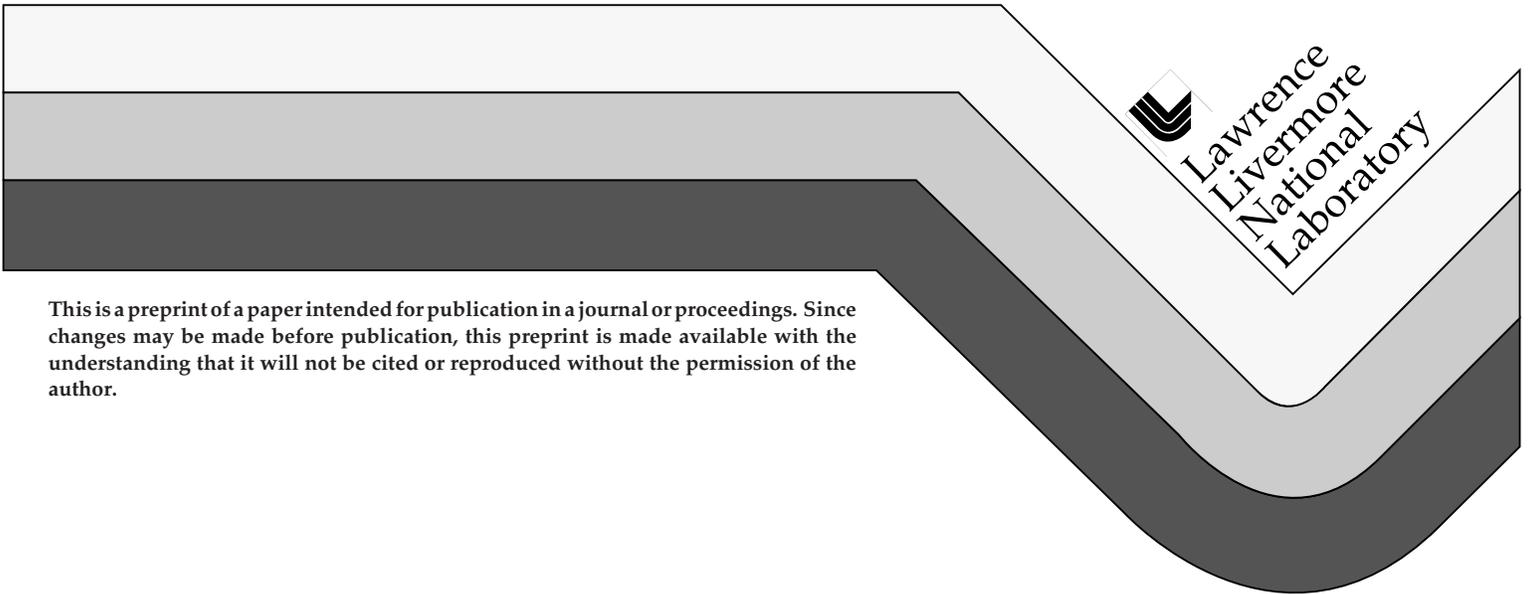


Periodic Boundary Conditions in a 3D Hydro Code

D. L. Morgan, Jr., J. R. Neely, H. C. Vantine

This paper was prepared for submittal to the
Modern Trends in Computational Physics
Dubna, Russia
June 15-20, 1998

September 18, 1998



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.

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.

Periodic Boundary Conditions in a 3D Hydro Code

David L. Morgan, Jr., J. Robert Neely, and Harry C. Vantine

Lawrence Livermore National Laboratory, Livermore, California 94551, U.S.A.
morgan6@llnl.gov

Abstract

We have modified a 3D hydrodynamics code so that it has the capability to impose periodic boundary conditions on the problem being considered. This capability allows it to treat only a basic symmetry unit of the problem when translational or rotational periodic symmetries are present. The code has been run successfully for two test problems involving rotational symmetries.

Introduction

When symmetries are present in hydrodynamics problems, the computational effort required to solve them can be substantially reduced. While most hydrodynamics computer codes can deal with reflective symmetries, it is less common that they can handle periodic symmetries through the use of periodic boundary conditions. Recently we have added this latter capability to a 3D, finite element, time-explicit hydrodynamics code capable of running on parallel computers. At present the capability is only available for pure Lagrangian problems, but when fully implemented to run with advection present, it will enable the code to solve efficiently problems such as jet-engine blade simulation, where it need include only a single blade in the calculation instead of all blades equally positioned around an axis.

Periodic boundary conditions arise when the symmetries involved in a problem are such that it can be fully described as consisting of two or more units which are identical in all of their features except for position and orientation. Under these circumstances, each smallest possible unit (basic symmetry unit) is a volume (in 3D) surrounded by a number of surfaces, each of which is a member of a pair within which the two surfaces are identical except for position and orientation. They are identical in shape as well as in the distributions upon them of intrinsic quantities, such as pressure, density, and velocity. Periodic boundary conditions consist of requiring that these identities are present in each basic unit. When the conditions are applied, it is then necessary to solve the problem for only a single such unit.

Periodic conditions can exist in conjunction with reflective conditions, in which case the basic symmetry units for each may or may not be the same. If they are the same, the results of applying either type of boundary condition will be the same, but applying reflective conditions is simpler.

Implementation

In the code, all hydrodynamic quantities are numerically represented by, or are derived from, the positions of a set of points in three dimensional space, called nodes, that form the mesh for the numerical model of the problem. These quantities include the node-centered quantities, position and velocity, as well as the element-centered quantities such as energy, density, and stress, whose values ultimately depend on the positions of the eight nodes surrounding the element. The motion of these nodes defines the motion of the material as time progresses.

Periodic boundary conditions are implemented by first requiring the user to specify the pairs of boundaries (surfaces) that are periodically symmetric. This is done by specifying their nodesets. A nodeset is simply a subset of the nodes in the mesh and can be defined either during mesh generation, which occurs before running the hydrodynamics code, or by using input commands built into the code itself. The user must, in addition, supply a description of how to translate and/or rotate the first nodeset of each pair so that it will overlay the second. It is required that after this translation and rotation, the nodes in each set match up one-to-one.

The motion of each node is determined by a force calculation. During this calculation, forces on the nodes on each boundary contribute to the forces on each of the corresponding nodes on the other boundary of each pair. If the boundaries are such that they are rotated relative to each other, then the force vector must be rotated as well before adding in the contribution. This exchange of partial forces between boundaries ensures that the boundaries will maintain the same shape relative to each other, differing only by their rotational relationship in 3D space.

In some problems, the boundaries in a symmetric pair intersect with each other. In the test problems described below, the periodic boundaries are 90 degrees to each other, with a 1D line of nodes which falls into both nodesets. Force calculations on these nodes are handled separately from the other non-intersecting nodes. It is required that the angle, theta, between the nodesets on either side of the intersection is such that it divides evenly into 360 degrees, as is the case for rotational periodic symmetry. Then the force vector is rotated and added to the original N times, where N is $360/\theta$. For example, the force along two of the intersections in the problem described below is $F + F_{90} + F_{180} + F_{270}$, where F_r is the original force, F , rotated around r degrees. This, of course, results in a net force of zero perpendicular to the axis of rotational symmetry (such axes are invariant) and provides a check on the accuracy of the code.

The implementation is fully parallelized, and the test problems described here were all run on the ASCI Blue Pacific machine at Lawrence Livermore National Laboratory using 32 processors.

Testing

We have tested the 3D periodic boundary condition capability by applying it to a problem consisting of an infinite volume of high explosive detonated by an infinite, repetitive array of lighting points. The basic symmetry units here are initially cubes. This high energy test case was chosen so that significant surface distortions would occur. Two lighting-point arrays were used, single layers of which are described in Figures 1 and 2. The array of cubes is shown in Fig. 3. The first lighting array is such that only periodic conditions are present. With the second, both periodic and reflective conditions are present, but some of the symmetry units are different.

In the test with the first array, ALE3D demonstrated the required surface identities at all times. Those results are not shown here because they are embodied in the results of the test with the second array. In that test the effects of periodicity were precisely reproduced (Figures 4 and 5), and even though only periodic boundary conditions were employed, the results show the reflective symmetries that exist in the problem as accurately as possible with a mesh which does not itself possess those symmetries (Figures 5 and 6).

Conclusion

We have successfully added a periodic-boundary-condition capability to our 3D Lagrangian hydrodynamics code. Work is in progress to generalize this capability to a version of the code that includes advection so that we can solve problems involving flow through the surfaces of the basic symmetry unit.

Acknowledgment

This work was performed in part by Lawrence Livermore National Laboratory under the auspices of the US Department of Energy under contract W-7405-ENG-48.

Keywords

3D, hydrodynamics, boundary conditions, periodic

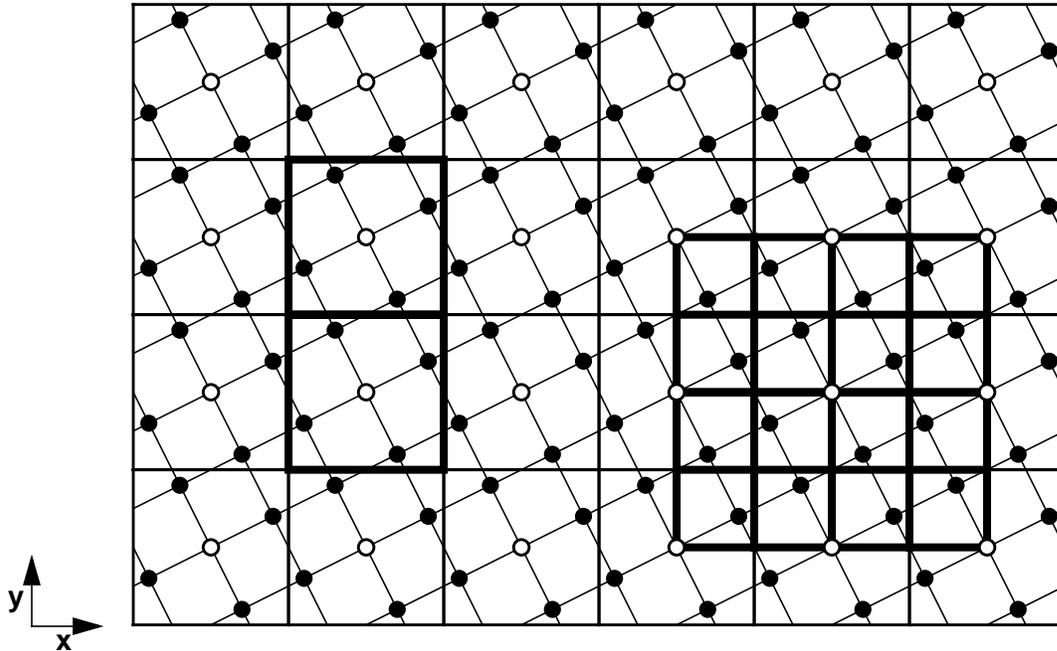


Fig. 1. Positions of lighting points (dots) in one particular x-y plane for the tests of the periodic-boundary-condition capability. Only the lighting points at the solid dots were employed in the first test. Such an array has no reflective symmetries. The large squares (two are highlighted on the left) are basic units of translational periodic symmetry for this array. When rotational periodic symmetry is considered, the basic units are smaller ones, some of which are shown on the right. Their symmetries are shown in Fig. 2. In the second test, lighting points at the positions of the open dots were included. This adds a number of obvious reflective symmetries while retaining all periodic ones. (Also added are more periodic symmetries. Of these, the rotational ones are around the corners, centers, and edge centers of each small square of the array of lighting points. These were not considered, but their effect can be seen in Figures 4., 5., and 6.)

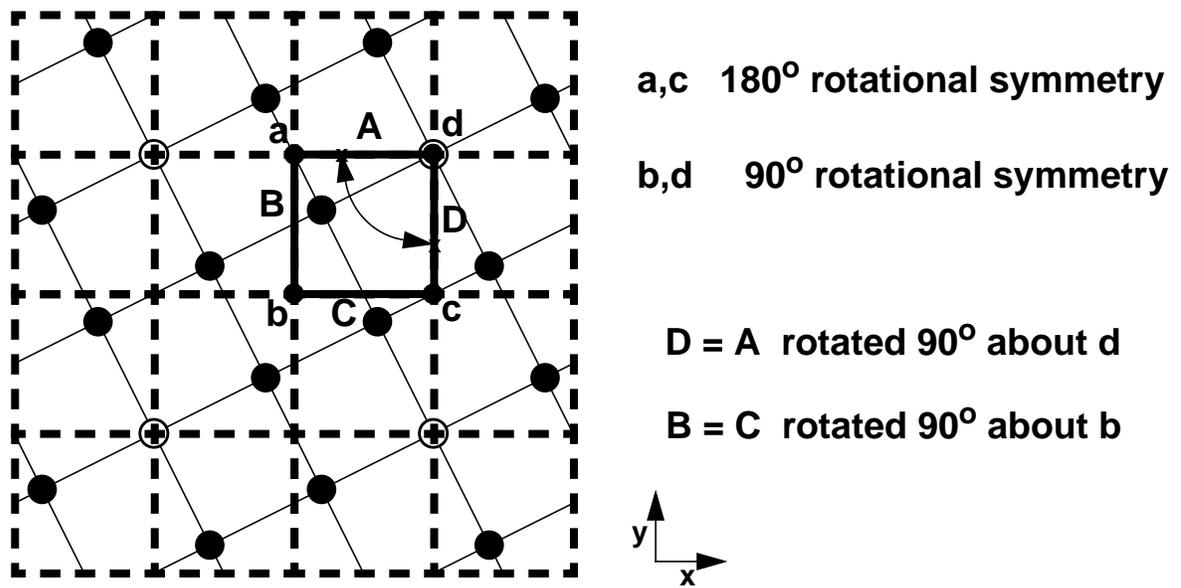


Fig. 2. The basic rotational periodic symmetry units (dashed edges) in an x-y plane containing lighting points (dots) for either array described in Fig. 1. The rotations indicated will move the highlighted unit into coincidence with neighboring units. A result of this is that all quantities on side A are identical to those on side D (after rotation of the vector quantities like velocity) at corresponding locations. The same is true for sides B and C.

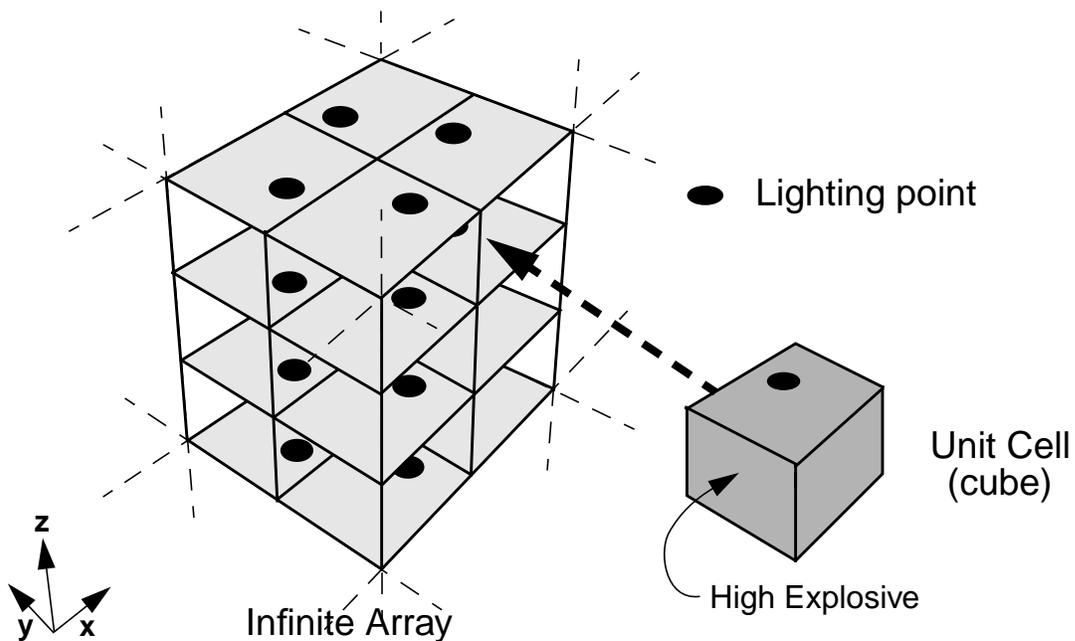


Fig. 3. Geometry for the test problems. The lighting points are shown as they were for the first test.

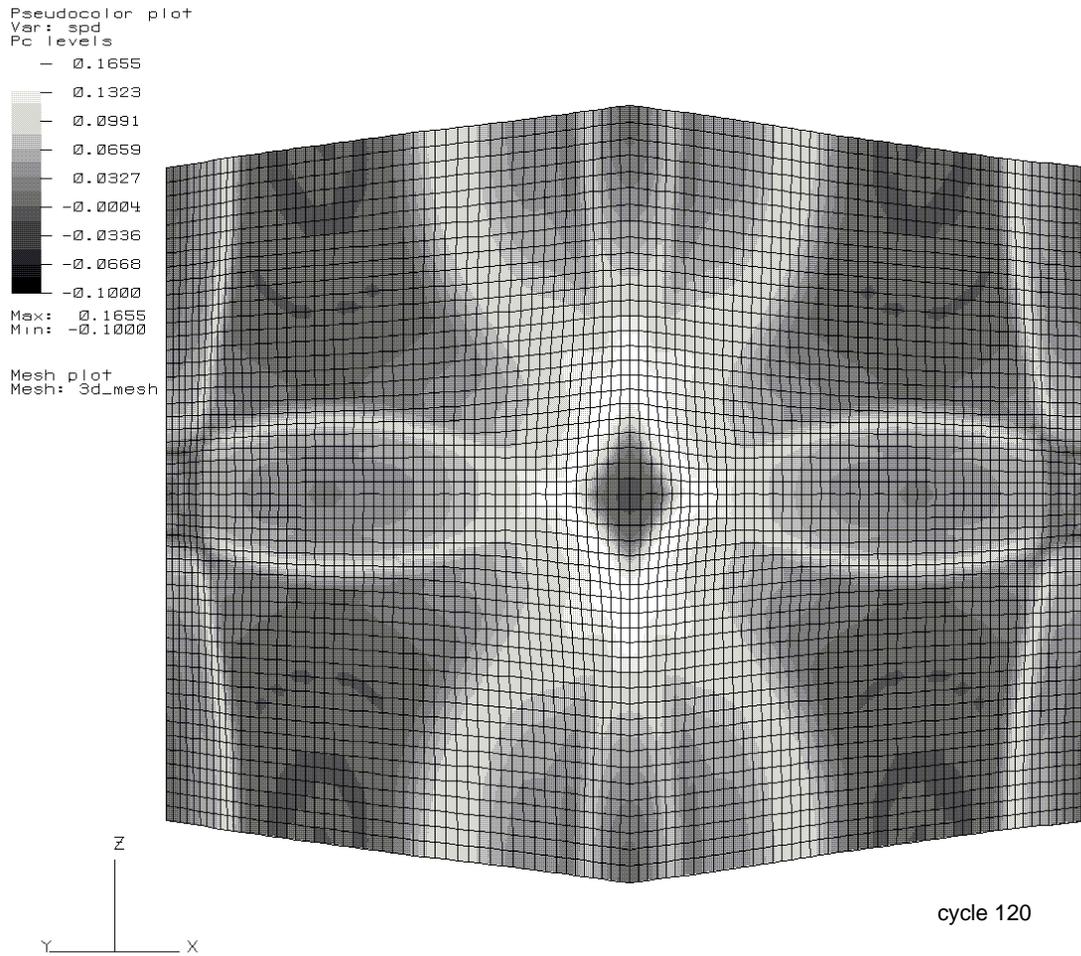


Fig. 4. A grey-scale intensity plot of material speed ($\text{cm}/\mu\text{s}$) in the second test of the code at a time when all of the high explosive has detonated with the exception of a small portion near the center of the figure. The view is of the two surfaces of the now distorted cube which correspond to lines B (left surface) and C (right surface) of Fig. 2. The periodic symmetry (invariance under rotation by 90 degrees about the near, vertical edge) requires that the left and right portions of the view be mirror images of each other, except for small variations due to distortions from flatness of the cube's faces. This is seen to be the case. Note that each vertical edge of the cube is an axis of rotational symmetry; thus they are invariant, and it may be seen that they have remained vertical and straight. Note also the reflective symmetry (unimposed) about the mid-z plane which results from the fact that the lighting point patterns are identical and equally separated in the z direction.

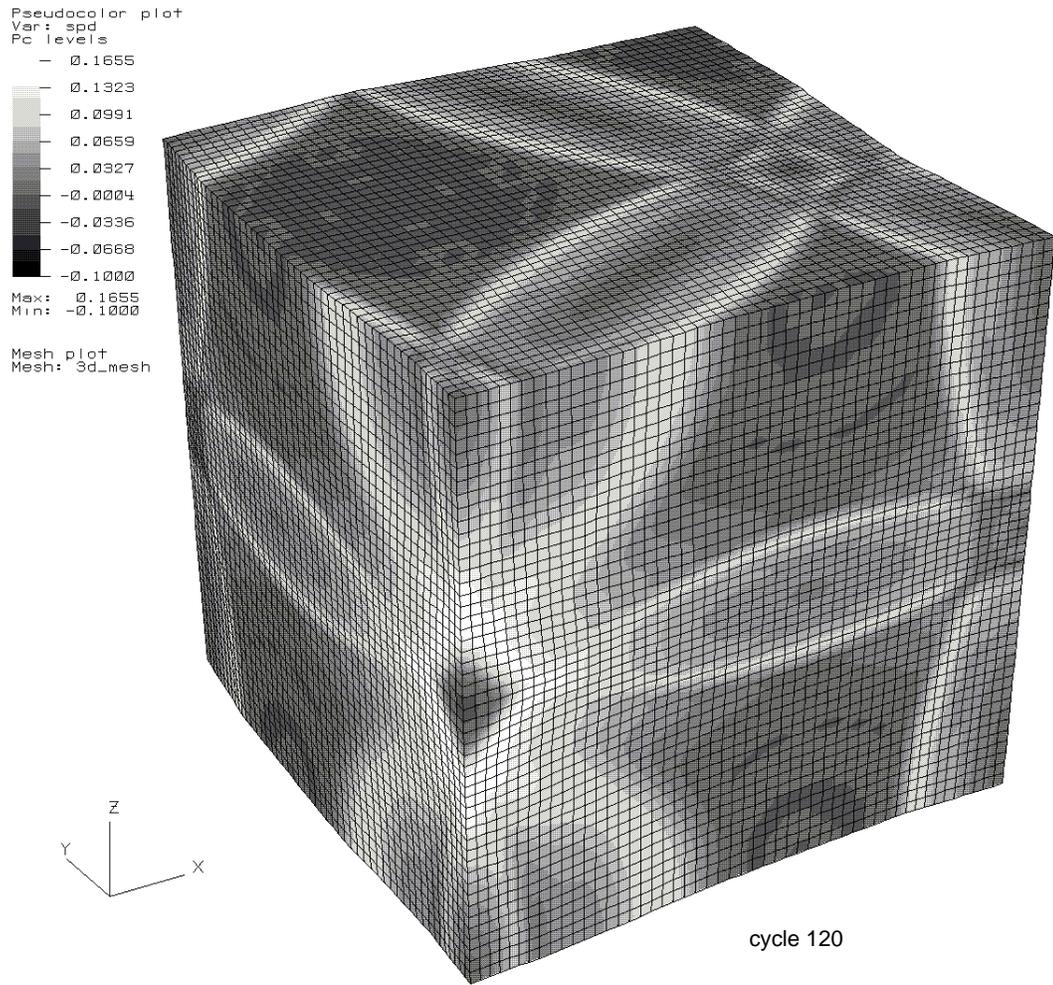
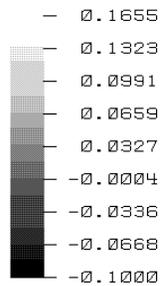


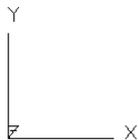
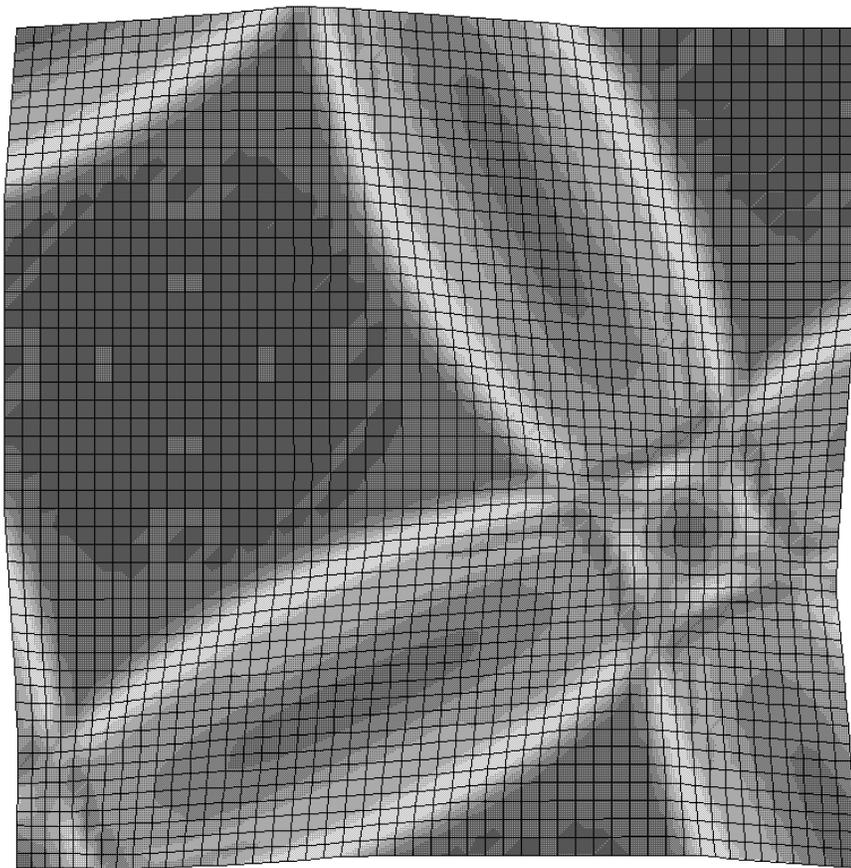
Fig. 5. A grey-scale intensity plot of material speed ($\text{cm}/\mu\text{s}$) at the same time as in Fig. 4. The distorted cube has been rotated relative to the view in Fig. 4 so that the top may be seen.

Pseudocolor plot
Var: spd
Pc levels



Max: 0.1655
Min: -0.1000

Mesh plot
Mesh: 3d_mesh



cycle 120

Fig. 6. A grey-scale intensity plot of material speed ($\text{cm}/\mu\text{s}$) at the same time as in Fig. 4. The view is of the upper surface (maximum z coordinate) looking toward negative z . The upper edge and right edge are the 90 degree rotations of one another as are the left and lower edges, thus demonstrating the imposed rotational periodicity. The pattern shows the unimposed (but present) reflective symmetry about various lines with $y-x$ slopes of -2 and $1/2$ as expected from the detonation pattern employed in the second test of the code.