Micro-mechanical Simulations of Soils using Massively Parallel Supercomputers

David W. Washington

Civil and Environmental Engineering, New Jersey Institute of Technology

Newark, NJ 07102, USA

and

Jay N. Meegoda

Civil and Environmental Engineering, New Jersey Institute of Technology

Newark, NJ 07102, USA

Abstract

In this research a computer program, Trubal version 1.51, based on the Discrete Element Method was converted to run on a Connection Machine (CM-5),a massively parallel supercomputer with 512 nodes, to expedite the computational times of simulating Geotechnical boundary value problems. The dynamic memory algorithm in Trubal program did not perform efficiently in CM-2 machine with the Single Instruction Multiple Data (SIMD) architecture. This was due to the communication overhead involving global array reductions, global array broadcast and random data movement. Therefore, a dynamic memory algorithm in Trubal program was converted to a static memory arrangement and Trubal program was successfully converted to run on CM-5 machines. The converted program was called "TRUBAL for Parallel Machines (TPM)." Simulating two physical triaxial experiments and comparing simulation results with Trubal simulations validated the TPM program. With a 512 nodes CM-5 machine TPM produced a nine-fold speedup demonstrating the inherent parallelism within algorithms based on the Discrete Element Method.

Key Words

Discrete Element Models, Micro-mechanics, Massively Parallel Machines, Super-computers, Granular Material, Geotechnical Problems, Simulations

1.0 INTRODUCTION

The use of micro-mechanical models on massively parallel machines in the Geotechnical field is relatively new. Ghaboussi [6] used neural networks, on the connection machine (CM-2) with 32,768 processors, to develop a micro-mechanical model. Their model was based on implicitly solving Newton's law of motion using the Wilson-θ method. The neural networks were trained to simultaneously detect particles in contact. Meegoda and Washington [8] employed an explicit numerical scheme to develop a micro-mechanical model to run on CM-2. They compared the results from the micro-mechanical model running on CM-2 with the serial version running on a VAX 8800. Senro [12] developed a micro-mechanical model to run on a small parallel machine, the Intel iPSC/860 with 16 processors. In this 2-dimensional case, the flow of sand was simulated in an expendable pattern casting process. Senro [12] did not exploit the full capabilities of a massively parallel machine like the previous two research efforts, but demonstrated the potential use of multiple processing machines. O'Connor developed a new object representation scheme to model complex 3D geometries on parallel distributed memory systems. All the above micro-mechanical models were based on the discrete element method.

Discrete Element Method

There are currently many DEM algorithms available both commercially and academically, hence choosing an algorithm to develop a DEM model for massively parallel computers is not a trivial matter. Since the primarily focus of this research was to discuss the merit of a parallel application of DEM algorithm, Cundall's [4&5] first generation of 3D DEM algorithm was selected for demonstration. The main advantage of using this algorithm was its availability as freeware. This algorithm was a result of a grant from the National Science Foundation. Other commercial programs such as PFC 3D were not considered, as they were very expensive and protected under various copyright laws. Since parallel applications of DEM models have not been commercially developed, this study hopes to widen the possibilities for parallel paradigms involving DEM.

The TRUBAL program [6&7] based on Discrete Element Method can numerically model a dry granular material by using an explicit finite difference formulation. A network of linear springs connects sphere-shaped masses to describe the contact laws. Although the system is dynamic, the transient state eventually approaches a static equilibrium condition. Each calculation cycle includes the application of Newton's Second Law to the center of

gravity of each sphere followed by the application of force-displacement laws between each contact. Accelerations and velocities of each sphere calculated from Newton's Second Law are assumed to be constant over the time-step. The net forces and moments acting on each sphere are updated from force-displacement laws applied at the contacts of neighboring spheres. Small time steps are used to assure that these forces and displacements cannot propagate from any particle further than its neighbors. Therefore, all the resultant forces on any particle are calculated exclusively by its interaction with its particle contacts.

The success of TRUBAL has validated the effectiveness of Discrete Element Method [6&7]. Further modification to this program by other researchers increased its capabilities. Bathurst [1] modeled anisotropic granular material. Ratnaweera and Meegoda [10] simulated the behavior of saturated granular soils. Gili and Alonso [7] modeled the behavior of unsaturated granular soils. Chang and Meegoda [2&3] modeled the behavior of asphalt concrete.

CM -5 Architecture

The algorithm discussed in this paper was designed to exploit the performance of the multiple processing capability and the interconnection network of any SIMD machine, because most of the computations were confined locally to each processor and its memory. However, when the global communication is performed the communication latency is entirely dependent upon the interconnection network of the machine. Since, the CM-5 architecture was the only machine tested for this paper, its machine topology is briefly summarized. The CM-5's *multiple processing capabilities* were designed differently than the CM-2. As opposed to 32,768 processors found in the CM-2, there are only 512 nodes in the CM-5 (see fig 1). The CM-2 used for TPM version 1.0 consisted of a low performance processor attached to its own memory, however the CM-5 nodes consisted of a high performance RISC processor (performing at 5 Mflops/32Mhz) attached to four vector processors (performing at 128Mflops/16Mhz). The RISC processor is able to perform all of the operations that the front-end machine can perform, only in this case it controls a single node instead of the entire machine.

The *control system* of the CM-5 can perform as a MIMD or SIMD architecture, because of its multiple network system. All of CM-5 nodes are interconnected by three major networks, the data network, the control network, and the diagnostic network (see figure 2). The control network is used for operations that involve all of the nodes at once, such as synchronization operations and broadcasting (SIMD). The data network is responsible for bulk data transfers, where each item has a single source and destination. It is this feature of the data network that enables the architecture to perform as a Multiple Instruction Multiple Data (MIMD) machine.

The *interconnection network* of the CM-5 system is within the control network and the data network. The control network and the data network are a binary tree and

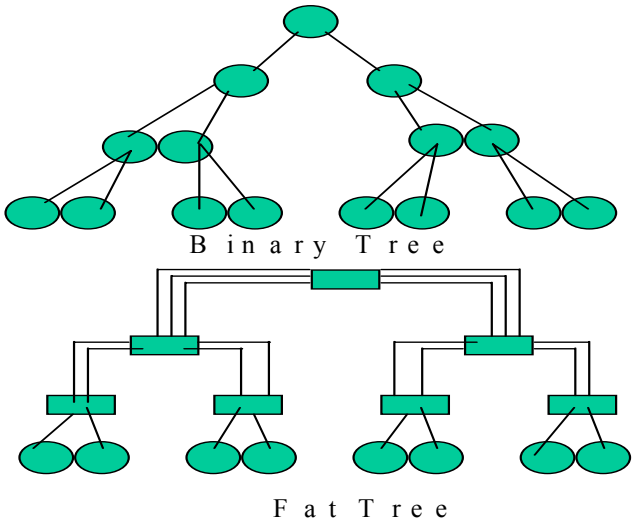

Figure 1 Control network system of the CM-5

fat tree interconnection network respectively (see fig.2). The fat tree in comparison with the binary tree is designed to produce fewer bottlenecks in communication as messages are sent to higher levels of the tree. The binary tree will inherently be slower, because there are less physical channel connections for data flow, creating signal blocks at the higher levels. Because TPM version 2.0 uses the SIMD architecture of the CM-5, the binary tree performs the global communication.

2.0 TRUBAL for Parallel Machines (TPM)

A program based on the Discrete Element Method termed "TRUBAL for massively Parallel Machines (TPM)" was developed at NJIT (TPM version 1.0 [10]). This algorithm was a modification of TRUBAL version 1.51(1989), which included the periodic spaces and non-linear contact laws from the original TRUBAL [7]. This approach involved changing the dynamic memory

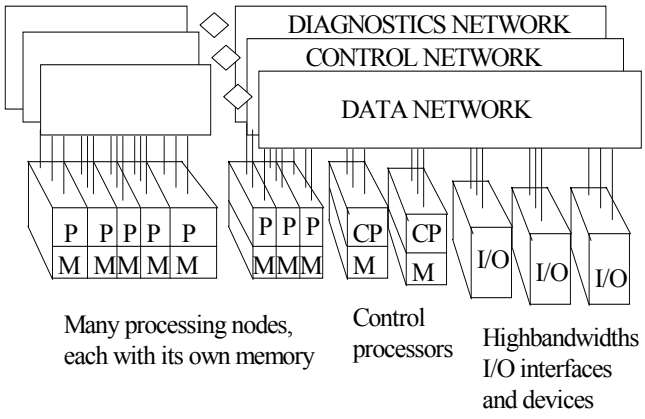

Figure 2 CM-5 Architecture

allocation involved in contact detection that consumes 60% of CPU time in most discrete models. This first

approach was performed on TPM version 1.0, which was developed on the CM-2, a Connection Machine with 32,768 processors, located at the Pittsburgh Supercomputing Center. TPM version 1.0 [10] used the parallel arrays that followed the TRUBAL data structure. However, in regard to contact detection, the dynamic memory allocation of particle regions found in Trubal was eliminated. TPM used a vector processing command "spread", to layout particle information (replicating it) over massive arrays. This function performed simultaneous computations for various particle regions without reverting back to a "link and list" methodology. This algorithm turned out to be very slow, because the data structure required random sorting and global array reductions in order to follow the sequential data structure in TRUBAL.

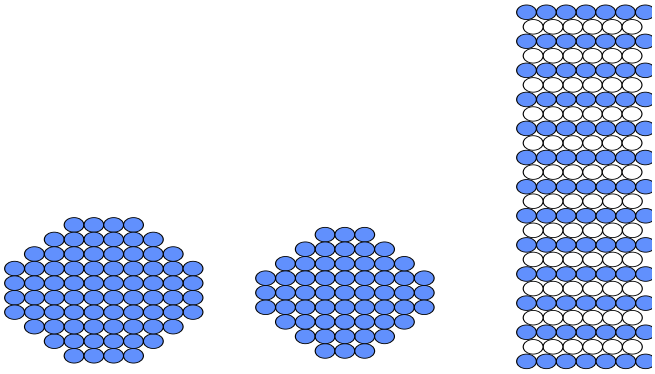

Figure 3 Layers of spheres in Rowe's physical test

The second approach involved creating a homogeneous data structure by allocating memory for all possible contacts in such a manner that all computations were uniformly handled within each processor. This scheme eliminated the particle regions, and thereby reduced the interprocessor communication for the mapping of spheres. TPM version 2.0 was the result of this development and was ported to the CM-5 Connection Machine with 512 nodes at the Pittsburgh Supercomputing Center. TPM version 2.0 assigns each processor a multiple number of contact pairs (two spheres in contact) existing within an assembly of spheres. Each contact pair is clearly defined by its location, material parameters as well as the shear and normal force interactions between the two spheres. To avoid excessive communication overhead, each sphere was paired with every other sphere in the assembly and the pair was placed within the same processor. This allowed all computations to run simultaneously, thus speeding up the algorithm. As a result of this static memory arrangement, the same sphere information was kept along the same dimension of the array. Therefore, the summations of forces and moments were handled globally by rapid data parallel inter-processor communication. Otherwise, summations for each individual sphere on an elemental basis would hinder the speed of this machine. Using this approach, the assembly size that was needed was the square of the number of particles in the assembly. Although a faster algorithm was obtained by squaring the number of particle parameters in the assembly, the extra memory that was required placed limitations on the size of the problem that could be simulated.

3.0 TPM's simulation of the Rowe's model

The objective of the second numerical triaxial simulation was to determine the effect on the overall performance with a larger simulation, and clearly validate the accuracy based on a physical experiment. Rowe [11] described the behavior of densely packed and loosely packed sands. The "uniform spheres in face-centered cubic packing" experiment was simulated with Trubal and TPM to validate the performance and accuracy of the TPM algorithm. The global behavior of a granular type sample under triaxial loading was clearly effected by the particle arrangement within the assembly. This was demonstrated with a physical laboratory triaxial test using an assembly of 1/4 in. diameter steel balls for the sample. Since the discrete element method is based on modeling the behavior of discrete element bodies within an assembly, Cundall [5] used the data in Rowe[11] to validate Trubal. Due to the limitation of the computer , this simulation was smaller in size than the actual model. In this paper, TPM simulations were performed for the actual model and for that used by Cundall [5].

Rowe's physical laboratory test

The laboratory model conducted by Rowe consisted of an octagonal shaped packing of "large" layers and "small" layers (figure 3). The 1,672 sphere sample consisted of 13 large layers with 76 spheres in each row and 12 small layers with 57 spheres of each row placed alternately on top of one another. The large layers at the top and bottom of the sample were in contact with the axial loading that was applied to the sample. The rubber membrane and the vacuum applied within the sample created the confining pressure.

Cundall's numerical simulation

Cundall's numerical sample of this model was slightly smaller in size, because of the limitation of the computer's memory capacity used at that time. Hence, the model's large layer was reduced from 76 spheres to 37 spheres and the small layer was reduced from 57 spheres to 24 spheres. In this case, seven large layers and six small layers placed alternately on top of one another with the cross-section as shown and was simulated. Table 1 shows the parameters used in the Trubal simulation, where the contact stiffnesses were chosen so that the elastic deformations would be small compared to the distortions arising from the slip between particles. The end platen used in the Rowe model to apply the load was simulated in the numerical simulation, by fixing the velocity of the top and bottom boundary particles in the z-

direction. During the compaction phase, the particle velocity is set to zero so that the assembly can achieve a

Density of each sphere:	2000
shear contact stiffness:	1.5×10^9
normal contact stiffness:	1.5×10^9
friction angle	: 7 degrees
cohesion	: 0
radius of each sphere	: 20
confining pressure(σ_2) :	5×10^4

Table 1 Parameters used in Cundall [5] numerical simulation

state of equilibrium. The rubber membrane is approximated to the form of an *ideal* membrane, and it is assumed to stretch between the particles. Since the membrane is only in contact with the boundary particles, the confining pressure is only applied individually to the particles on the outer perimeter of the larger layers. It was noted that the fix forces on the boundary pressure were only valid for small strains and displacements, since the actual forces will vary as the geometry changes. In order to make a correlation to Rowe's results, the graph of the axial strain , e_1, vs. the stress ratio, R, used in Rowe's paper was compared. The stress ratio is defined as follows:

$$R = \frac{\sigma_1'}{\sigma_2'} = \frac{F_1}{A * \sigma_2'} \quad \text{.....1}$$

where A is the area of the octagonal shape formed by the assembly, F_1 is the measured platen force, and σ_2 is the confining pressure on the boundary particles equivalent to $5*10^4$. The axial strain, e_1 is defined as follows:

$$e_1 = \frac{2 * \delta}{\Delta h} \quad \text{.....2}$$

where δ is the measured displacement at either boundary layer, and Δh is the distance between the centers of the top and bottom layers. The factor of 2 appears due to the movement of the top and bottom layers. Because the rubber membrane produced a confining pressure,

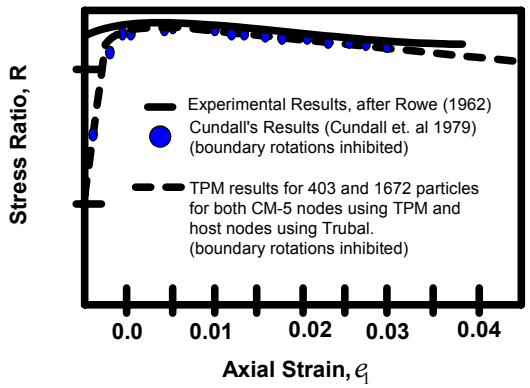

Figure 4 Results presented by Rowe, Cundall, and TPM version 2.0

Cundall's numerical results were improved when the rotations of articles in contact with the membrane were fixed. Therefore, all of the simulations reported in this paper restrict particle rotation in contact with the membrane.

TPM simulation

Cundall [5] was duplicated with 403 spheres on the control processor of the CM-5 and using TPM version 2.0. TPM was run with 32, 64, 128, 256, and 512 nodes respectively. Also, a full scale model simulation of 1672 spheres was tested and the results of this simulation were compared with the results of the 403 sphere simulations. The parameters for TPM's algorithm were the same as the parameters used by Cundall, with an exception. The damping in Cundall's original simulation used the Rayleigh damping of λ_{min}=0.05 and f_{min}=0.5, for the fraction of critical damping and the modal frequency respectively. However, when these damping values were used for TPM version 2.0 the simulation became unstable. When the values for the damping were increased to one for both λ_{min} and f_{min}, stable results were achieved. The results as presented by Rowe [11] and Cundall [5] are compared with TPM's results for the two assembly sizes in figure 4. A good agreement can be found amongst all of the results in this figure.

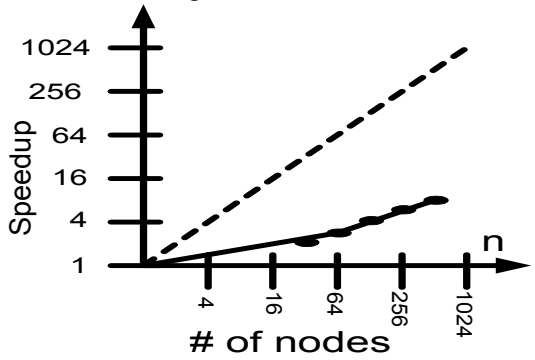

Figure 5 Speed-ups plotted for the TPM algorithm simulating the Rowe's model

Performance evaluation

The speedup was calculated and plotted for the 403 and 1672 sphere simulation in order to analyze the performance of TPM. The 403 sphere simulation time was recorded from the control processor, 32 nodes, 64 nodes, 128 nodes, 256 nodes, and 512 nodes. After the speedups and efficiencies were calculated based on equation (3) and (4) respectively, the speedups were plotted. The performance evaluation of TPM was based on the following formulas;

$$Sp(N) = T^*(N)/Tp(N) \quad \text{.....3}$$

$$Ep(N) = Sp(N)/p \quad \text{.....4}$$

which determined the speedup, Sp(N), and efficiency Ep(N) of the algorithm. Where T*(N) is the CPU time for the best serial version of the algorithm running on a single processor, Tp(N) is the CPU time taken by the parallel algorithm with *p* processors, and N is the problem size or number of elements. Then the CPU time for the 1672 sphere simulation was recorded, but due to the memory

consumption, only the control processor, 256 nodes, and the 512 nodes could perform this simulation. Within the program, larger arrays were allocated to multiple parameters to conserve on memory, which enabled a larger assembly to be simulated. Table 2 shows the times from each simulation along with their corresponding speedups and efficiencies.

Figure 5 shows the curve of actual speedup using the 403 spheres of the Rowe simulation along with the ideal speedup. The ideal speedup (the 45 degree dotted line) is based on the premise that the code is completely parallel and there are no overheads in communication processes. Since processors are doing the same work, a speedup becomes a multiple of the number of processors used. In the case of TPM version 2.0, global communication was a major bottleneck within the algorithm producing a lower efficiency as the number of processors increased.

4.0 Summary and Conclusions

The program TRUBAL for massively Parallel Machines (TPM) was modified and ported to the CM-5 to run as a SIMD program. The faster TPM version 2.0 assigns each processor a multiple number of contact, each sphere was paired with every other sphere in the assembly and the pair was placed within the same processor. As a result of this static memory arrangement, the assembly size that was needed was the square of the simulated model assembly. The extra memory that was required placed limitations on the size of the problem that could be simulated.

Three basic advantages were obtained from the results presented in this paper. First, TPM can exploit a SIMD machine architecture with its static memory arrangement

	403 spheres speedup(eff) (Time)	1672 spheres speedup(eff) (Time)
control processor	1(100%) (2.3 hrs)	1(100%) (5.4 hrs)
32 nodes	1.6(5%) (35 min)	---
64 nodes	2.8(4.4%) (20min)	---
128 nodes	4.4(3.4%) (12 min)	---
256 nodes	6.5(2.5%) (8 min)	5(2.0%) (67.1min)
512 nodes	7.9(1.5%) (7 min)	8.7(1.7%) (37.1min)

Table 2. Speedup chart for the 403 and 1672 sphere simulation

and obtain a speedup of up to nine times faster for a 1672 particle simulation. However, with Rowe's model, the damping constants had to be increased, in order to achieve stable results. Secondly, a drastic increase in the problem size does not decrease the overall speedup as expected. In the case presented, the problem size was increased by a factor of four, however Table 2 shows small differences in the speedup between the two problem sizes. Lastly, the size of assembly can exceed over a thousand particles, even though TPM's memory requirement restricts very large problem sizes from being simulated. This restriction was overcome during the 1672 particle simulation by repeatedly using arrays that were the square of the problem size for different parameters. As a result, the memory consumption is reduced, and a simulation using over a thousand spheres can be simulated effectively with this technique. In order to overcome excessive global communications and the excessive memory requirements, TPM version 3.0 based on MIMD architecture is currently being developed.

5.0 Acknowledgments

The research described in this paper was supported by a grant from the Pittsburgh Supercomputer Center in Pennsylvania. Authors wish to acknowledge this support. Authors also wish to acknowledge Dr. Peter Cundall for providing the TRUBAL source code and allowing to modify it.

6.0 References

[1]Bathurst, R. J. (1985), "A Study of Stress and Anisotropy in Idealized Granular Assemblies" Ph.D. dissertation, Civil Engineering, Queen's University at Kingston, Ontario, Canada

[2]Chang, K. G. and Meegoda, J. N. (1997), "Micro-mechanics of Hot Mix Asphalt Concrete" ASCE Engineering Mechanics Journal, Vol. 123, #5, May 1997, pp. 495-503.

[3] Chang, K. G. and Meegoda, J. N. (1999), "Micro-mechanical Model for Temperature Effects of Hot Mix Asphalt Concrete," Transportation Research Record #1687, Journal of the Transportation Research Board, TRB 1999, pp.95-103.

[4] Cundall, P. A. and Strack, O. D. L. (1978) "The Distinct Element Method As A Tool For Research In Granular Media" Report to the National Science Foundation Concerning NSF Grant ENG-76-20711, Part I

[5] Cundall, P. A. and Strack, O. D. L. (1979) "The Distinct Element Method As A Tool For Research In Granular Media" Report to the National Science Foundation Concerning NSF Grant ENG-76-20711, Part II.

[6] Ghaboussi, J., Basole, M., Ranjithan, S. (1993) "Three Dimensional Discrete Element Analysis on Massively Parallel Computers," *Proceedings of the Second International Conference on Discrete Element Methods*, Massachusetts Institute of Technology, Cambridge, MA, March 18-19

[7] Gili, J. A., and Alonso, E. E.(1988) "Discontinuous numerical model for partially saturated soils at low saturation," *Proc. 6th Int. Conf. on Numerical*

Methods in Geomechanics. Swoboda (edt.). Balkema Rotterdam, pp. 365-372

[8] Meegoda, N. J., and Washington, D.(1994) "Massively Parallel Computers for Microscopic Modeling of Soils," *Proceedings of the 8th. International Conference of the Association for Computer Methods and Advances in Geomechanics*, H. J. Siriwardane & M. M. Zaman Edts., A. A. Balkema Publishers, pp. 617-622, May, Morgan Town WV.

[9] O'Connor, R.[1995], A Distributed Discrete Element Modeling Environment –Algorithms, Implementation and Applications. Ph.D. dissertation, Civil Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts

[10]Ratnaweera, P. and Meegoda, N. J. (1991) "Comparison of a Microscopic Model and Macroscopic Model Prediction on the Stress-Strain Behavior of a Soil", Mechanics Computing 1990's and Beyond, Editors, H. Adeli and R. L. Sierakowski, Vol. #2, pp. 1199-1203, May.

[11]Rowe, P.W. (1962), "The stress dilatancy relation of static equilibrium of an assembly of particles in contact", *Proc. of the Royal Society*, 269,No 1339, 500-527

[12]Senro, K. (1994), "Anisotropic Stiffness and Circulation Flow of Sand: Application For the Expendable Pattern Casting," Ph.D. dissertation, Civil and Environmental Engineering at University of Wisconsin-Madison