Accomplishments and Challenges in Code Development for Parallel and Multimechanics Simulations

Tony Degroot, Robert Ferencz, Mark Havstad, Neil Hodge, Jerry Lin, Dennis Parsons, Michael Puso, Jerome Solberg, Ed Zywicz

Lawrence Livermore National Laboratory
8000 East Ave. Livermore CA 94550

Abstract.

The Methods Development Group at Lawrence Livermore National Laboratory has historically developed and supported software for engineering simulations, with a focus on nonlinear structural mechanics and heat transfer. The quality, quantity and complexity of engineering analyses have continued to increase over time as advances in chip speed and multiprocessing computers have empowered this simulation software. As such, the evolution of simulation software has seen a greater focus on multimechanics and the incorporation of more sophisticated algorithms to improve accuracy, robustness and usability. This paper will give an overview of the latest code technologies developed by the Methods Development group in the areas of large deformation transient analysis and implicit coupled codes. Applications were run on the state of the art hardware available at the national laboratories.

INTRODUCTION

The Methods Development Group (MDG) supports a group of roughly seventy two engineering analysts at Lawrence Livermore National Laboratory (LLNL) . It supports analysts at Los Alamos National Laboratory (LANL) and limited DoD sites. LLNL is the home of some of the fastest supercomputers in the world including the world's fastest: Blue Gene/L. Much of the hardware and software development at the LLNL has been driven by the Advanced Simulation and Computing Program (ASC). ASC was created to help maintain the United States nuclear arsenal after the 1992 moratorium on nuclear testing. The MDG group has traditionally supported roughly seventy-five analysts around the laboratory in the areas of weapons thermo-structural analysis, lasers and various physics groups. The group's flagship codes: DYNA3D (explicit structural mechanics), NIKE3D (implicit structural mechanics) and the TOPAZ3D (thermal mechanics) were originally developed in the 1970's. The latest parallel codes: PARADYN and DIABLO are the parallel explicit and implicit versions of the original codes. DIABLO, is the newest code and features support for coupled structural, thermal, diffusion and electromagnetic analyses. This paper presents the latest code technologies incorporated into these codes.

The paper is organized as follows. First the issues and technologies most related to explicit/transient dynamics and the PARADYN code will be presented including: automatic/dynamic contact, extreme material deformations and coupled finite element/meshless methods. Second, the underlying technologies and features unique to the implicit statics/dynamics multimechanics code DIABLO will be presented. The treatment of contact is still one of the group's biggest challenges, thus much of the focus of this paper. Additional aspects are presented in the areas of parallelization, solution schemes and adaptive mesh refinement. Numerous examples illustrating the latest features and run on Lab's fastest hardware will be presented.

Explicit Finite Elements: PARADYN

The main focus of this code is in the area of transient structural mechanics with limited thermal and fluid mechanics coupling. Many of the parallel methods underlying PARADYN are well document and very good scaling has been observed on very massively parallel runs (Fig. 1). The primary applications include the simulation of container drop tests (Fig. 2), pressure vessels (Fig. 3), infrastructure failure, (Fig. 4) automobile crash (Fig. 5) and penetration (Fig. 6). One of the main challenges in most of these problems typically is how the contact (interpenetration) constraints are handled. Issues regarding explicit contact and searching are presented here. Furthermore, because meshless methods accommodate evolving connectivity, the same dynamic partitioning can be applied to the meshless implementation.

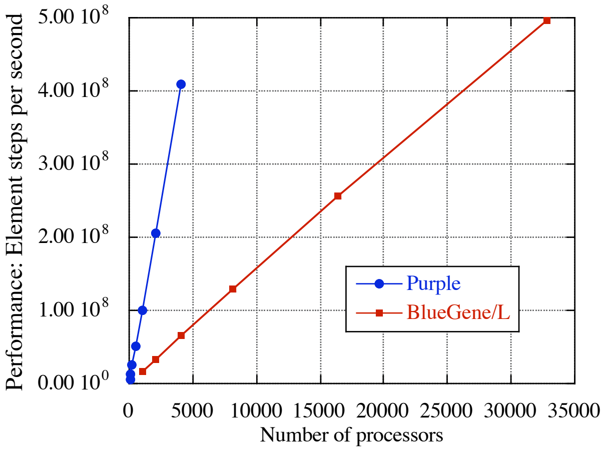

Fig. 1. Performance in element-steps per second versus number of processors of a simple 90 million element simulation using PARADYN on the ASC Purple and Blue Gene/ L platforms.

Explicit node-on-surface contact formulation

. The explicitly time integrated equations of motion specific to node-on-surface contact are given in Eq. 1.

$$Ma_{n+1} + f_n^{\text{int}}(x_n) + f^c = f_n^{ext} \tag{1}$$

M – diagonal mass matrix
f - force vector (internal, contact and external)
x_n,v_n,a_n nodal position, velocity and acceleration vectors at time t_n

The internal forces are the forces elements apply to nodes, the external forces are the applied nodal loads and the contact forces enforce the interpenetration constraints. Here, all time t_n quantities are known and the acceleration at time t_{n+1} is sought. The node-on-surface method computes a traction λ that forces penetrating nodes onto nearby facets. For example, in Fig. 5, the slave node S1 is forced onto segment M1-M2 and the contact force f^c is computed from the contact pressure λ at S1 and its distribution to node M1 and M2 based on where the closest point projection is. The single pass version of the method only forces slave nodes S onto opposing master segments. The symmetric or double pass version also forces master nodes M onto opposing slave segments. The contact pressure can be computed from the contact gap at time t_n using a penalty method i.e. $\lambda = \kappa\, g_n$. Since this gap is based on known time t_n,the contact force f^c is known and the acceleration is found easily since the mass matrix is diagonal

$$a_{n+1} = M^{-1}(f_n^{ext} + f_n^{\text{int}}(x_n) + f^c) \tag{2}$$

The velocities and positions are updated

$$v_{n+1/2} = v_{n-1/2} + \Delta t_n a_{n+1} \quad x_{n+1} = x_n + \Delta t_n v_{n+1/2} \tag{3}$$

The penalty method is often not sufficient to eliminate penetration since very high penalty values κ lower the stable time step. Here, Lagrange multipliers are used since they eliminate penetration, but don't affect the time step. The predictor-corrector strategy [1] used here computes a predictor such that no contact is active $f^c = 0$ in Eq. 1. This step yields a predicted configuration x'_{n+1} with gaps g'_{n+1}. In the corrector step, the contact tractions are treated as unknowns such that $f^c = G\,\lambda$ in Eq. 1 and the corrected displacement is computed due to these unknown contact forces

$$x_{n+1} = x'_{n+1} - \Delta t_n{}^2 M^{-1} G_{n+1} \lambda_{n+1} \tag{4}$$

Here the contact matrix G is based on segment normal vectors (Fig. 5) and is also used to compute the nodal gap vector i.e. $g = G\,x$. Multiplying Eq. 4 by G yields the nodal gap vector based on the predicted configuration and unknown contact pressure

$$g_{n+1} = g'_{n+1} - \Delta t^2 G_{n+1}{}^t\; M^{-1} G_{n+1}\; \lambda_{n+1} \tag{5}$$

The nodal gaps must satisfy the Kuhn Tucker conditions:

$$\lambda_{n+1} \geq 0, \quad g_{n+1} \geq 0, \quad g_{n+1}\, \lambda_{n+1} \geq 0 \tag{6}$$

i.e. contact pressure must be compressive, gaps must be open and only closed gaps have contact pressure. This is a mathematical programming problem and is solved via a parallel, constrained, preconditioned conjugate gradient method.

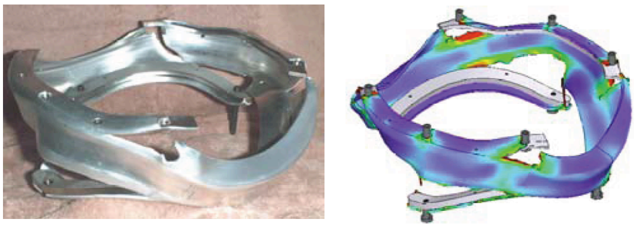

Fig. 2. Transportation container flange detail (Courtesy of Dan Badders, LLNL).

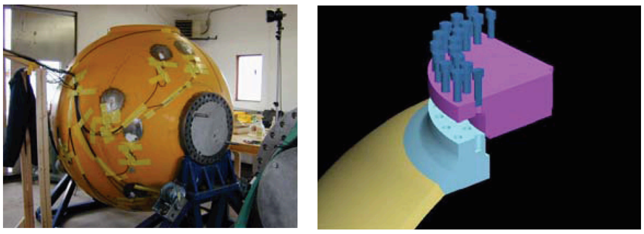

Fig. 3. Hydrodynamic containment vessel (LANL Weapons Engineering).

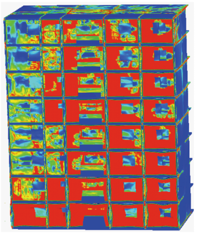

Fig. 4. Blast loading on apartment building, 30 millioin degrees of freedom (P. Papados, U.S. Army ERDC).

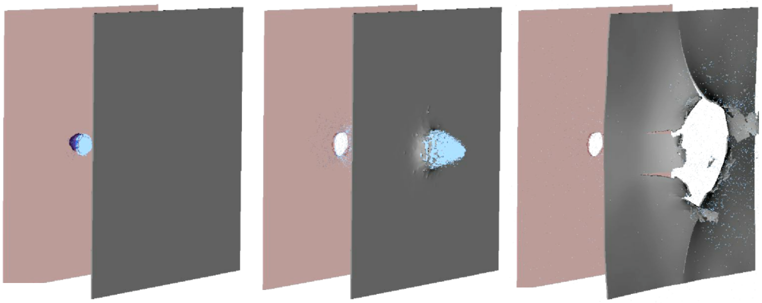

Fig. 5. Sequence of deformation as penetrator goes through two plates. The secondary damage is a result of the inertia from the fragments from the first penetration. That is, new contact surface (fragments) and particles from the first penetration will impact the second plate.

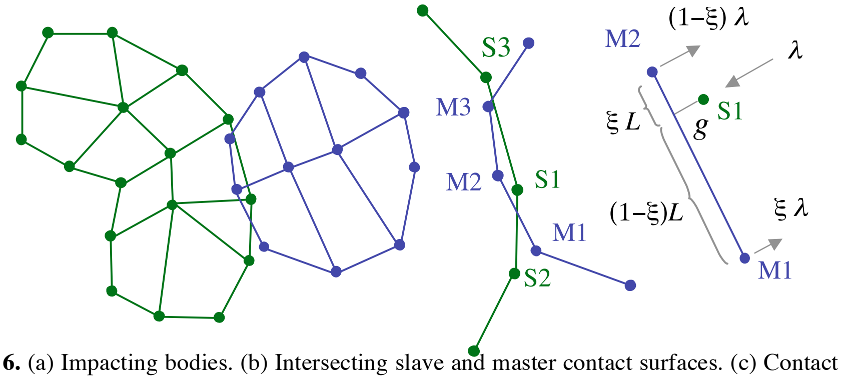

Fig. 6. (a) Impacting bodies. (b) Intersecting slave and master contact surfaces. (c) Contact gap based on closest point, contact forces at slave node S1 and master nodes M1 and M2.

Parallel Contact Search Algorithm

The node-on-segment contact methods described in the previous section require a search for the "nearest" master segments for each slave node (and slave segment for each master node for the double pass version). This process can be one of the most time consuming in an explicit finite element code. Master segments are generated automatically by computing the free/exposed element faces on the mesh. For large deformation problems, elements will become damaged and no longer active. This "element erosion" defines new master segments that must be accounted for. The parallel contact algorithm is based on a static domain decomposition of the mesh and a dynamic decomposition of the contact segments as follows:

1. Compute static decomposition of the mesh using METIS [2]. This is done once at problem initialization. Nodes on partition boundaries are assigned a home processor and are considered shared on remaining partitions.
2. On each static partition, define all free facets to be candidate contact segments and define all nodes attached to these facets to be candidate slave nodes. This is a double pass algorithm.
3. Based on a characteristic distance, subdivide the entire domain into bins and loop over all slave nodes and master segments to determine which bin it resides in.
4. A graph structure where each bin is a vertex and each master segment that overlaps bins is considered a connective edge,
5. Use METIS [2] to partition this graph structure to minimize edge cuts and assign bins to a partition.
6. Assign each "dynamic" partition to the processor that most of its contact nodes have for the "static" partition.
7. A serial algorithm (e.g. bucket sort) is done in each dynamic partition to find contact node-segment pairs. Segments that overlap partitions are shared, hence the METIS [2] partitioning reduces the search size.

8. Contact forces are computed on each partition and then communicated to the home processor. The home processor then communicates the contact forces to the remaining shared partitions (processors) if any.

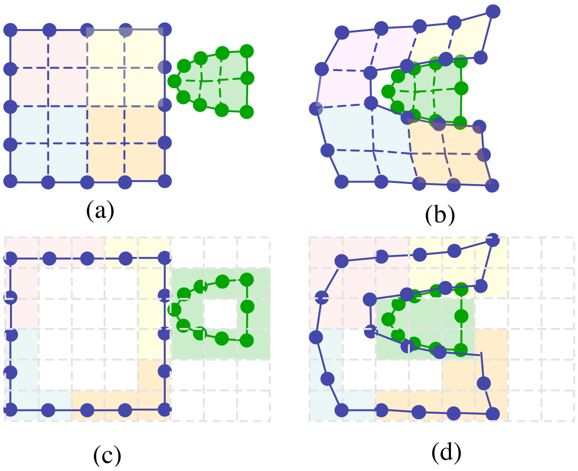

Fig. 7. (a) Static decomposition of mesh in un-deformed configuration. (b) Large deformations cause element erosion due to damage i.e. elements are eliminated. (c) The candidate contact segments are binned, partitioned according to METIS and placed on processors based on the home processors of the nodes in each bin. (d) This process is done dynamically as element erosion defines new candidate master segments and large motions change the graph structure of the contact bins.

This process is illustrated in Fig. 7. The simulation in Fig. 5 used this parallel search algorithm with the element erosion and the Lagrange multiplier contact. In that, the elements fail but the surrounding nodes are still considered point masses so that momentum is conserved. These point masses are included as potential contact nodes in the process described above and cause secondary damage in the following plate.

Meshless Methods

The meshless methods were developed to handle very large deformation problems where mesh tangling becomes an issue. That is, when elements get so distorted due to deformation, they are no longer usable (e.g. the invert). Meshless methods do not rely on elements to parameterize deformation but instead use shape functions that can operate on arbitrary point clouds. The overlap (Fig. 8) of elliptical meshless shape functions defines the graph structure of the discretization [3]. Since the meshless particles flow such that new support overlaps need to be defined on a regular basis for large deformation problems, a dynamic partitioning method identical to that for contact should be used. Fig. 9 shows a simulation of a steel penetrator going through a concrete slab with rebar.

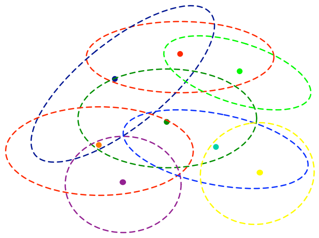

Fig. 8. Overlapping elliptical supports of meshless method.

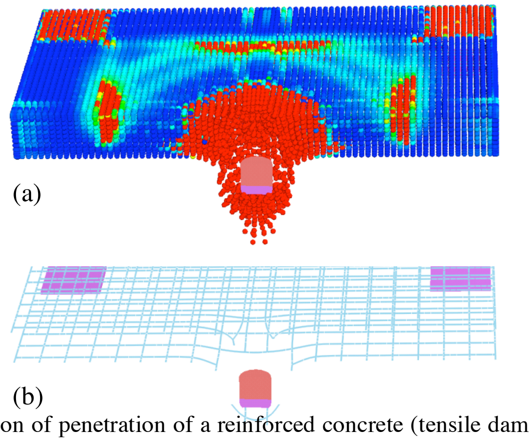

Fig. 9. (a) Simulation of penetration of a reinforced concrete (tensile damage shown in red) (b) Rebar (shown alone) was attached to particles in the simulation and eventually fails upon exit.

Implicit Finite Elements: DIABLO

Different strategies exist for how mechanics coupling is implemented in simulations codes. One of the simplest strategies is to take independent codes and then pass a limited amount of data (e.g. node positions, node temperatures and time) through an interface. This is the strategy for the coupling the DYNA3D structural mechanics mechanics code with the TOPAZ thermal mechanics code and the finite volume fluids code GEMINI. The codes themselves are standalone with minimum sharing of data structures and code reuse. This model can get more complicated when adaptive meshing (AMR) is employed since some common definition for the octree mesh needs to be defined. Because many of the operations needed to do error estimating and data remapping need to be written for data manipulation, a model that employs an ample amount of code reuse between the different mechanics becomes more tractable. As such, a single data structure for all the different mechanics would make this code re-use more practical. DIABLO is the latest MDG code project and employs the latter strategy. DIABLO currently has coupled solid mechanics, thermal mechanics, electromagnetics and diffusion along with adaptive meshing. Fig. 6 shows an example where AMR is applied to a coupled thermal structural simulation. Example applications include metal forming, rail gun etc. As with PARADYN, the contact algorithms are considered a key component of the code development of DIABLO and are highlighted here.

Implict finite elements: segment-to-segment contact

Node-on-segment algorithms are simple but have a number of flaws. In particular, they don't transmit stresses or fluxes smoothly across the boundary. This is particularly important for solid mechanics where non-smooth force transmission hinders converge of the implicit non-linear algorithm and in electromagnetics where Nedelec edge elements are used. Referring to Fig. 11, the mortar segment-to-segment algorithm for solid mechanics [4,5] computes a nodal gap based on the integral

$$g_A = \int N_A (x^s - x^m)\, d\Gamma \qquad \textbf{(7)}$$

Here the contact traction can be computed via a penalty $\lambda_A = \kappa g_A$ or an augmented Lagrangian $\lambda_A^{i+1} = \kappa g_A^i + \lambda_A^i$ computed through an Uzawa algorithm. Now the static, implicit discrete equations of motion are written

$$f_{n+1}{}^{\text{int}}(x_{n+1}) + f_{n+1}^c - f_{n+1}{}^{ext} = 0 \qquad \textbf{(8)}$$

where the unknown configuration x_{n+1} is typically solved for using the linearized form of Eq. 8 in a Newton Raphson (or Quasi-Newton) scheme. The additional mechanics types compute an analogous discrete balance equation. Fig. 12 illustrates the nonlinear algorithm used in DIABLO. Fig. 13 demonstrates the robustness of mortar segment-to-segment contact method.

Parallel contact search algorithm

The parallel implementation of this contact was kept simple for the sake of development and because communication costs for an implicit code are small compared to the solution of simultaneous equations. Here we choose to build an entirely static decomposition using METIS [2] and then add shared nodes from relevant contact surfaces on each partition (Fig. 14). Now a serial algorithm (e.g. bucket sort) can be performed on each processor for the contact search. Contact forces are only computed on contact home nodes and then scattered via point-to-point communications to shared nodes.

Multimechanics Examples

One of the main applications for the coupled solid electromagnetics is the rail gun. Here a high voltage is applied between two rails and an armature carries current between the rails (Fig. 15). Because of the transient time for the current to penetrate the rails and armature (magnetic diffusion), the current travels along the skin of the rails (i.e. skin effect). Ampere's law $\nabla \times \mathbf{B} = \mu \mathbf{J}$ predicts that the resulting magnet **B** field will be relatively high (pointing out) on the inside of the circuit (red) but small inside the surrounding iron armature and rails due to the low magnet permeability μ of the iron and high permeability μ of the air. This produces a Lorentz force per unit length, $\mathbf{F} = \mathbf{J} \times \mathbf{B}$ on the armature propelling it outward. Referring to Fig. 15, the model uses a grid for the armature (blue), two grids for the rails and four grids for the air. Only the rail and armature meshes consist of solid elements. All meshes are required to capture the magnetic field. Two fine (inner) air meshes move with the armature and interact with the surrounding air and rail through electromagnetic (mortar) contact surfaces. Both mechanical and electromagnetic contact is used between the armature and the two rails Snapshots of the 3D simulation are shown in Fig. 16 as the armature slides across the rails. Quarter symmetry is used to reduce the model size such that only the top, left-hand side of the mesh is shown. Here it is confirmed that the **B** field is very high at the back of the armature (i.e. fine mesh in Fig. 15) and low in the surrounding air. The problem had ~4.5 million degrees of freedom and was run on 16 processors on the ASC Purple platform in 20 hours.

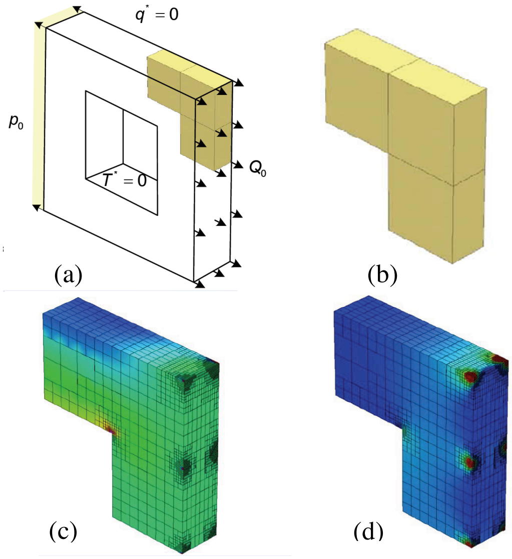

Fig. 10. Thermal-structural AMR example: (a) ring with (tensile) pressure loading (over faces) and concentrated thermal fluxes Q, (on nodes) on left and right side. (b) initial quarter symmetry mesh (c) effective stress and (d) temperature on final adapted mesh. Note that refinement was made near stress concentrations on the inner corner and locations where nodal flux loads reside.

Another important application is in nuclear engineering (Fig. 17). Here, the fuel bundle is composed of hexagonal fuel rods (metal alloy tubes filled with nuclear fuel). The fuel is meshed as a homogenous material and acts as a neutron heat source. The rods are inserted into three restraint plates with hexagonal slots to constrain the individual rods. The model in Fig. 17 is one-third symmetry and includes 390 contact surfaces between each of the 75 rods and also the slots in the restraint plates. The boundary conditions include neutron heat sources near the top center of the fuel bundle and thermal boundary conditions on the outer edges of the constraint plates. A separate neutronics package determined the heat sources in the model. The model was over 2 million degrees of freedom and was solved using 32 partitions on 32 (8 processor) nodes on the ASC Purple machine. Each partition had a dedicated processor and four threads per node were used for the parallel linear direct solver. The thermal gradient that results from the heating causes bending in the individual rods such that many large gaps open up between the adjacent rods near the top.

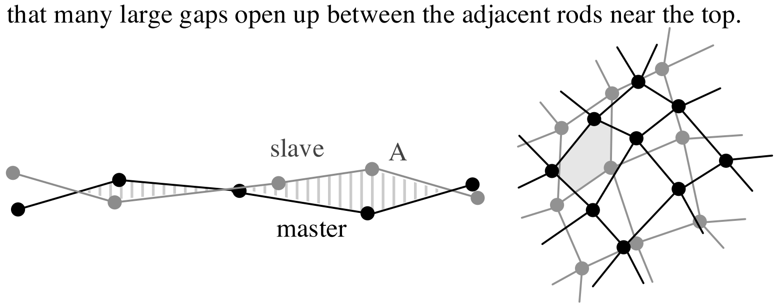

Fig. 11. Segment-to-segment contact computes the weighted volume between adjacent facets to get the nodal gap. This is simple in 2D (left). More sophisticated 3D algorithms require the intersection of adjacent slave and master segments to compute the nodal gap.

Discussion

Different methods were presented for doing parallel contact with applications in both large deformations with failure and multimechanics. A novel dynamic partitioning algorithm was presented for the contact searching and meshless particle methods. An overview of a one-of-a-kind parallel implicit solid-thermal-electromagnetics code was presented along with its novel contact algorithms capabilities. Future work includes embedded mesh techniques and coupled mechanics on different meshes.

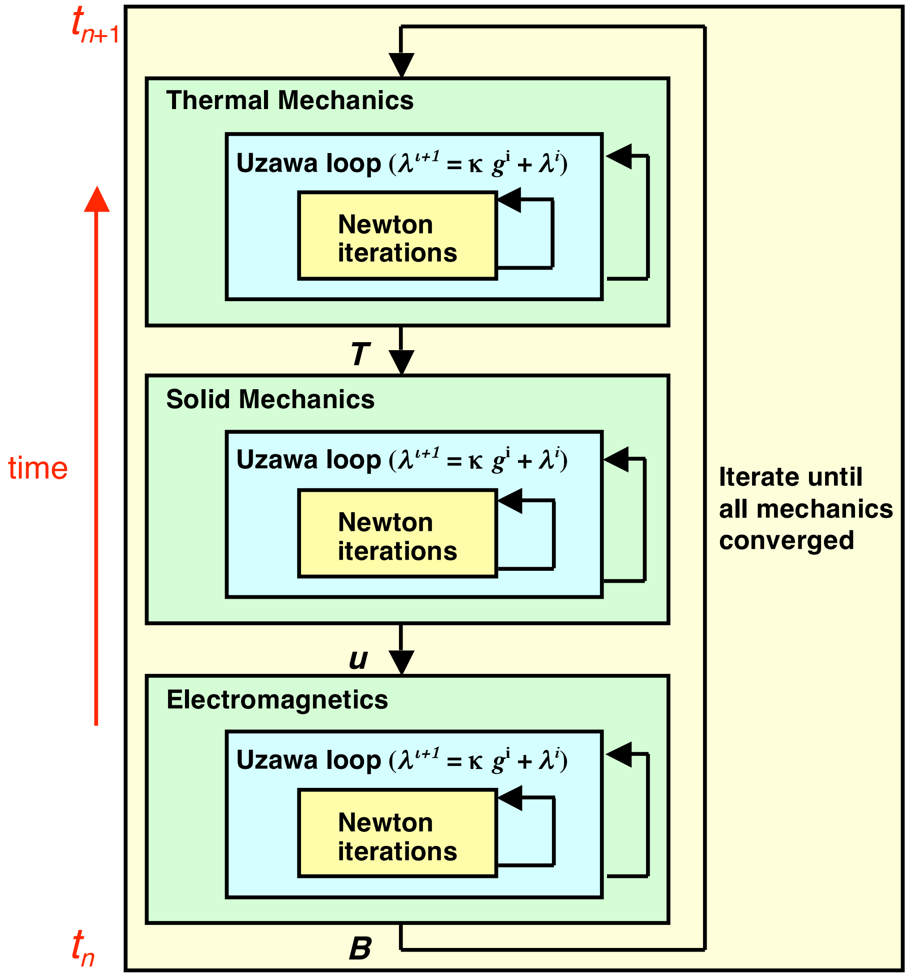
t_{n+1}
Thermal Mechanics
Uzawa loop ($\lambda^{\iota+1} = \kappa\, g^{i} + \lambda^{i}$)
Newton iterations
T
Solid Mechanics
Uzawa loop ($\lambda^{\iota+1} = \kappa\, g^{i} + \lambda^{i}$)
Newton iterations
u
Electromagnetics
Uzawa loop ($\lambda^{\iota+1} = \kappa\, g^{i} + \lambda^{i}$)
Newton iterations
B
t_n
time
Iterate until all mechanics converged

Fig. 12. Flow of nonlinear solution algorithm over a time step. A Uzawa loop is included to enforce augmented Lagrange type contact for each mechanics.

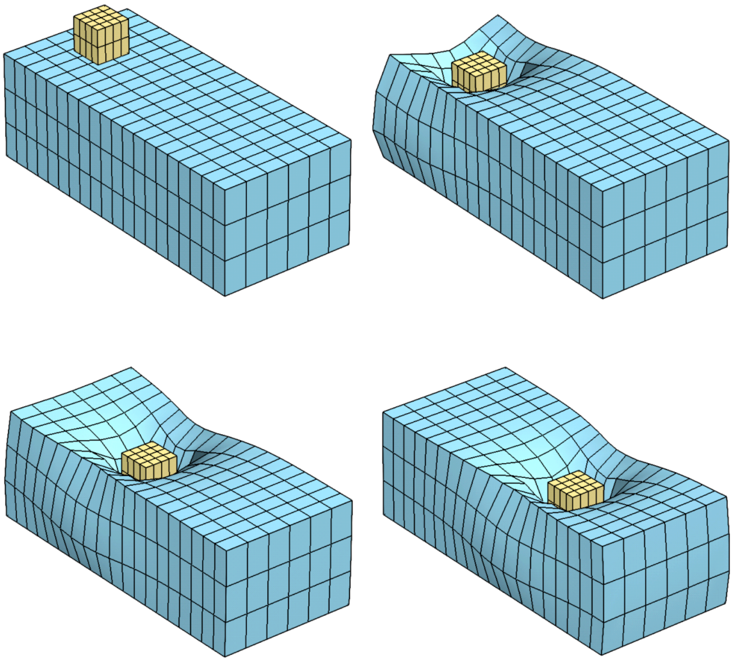

Fig. 13. Sequence of deformations of steel block penetrating rubber block and sliding. Because of the sharp corners and the "bumpy" segments, this problem can't be solved using node-on-segment with an implicit scheme.

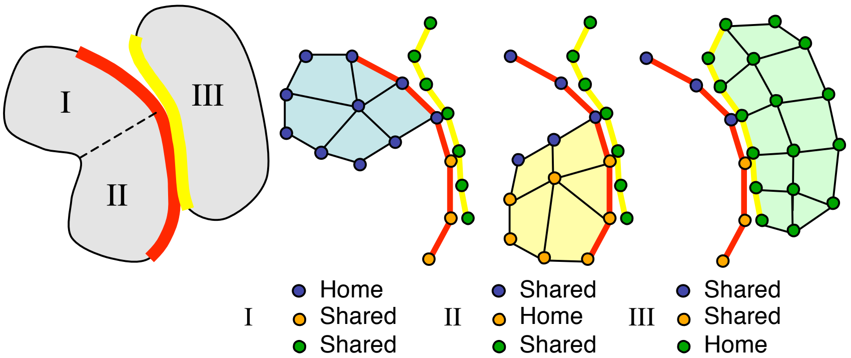

Fig. 14. DIABLO builds a static domain decompositions and then adds shared nodes to each partition so that all relevant contact nodes reside on the partition. So, for example, partition II computes contact gaps on its three (orange) home contact nodes using coordinates from its shared (green,blue) nodes. Forces are then communicated back to shared nodes..

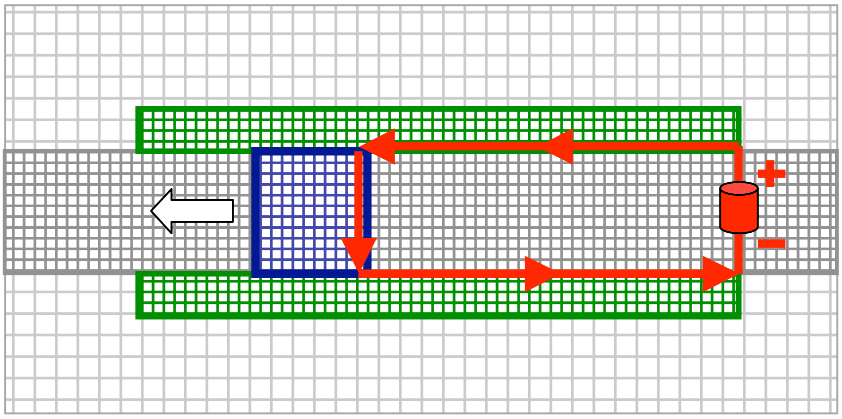

Fig. 15. A 2D slice of a 3D mesh used for electromagnetic rail gun. A large voltage is applied across the rails (green) and the resulting current (red) stays close to the surface as it passes across the armature (blue).

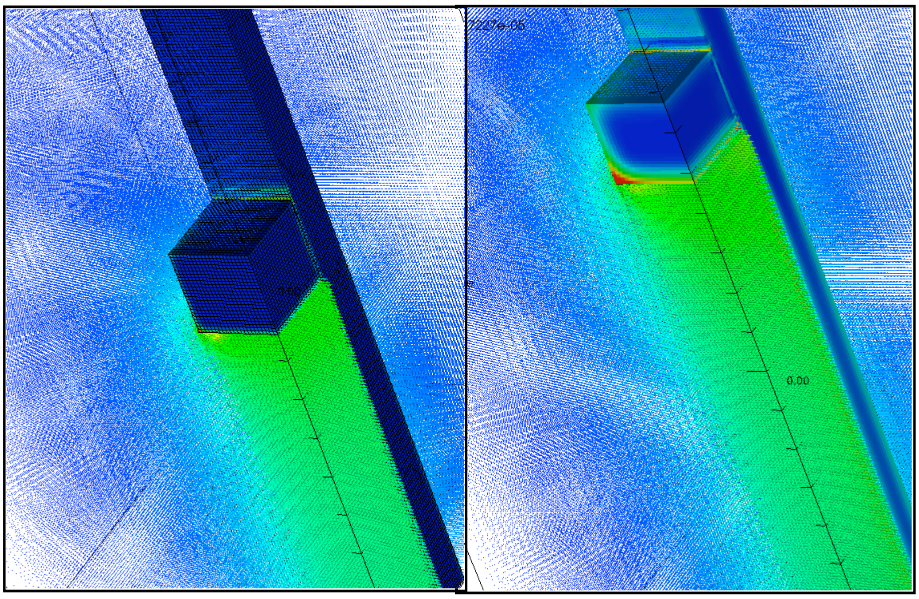

Fig. 16. A quarter symmetry simulation of the 3D rail gun simulation. The trailing **B** field is very high (green) whilst the **B** field in the air (blue) is very low. Only a little current is visible in the armature and rail at the early time, but it becomes particularly apparent in the outside portion (red) of the armature at the later time.

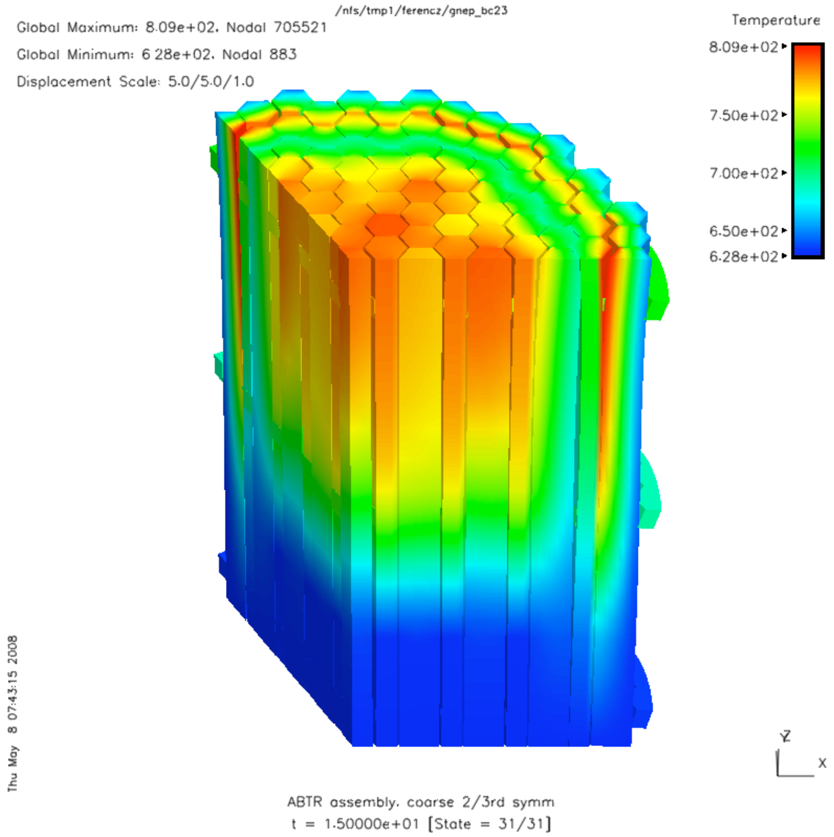
/nfs/tmp1/ferencz/gnep_bc23
Global Maximum: 8.09e+02. Nodal 705521
Global Minimum: 6.28e+02. Nodal 883
Displacement Scale: 5.0/5.0/1.0
Temperature
8.09e+02
7.50e+02
7.00e+02
6.50e+02
6.28e+02
Thu May 8 07:43:15 2008
Z
X
ABTR assembly. coarse 2/3rd symm
t = 1.50000e+01 [State = 31/31]

Fig. 17. Nuclear fuel bundle (one-third symmetry). Neutron heat sources cause thermal gradient and bending of fuel rods forming large gaps in contact surfaces between adjacent rods.

References

1. Zywicz, E., Puso, M.A.: A General Predictor-Corrector Solver for Explicit Finite-Element Contact. Int. J. Numer. Meth. Eng. 44, 439-459 (1999).
2. Karypis G, Kumar V. Metis: A software package for partitioning unstructured graphs, partitioning meshes and computing fill-reducing orderings of sparse matrices, version 4.0, University of Minnesota, Department of Computer Science, (1998).
3. Puso, M.A., Chen, J.S., Zywicz, E., Elmer, W.: Meshfree and Finite Element Nodal Integration Methods. Int. J. Numer. Meth. Eng. 74, 416-446 (2008).
4. Puso, M.A., Laursen, T.A., A Mortar Segment-to-Segment Contact Method for Large Deformation Solid Mechanics. Comput. Meth. Appl. M. 193, 601-629 (2004).
5. Puso, M.A., Laursen, T.A., A Mortar Segment-to-Segment Frictional Contact Method for Large Deformations. Comput. Meth. Appl. M. 193, 4891-4913 (2004).

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.