

Simulation of Earthquake Liquefaction Response on Parallel Computers

J. Peng¹, J. Lu², K. H. Law³ and A. Elgamal⁴

¹ Research Associate, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305; PH (650) 725-1886; email: junpeng@stanford.edu

² Ph.D. Student, Department of Structural Engineering, University of California, San Diego, CA 92093; PH (858) 822-0058; e-mail: jinlu@ucsd.edu

³ Professor, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305; PH (650) 725-3154; email: law@stanford.edu

⁴ Professor, Department of Structural Engineering, University of California, San Diego, CA 92093; PH (858) 822-1075; e-mail: elgamal@ucsd.edu

Abstract

This paper presents a parallel nonlinear finite element program, ParCYCLIC, which is designed for the analysis of cyclic seismically-induced liquefaction problems. Key elements of the computational strategy employed in ParCYCLIC include the deployment of an automatic domain decomposer, the use of the multilevel nested dissection algorithm for the ordering of finite element nodes, and the development of a parallel sparse direct solver. Simulation results of grid models and a centrifuge test model using ParCYCLIC are presented. Performance results show that ParCYCLIC is efficiently scalable to a large number of processors.

Introduction

Large-scale finite element simulations of earthquake ground response including liquefaction effects often require a lengthy execution time. This is necessitated by the complex algorithms of coupled solid-fluid formulation, the associated highly nonlinear plasticity-based constitutive models, and the time domain step-by-step earthquake computations. In view of the finite memory size and the limitation of current operating systems (e.g. Linux, MS Windows, and so forth), large-scale earthquake simulations may not be feasible on single-processor computers. Utilization of parallel computers, which combine the resources of multiple processing and memory units, can potentially reduce the solution time significantly and allow simulations of large and complex models that may not fit into a single processor.

Sequential application software, such as traditional finite element programs, needs to be re-designed to take full advantage of parallel computers. This paper presents the development of a state-of-the-art nonlinear parallel finite element program, ParCYCLIC, for earthquake ground response and liquefaction simulation (Lu et al. 2004; Peng et al. 2004). ParCYCLIC is implemented based on a serial program CYCLIC, which is a nonlinear finite element program developed to analyze liquefaction-induced seismic response (Parra 1996; Yang and Elgamal 2002). Extensive calibration of CYCLIC has been conducted with results from experiments

and full-scale response of earthquake simulations involving ground liquefaction. In ParCYCLIC, the calibrated serial code for modeling of earthquake geotechnical phenomena is combined with advanced computational methodologies to facilitate the simulation of large-scale systems and broaden the scope of practical applications.

Finite Element Formulation

In ParCYCLIC, the saturated soil system is modeled as a two-phase material based on the Biot (1962) theory for porous media. A numerical framework of this theory, known as u - p formulation, was implemented (Parra 1996; Yang 2000; Yang and Elgamal 2002). In the u - p formulation, displacement of the soil skeleton u and pore pressure p , are the primary unknowns (Chan 1988; Zienkiewicz et al. 1990). The implementation is based on the following assumptions: small deformation and rotation, constant density of the solid and fluid in both time and space, locally homogeneous porosity which is constant with time, incompressibility of the soil grains, and equal accelerations for the solid and fluid phases.

The u - p formulation as defined by Chan (1988) consists of an equation of motion for the solid-fluid mixture and an equation of mass conservation for the fluid phase. These two governing equations can be expressed in the following finite element matrix form (Chan 1988):

$$\mathbf{M}\ddot{\mathbf{U}} + \int_{\Omega} \mathbf{B}^T \boldsymbol{\sigma}' d\Omega + \mathbf{Q}\mathbf{p} - \mathbf{f}^s = \mathbf{0} \quad (1a)$$

$$\mathbf{Q}^T \dot{\mathbf{U}} + \mathbf{S}\dot{\mathbf{p}} + \mathbf{H}\mathbf{p} - \mathbf{f}^p = \mathbf{0} \quad (1b)$$

where \mathbf{M} is the mass matrix, \mathbf{U} the displacement vector, \mathbf{B} the strain-displacement matrix, $\boldsymbol{\sigma}'$ the effective stress tensor (determined by the soil constitutive model described below), \mathbf{Q} the discrete gradient operator coupling the solid and fluid phases, \mathbf{p} the pore pressure vector, \mathbf{S} the compressibility matrix, and \mathbf{H} the permeability matrix. The vectors \mathbf{f}^s and \mathbf{f}^p represent the effects of body forces and prescribed boundary conditions for the solid-fluid mixture and the fluid phase respectively.

Extensive calibration of ParCYCLIC has been conducted with results from experiments and full-scale response of earthquake simulations involving ground liquefaction. Calibration was based on results of monotonic and cyclic laboratory tests (Arulmoli et al. 1992), as well as data from dynamic centrifuge model simulations (Dobry et al. 1995). For the liquefaction model currently employed, emphasis is placed on controlling the magnitude of cycle-by-cycle permanent shear strain accumulation in clean medium-dense sands. Following the classical plasticity formulation, it is assumed that material elasticity is linear and isotropic, and that nonlinearity and anisotropy are the results of plasticity. The selected yield function forms a conical surface in stress space with its apex along the hydrostatic axis. During shear loading, the soil contractive/dilatative behavior is handled by a non-associative flow rule (Parra 1996) so as to achieve appropriate interaction between shear and volumetric response.

Computational Procedures

The computational procedure of ParCYCLIC is illustrated in Figure 1. The procedure can be divided into three distinct phases: the initialization phase, the nonlinear solution phase, and the output and postprocessing phase. The initialization phase consists of reading input files, performing mesh partitioning, and symbolic factorization. METIS (Karypis and Kumar 1998), which is a set of libraries for graph partitioning developed at the University of Minnesota, is used to partition the finite element mesh at this phase. Specifically, the multilevel nested dissection algorithm in METIS is employed to perform the mesh partitioning. An automatic domain decomposer for performing domain decomposition, based on the METIS ordering, is also implemented in ParCYCLIC.

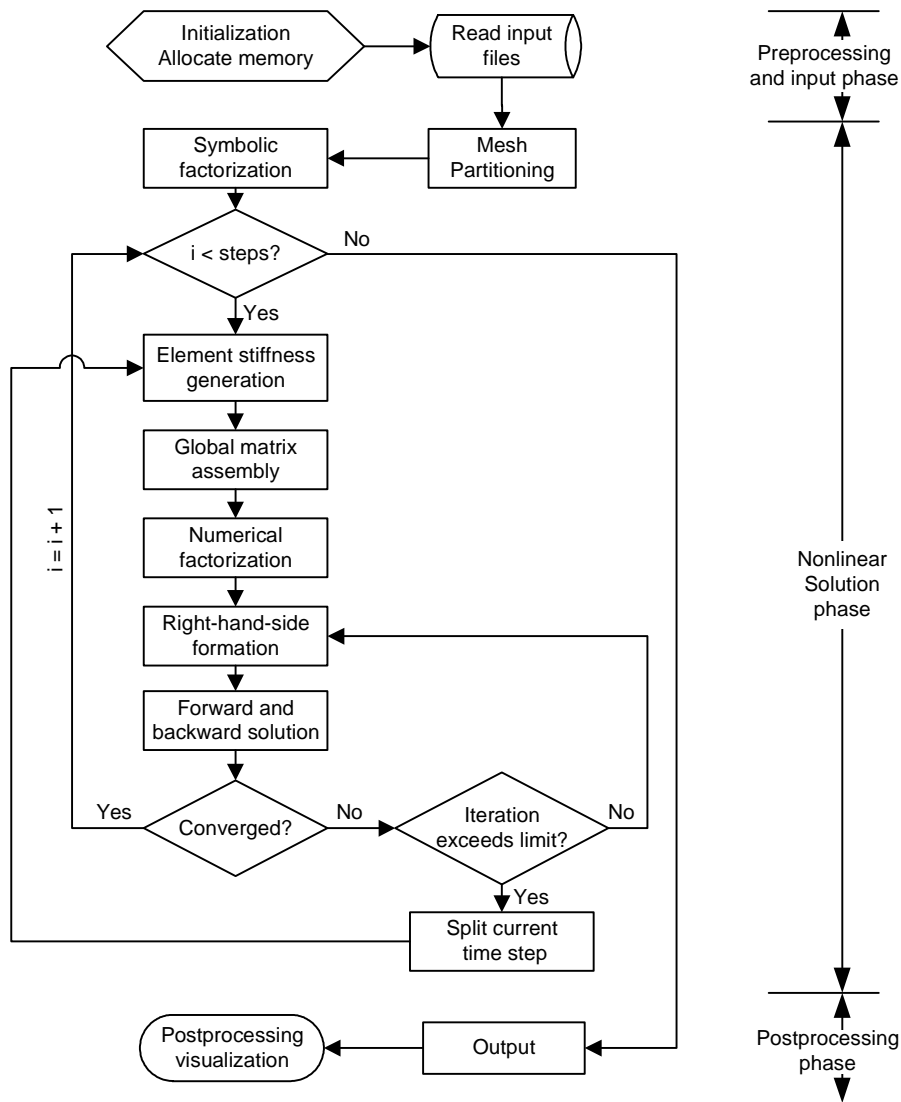


Figure 1. Flowchart of computational procedures in ParCYCLIC

The symbolic factorization is performed after the initialization phase to determine the nonzero pattern of the matrix factor. The storage space for the matrix factor is also allocated during the symbolic factorization. Since all the processors need to know the nonzero pattern of the global stiffness matrix and symbolic factorization generally only takes a small portion of the total runtime, the domain decomposition and symbolic factorization are performed within each processor based on the global input data. In the nonlinear analysis solution phase, the program essentially goes through a while loop until the number of increments reaches the pre-set limit. In the nonlinear solution phase, the modified Newton-Raphson algorithm is employed, that is, the stiffness matrix at each iteration step uses the same tangential stiffness from the initial step of the increment. The convergence test is performed at the end of each iteration step. If the solution is not converged after a certain number of iterations (e.g., 10 iterations) within a particular time step, the time step will be divided into two steps to expedite convergence. If the new step still does not converge after the certain number of iterations, then further time step splitting will automatically take place, until convergence is achieved.

The final phase, output and postprocessing, consists of collecting the calculated node response quantities (e.g. displacements, acceleration, pore pressure, and etc.) and element output (includes normal stress, normal strain, volume strain, shear strain, mean effective stress, and etc.) from the different processors. The response quantities and timing results are then written into files for future processing and visualization.

Parallel Sparse Solver

Nonlinear finite element computations of earthquake simulations involve the iterative solution of sparse symmetric systems of linear equations. Solving the linear system is often the most computational intensive task of a finite element program, especially when an implicit time integration scheme is employed. The parallel sparse solver implemented in ParCYCLIC is based on a row-oriented storage scheme that takes full advantage of the sparsity of the stiffness matrix (Law and Mackay 1993). A direct solver is preferred in ParCYCLIC over an iterative solver because even the best-known iterative solver (e.g. the Polynomial Preconditioned Conjugate Gradient method (PPCG)) may exhibit instabilities under certain conditions. For instance, in a nonlinear analysis, an iterative solver may diverge (Romero et al. 2002). The direct solution method is a more stable approach to achieve solution convergence. The concept of the sparse solver incorporated in ParCYCLIC is briefly described below.

Given a linear system of equations $Kx = f$, the symmetric sparse matrix K is often factored into the matrix product LDL^T , where L is a lower triangular matrix and D is a diagonal matrix. The solution vector x is then computed by a forward solution, $Lz = f$ or $z = L^{-1}f$, followed by a backward substitution $DL^T x = z$ or $x = L^{-T}D^{-1}z$. Sparse matrix factorization can be divided into two phases: symbolic factorization and numeric factorization (Law and Mackay 1993). Symbolic factorization determines the structure of matrix factor L from that of K (i.e. locations of nonzero entries). Numeric factorization then makes use of the data structure determined to compute the numeric

values of L and D . By topologically postordering the elimination tree, the nodes in any subtree can be numbered consecutively (Law and Mackay 1993). The resulting sparse matrix factor is partitioned into block submatrices where the columns/rows of each block correspond to the node set of a branch in the elimination tree. Figure 2 shows a simple finite element grid and its post-ordered elimination tree representation.

For parallel implementation of the sparse matrix factorization, the processor assignment strategy can be based on matrix partitioning according to the post-ordered elimination tree. The coefficients of a sparse matrix factor are distributively stored among the processors according to the column blocks. Essentially, the strategy is to assign the rows corresponding to the nodes along each branch (column block) of the elimination tree to a processor or a group of processors. Beginning at the root of the elimination tree, the nodes belonging to this branch of the tree are assigned among the available processors in a rotating round robin fashion or a block wrap mapping. As we traverse down the elimination tree, at each fork of the elimination tree, the group of processors is divided to match the number and the size of the subtrees below the current branch. A separate group of processors is assigned to each branch at the fork and the process is repeated for each subtree.

The parallel numerical factorization procedure is divided into two phases (Peng et al. 2004). In the first phase, each processor independently factorizes certain portions of the matrix that assigned to a single processor. In the second phase, other portions of the matrix shared by more than one processor are factored. Following the parallel factorization, the parallel forward and backward solution phases proceed to compute the solution to the global system of equations.

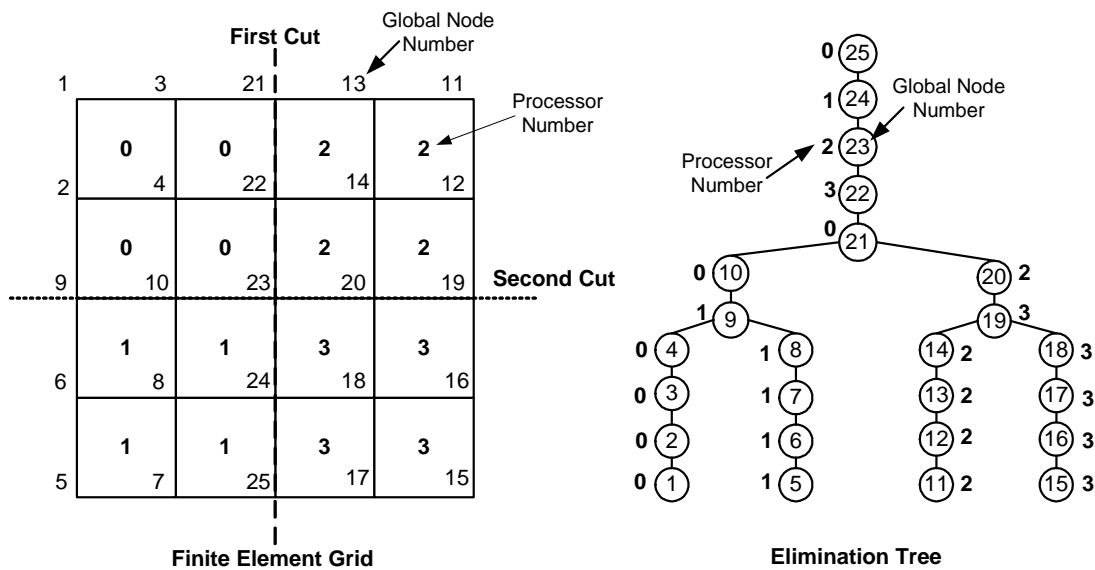


Figure 2. A finite element mesh and its elimination tree representation

Parallel Performance

ParCYCLIC has been successfully ported on many different types of parallel computers and workstation clusters, including IBM SP machines, SUN super computers, and Linux workstation clusters. To demonstrate the parallel performance of ParCYCLIC, the following shows the performance of ParCYCLIC on the Blue Horizon machine at San Diego Supercomputer Center. Blue Horizon is an IBM Scalable POWERparallel (SP) machine with 144 compute nodes, each with 8 POWER3 RISC-based processors and with 4 GBytes of memory. Each processor on the node has equal shared access to the memory.

Solution of finite element grid models. The first experiment deals with the solution of a number of 2D plane strain finite element grid models of sizes ranging from 150x150 to 300x300 elements, as well as 3D grid models of sizes ranging from 20x20x20 to 35x35x35 elements. The multilevel nested dissection (Karypis and Kumar 1998) for the grid problems is able to generate well-balanced workload distribution for running on a parallel computer. Each processor is responsible for approximately the same number of elements and equations. When there is good load balance, each processor will complete its tasks at about the same time and synchronization costs are low. Table 1 summarizes the execution times of the solution phase (numerical factorization, forward solve and backward substitution) for these 2D and 3D grid models for one time step. Excellent parallel speedup is achieved for these grid models up to a certain number of processors, as shown in Table 2. However, the speedup tends to saturate or peak at a certain limit and the performance does not continue to improve with increasing number of processors. This is due to the increased communication overhead as the number of processors continues to increase. It may be noted that some of the grid models, e.g. the 30x30x30 mesh and the 35x35x35 mesh, are too large to fit into the memory of a low number of processors. The execution time for these situations is denoted as N/A in Table 1.

Table 1. Execution time of solution phase for grid models (time in seconds)

Num. Proc.	2D grid models				3D grid models			
	150x150 ¹	200x200 ²	250x250 ³	300x300 ⁴	20x20x20 ⁵	25x25x25 ⁶	30x30x30 ⁷	35x35x35 ⁸
1	19.66	46.11	98.14	185.70	188.00	674.60	N/A	N/A
2	9.34	19.83	37.60	66.38	93.97	357.00	1016.78	N/A
4	4.57	9.73	17.70	28.68	47.26	187.10	492.82	1359.26
8	2.69	5.56	9.78	16.16	25.59	96.16	248.53	714.91
16	1.81	3.49	5.81	9.22	16.74	54.72	132.73	365.88
32	1.52	2.55	4.05	6.17	12.52	35.14	76.85	188.32
64	2.36	3.60	7.02	6.84	14.80	34.70	58.98	127.23
128	4.32	6.01	11.50	8.94	18.97	50.88	64.81	119.12
¹ 68,403 equations; 7,457,460 non-zeros					⁵ 37,044 equations; 27,214,674 non-zeros			
² 121,203 equations; 14,189,580 non-zeros					⁶ 70,304 equations; 66,365,120 non-zeros			
³ 189,003 equations; 23,724,231 non-zeros					⁷ 119,164 equations; 138,930,582 non-zeros			
⁴ 271,803 equations; 35,706,300 non-zeros					⁸ 186,624 equations; 258,680,240 non-zeros			

Table 2: Parallel speedup factors of solution phase for grid models

Num. Proc.	2D grid models				3D grid models			
	150x150	200x200	250x250	300x300	20x20x20	25x25x25	30x30x30 ¹	35x35x35 ²
1	--	--	--	--	--	--	--	--
2	2.10	2.33	2.61	2.80	2.00	1.89	--	--
4	4.30	4.74	5.54	6.47	3.98	3.61	2.06	--
8	7.31	8.29	10.03	11.49	7.35	7.02	4.09	1.90
16	10.86	13.21	16.89	20.14	11.23	12.33	7.66	3.72
32	12.93	18.08	24.23	30.10	15.02	19.20	13.23	7.22
64	8.33	12.81	13.98	27.15	12.70	19.44	17.24	10.68
128	4.55	7.67	8.53	20.77	9.91	13.26	15.69	11.41

¹Relative to 2 processors
²Relative to 4 processors

Numerical simulation of centrifuge experiment. Figure 3 shows a Rensselaer Polytechnic Institute centrifuge test model (Abdoun and Doubry 2002) to investigate the response of a single-pile foundation in a liquefied gently sloping ground, subjected to dynamic base excitation. The experiment was conducted using a rectangular, flexible-wall laminar box container. The soil profile consists of a saturated loose liquefiable sand layer (relative density $D_r = 40\%$), underlain by a slightly cemented non-liquefiable sand layer (Abdoun and Doubry 2002). The prototype single pile in the middle of the soil domain is 0.6m in diameter, 8m in length, and is free at the top. The model was inclined in 2 degrees and subjected to a predominantly 2Hz harmonic base excitation with a peak acceleration of 0.3g.

The centrifuge model is numerically simulated using ParCYCLIC on Blue Horizon. The soil domain and the pile were discretized with 8-node brick elements, as shown in Figure 3. A half mesh configuration was used due to geometrical symmetry. ParCYCLIC is used to simulate the model for roughly 2200 time-steps. As shown from the results in Figure 4 and Figure 5, good agreement has been achieved between the computed and the recorded acceleration and pore pressure responses. Salient liquefaction response characteristics, including excess pore pressure generation and dissipation and acceleration spikes, were captured by the ParCYCLIC model with reasonable accuracy.

Table 3 summarizes the timing measurements for performing the simulation of the centrifuge test model on 8, 16, and 32 processors. Generally speaking, significant decrease in both numerical factorization and the total execution time can be observed. Since there is no element duplication in the phases of the right-hand-side (RHS) formation and the stress update, these two procedures scale very well, and achieve nearly linear speed-up. In the hybrid element assignment strategy, element duplication occurs in the stiffness matrix formation (Peng et al. 2004). As the number of processors increases, the ratio of the duplicated portion (i.e., number of duplicated elements) over the non-duplicated portion becomes larger, and thus the parallel speedup of the left-hand-side (LHS) formation decreases.

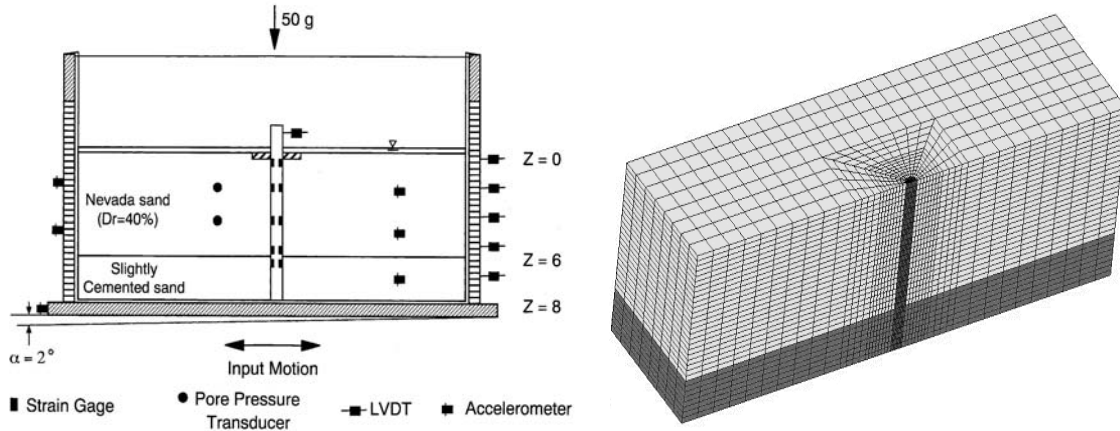


Figure 3. Centrifuge test model and its finite element mesh

Table 3: Simulation time of the centrifuge test model (time in second)

#procs	LHS formation	RHS formation	Stress update	Factorization	For.& back solve	Total Time	Speedup
8	273.5744	2446.7949	254.423	2736.6251	490.8859	6406.2478	1.00
16	164.1032	1224.6193	128.2848	1491.3682	341.3722	3579.8026	1.79
32	104.4461	622.0131	64.1883	813.7018	320.2453	2157.3544	2.97
64	85.4338	445.0552	40.5882	629.3023	517.6617	2000.4592	3.20

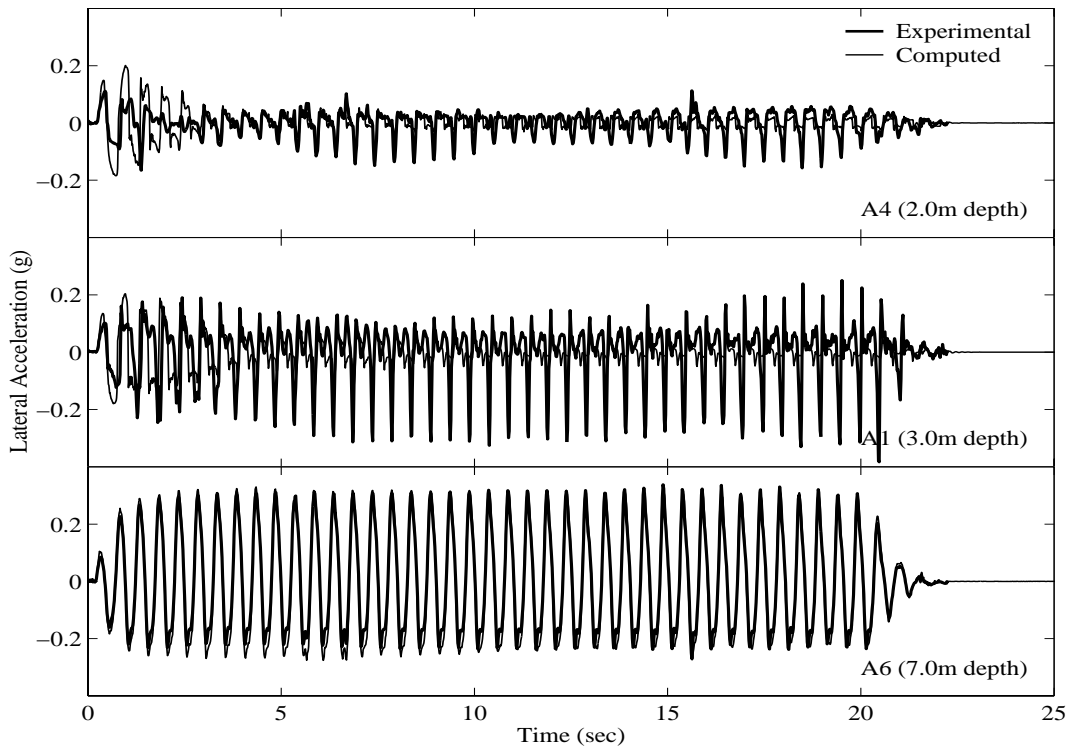


Figure 4. Computed and recorded lateral acceleration time histories

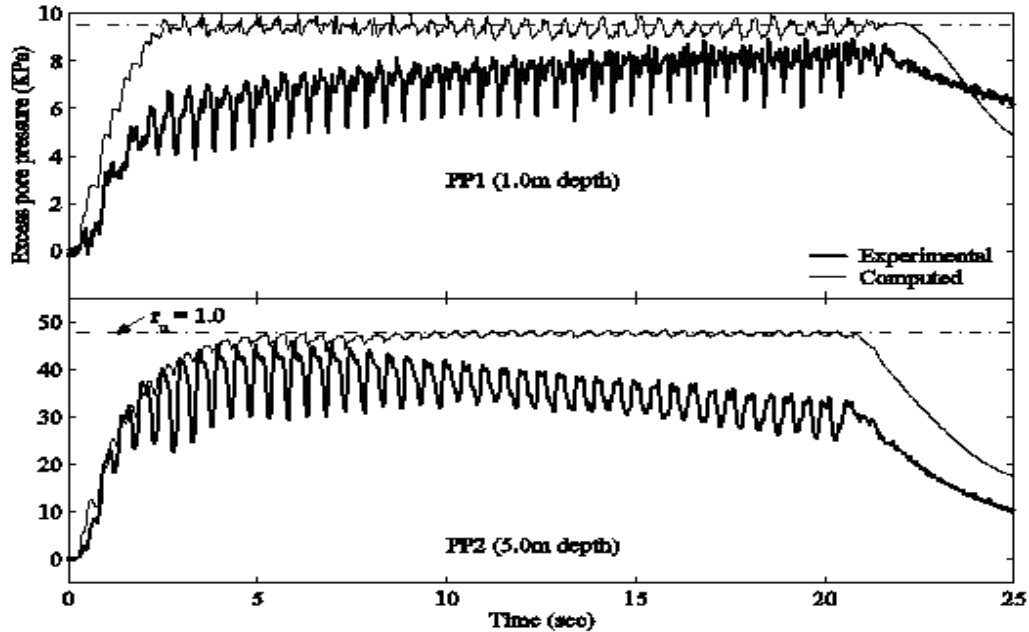


Figure 5. Computed and recorded excess pore pressure time histories

Conclusion

The finite elements employed in ParCYCLIC are modeled as an incremental plasticity coupled solid-fluid formulation. The solution strategy in ParCYCLIC is based on the enhancement of a parallel row-oriented sparse solver (Law and Mackay 1993). An automatic domain decomposer based on METIS routines (Karypis and Kumar 1998) is implemented in ParCYCLIC, in which the multilevel nested dissection algorithm is used to order the finite element nodes.

Large-scale experimental results for gird models and 3-D geotechnical simulations are presented to demonstrate the performance of ParCYCLIC. Excellent parallel speedups are reported from the simulation results. Furthermore, the results show that ParCYCLIC is scalable to a large number of processors, e.g., 64 or more. It is also shown that ParCYCLIC can be used to simulate large-scale problems, which would otherwise be infeasible using single-processor computers due to the limited memory. The parallel computational strategies employed in ParCYCLIC are general and can be adapted to other similar applications without difficulties.

Acknowledgements

This research was supported in part by the National Science Foundation Grants CMS-0084616 to University of California at San Diego, and CMS-0084530 to Stanford University. Additional support was also provided by the NSF cooperative agreement ACI-9619020 through computing resources provided by the National Partnership for Advanced Computational Infrastructure at the San Diego Supercomputer Center.

References

- Abdoun, T., and Doubry, R. (2002). "Evaluation of Pile Foundation Response to Lateral Spreading." *Soil Dynamics and Earthquake Engineering*, 22(9-12), 1069-1076.
- Arulmoli, K., Muraleetharan, K. K., Hossain, M. M., and Fruth, L. S. (1992). "VELACS: Verification of Liquefaction Analyses by Centrifuge Studies, Laboratory Testing Program, Soil Data Report." No. 90-0562, The Earth Technology Corporation, Irvine, CA.
- Biot, M. A. (1962). "The Mechanics of Deformation and Acoustic Propagation in Porous Media." *Journal of Applied Physics*, 33(4), 1482-1498.
- Chan, A. H. C. (1988). "A Unified Finite Element Solution to Static and Dynamic Problems in Geomechanics," Ph.D. Thesis, University of Wales, Swansea, U.K.
- Dobry, R., Taboada, V., and Liu, L. (1995). "Centrifuge Modeling of Liquefaction Effects During Earthquakes." *Proceedings of 1st International Conference on Earthquake Geotechnical Engineering (IS-Tokyo)*, Keynote Lecture, Tokyo, Japan, 1291-1324.
- Karypis, G., and Kumar, V. (1998). "METIS Version 4.0: A Software Package For Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices." Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN.
- Law, K. H., and Mackay, D. R. (1993). "A Parallel Row-Oriented Sparse Solution Method for Finite Element Structural Analysis." *International Journal for Numerical Methods in Engineering*, 36(17), 2895-2919.
- Lu, J., Peng, J., Elgamal, A. W., Yang, Z., and Law, K. H. (2004). "Parallel Finite Element Modeling of Earthquake Ground Response and Liquefaction." *Journal of Earthquake Engineering and Engineering Vibration*, accepted.
- Parra, E. (1996). "Numerical Modeling of Liquefaction and Lateral Ground Deformation Including Cyclic Mobility and Dilation Response in Soil Systems," Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, NY.
- Peng, J., Lu, J., Law, K. H., and Elgamal, A. W. (2004). "ParCYCLIC: Finite Element Modeling of Earthquake Liquefaction Response on Parallel Computers." *International Journal for Numerical and Analytical Methods in Geomechanics*, in review.
- Romero, M. L., Miguel, P. F., and Cano, J. J. (2002). "A Parallel Procedure for Nonlinear Analysis of Reinforced Concrete Three-dimensional Frames." *Computers and Structures*, 80(16-17), 1337-1350.
- Yang, Z. (2000). "Numerical Modeling of Earthquake Site Response Including Dilation and Liquefaction," Ph.D. Thesis, Columbia University, New York, NY.
- Yang, Z., and Elgamal, A. (2002). "Influence of Permeability on Liquefaction-Induced Shear Deformation." *Journal of Engineering Mechanics, ASCE*, 128(7), 720-729.
- Zienkiewicz, O. C., Chan, A. H. C., Pastor, M., Paul, D. K., and Shiomi, T. (1990). "Static and Dynamic Behavior of Soils: A Rational Approach to Quantitative Solutions: I. Fully Saturated Problems." *Proceedings of the Royal Society London, Series A, Mathematical and Physical Sciences*, 429, 285-309.