8th World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

PARALLEL PROCESSING OF MATERIAL POINT ANALYSES USING OpenMP

Zdzisław Więckowski

Technical University of Łódź
Politechniki 6, 90-924 Łódź
e-mail: zwi@p.lodz.pl

Key Words: *Material Point Method, Parallel Processing, OpenMP, Arbitrary Lagrangian–Eulerian Formulation, Granular Flow.*

ABSTRACT

The material point method is a computational approach well-known in fluid mechanics as a particle-in-cell method. The particle-in-cell method was introduced by Harlow in 1964 [1] and then used in many fields of computational physics. In the last decade of the 20th century, the method was adapted to problems of solid mechanics, e.g. [2]. The method allows one to analyse problems of large strains. Two kinds of space discretisation are used in the method: the Lagrangian and Eulerian ones. Beside the division of the region representing an analysed body into a set of subregions, a computational finite element mesh—covering the virtual position of the body—is introduced. Each subregion is represented by one of its points called a material point; it is assumed that the mass of the specific subregion is concentrated in the corresponding material point. The motion of the material points is traced together with the values of state variables using the interpolation functions and their derivatives defined by the use of the computational mesh which can remain constant or be changed during the computations. As the computational mesh can be defined in an arbitrary way, the problem of element distortions, which appears in the purely Lagrangian formulation of the finite element method, is avoided. The material point method can be regarded as the finite element method formulated in an arbitrary Lagrangian–Eulerian description of motion. On the other hand, it can also be interpreted as a point based or meshless method as the material points—where the state variables are traced—are defined independently of the computational mesh. Due to its features, the method is well-suited for modelling the large strain engineering problems like granular flow and plastic forming problems [3,4].

The analyses of dynamic, large strain problems, to which the material point method can be applied, are time consuming. Therefore, there is a need to use parallel computations for such tasks. A parallelism technique used in the present paper is based on the OpenMP standard [5,6]. OpenMP enables, in simple cases, to make a computer code parallel using only OpenMP constructs, directives and clauses (without changing the source text of a program written in FORTRAN or C language) which play the role of commenting lines when the program is intended to run in the serial fashion. As, in many cases, OpenMP may require only small modifications of the serial code, it is very attractive especially due to the fact that, at present, PC computers and notebooks allows to use multi-core and hyper threading technologies.

In the present paper, a part of the existing finite element program related to the material point analysis has been parallelised using loop-level parallelism in the following procedures:

- assembly of global matrices and vectors (the loop over the elements),
- updating stresses (the loop over the elements),
- searching elements in which the material points are located (the loop over the material points),
- solution of the diagonalised global system of equations.

The efficiency of the code parallelisation is shown by analysing the process of granular flow. An axisymmetric silo discharge problem is considered; some phases of the discharge process are shown in Fig. 1. In the same figure, the speedup factor is shown with respect to the number of threads used during the computations. The results are obtained by the use of a computer equipped with an Intel Core Quad 2.4 GHz processor.

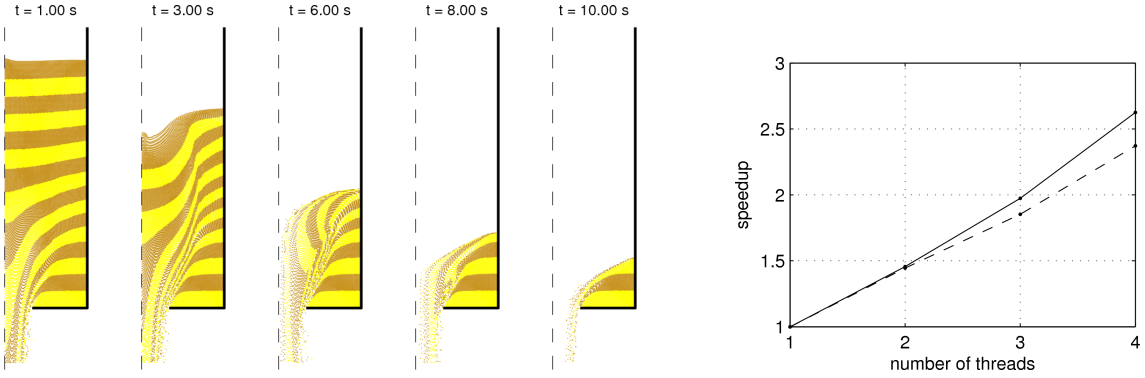

Figure 1: Flow of a granular material in a silo, phases of the discharge process (left). Efficiency of program parallelisation (right), solid line: task with 6487 nodes, 12595 elements and 45592 material points; dashed line: task with 1812 nodes, 3428 elements and 12356 material points. Real time of calculations in case of one thread: 10.25 and 1.03 hours, respectively.

Although only a portion of the computer program has been parallelised (however, it seems that the most essential parts of the code have been made parallel), a significant reduction of the real computational time is observed.

REFERENCES

[1] F.H. Harlow, "The particle-in-cell computing method for fluid dynamics." in: B. Adler, S. Fernbach and M. Rotenberg (eds.), *Methods for Computational Physics*, Vol. **III**, Academic Press, 319–343, 1964.

[2] D. Sulsky and H.L. Schreyer, "Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems." *Comput. Methods Appl. Mech. Engrg.*, Vol. **139**, 409–429, 1996.

[3] Z. Więckowski, S.K. Youn and Y.H. Yeon, "A particle-in-cell solution to the silo discharging problem." *Int. J. Numer. Meth. Engrg.*, Vol. **45**, 1202–1225, 1999.

[4] Z. Więckowski, "The material point method in large strain engineering problems." *Comput. Methods Appl. Mech. Engrg.*, Vol. **193**, 4417–4438, 2004.

[5] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald and R. Menon, *Parallel Programming in OpenMP*, Academic Press, 2001.

[6] B. Chapman, G. Jost and R. van der Pas, *Using OpenMP. Portable Shared Memory Parallel Programming*, The MIT Press, 2008.