

A Parallel Multigrid Method for the Finite Element Analysis of Mechanical Contact

J. D. Hales, I. D. Parsons

This article was submitted to
International Conference on Computational Engineering and
Sciences, Reno, NV., July 31- August 2, 2002

U.S. Department of Energy

March 21, 2002

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

This report has been reproduced directly from the best available copy.

Available electronically at <http://www.doc.gov/bridge>

Available for a processing fee to U.S. Department of Energy
And its contractors in paper from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-mail: reports@adonis.osti.gov

Available for the sale to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-mail: orders@ntis.fedworld.gov
Online ordering: <http://www.ntis.gov/ordering.htm>

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
<http://www.llnl.gov/tid/Library.html>

A Parallel Multigrid Method for the Finite Element Analysis of Mechanical Contact

J. D. Hales¹ and I. D. Parsons²

Summary

A geometrical multigrid method for solving the linearized matrix equations arising from node-on-face three-dimensional finite element contact is described. The development of an efficient implementation of this combination that minimizes both the memory requirements and the computational cost requires careful construction and storage of the portion of the coarse mesh stiffness matrices that are associated with the contact stiffness on the fine mesh. The multigrid contact algorithm is parallelized in a manner suitable for distributed memory architectures; results are presented that demonstrates the scheme's scalability. The solution of a large contact problem derived from an analysis of the factory joints present in the Space Shuttle reusable solid rocket motor demonstrates the usefulness of the general approach.

Introduction

A combination of several factors during the last fifty years has dramatically increased the speed and accuracy of engineering analysis and design of complex systems. These include the development of computers, the growth of analysis methods such as the finite element method, and the progress made in the field of linear solvers. Engineers now combine these three tools in ways that allow them to solve incredibly large, previously intractable problems. This paper explains how a multigrid method, finite elements and parallel computers can work together to solve one of today's more difficult and important engineering analysis problems: mechanical contact.

Finite element contact has received a significant amount of research attention. The fact that contact problems are geometrically non-smooth as well as algebraically non-linear makes them difficult to solve. The geometric multigrid method is an optimal iterative method for solving linear matrix equations since it is able to arrive at a solution with a computational effort proportional to the problem size. Enabling a multigrid method to solve mechanical contact problems has the potential for significantly increasing the power of the analytical tools available to the engineering analyst.

The mathematical formulation of finite element contact employed, the geometric multigrid method, and the parallel implementation of the combined multigrid contact algorithm are described in the following sections. The paper also demonstrates the scalability of the resulting scheme on a distributed memory parallel computer and presents results for a non-trivial test problem.

Finite Element Contact

In a finite element contact simulation, the goal is to identify specific instances of contact and to model accurately the resulting contact forces. This section outlines the underlying mechanics adopted to treat contact, the search algorithm employed to identify contact, and the governing equations that must be solved to evolve a mechanical system that suffers from contact constraints.

¹ Sandia National Laboratory, Albuquerque, NM 87185, USA; hales@ sandia.gov

² Lawrence Livermore National Lab, Livermore, CA 94551, USA; parsons14@ llnl.gov

Contact Mechanics

When two bodies are in contact, forces develop that push the bodies apart. The amount of force required to prevent the bodies from penetrating one another is directly related to how much the bodies would penetrate if the contact forces were not present. Many formulations have been developed to describe the contact force. The most common are the penalty, Lagrange multiplier, and augmented Lagrange multiplier approaches. The three approaches make different choices for the contact force, t_n (in this paper, we focus solely on frictionless contact; however, the methodology employed should be equally applicable when friction is included).

The augmented Lagrange multiplier approach largely avoids the ill-conditioning of the penalty approach while providing an update scheme for the solution of the Lagrange multipliers. In this approach,

$$t_n = \lambda + \varepsilon g \quad (1)$$

Here, the penalty parameter ε need not be extremely large for a good solution. When the gap is removed, $g = 0$, and the contact force is the determined value of the Lagrange multiplier, λ . A benefit of this approach is the updating scheme for the Lagrange multipliers. Instead of the multipliers being added to the list of unknowns, the multipliers are found separately. In an iteration loop, the update scheme

$$\lambda^{(n+1)} = \lambda^{(n)} + \varepsilon g^{(n+1)} \quad (2)$$

improves the multipliers. With the augmented Lagrange multiplier approach, the penetration can be removed to within a user-defined tolerance simply by continuing in the iteration loop.

Contact Identification

A visual inspection of two bodies makes clear whether contact occurs. However, enabling a computer to recognize contact is somewhat complicated. The procedure employed in this study follows the methodology in [3], and consists of three phases: exterior identification, locating nearby nodes and checking whether contact exists.

The exterior of a finite element mesh is identified by assigning each face of each element a number computed as $n_{diag} + (n_{min} \times n_{max})$, where n_{min} is the minimum node number of the face, n_{diag} is the node number diagonally opposite n_{min} , and n_{max} is the total number of nodes. The exterior faces are those with unique identity numbers. Determining the exterior nodes is simply a matter of identifying the nodes belonging to the external faces.

The next step is determining which exterior nodes are near an exterior face using a capture box. A capture box is an imaginary box built around an exterior face large enough to capture all exterior nodes that may penetrate the face in a given time step. A binary search identifies the nodes at the maximum and minimum coordinates of the capture box, thereby locating all of nodes in the bounding box. Excluding the four nodes belonging to the face from this list of nodes identifies the nodes that may be in contact with the element face.

For each exterior node, a list of exterior faces near that node is created. Then, which, if any, faces a node has penetrated is determined by computing the point of projection of the node to the face (i.e., the contact point) and whether the node is above or below the face. Thus, a list of contact node-face pairs is

created. Each of these pairs constitutes a contact element that produces a contact force and contact stiffness.

Contact Governing Equations

Computation of the contact stiffness and force follows the methodology given in [2]. In summary, the variational statement of the problem is written as

$$G(\mathbf{u}, \mathbf{v}) + G_c(\mathbf{u}, \mathbf{v}) = 0, \quad (3)$$

where \mathbf{u} are the displacements, \mathbf{v} belong to the set of admissible test functions, $G(\dots)$ is the usual expression for internal and external virtual work, and $G_c(\dots)$ is the virtual work done by the contact forces. For a linear elastic system, standard implicit finite element discretization produces the equation of motion

$$\mathbf{M}\ddot{\mathbf{u}}^{i+\nu} + \mathbf{K}\mathbf{u}^{i+\nu} = \mathbf{f}^{i+\nu} + \mathbf{f}_c^{i+\nu} \quad (4)$$

at time $t + \Delta t$, where \mathbf{M} and \mathbf{K} are the mass and stiffness matrices, respectively, \mathbf{f} is the external applied force and \mathbf{f}_c is the force resulting from contact. Discretization in time using Newmark's method produces

$$\left(\frac{1}{\beta\Delta t^2}\mathbf{M} + \mathbf{K}\right)\mathbf{u}^{i+\nu} = \mathbf{f}^{i+\nu} + \mathbf{f}_c^{i+\nu} + \mathbf{M}\left(\frac{1}{\beta\Delta t^2}\mathbf{u}^i + \frac{1}{\beta\Delta t}\dot{\mathbf{u}}^i + \left(\frac{1}{2\beta} - 1\right)\ddot{\mathbf{u}}^i\right), \quad (5)$$

which can be written as

$$\bar{\mathbf{K}}\mathbf{u}^{i+\nu} = \bar{\mathbf{f}}^{i+\nu}. \quad (6)$$

Since the contact forces depend on the amount of penetration, Equation (6) is nonlinear in \mathbf{u} . Linearization allows Newton iteration to be used for updating estimates of $\mathbf{u}^{i+\nu}$ with the addition of increments $\Delta\mathbf{u}^{(k)}$ obtained by solving

$$\left(\bar{\mathbf{K}}^{(k)} - \mathbf{K}_c^{(k)}\right)\Delta\mathbf{u}^{(k)} = \mathbf{r}^{(k)} = \bar{\mathbf{K}}^{(k)}\mathbf{u}^{(k)} - \bar{\mathbf{f}}^{(k)}, \quad (7)$$

where the contact stiffness matrix $\mathbf{K}_c = \frac{\partial \mathbf{f}_c}{\partial \mathbf{u}}$.

Review of Geometric Multigrid Solvers

Multigrid methods are based on the following observation: basic iterative methods are quick to reduce the high frequency components of the error in an approximate solution but slow to reduce the low frequency components. The low frequency components can be represented on a coarse grid, where solution is relatively inexpensive. The combination of fine grid smoothing and coarse grid correction forms the basis of multigrid methodology [1,5]. Recursive application of this approach to the coarse mesh problem produces a true multigrid method, which has the property that a problem can be solved in $O(n)$ operations, where n is the number of unknowns on the finest grid. This makes it particularly suitable for large-scale discretizations. In summary, a relaxation method (usually preconditioned conjugate gradients) is applied on fine meshes to smooth the errors in the solution. A restriction operator transfers forces from a fine mesh to a coarse mesh, whereas an interpolation operator transfers

displacements between the meshes. A solution of a small system of linear matrix equations is required on the coarsest mesh.

The majority of the computational effort is expended in computation of matrix-vector products of the form Kp during the conjugate gradient fine mesh relaxations. Generally, an element-level, matrix free algorithm is used to minimize the number of operations and the storage required so that Kp is computed as

$$Kp = \sum K_i p_i \quad (8)$$

A Geometric Multigrid Method for Mechanical Contact

The section explains how the geometric multigrid method was used to solve Equation (7), the linearized form of the governing equations. Specifically, it addresses the issue of enabling the multigrid method, built upon the idea of multiple meshes, to manage contact stiffness matrices, matrices which are inherently finest mesh objects.

The work required to determine $K_i p_i$ on a given mesh is always independent of the number of degrees-of-freedom on the current mesh and always dependent on the number of contacts. Thus, the work associated with the contact stiffness will not decrease as the mesh hierarchy is traversed from fine to coarse, unlike the remainder of the computations. To avoid this difficulty, element-level contact stiffness matrices from each contact condition are formed on the finest mesh and assembled. For each coarser mesh, the matrices are transformed and assembled into that mesh's global contact stiffness matrix. This produces K_i specific for each mesh in a compact storage form, which can then be used to compute $K_i p_i$ when necessary.

In summary, the portion of the stiffness matrix on each mesh resulting from the contact conditions present on the finest mesh is computed as follows.

- Compute the contact element stiffness matrices on the finest mesh, $K_{i,c}$.
- Compute the contact element stiffness matrices on the coarse meshes, $K_{i,c}$, using the constraints employed for inter-mesh transfers, i.e., $K_{i,c} = T^T K_{i,c} T$.
- Assemble the coarse mesh contact stiffness $K_{i,c}$ from $K_{i,c}$ stored as a vector representing the non-zero entries of the upper triangular part of $K_{i,c}$, denoted as $\hat{K}_{i,c}$.

Then, the computation of $K_{i,c} p_i$ on a given coarse mesh is

$$K_{i,c} p_i = (\hat{K}_{i,c} + \hat{K}_{i,c}^T - \text{diag}(K_{i,c})) p_i \quad (9)$$

Note that the diagonal part of the stiffness matrix is stored separately to be used as a preconditioner.

Parallel Implementation

Metis [4] is used to produce a volumetric partition of the finite element mesh as part of the preprocessing stage. This partition is constructed to provide load balancing in terms of the number of elements assigned to each processor, thereby producing a scalable implementation of the geometric

multigrid solver. However, this volumetric partition is unsuitable for the contact calculations. Recursive coordinate bisection (RCB) is employed to give each processor an equal portion of the work associated with contact detection [3]. RCB divides items based on geometric information; this is ideal for contact detection since nodes and faces near one another are contact candidates. A significant amount of communication must occur during the contact identification phase. Communication occurs to evenly distribute the contact identification work; further communication is required so that every processor knows which nodes from which processors are in contact with its faces and which of its nodes are in contact with which other processors. Communication for the calculation of K_p is also required.

Parallel Performance

The parallel performance of the multigrid contact algorithm described above is measured using a fixed size problem in which a given hierarchy of meshes are partitioned for different numbers of processors. The specific problem consists of two square plates, one on the other. Each plate is 80 units square and 10 units thick and was discretized with eight node hexahedral elements. A three-mesh hierarchy was built with each plate on the coarsest mesh having one element through the thickness and being eight elements square. Using uniform refinement, each plate's finest mesh is $4 \times 32 \times 32$ elements, for a total problem size on the finest mesh of 8,192 elements.

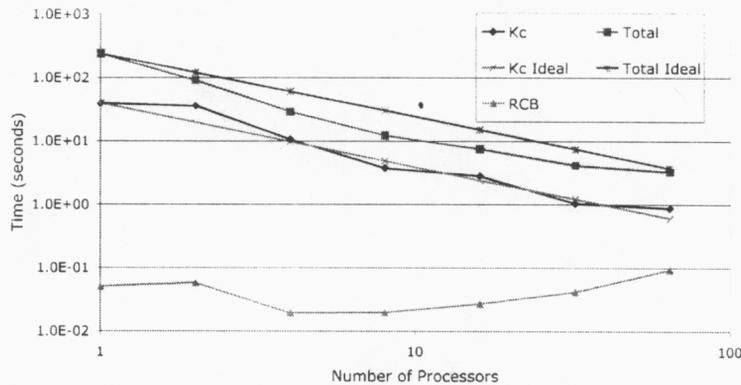


Figure 1: Timings for the fixed problem size problem.

Figure 1 reports the timings for a single linear solve on an Origin 2000. Timings are shown for the contact mechanics computations ("Kc"), the total solution time ("Total") and the RCB time ("RCB"). The ideal times for perfect scalability for the contact mechanics computations and the total solution time are also shown ("Kc Ideal" and "Total Ideal", respectively). The two straight lines give the theoretically perfect times based on the time for a single processor. The figure clearly shows excellent performance. The time required for RCB is significantly lower than even the contact time, which is much lower than the total time.

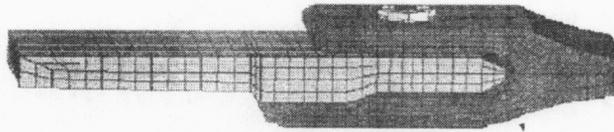


Figure 2: Finite element model of simplified solid rocket booster joint.

Demonstration Problem

The analysis of the factory joints present in the Space Shuttle reusable solid rocket motor are used to demonstrate the usefulness of the general approach described in this paper. In order to simplify the modeling and reduce the problem size, the analysis model involves the tang, clevis and pin in a factory joint. A model of part of the circular joint involving just one pin is shown in Figure 2. The complete model consists of 177 pins around the circumference. A small detail called a leak port prevents cyclic symmetry. The complete model has two meshes consisting of 867,520 nodes and 664,120 elements on the finest mesh. The upper face of the tang is restrained against vertical motion and a downward pressure field is applied to the lower face of the clevis. The dynamic simulation, employing a consistent mass matrix, used 256 processors on the Origin 2000.

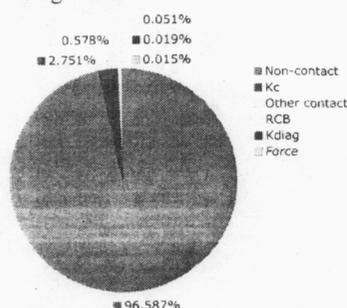


Figure 3: Percent of solution time for several categories of time step 27.

Figure 3 shows the percentage of the execution time various portions of the code required for time step number 27. The total execution time for that time step was about 13 minutes. "Non-contact" is the time required for all calculations not directly associated with contact. "Kc" is the time required for the K_p calculation. "RCB" is the time spent for recursive coordinate bisection. "Kdiag" represents the time for determining the contact stiffness diagonal, and "Force" is the time spent calculating contact forces. "Other contact" is the time associated with work required for contact that does not fit into one of the other categories. The sum of all contact categories totals less than 4%. The dominant contact cost is the cost for the K_p , as expected. This simulation shows that the methodology presented in this paper can solve practical implicit contact problems large enough to require the use of parallel computers.

References

- 1 Briggs, W. L., 1987. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics.
- 2 Crisfield, M. A., 1997. Contact with friction. In *Non-Linear Finite Element Analysis of Solids and Structures, Volume 2: Advanced Topics*, Chapter 23, pp 411-446. John Wiley & Sons.
- 3 Heinstein, M. W., S. W. Attaway, J. W. Swegle, and F. Mello, 1993. A general-purpose contact detection algorithm for nonlinear structural analysis codes. *Technical Report SAND92-2141*, Sandia National Laboratories.
- 4 Karypis, G., 2001. *METIS: Family of Multilevel Partitioning Algorithms*. <http://www-users.cs.umn.edu/~karypis/metis/>.
- 5 Parsons, I. D., 1997. Parallel adaptive multigrid methods for elasticity, plasticity, and eigenvalue problems. In *Parallel Solution Methods in Computational Mechanics*, M. Papadrakakis (editor), Chapter 4, pp 143-180. John Wiley & Sons Ltd.