Assessment of Techniques for Contact-Pair Detection in Solid Mechanics

F. Quaranta[1] and J. L. D. Alves[2] (chico@coc.ufrj.br, jalves@coc.ufrj.br)
[1]Graduate Student, [2]Associate Professor, PEC / COPPE / UFRJ

Abstract

This work presents some basic concepts of a technique used for modeling contact-impact problems between deformable solids through penalized finite element formulation handling the impenetrability condition and conserving *momentum*. This technique uses the concept of discrete particles and provides a way to treat general contact configurations due an automatic search for the elements involved in the phenomenon. An efficiency analysis of different algorithms used in this search is also presented. The algorithms studied were used in a finite element code, developed at PEC/COPPE, for explicit transient analysis of the non-linear behavior of axyssimetric solids.

Introduction

Contact-Impact problems are very usual in several engineering applications such as vehicle crashworthiness simulation in automotive industry, ballistics simulations, nuclear reactor safety, stamping and other metal forming processes [1]. This work reports the continuing efforts at the Civil Engineering Department of COPPE/UFRJ towards the development of finite element codes for dealing with this kind of phenomena. In particular we address some of the key issues and techniques we have used for modeling contact-impact between deformable bodies through penalty finite element formulations [1,2,3]. The basis for the implementation was a finite element code for transient dynamics, considering rate-independent elastic-plastic behavior, large deformation, finite strains and rotations, fully developed for shared memory vector/multiprocessors [4].

The semi-discrete finite element approximation of the momentum equations in transient dynamics, also including contact-impact conditions can be given by:

$$\mathbf{M\,a} + \mathbf{F}^{int} - \mathbf{F}^{b} - \mathbf{F}^{p} - \mathbf{F}^{c} = \mathbf{0} \qquad (1)$$

where $\mathbf{M\,a}$ are the inertial forces and $\mathbf{F}^{int}$, $\mathbf{F}^{b}$, $\mathbf{F}^{p}$ **and** $\mathbf{F}^{c}$ are respectively the internal forces, volume distributed forces, applied external forces and the contact forces. Contact-Impact conditions (impenetrability of bodies) are approximately enforced by penalty tractions applied to the boundaries of the bodies, at the contact region. The contact-region in not known *a-priori* and, as the problem evolves, the contact region between the bodies may change dramatically requiring frequent searches and updatings of the contact boundaries. This issue poses one of the worst and time consuming tasks for the whole analysis. Therefore, in the present work we focus our attention on this point: the search for boundaries in contact. In our implementation we have used a simple and smart technique, namely The Pinball Technique [2,3], explained in the next section.

The Pinball Algorithm

The main idea in the pinball algorithm is approximately enforce the contact conditions (impenetrability of bodies) on a set of spheres (pinballs) which are embedded in the finite elements. With this trick, the determination of whether interpenetration between the bodies (elements) has occurred becomes a simple check of the distance between two pinballs. In the pinball algorithm, the penalty force is computed at the centers of each interpenatring pinball as,

$$\mathbf{F}^{c} = \mathrm{p\,g}\,\mathbf{n} \qquad (2)$$

where p is the penalty factor, g is the amount of penetration, **n** is the average normal vector for the considered pinball pair. The penalty factor p depends on the radius R_i and the bulk modulus B_i of each pinball and on a limiting factor β as follows:

$$p = \frac{1}{2}\beta(B_1 R_1 + B_2 R_2) \qquad (3)$$

The material properties for each pinball are inherited from the enclosing element. The radius of each pinball is evaluated in order to provide the same volume as the enclosing element.

Searching of Impacting Elements

As it can be seen in the previous section, the computation of the amount of interpenetration and the penalty force applied between two pinballs can be accomplished without difficulty. However, determination of the pairs for which penetration check is required is not so easy. When the two bodies undergo large motions, it is impossible to know in advance which elements will come into contact. The search and detection method for the contacting pinball pairs is critical for the analysis and can greatly affect the time required for the total simulation. In order to locate the contacting pinballs, the most obvious method is to calculate the distance between the centers of each slave pinball and every master pinball and determine which ones have interpenetrated. This algorithm can be summarized as follows:

```
npair = 0
do i = 1, nslave
    do j = 1, nmaster
        compute the distance between pinballs i and j, and determine if they have impacted
        if they have impacted
            npair = npair + 1
            lslave(npair) = i
            lmaster(npair) = j
        endif
    end loop j
end loop i
```

Fig. 1 - The Brute Force Search

As it is likely that not always all the master and slave pinballs will participate in the contact, we can reduce the list of pinballs that need to be checked every time. This is accomplished defining a rectangular region of the space where all contacts between the bodies are expected. This region is subdivided in small parts called cells, with sides parallels to the references axis. In the cell algorithm, for each of these cells a list is made with all the master pinballs whose centers are located in the cell. After this, for each slave pinball considered, the cell in which its center is located is determined and the master pinballs which are located in that cell and the 8 surrounding cells are searched to see which, if any, impact the slave pinball. This procedure is represented in Fig. 2,

```
do i = 1, nslave
    loop through this slave cell and surrounding cells (8 in all)
        loop through the master pinballs in this cell
            if master and slave impact then
                npair = npair + 1
                lslave(npair) = slave pinball
                lmaster(npair) = master pinball
            endif
        continue loop of masters
    continue loop of cells
end loop i
```

Fig. 2 - The Standard Cell Algorithm

A more efficient means to perform the cell algorithm is given as in Fig. 3. The idea here is explore de fact that the loop which starts at line 4 is totally vectorizable, while the loop that starts at line 8 is not vectorizable. Although this second loop performs a redundant search through the master pinballs of a cell, the main advantage of this method can be appreciated by examining the frequency with which the program will perform this second unvectorized loop. Using the optimal cell size, it is highly unlikely that a given slave pinball will impact more than four or five master pinballs in a given time step. Therefore, at most, searches through only four or five of the cells will result in an impacting master element being discovered. Then, the searches through the remaining cells will prove negative and therefore it is very advantageous to conduct this fruitless searches in a vectorized manner.

```
1. do i = 1, nslave
2.       loop through this slave cell and surrounding cells (8 in all)
3.             iflag = 0
4.             loop through the master pinballs in this cell
5.                   if master and slave impact then iflag = 1
6.             continue loop of masters in this cell
7.             if iflag is not 0 then
8.                   loop through the masters pinballs in this cell
9.                         if master and slave impact then
10.                              npair = npair + 1
11.                              lslave(npair) = slave pinball
12.                              lmaster(npair) = master pinball
13.                        endif
14.                  continue loop of masters in this cell
15.            endif
16.      continue loop of cells
17. end loop i
```

Fig. 3 - The Optimized Cell Algorithm

The Paradigm Test

In order to study the efficiency of the different schemes described herein, a generic paradigm was constructed as in Fig. 4. This paradigm resembles that of a projectile (the slave domain) impacting and penetrating a target (the master domain). Rather than generating actual finite element meshes, the coordinates of the slave and master pinballs were generated randomly to lie within the specified domains. This allowed us to easily change the parameters of the problem.

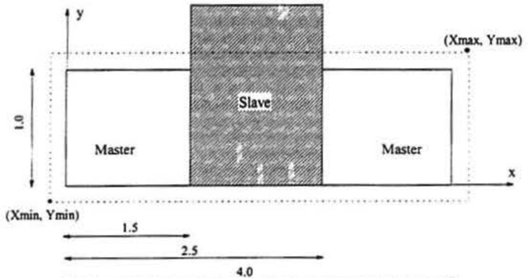

Fig. 4 - Paradigm Definition

Four different situations were tested with the two algorithms of sections 3.2 and 3.3, as show the table below

	Case 1	Case 2	Case 3	Case 4
N. of Pinballs	200 slaves 400 masters	500 slaves 1000 masters	800 slaves 1600 masters	2000 slaves 4000 masters
N. of Cells	1 x 1	5 x 5	10 x 10	20 x20

The graphics obtained are depicited below in Fig.5. As we can see, the graphics confirms the gain expected when the optimal cell size is used (case 4).

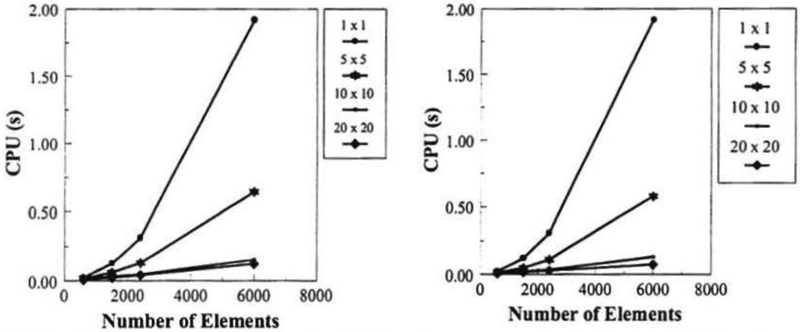

Fig. 5 - Effect of Cell Structure on Search Time for Standard and Optimized Cell Algorithm

Conclusions

In this paper we presented some stretagies for reducing the CPU resources required to contact-pairs detection in contact-impact simulations. The developed techniques are keystones for successful large scale simulations of such problem class. The observed results clearly illustrate the importance of these techniques and encourages the continuing efforts towards the implementation of new code capabilities for tackling large scale real world problems.

Acknowledgements

Computer time on the Cray J90 was provided by the Center for Parallel Computations of the Federal University of Rio de Janeiro (NACAD/COPPE). This ressearch is partially supported by a RECOPE/FINEP/MCT research grant. The authors also gratefully acknowledge the support of CNPq for the development of their reasearch activities at COPPE/UFRJ.

References

[1] Quaranta Neto, F. - "Estratégias Computacionais para a Simulação de Problemas de Contato-Impacto via Formulações Penalizadas de Elementos Finitos" - M.Sc. Thesis, in portuguese, Civil Engng. Dept. COPPE/UFRJ, (1995).

[2] Neal, M. O. - "Contact-Impact by The Pinball Algorithm with Penalty, Projection and Augmented Lagrangean Methods" - Ph.D. Dissertation, Northwestern University, Evanston IL, (1989).

[3] Costa, Luiz. A. G., Quaranta Neto, Francisco and Alves, José L. D. - "An Implementation for The Analysis of Tridimensional Contact-Impact Problems by The Pinball Algorithm" - XVIII CILAMCE, Brasilia/DF, (1997).

[4] Alves, José Luis Drummond and Quaranta Neto, Francisco - "Penalized Techniques for Finite Element Contact-Impact Analysis" - Joint Conference of Computational Mechanics and Ibero Latin American Association of Computational Methods in Engineering, Padova, Italy, (1996).