, and vector, \mathbf{f}_n , at t_n :

1. Solve residual equation for \mathbf{u}_{n+1} using appropriate method

$$\mathbf{Z}_1 \mathbf{u}_{n+1} = \mathbf{f}_n - \mathbf{Z}_2 \mathbf{u}_n - \mathbf{Z}_3 \mathbf{v}_n.$$

2. Update velocity field

$$\mathbf{v}_{n+1} = \mathbf{H}_1 \mathbf{u}_{n+1} + \mathbf{H}_2 \mathbf{u}_n + \mathbf{H}_3 \mathbf{v}_n.$$

3. Assess constraint statuses:

If any changes status:

- Locate exact occurrence of earliest event.
- Update set \mathcal{C}_{n+1} and form iteration matrices and vector accordingly.

Else:

- Proceed to next time step unless final time has been reached.
-

Though it ensures the energetic stability, the contact force defined by Armero and Petőcz introduces a nonlinearity in the governing equations. While this does not have a major computational impact in the framework of finite transformations (for which it was originally developed), it does in the linear setting of equation (1). To conserve the linearity of the governing equations, we propose instead to use the usual linear definition of the penalty-based contact force in combination with a dissipative midpoint scheme while accurately tracking the closures and openings of the contact interfaces. This enables to reduce the work of the contact force over a closure/opening cycle down to a chosen precision threshold while retaining the linear structure of the governing equations.

2.1 INTEGRATION PROCEDURE

In virtue of the definition of the contact forces, equations of motion (1) is piecewise linear, the points of non-smoothness corresponding to the zeros of the gap functions $g_i = {}^0g_i + \mathbf{s}^T \mathbf{u}$, $i \in \mathcal{C}$. If, however, the equation is considered on a restricted domain over which no contact interfaces change status, it can be treated as a linear equation. This naturally calls for the integration procedure given in Algorithm 1: given the index set of active constraints \mathcal{C}_n , a subset of \mathcal{C} , integrate the linear equations of motion using either the conservative or dissipative midpoint rule until the next event (any change of contact status) occurs in the system; in that case, update constraint statuses and proceed further with the updated index set of active constraints.

To derive the iteration matrices and vector required by the integration procedure, we apply the discretisation rule of the dissipative midpoint scheme as proposed by Armero and Romero [3]. This leads to the following set of equations, to be solved for $\mathbf{u}_{n+1}, \mathbf{v}_{n+1}$,

$\tilde{\mathbf{u}}_n$ and $\tilde{\mathbf{v}}_n$ at each time step

$$\left\{ \begin{array}{l} \mathbf{M}\dot{\mathbf{v}}_{n+1/2} = \mathbf{f}_{n+1/2}^{ext} - \sum_{i \in \mathcal{C}_n} k_i^{con} g_i \mathbf{s}_i - \mathbf{C}\mathbf{v}_{n+1/2} - \mathbf{K}\mathbf{u}_{n+1/2} - \mathbf{K}^T \frac{\tilde{\mathbf{u}}_n - \mathbf{u}_n}{2}, \\ \dot{\mathbf{u}}_{n+1/2} = \frac{\mathbf{v}_{n+1} + \tilde{\mathbf{v}}_n}{2}, \\ \tilde{\mathbf{u}}_n = \mathbf{u}_n + \chi \Delta t (\tilde{\mathbf{v}}_n - \mathbf{v}_{n+1}), \\ \mathbf{M}\tilde{\mathbf{v}}_n = \mathbf{M}\mathbf{v}_n - \chi \Delta t \mathbf{K} (\tilde{\mathbf{u}}_n - \mathbf{u}_{n+1}), \end{array} \right. \quad (3)$$

where $\mathbf{x}_{n+1/2} = (\mathbf{x}_{n+1} + \mathbf{x}_n)/2$ and $\dot{\mathbf{x}}_{n+1/2} = (\mathbf{x}_{n+1} - \mathbf{x}_n)/\Delta t$, $\mathbf{x} \in \{\mathbf{u}, \mathbf{v}\}$, and Δt denotes the time step. Rather than solving this system of dimension twice that of the state space, we rewrite the force residual and the velocity update as a function of the past state variables and the unknown displacement field at the end of the timestep, \mathbf{u}_n , \mathbf{v}_n and \mathbf{u}_{n+1} respectively,

$$\left\{ \begin{array}{l} \mathbf{0} = \mathbf{Z}_1 \mathbf{u}_{n+1} + \mathbf{Z}_2 \mathbf{u}_n + \mathbf{Z}_3 \mathbf{v}_n - \mathbf{f}_n, \\ \mathbf{v}_{n+1} = \mathbf{H}_1 \mathbf{u}_{n+1} + \mathbf{H}_2 \mathbf{u}_n + \mathbf{H}_3 \mathbf{v}_n, \end{array} \right. \quad (4)$$

by exploiting relations (3b)-(3d). After a bit of algebra, the iteration matrices and vector are given by

$$\left\{ \begin{array}{l} \mathbf{Z}_1 = \frac{1}{\Delta t} \left(\frac{1}{\Delta t} \mathbf{M} + \frac{1}{2} \mathbf{C} \right) (2\mathbf{I} - \tilde{\mathbf{Q}}_{21}) + \frac{1}{2} \left(\mathbf{K} + \sum_{i \in \mathcal{C}_n} k_i^{con} \mathbf{s}_i \mathbf{s}_i^T + \mathbf{K}^T \tilde{\mathbf{Q}}_{11} \right), \\ \mathbf{Z}_2 = -\frac{1}{\Delta t} \left(\frac{1}{\Delta t} \mathbf{M} + \frac{1}{2} \mathbf{C} \right) (2\mathbf{I} + \tilde{\mathbf{Q}}_{22}) + \frac{1}{2} \left(\mathbf{K} + \sum_{i \in \mathcal{C}_n} k_i^{con} \mathbf{s}_i \mathbf{s}_i^T + \mathbf{K}^T (\tilde{\mathbf{Q}}_{12} - \mathbf{I}) \right), \\ \mathbf{Z}_3 = -\frac{1}{\Delta t} \mathbf{M} (\mathbf{I} + \tilde{\mathbf{Q}}_{23}) + \frac{1}{2} \mathbf{C} (\mathbf{I} - \tilde{\mathbf{Q}}_{23}) + \frac{\Delta t}{2} \mathbf{K}^T \tilde{\mathbf{Q}}_{13}, \end{array} \right. \quad (5)$$

and

$$\mathbf{H}_1 = \frac{1}{\Delta t} (2\mathbf{I} - \tilde{\mathbf{Q}}_{21}), \quad \mathbf{H}_2 = -\frac{1}{\Delta t} (2\mathbf{I} + \tilde{\mathbf{Q}}_{22}), \quad \mathbf{H}_3 = -\tilde{\mathbf{Q}}_{23}, \quad (6)$$

and

$$\mathbf{f}_n = \mathbf{f}_{n+1/2}^{ext} - \sum_{i \in \mathcal{C}_n} k_i^{con} g_i \mathbf{s}_i, \quad (7)$$

with

$$\left\{ \begin{array}{ll} \tilde{\mathbf{Q}}_{11} = 2\chi \mathbf{S}_c^{-1} (\chi \Omega - \mathbf{I}), & \tilde{\mathbf{Q}}_{21} = \chi \Omega (\mathbf{I} - \tilde{\mathbf{Q}}_{11}), \\ \tilde{\mathbf{Q}}_{12} = (1 + 2\chi) \mathbf{S}_c^{-1}, & \tilde{\mathbf{Q}}_{22} = -\chi \Omega \tilde{\mathbf{Q}}_{12}, \\ \tilde{\mathbf{Q}}_{13} = 2\chi \mathbf{S}_c^{-1}, & \tilde{\mathbf{Q}}_{23} = \mathbf{I} - \chi \Omega \tilde{\mathbf{Q}}_{13}, \end{array} \right. \quad (8)$$

$$\Omega = \Delta t^2 \mathbf{M}^{-1} \mathbf{K} \text{ and } \mathbf{S}_c = (\mathbf{I} + 2\chi^2 \Omega).$$

The energy-conservative version of the formulation is obtained by setting $\chi = 0$ in the above equations; in which case, the iteration matrices simplify to

$$\mathbf{Z}_1 = \frac{2}{\Delta t^2} \mathbf{M} + \frac{1}{\Delta t} \mathbf{C} + \frac{1}{2} \left(\mathbf{K} + \sum_{i \in \mathcal{C}_n} k_i^{con} \mathbf{s}_i \mathbf{s}_i^T \right), \quad (9)$$

$$\mathbf{Z}_2 = -\frac{2}{\Delta t^2} \mathbf{M} - \frac{1}{\Delta t} \mathbf{C} + \frac{1}{2} \left(\mathbf{K} + \sum_{i \in \mathcal{C}_n} k_i^{con} \mathbf{s}_i \mathbf{s}_i^T \right), \quad (10)$$

$$\mathbf{Z}_3 = -\frac{2}{\Delta t} \mathbf{M}, \quad (11)$$

and

$$\mathbf{H}_1 = \frac{2}{\Delta t} \mathbf{I}, \quad \mathbf{H}_2 = -\frac{2}{\Delta t} \mathbf{I}, \quad \mathbf{H}_3 = -\mathbf{I}. \quad (12)$$

The iteration matrices depend on the time step of the integration scheme as well as on the set of active constraints through the modification of the stiffness matrix due to closed contact interfaces. Accordingly, they must be formed or updated every time one (event-localisation procedure) or the other (contact status change) varies. If we restrict ourselves to a scope of problems with a limited number of degrees of freedom and a limited number of contact interfaces, and consider a constant time step algorithm, it might be worthwhile to follow the unusual path of computing the inverse of matrix \mathbf{Z}_1 and solve the force residual (4a) by left multiplication by \mathbf{Z}_1^{-1} rather than by solving a linear system at each increment, in which case the integration scheme can be seen as an explicit one. The inverse matrix can then be updated at each change of contact status using the Sherman-Morrison formula [4] for rank-1 update of an inverse matrix. If storage permits, they can also be precomputed for each possible contact configuration and stored in memory. It is the second option we follow in our example problems.

2.2 EVENT-DETECTION AND EVENT-LOCALISATION PROCEDURE

Event detection and localisation is a difficult task. Whereas several approaches have been developed for higher order integration schemes designed to integrate first-order ODEs, e.g. relying on the finding of the zeros of a polynomial interpolation of the event functions over the time step [5], there is, to our knowledge, no such richness for integration schemes dedicated to second-order ODEs. Though not foolproof, we base our detection algorithm on the tracking of sign changes of the functions representative of events, e.g. the gap function in the case of unilateral contact. Such an approach indeed fails to detect an event should it occur an even number of times over the time step; too large time steps should therefore be avoided, in accordance with the eigenfrequencies of the system under scrutiny. Once the event has been detected, we then locate it using a bracketing procedure based on inverse linear interpolation (secant method).

Let us consider the vectors of event functions evaluated at times t_L and t_R ($t_L < t_R$), \mathbf{g}_L and \mathbf{g}_R . We define the set of events changing status over the time increment $t_R - t_L$ as

$$\mathcal{I}_n = \{i \in \mathcal{C} : (\mathbf{g}_L)_i \cdot (\mathbf{g}_R)_i < 0\}, \quad (13)$$

i.e. all event functions with indices $i \in \mathcal{I}_n$ have an odd number (minimum one) of zero crossings over that time increment. For all status changing events, we then perform an inverse linear interpolation to identify the earliest event occurrence under the linear

Algorithm 2 Event-localisation procedure.

Given state vectors, $\mathbf{u}_L, \mathbf{v}_L, \mathbf{u}_R, \mathbf{v}_R$, and event functions, $\mathbf{g}_L, \mathbf{g}_R$, at t_L and t_R , and index set, \mathcal{I}_n :

1. Compute event time by inverse linear interpolation method $\forall i \in \mathcal{I}_n$

$$t_i^* = t_L - (\mathbf{g}_L)_i \frac{t_R - t_L}{(\mathbf{g}_R)_i - (\mathbf{g}_L)_i}.$$

2. Identify earliest event

$$t^* = \min_{i \in \mathcal{I}_n} t_i^*.$$

3. Compute state vectors using the leftmost state and time step $\Delta t^* = t^* - t_L$: $\mathbf{u}^*, \mathbf{v}^*$.
4. Compute event functions using updated state $\forall i \in \mathcal{C}$: \mathbf{g}^* .
5. Calculate index set \mathcal{I}_n according to equation (13).

6. Assess convergence:

If $\|\mathbf{g}_{\mathcal{I}_n}^*\| < \text{tol}$:

- $t_{n+1} \leftarrow t^*, \mathbf{u}_{n+1} \leftarrow \mathbf{u}^*, \mathbf{v}_{n+1} \leftarrow \mathbf{v}^*$.
- Update \mathcal{C}_{n+1} .
- Proceed with the integration.

Else:

- Update bracket:

If $\exists i \in \mathcal{I}_n : (\mathbf{g}_L)_i \cdot (\mathbf{g}^*)_i < 0$:

$$\star t_R \leftarrow t^*, \mathbf{u}_R \leftarrow \mathbf{u}^*, \mathbf{v}_R \leftarrow \mathbf{v}^*, \mathbf{g}_R \leftarrow \mathbf{g}^*.$$

Else:

$$\star t_L \leftarrow t^*, \mathbf{u}_L \leftarrow \mathbf{u}^*, \mathbf{v}_L \leftarrow \mathbf{v}^*, \mathbf{g}_L \leftarrow \mathbf{g}^*.$$

- Go to 1.
-

assumption, bracket the zero accordingly and reevaluate the set \mathcal{I}_n over all event functions. We then iteratively proceed as such until the norm of the changing event functions becomes smaller than a given threshold. After convergence, the state vectors, as well as the set of active contacts \mathcal{C}_{n+1} are updated and the integration procedure proceeds. Details of the procedure can be found in Algorithm 2.

As the time step changes at each iteration, the iteration matrices must be formed anew at each iteration, contrary to the case of constant time step and constant set of active constraints \mathcal{C}_n that takes place between events. This part accounts for a substantial amount of the computational effort, in particular with the dissipative formulation as the forming of the $\tilde{\mathbf{Q}}_{ij}$ can prove costly. In certain applications, it might be cheaper to directly solve system (3). Though its dimension is four times the number of degrees of freedom, it prevents from forming the costly $\tilde{\mathbf{Q}}_{ij}$ matrices. This implementation is referred to as the alternative formulation of Algorithm 1 in the following example.

3 BENCHMARK EXAMPLES

3.1 BOUNCING BAR

In their review paper on integration schemes for the dynamic Signorini problem, Doyen et al. [1] introduce the benchmark problem of a bar of constant properties bouncing against a rigid wall under the action of gravity. Upon proper adjustment of the problem

parameters, the bar motion follows a periodic sequence of persistent contact and free flight phases. We refer the reader to their paper for the complete problem definition and only provide a graphical representation of its analytical solution in Figure 1.

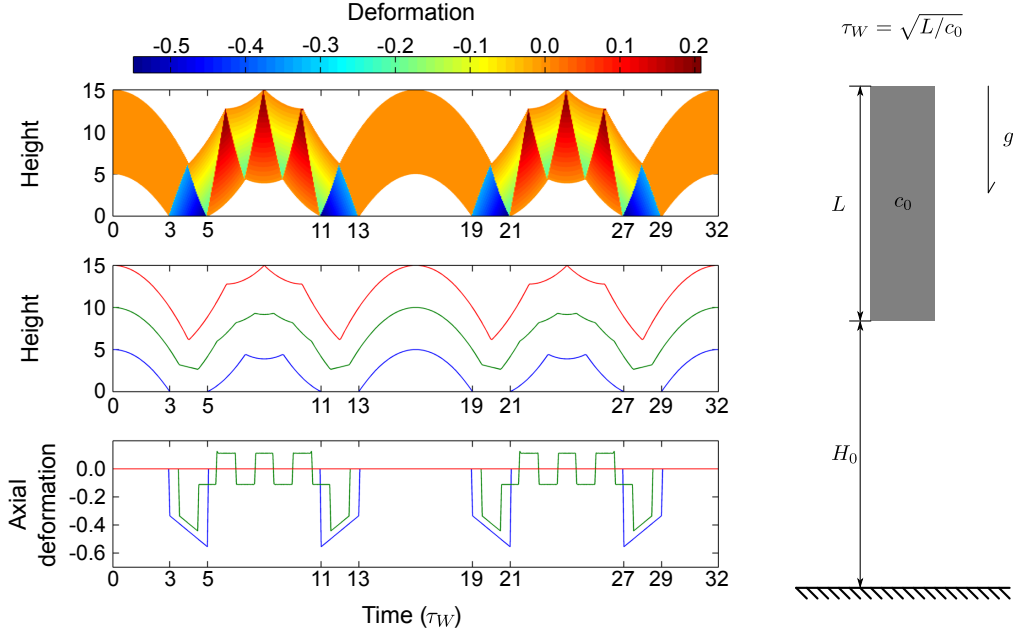


Figure 1: Analytical solution to the bouncing bar benchmark – The periodic motion comprises four phases: (i) non-oscillatory free flight phases represented by parabolas ($t/\tau_W \in [0, 3] \cup [13, 19] \cup [29, 32]$), (ii) compressive contact phases ($t/\tau_W \in [3, 5] \cup [19, 21]$), (iii) oscillatory free flight phases ($t/\tau_W \in [5, 11] \cup [21, 27]$), and (iv) expansive contact phases ($t/\tau_W \in [11, 13] \cup [27, 29]$). Parameters: $(H_0, L, c_0, g) = (5, 10, 30, 10)$.

For this example, the semi-discrete governing equation and initial conditions read

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{K}\mathbf{u} = \mathbf{f}^{ext} - k^{con}\mathbf{s}[\mathbf{s}^T\mathbf{u}]_-, \quad \mathbf{u} = H_0 + x, \quad x \in [0, L], \quad \dot{\mathbf{u}} = 0, \quad (14)$$

with the mass and stiffness matrices obtained from the assembly of the usual linear truss elements serially connected; a uniform mesh is used. Only one constraint is to be considered given the unique contact interface; it is given by the gap function $g = \mathbf{s}^T\mathbf{u}$, with $\mathbf{s} = [1 \ 0 \ \dots \ 0]^T$.

To assess the performance of the proposed algorithms, we compare to the analytical solution the computed displacements at the extremities and at the center of the bar, as well as the computational time required by our implementation of the different procedures. For indicative purposes, the formulations of [2, 3] have also been implemented as state-of-the-art references. Simulations were conducted in the MATLAB software [6]; all codes have been profiled. Figure 2 illustrates the results:

- All formulations significantly lose track of the exact solution after two periods ($t > 32\tau_W$).
- In terms of energy, the difference between the conservative and dissipative formulations is clearly visible. The numerical dissipation mainly affects high frequency vibrations; energy is seen to be most dissipated every time the bar enters a phase of persistent contact, i.e. every time a wave front propagates in the bar (the discontinuity at the wave front corresponds to an infinite frequency).

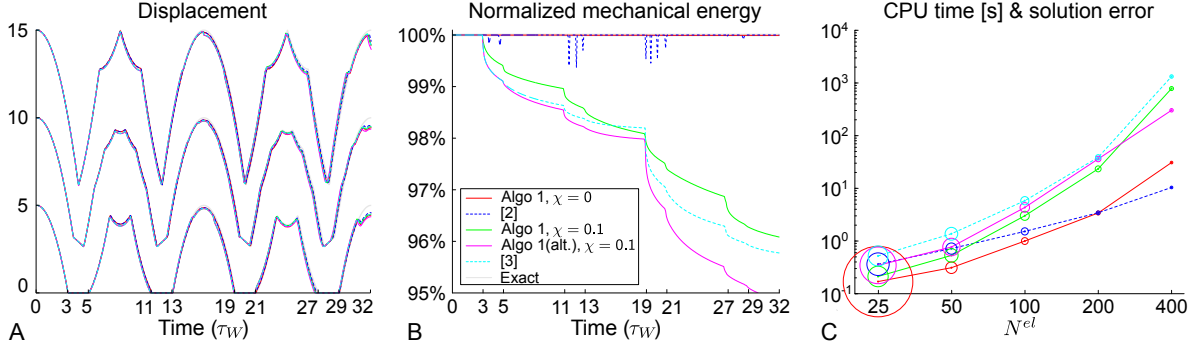


Figure 2: Numerical results for the bouncing bar example. Physical parameters are chosen identical to the ones of Figure 1 while the numerical ones are $CFL = 1.0$, $k^{con} = 10^6 c_0^2 N^{el}/L$, $\tau_{ol} = 10^{-6}$. Plot A (left) shows, for all formulations, the computed displacements at the top and bottom extremities of the bar, as well as at its center of gravity; $N^{el} = 100$. Plot B (center) displays the evolution of the mechanical energy in the system; $N^{el} = 100$. Plot C (right) gives the computational time required to simulate the system using different mesh refinements and integration procedures; the error level of the solution is represented by the size of the circle at each computed point.

- On this specific example, the schemes based on event-detection provide solutions as accurate as the ones obtained using the formulations from [2, 3] (the magnitude of the circles in the right plot is proportional to the error on the displacements of the left plot). With respect to the computational expense, they do generally perform better, with a lower burden.

3.2 DTH SYSTEM

A schematic representation of the bottom parts of a DTH percussive system is given in Figure 3, as well as the model parameters. It consists of two moving parts: (i) a drill bit in unilateral contact with the rock whose interaction is described by a bilinear stiffness model, and (ii) a piston subject to pneumatic activation that impacts the drill bit. The bilinear nature of the interaction model requires the tracking of the state at the bit/rock interface, whence the event-driven scheme proposed in this paper. The event functions are associated with the gaps at the two contact interfaces, and the velocity of the contact interface penetrating the rock. We limit our consideration to a single percussive activation given uniform initial velocities and a closed contact with null indentation at the bit/rock interface at initial time. The bit and the piston are modeled as steel cylinders in this example, but can take any shape within the restriction of 1D description.

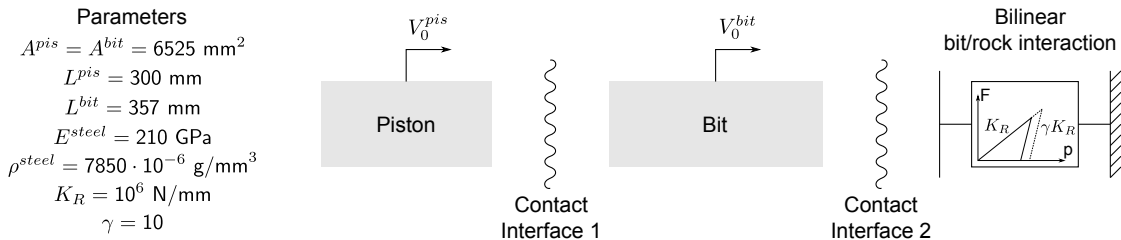


Figure 3: Simplification of a down-the-hole percussive system to two bodies moving axially: the piston and the bit. The bit/rock interaction model is based on a bilinear representation of the force/penetration relationship.

Figure 4 depicts the typical evolution of the velocities of the piston and bit centers

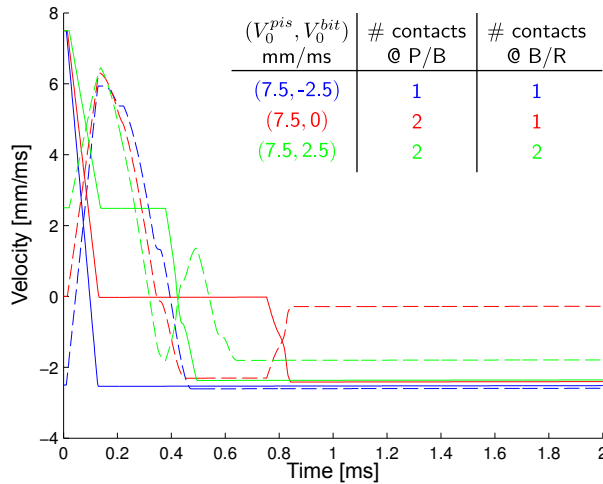


Figure 4: Typical evolution of the velocities of the centers of gravity of the piston (solid lines) and of the drill bit (dashed lines); three impacting configurations are considered. Transfer of momentum between the two bodies corresponds to an impact between the bit and the piston while the sole variation of the bit momentum is due to the interaction with the rock. The number of persistent contact phases varies with the impact velocities.

of gravity as computed using Algorithm 1 with $\chi = 0.1$. It appears that the number of persistent contact phases varies with the impacting velocities, underscoring the difficulty, not to say the impossibility, to describe the interaction of the piston and the bit with a single coefficient of restitution, as would typically be done in rigid body dynamics. This difficulty is further highlighted in Figure 5 that represents the velocities of the piston and bit centers of gravity after the interaction between the two bodies has completed (we assume it has after 0.75 ms of free flight). Abrupt changes are observed with respect to the initial velocities of the bodies. These correspond to a change in the number of persistent contact phases between the bodies or, equivalently, a modification of the underlying interaction mechanism.

4 CONCLUSIONS

Per requirement of the bilinear bit/rock interface model we use, wave propagation simulation in the elements of down-the-hole percussive drilling systems necessitates the implementation of an event-driven integration scheme. Under that constraint, we have developed a time marching procedure to integrate the linear equations of motion associated with the representation of stress waves propagating in 1-dimensional colliding elastic solids.

The introduced formulation can be declined into an energy conservative or dissipative scheme according to the choice of the numerical damping parameter χ . Comparison of these two declinations to state-of-the-art integration methods was conducted on the benchmark of a bar bouncing against a rigid wall in the field of gravity. Both have proven as accurate and as efficient as the reference formulations for this example involving a low number of contact interfaces.

Following the bouncing bar problem, the simulation algorithm has been invoked to compute the post-interaction velocities of the colliding piston and drill bit of a down-the-hole percussive drilling system, while accounting for a bilinear bit/rock interaction. This

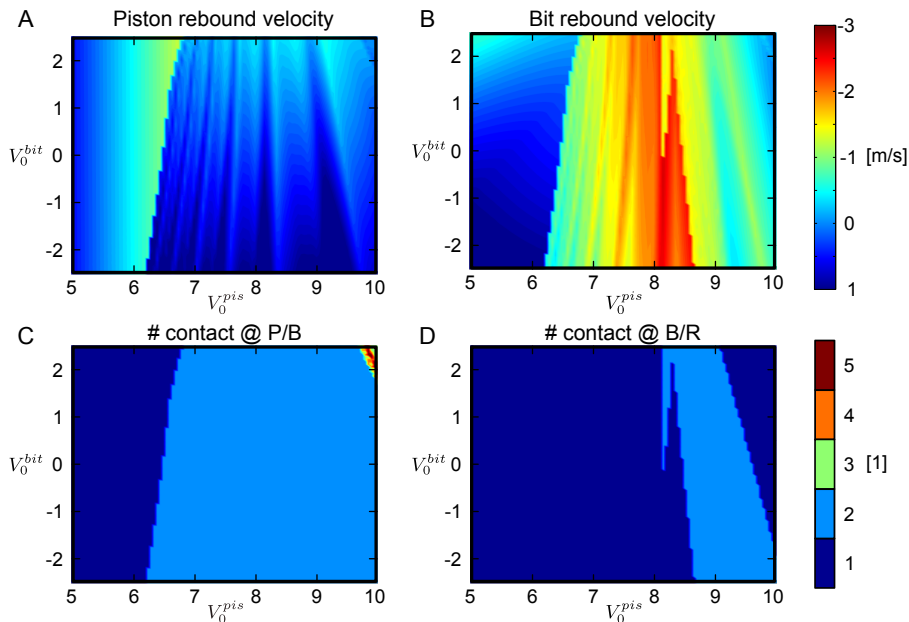


Figure 5: The post-interaction velocities of the piston (A) and the drill bit (B) depend piecewise continuously on the initial velocities of the colliding bodies. Discontinuities arise from the alteration of the underlying interaction mechanism, as indicated by the variations of the number of persistent contact phases between the piston and the bit (C) or the bit and the rock (D).

analysis has highlighted the difficulties associated with the prediction of these velocities due to the variability of the interaction chain between the bodies as the initial velocities of the colliding bodies are varied.

REFERENCES

- [1] D. Doyen, A. Ern, and S. Piperno. Time-integration schemes for the finite element dynamic Signorini problem. *SIAM Journal on Scientific Computing*, 33(1):223–249, 2011.
- [2] F. Armero and E. Petőcz. Formulation and analysis of conserving algorithms for frictionless dynamic contact/impact problems. *Computer Methods in Applied Mechanics and Engineering*, 158(3-4):269–300, 1998.
- [3] F. Armero and I. Romero. On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. part i: low-order methods for two model problems and nonlinear elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 190(20-21):2603–2649, 2001.
- [4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C : the art of scientific computing – 2nd ed.* Cambridge University Press, 2002.
- [5] T. Park and P.I. Barton. State event location in differential-algebraic models. *ACM Transactions on Modeling and Computer Simulation*, 6(2):137–165, April 1996.
- [6] MATLAB Release 2013a, The Mathworks, Inc., Natick, Massachusetts, USA.