Periodic Boundary Conditions in the ALEGRA Finite Element Code

A. C. Robinson, J. R. Weatherby and J. B. Aidun

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories
Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent 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, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (703) 605-6000
Web site: http://www.ntis.gov/ordering.htm

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
Periodic Boundary Conditions in the ALEGRA Finite Element Code

A. C. Robinson and J. R. Weatherby
Computational Physics Research and Development

J. B. Aidun
Material Mechanics

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0819

Abstract

This document describes the implementation of periodic boundary conditions in the ALEGRA finite element code. ALEGRA is an arbitrary Lagrangian-Eulerian multi-physics code with both explicit and implicit numerical algorithms. The periodic boundary implementation requires a consistent set of boundary input sets which are used to describe virtual periodic regions. The implementation is noninvasive to the majority of the ALEGRA coding and is based on the distributed memory parallel framework in ALEGRA. The technique involves extending the ghost element concept for interprocessor boundary communications in ALEGRA to additionally support on- and off-processor periodic boundary communications. The user interface, algorithmic details and sample computations are given.
Acknowledgments

The authors wish to thank Mark Blanford, Randall Summers and Rebecca Brannon for their careful reviews of this manuscript and insightful comments. Developing a parallel finite element code framework is a major undertaking and we recognize the contributions of the many ALEGRA team members to the code infrastructure necessary for this work. Chris Garasi and Bryan Oliver have been users of the periodic boundary conditions and have not been shy in pointing out observed deficiencies.
# Table of Contents

1. Introduction ............................................................................................................ 11

2. The Supported Transformations ............................................................................. 12
   2.1 Translational periodicity .................................................................................................. 12
   2.2 Rotational Periodicity ....................................................................................................... 13
   2.3 “Potential” Dependent Variables .................................................................................. 14
   2.4 Number of Required Node Sets ..................................................................................... 14

3. The Periodic Boundary Condition Implementation ............................................... 17
   3.1 Updating ALEGRA to allow for inter-processor communications. ............................ 17
   3.2 Identifying matching nodes sets .................................................................................... 17
   3.3 Internal implementation of periodic boundary conditions ......................................... 19
   3.4 Message Tags .................................................................................................................. 20
   3.5 Scratch Maps ................................................................................................................... 20
   3.6 Implicit Algorithms ......................................................................................................... 20

4. Example Calculations ............................................................................................ 24
   4.1 Exploding wire in a wedge. (Verification/2D/periodic/rotper) ...................................... 24
   4.2 Exploding wires (Verification/2D/periodic (lag, eul, advect))....................................... 24
   4.3 Anisotropic shock propagation (Verification/3D/periodic/hexqzt) ............................... 26
   4.4 Periodic Electric Field (Verification/3D/qsem_physics/ballbox) .................................... 29
   4.5 Periodic Magnetohydrodynamics (Verification/2D/mhd_physics/cnvct45) .................... 30

5. Unsupported Features and Unresolved Problems .................................................. 32

6. Conclusions ............................................................................................................ 33
List of Figures

Figure 1 Basic Translational and Rotational Periodicity Cases ...........................................12
Figure 2 2D node sets for a case where there are no point boundaries and therefore no point node sets .................................................................15
Figure 3 2D required node sets for a box. a) single periodic direction b) the fully periodic case .................................................................16
Figure 4 Sample decomposition for description of parallel noderset matching ........18
Figure 5 Periodic Extension. Center region has 8 neighboring images. Ghost element numbers are labeled. Circled nodes and edges (faces in 3D) are redundant and are not included in linear system solve ........................................21
Figure 6 Offset pressurized wire in a 2D periodic wedge .............................................24
Figure 7 Point sources in a periodic box (left:lag, center:eul; right:advect) ..........25
Figure 8 Two wave structure and transverse motion in the horizontal periodic direction is seen for a wave whose structure depends only on the vertical coordinate. The wave is propagating downward and displacements have been exaggerated .........................................................28
Figure 9 Plots of (solid), (dotted) and electric field component (dashed) for dielectric sphere in a periodic medium ..............................................29
Figure 10 Power rates in cnvct45 test problem .........................................................30
Figure 11 Energy histories from cnvct45 test problem .............................................31
Figure 12 Temperature distribution at early time and late times ..............................31
List of Tables

TABLE 1: Required node sets for a 3D periodic box..................................................15
TABLE 2: Plane Wave Propagation at 73° to c-axis in “Hexagonal Quartz” .............27
TABLE 3: Plane Wave Propagation at 73° to c-axis in “Hexagonal Quartz” .............27
1. Introduction

Many problems in engineering analysis can effectively utilize periodic boundary conditions. For example, anisotropic materials allow for waves which depend on only one spatial dimension yet at the same time admit transverse motion. Periodic boundary conditions in the transverse direction can reduce the lateral extent required to model this one-dimensional motion using a two- or three-dimensional code. Magnetohydrodynamics is another set of equations which admits such solutions. Other problems require periodic boundary conditions as the natural means of achieving boundary independence such as in the study of scale cascades in turbulence or in microstructural simulations. In some cases the problem may be required to be periodic in an angular direction such as a pie slice geometry for z-pinch modeling. In most cases, the periodicity reduces to simple translations.

ALEGRA is being developed at Sandia National Laboratories for modeling a great variety of shock wave physics phenomena[1]. It is desirable that ALEGRA support within its basic infrastructure these various periodic boundary conditions for both explicit hydrodynamics, including Lagrangian and Eulerian modeling, as well as modeling of physics which requires the solution of linear systems (i.e implicit algorithms). This report describes the implementation approach for periodic boundary conditions in ALEGRA.

A standard implementation of periodic boundary conditions in explicit transient dynamics finite element codes is to first identify the set of corresponding periodic nodes and then follow any assembly of forces to these nodes by an additional summation of the forces on the corresponding periodic boundary nodes. Unfortunately this straightforward algorithm does not carry through either for implicit algorithms or for Eulerian algorithms. One solution which has been implemented for the Sandia code MP-SALSA is to define a truly periodic mesh by modification of the element connectivity lists and explicitly updating nodal locations when needed in order to get the correct gradients and volumes[2]. This approach seems too invasive for use in ALEGRA. It requires that the user apply special software to modify the initial mesh. Postprocessing also appears to require special modifications. It is desirable to have a periodic mesh capability in ALEGRA which can be applied to a normal mesh satisfying proper periodicity conditions on the nodes of the finite elements which lie on periodic faces, edges and nodes. The user may also want to apply different boundary conditions to the same mesh. The complexity of modifying the mesh for the periodic boundary conditions seems undesirable and perhaps error prone. Another code at Sandia which supports periodic boundary conditions in serial is the JAS3D code which is a quasistatic explicit Lagrangian mechanics code [3].

One technique for implementing the periodic boundary conditions in ALEGRA is to extend the ghost element infrastructure inherent in distributed memory parallel processing. This is minimally invasive to the code since only boundary communications are affected and periodic boundary conditions can be effectively partitioned from the rest of the code. The ghost element technique does require additional memory and processing but is minor compared to the advantages available. This approach is taken in ALEGRA and will be described in detail in this report.
2. The Supported Transformations

The ALEGRA user must supply an initial mesh which supports the periodic boundary conditions requested. This means that consistent matching node sets are required for each set of periodic boundaries. For the mesh to be allowable, the number of nodes in matching node sets must be the same and each node in one node set must correspond one-to-one to a node in the matching node set. The user may pick either a translation or a rotation for each periodic boundary condition as illustrated in Figure 1 below. If periodic boundary conditions are specified on a particular set of nodes then these are the only boundary conditions which may be specified there. This is because from the point of view of the code there is no boundary in the periodic directions. Attempts to specify additional boundary conditions on the periodic boundaries will result in incorrect results.

2.1 Translational periodicity

User input for translational periodicity is

PERIODIC BC, {NS1}, TRANSLATE {u}, {NS2} [TOLERANCE &]

where NS1 and NS2 are the first and second node sets, u is the translation vector which maps the nodes in NS1 to NS2. & is a tolerance value described later. That is,

\[ x_2 = x_1 + u \]  

For example, user input for an x-periodic box with unit dimensions would be

periodic bc, nodeset 1, translate, x 1. y 0., nodeset 2

with the node set on the left and right boundaries identified by 1 and 2, respectively. The tolerance value is optional and is described in Section 3.2. Whenever node and element quantities are mapped across periodic boundary for the translational periodicity case, only the physical coordinates are modified. This is much simpler than the rotational periodicity case described below in Section 2.2.
2.2 Rotational Periodicity

User input for rotational periodicity in two dimensions is

\[
\text{PERIODIC BC, \{NS1\}, ROTATE \{\theta\} POINT \{p\} \{NS2\} [TOLERANCE \varepsilon]}
\]

and in three dimensions

\[
\text{PERIODIC BC, \{NS1\}, ROTATE \{\theta\} POINT \{p\} AXIS \{a\},\{NS2\} [TOLERANCE \varepsilon]}
\]

\(a\) is the rotation axis passing through the point \(p\) which rotates \(\text{NS1}\) to \(\text{NS2}\) through an angle \(\theta\) given in degrees. Figure 1 shows an example which would be described by

\[
\text{periodic bc, nodeset 3, rotate 35.0, point, x 0. y 0., nodeset 4}
\]

In general, this case requires transformations of vector and tensor variables. That is,

\[
x_2 - p = R(x_1 - p) = V \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T(x_1 - p)
\]

(2)

where \(R\) is the rotation matrix. \(R\) is generated as shown using the matrix \(V = \{b, c, a\}\) composed of the column vectors \(b, c\) and \(a\) where the vectors \(b\) and \(c\) are orthogonal unit vectors perpendicular to the unit vector \(a\) called the axis of rotation. \(V\) is a real unitary matrix. The code ensures that the input vector \(a\) is converted to a unit vector even if it is not input as such. In two dimensions we can take \(V\) to be the identity tensor since the axis of rotation is assumed to be perpendicular to the two-dimensional plane. In three dimensions we define

\[
b = \frac{a \times \hat{s}}{|a \times \hat{s}|}
\]

(3)

where \(\hat{s}\) is the unit vector corresponding to the smallest absolute value coordinate of \(a\) and

\[
c = a \times b
\]

(4)

For rotational periodicity, scalars transform unchanged, vectors transform according to

\[
v_2 = Rv_1
\]

(5)

and tensors transform according to

\[
T_2 = RT_1R^T
\]

(6)
The Supported Transformations

since it must be that

\[ T_2y_2 = RT_1y_1. \]  

(7)

2.3 “Potential” Dependent Variables

Implicit in the design of the periodic boundary condition implementation in ALEGRA is that only the above transformations are required. However, there are some special boundary value problems for which true periodicity does not extend to all the dependent variables. For example, consider a pressure driven pipe flow in which periodicity in the flow direction might be assumed on top of a constant pressure gradient. The forces are periodic but the pressure has a term which is linear (not periodic) in the flow direction. Similarly, a periodic electric field potential cannot support a non-zero mean electric field. When non-zero mean fields are required, the best way to handle these cases is to reformulate the equations to include the mean field explicitly in order that all dependent variables are periodic unknowns. Such capabilities have been added in two important instances to be described in Section 3.6.1 and Section 3.6.2.

2.4 Number of Required Node Sets

The user must define enough node sets to completely specify all adjacent periodic image regions. Each bounding image will correspond to a single node set. Think of picking up the primary region and placing it in one of the bounding image regions. The two nodes sets that exactly overlap in this process are the matching node sets which must be associated with each other in an input command line. The proper number of node sets that appear in all the periodic boundary input command lines is equal to the number of periodically extended regions which touch the primary region. The number of periodic boundary input lines is exactly half this number. This count must be done carefully since the node sets do not necessarily correspond geometrically to objects of the same dimension. In two dimensions it is possible to have bounding node sets that are all lines. In other cases some node sets are lines and some are points. In three dimensions periodic boundary node sets can be surfaces, lines or points. An example of a periodic extension in 2D which does not include any point node sets is given in Figure 2. There are six neighboring images (A-F) of the primary region (Z) and thus six node sets. The matching node sets are (1,2),(3,4) and (5,6) with corresponding translation vectors (0.25,1.0), (1.0,0.7) and (0.7,0.3), respectively. A two-dimensional square requires only 2 node sets to describe periodicity in one direction as shown in Figure 3a. Periodicity in both directions would require 8 node sets since there are 8 neighboring periodic image regions. In Figure 3a we see that node set 1 matches node set 2 for a single periodic direction. For the fully periodic 2D case shown in Figure 3b the matching node sets are (1,2),(3,4),(5,6) and (7,8). Node sets 5,6,7 and 8 each consist of only a single node. Table 1 describes the required node sets for a 3D periodic “box”. The box is the most obvious object which can be extended periodically. It is however a special case. A general periodic object will more likely have fewer neighbors than the box and thus fewer required node sets to describe the periodicity. Again, the required number of node sets is equal to the number of periodic images that border the actual or primary region.
The Supported Transformations

Figure 2  2D node sets for a case where there are no point boundaries and therefore no point node sets.

<table>
<thead>
<tr>
<th>Number of Periodic Directions</th>
<th>Faces</th>
<th>Edges</th>
<th>Corners</th>
<th>Required node sets</th>
<th>Input command lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>24</td>
<td>12</td>
</tr>
</tbody>
</table>

TABLE 1: Required node sets for a 3D periodic box
The Supported Transformations

Figure 3 2D required node sets for a box. a) single periodic direction b) the fully periodic case.
3. The Periodic Boundary Condition Implementation

Implementation of the periodic boundary condition algorithm in ALEGRA required several steps as described below. The term “processor” is used exclusively to describe a single networked computer running a single instance of ALEGRA. The term “node” refers to a nodal quantity in the finite element sense and is the same as “vertex” in common ALEGRA terminology. The following steps were required:

1) Update coding which switches based on a multiple processor check to instead switch on a communication requirement. Periodic boundary conditions require communication even on a single processor.

2) Find the processor locations of matching nodes and create lists of nodes associated with each adjacent processor. A processor may adjacent to itself and thus is required to communicate with itself.

3) Use the identified processor lists to set up communications. Create temporary virtual node numbers for nodes “residing” on virtual processors in order to set up communication sets and ghost elements. These communications are not always off processor. Sometimes they are on a single processor.

4) Renumber the global node numbers so that they are labeled periodically. That is, nodes which map periodically to each other have exactly the same global node numbers. This allows for the correct creation of global sparse linear systems. The global node numbers are required for example by the AZTEC linear solver package [5].

5) Test the code for various configurations in both serial and parallel.

Further details of the required changes are described below.

3.1 Updating ALEGRA to allow for inter-processor communications.

The ALEGRA code had several locations which implement single processor optimizations. Thus code which used to look like

```c
if(number_of_processors() > 1) {} 
```

was replaced in some instances by

```c
if (Boundary_Communication_Is_Required ()) {...} 
```

where the function returns true if more than one processor is in use or if there are periodic boundary conditions.

3.2 Identifying matching nodes sets

The ALEGRA implementation assumes that the user has provided a mesh that contains matching node sets in a periodic sense. After mesh decomposition, a parallel node matching algorithm must be executed in order to create periodic boundary node sets. The pro-
The Periodic Boundary Condition Implementation

Processor boundary node sets are provided by the mesh decomposition software in the parallel case for normal processor boundaries but not for the periodic boundaries. In essence the mesh must wrap across the periodic interface. Two nodes match if using the standard Euclidean norm

\[ |x_2 - (x_1 + u)|^2 < \varepsilon^2 l^2 \]  

for translational periodicity, or

\[ |x_2 - (p + R(x_1 - p))|^2 < \varepsilon^2 l^2 \]  

for rotational periodicity. The value \( \varepsilon \) is a user specified tolerance (default \( 10^{-5} \)) and \( l \) is a dimension given by the diagonal of the cartesian box containing the two node sets.

The parallel node matching algorithm is developed below and the results are described relative to Figure 4. The actual coding does not actually send or receive a message from a processor to itself but the affect is the same and the algorithm is described as if the message was actually sent.

1) Find those processors that have nodes in node set 1 (NS1) and node set 2 (NS2). This can be done by a global operation. The number of nodes in NS1 and NS2 on each processor is then known by all the processors. In the example, all processors know that there are 3 nodes in NS1 processor 1, 3 nodes in NS1 on processor 2, 3 nodes in NS2 on processor 1 and 3 nodes in NS2 on processor 3.

2) Post an asynchronous send of the NS2 information including global number and local location to all processors containing NS1 nodes. In the example, processor 1 sends infor-

![Figure 4 Sample decomposition for description of parallel nodeset matching.](image-url)
mation on nodes 6, 7, 8 in NS2 to processors 1 and 2 while processor 3 sends information on nodes 8, 9, 10 to processor 1 and processor 2.

3) All processors containing NS1 nodes request the remote NS2 list from all NS2 processors using a blocking receive. After each receive then loop through the local NS1 to find a matching node in the remote NS2. Quit search of remote NS2 if a single matching node is found. Place the matching node on the NS1 matching node list and sort by the NS1 global node number. There may be nodes which are matched on more than one processor but each node is only matched once on each processor. In the example, processor 1 receives the NS2 list from itself and finds the matches in the order (1, 6), (2, 7), (3, 8). It also receives the NS2 list from processor 3 but matches only (3, 8). Processor 2 receives the NS2 list from processor 1 and finds the match (3, 8). It also receives the NS2 list from processor 3 and creates a match list in the order (3, 8), (4, 9), (5, 10).

4) Check on completion of asynchronous sends of the NS2 array. In the example, processors 1 and 3 check that their NS2 messages described in 2 above have been received.

Now repeat steps 2 through 4 swapping NS1 and NS2 but sorting always on NS1 in step 3. The details of the matching in the example case are left to the reader. Each processor will now have NS1 and NS2 matching node lists that are sorted correctly. Both the local and global node numbers are communicated in the ALEGRA coding.

3.3 Internal implementation of periodic boundary conditions

Ghost elements, faces, edges and nodes must be created in the periodic ghost elements and communications must be set up with the virtual periodic processors. To achieve this sleight of hand, a periodic mesh interprocessor communication object (PMeshIPC) is derived from the ALEGRA MeshIPC class. The MeshIPC class is responsible for nearest neighbor communications in ALEGRA. The PMeshIPC class thus handles the details of communications and variable updates for the special case of periodic communication. The processor set class is updated to allocate and store the PMeshIPC objects. The required methods and algorithms of the MeshIPC class are rewritten to handle PMeshIPC communications and updates (i.e. translations and rotations) as necessary. The on-processor updates are handled either through using the MeshIPC buffering scheme or by a direct copy of variables from one location to another. “Swap and add functions” must first swap, then correct the information for periodicity conditions and then add.

The current ghost element creation code in ALEGRA assumes that all global node numbers are unique. This causes complications because for this algorithm to work the periodic ghost nodes must be given a unique identifier rather than the their periodic image numbers. Currently, this temporary numbering is accomplished by boxing a computational space which encloses the whole periodic mesh including all associated ghost elements attached to the mesh and then mapping an integer to each ghost node. The algorithm assumes 32 bit integers. In two dimensions 15 bits of this integer are reserved for the x coordinate and 15 bits to the y coordinate. In three dimensions there are 10 bits available for each coordinate. The packed integer is then set to a negative value to identify the node as a periodic ghost node. This works in most cases but will fail for meshes with extreme variations in edge lengths or for extremely large meshes (~ 1 billion nodes). In addition,
The Periodic Boundary Condition Implementation

the algorithm for calculating ghost node numbers needs the matching node number from
the matching node set in order for the unique node algorithm to work correctly. The com-
munication set lists must also be sorted. For this sorting to be done correctly the nodes ly-
ing on a processor boundary are sorted according to the global number of only one of the
matching node sets. The ghost nodes are sorted according to their true periodic node num-
bers.

3.4 Message Tags

Each side of the matched periodic boundary has a unique message tag for parallel commu-
nications. The standard message tag for boundary updates is incremented by a unique
identifier for sending and receiving for each periodic boundary pair in order that the mes-
sages arriving from the same processor but from opposite sides of the periodic boundary
might be distinguished.

3.5 Scratch Maps

Eulerian updating of variables uses a scratch storage. This requires that the scratch storage
be mapped according to type so that appropriate transformations can be made. Symmetric
tensors always transform to symmetric tensors and anti-symmetric tensors always trans-
form to anti-symmetric tensors.

3.6 Implicit Algorithms

Since nodes are matched on periodic boundaries the nodal degrees of freedom are reduct-
ant. If the redundant nodes are kept as separate unknowns this will result in the formation
of a non-symmetric matrix. This is highly undesirable for iterative solver efficiency and so
it is important that the redundant degrees of freedom are excluded from the matrix formu-
lation process. This is accomplished by first properly tagging boundary nodes as owned or
not owned in the periodic sense and then communicating ownership information so that
the other communication sets can determine where to obtain correct information for sub-
sequent updates. Redundant entities must then be labeled with the corresponding periodic
global identifier so that the proper periodic matrix may be created. Figure 5 illustrates a
two-dimensional periodic mesh showing ghost elements and redundant nodes and edges.
In the next two sections the special formulations for modeling electric fields with a scalar
potential and the magnetic field with a vector or scalar potential are described. These spe-
cial formulations are required in order to maintain a nonzero mean field capability.

3.6.1 Electric Field Modeling with ALEGRA-EMMA

Electromechanical modeling of quasistatic electric fields in ALEGRA-EMMA (EMMA)
assumes that the electric field is the gradient of a scalar potential. However, a periodic po-
tential cannot represent a nonzero mean electric field. Since we wish to allow nonzero
mean fields, a slight reformulation of the standard EMMA configuration is implemented
to allow for an arbitrary mean external field. This is accomplished by representing the
electric field as the gradient of a potential $\phi$ such that
The Periodic Boundary Condition Implementation

Figure 5 Periodic Extension. Center region has 8 neighboring images. Ghost element numbers are labeled. Circled nodes and edges (faces in 3D) are redundant and are not included in linear system solve.

\[
\phi = \sum V_c \phi_c + \phi_p - E_{ext} \cdot x,
\]  

\[(10)\]

so that

\[
E = - \sum_c V_c \nabla \phi_c - \nabla \phi_p + E_{ext}
\]

\[(11)\]

where \(E_{ext}\) is an externally applied electric field and the summation is over the homogeneous solutions \(\phi_c\) associated with each constant potential conducting surface and \(\phi_p\) is a particular solution [2]. The zero free charge assumption implies that the electric displacement, \(D\), satisfies

\[
\nabla \cdot D = \nabla \cdot (\varepsilon E + p) = 0
\]

\[(12)\]
The Periodic Boundary Condition Implementation

where $p$ is the spontaneous or remanent or mechanical polarization and $\varepsilon$ is the permittivity tensor. This leads to the set of problems:

$$\nabla \cdot (\varepsilon \nabla \phi_c) = 0 \text{ with } \phi_c = \delta_{c, j} \text{ on conductor } j$$  \hspace{1cm} (13)

$$\nabla \cdot (\varepsilon \nabla \phi_p) = \nabla \cdot (p + \varepsilon E_{ext}) \text{ with } \phi_p = E_{ext} \cdot x \text{ on conductor } j.$$  \hspace{1cm} (14)

Zero electric displacement boundary conditions are imposed as natural boundary conditions on any free surface. Periodic conditions are imposed on periodic boundaries. The external field can be considered to appear in the equations as an effective polarization. $E_{ext}$ can be given arbitrarily, and, if all boundaries are periodic, the mean field will be exactly $E_{ext}$. Moreover, $E_{ext}$ will only affect the solution if periodic boundary conditions are enabled. Additional standard EMMA boundary conditions for a given problem will still work and thus impose conditions on the electric field and may have the effect of nullifying part of the imposed field.

If there are only periodic or natural (Neumann) boundary conditions, then the elliptic boundary value problem is not well posed. In this case ALEGRA specifies the finite element node with the minimum global number to a fixed zero potential. This does not affect the gradients of the solution. In this case it also required that the whole mesh be connected, i.e. fully contiguous, otherwise the solver will fail because the matrix associated with any disconnected region not containing the single fixed potential point will be singular.

3.6.2 Magnetic Field Modeling with ALEGRA

Where a potential is used for transient magnetics modeling it is not possible to represent a non-zero mean magnetic field with a potential alone. As in the electric field case, a solution decomposition allows the presence of a non-zero mean field. We represent the magnetic induction $B$ as

$$B = \nabla \times A + B_0$$  \hspace{1cm} (15)

where $A$ is the vector potential and $B_0$ is a constant vector independent of time. It represents that part of the initial conditions which cannot be represented by a potential and is invariant in time. This invariance can be shown by integration of the magnetic induction equation. For a 3D periodic box all vector potential equation becomes

$$\nabla \times (\nabla \times (A + B_0)) = J = \sigma \left( -\nabla \phi - \frac{2A}{\sigma} + u \times (\nabla \times A + B_0) \right)$$  \hspace{1cm} (16)

Given the gauge

$$\phi = u \cdot A + \frac{\nabla \cdot A}{\sigma}$$  \hspace{1cm} (17)
The Periodic Boundary Condition Implementation

the corresponding weak form of Equation (16) is

\[ \int (\nabla \times N) \cdot \mathbf{v}(\nabla \times A + B_0) d\Omega + \int \nabla \cdot (\sigma N) \cdot \frac{\nabla A}{\sigma} d\Omega \]  
\[ + \int \sigma N \cdot \frac{\partial A}{\partial t} d\Omega + \int \sigma N \cdot (A \cdot (\nabla u)^T) d\Omega \]  
\[ = \int \mathbf{N} \times \mathbf{H}_b \cdot n d\Gamma + \int \nabla (\nabla \times A) \mathbf{N} \cdot n d\Gamma + \int \sigma N \cdot (\mathbf{u} \times B_0) d\Omega \]  
\[ \text{(18)} \]

The \( B_0 \) terms leads to two additional right-hand-side contributions to the linear system. Given \( A \) the calculation of \( \mathbf{J} \) is based on

\[ \int (\nabla \times N) \cdot \mathbf{v}(\nabla \times A + B_0) d\Omega \]  
\[ = \int \mathbf{N} \times \mathbf{H}_b \cdot n d\Gamma + \int \mathbf{N} \cdot \mathbf{J} d\Omega \]  
\[ \text{(21)} \]

A similar set of equations follows in 2D x-y plane geometry for the vector potential component perpendicular to the plane. The scalar equation to be solved in this case is in weak form

\[ \int \sigma N \cdot \frac{\partial A_z}{\partial t} d\Omega + \int \nabla N \cdot \mathbf{v}(\nabla A_z + B^\perp_0) d\Omega = \int \mathbf{N} H^\perp_b \cdot n d\Gamma - \int \sigma (\mathbf{u} \cdot B^\perp_0) N d\Omega \]  
\[ \text{(23)} \]

where \( B^\perp = (\mathbf{B_y}, \mathbf{B_x}) \). Similarly the calculation of the current density utilizes the equation

\[ \int \nabla N \cdot \mathbf{v}(\nabla A_z + B^\perp_0) d\Omega = \int \mathbf{N} H^\perp_b \cdot n d\Gamma + \int \mathbf{J} N d\Omega \]  
\[ \text{(24)} \]

The key point here again is that it is possible to use a periodic potential for a general periodic problem and the symmetry of the solution matrices can be maintained provided we augment the equations slightly to represent the mean field explicitly.
4. Example Calculations

The following example problems illustrate the use of periodic boundary conditions in ALEGRA. These problems run in both serial and parallel and test various features of the periodic boundary condition capability. The test problems shown are found in the ALEGRA 2D and 3D verification suites and are referenced appropriately in the section headers.

4.1 Exploding wire in a wedge. (Verification/2D/periodic/rotper)

This problem tests the periodic boundary conditions for a rotationally periodic configuration. The initial setup is an offset overdense (pressurized) wire in a wedge is illustrated in Figure 6. The interior and exterior edges expand at the same time as the wire explodes. A reflected shock is seen due to the image wire expansion.

4.2 Exploding wires (Verification/2D/periodic (lag, eul, advect))

This set of three test problems are all point source explosions in a periodic mesh as shown in Figure 7. The calculations are shown a five times. A Lagrangian calculation (lag) shows the mesh distortion. An Eulerian calculation (eul) of the same setup gives the results from an Eulerian approach to solving the same problem. A final Eulerian calculation (advect) imposes a constant velocity on the whole system and is timed to give the same configuration at the final time in periodic space.

Figure 6 Offset pressurized wire in a 2D periodic wedge.
Figure 7  Point sources in a periodic box (left:lag, center:eul; right:advect)
Example Calculations

4.3 Anisotropic shock propagation (Verification/3D/periodic/hexqtz)

Following Johnson [9], we simulated the propagation of elastic waves that would be produced in a linearly elastic, anisotropic medium by parallel plate impact. Ignoring edge effects, this loading generally produces one-dimensional plane waves of mixed longitudinal and transverse character in an elastically anisotropic medium. The wave that produces predominantly longitudinal particle motion is called the quasi-longitudinal wave; the ones producing predominantly transverse particle motion are called quasi-transverse waves. The particle displacement directions in the three distinct waves that can be generated in an anisotropic elastic material form an orthogonal triad, just as for the longitudinal and transverse waves familiar in isotropic elastic bodies.

The chosen wave propagation simulation provides a test of the periodic boundary conditions because only they are compatible with the resulting particle motion in a 3-D simulation. This is because, unlike an isotropic body, the deformation produced by plane wave propagation is an elastically anisotropic body is not one of dynamic uniaxial strain. The axial and transverse motions are not simple to describe and will generally not be known in advance.

The simulation tests the transverse isotropic elasticity portion of the ferroelectric-antiferroelectric material model for lead zirconate titanate which has been implemented in ALEGRA [7][8]. Transverse isotropy is the same as hexagonal symmetry to second order in the elastic response. The effects of anisotropy were less than 1% in Johnson’s examples of wave propagation in the hexagonal materials Ti and Be [9]. To amplify the effect of hexagonal anisotropy we fabricated a new model material using the density and elastic constant values for quartz but omitting the quartz \( C_{14} \) \( C_{1123} \) elastic constant. The resulting hexagonal material supports the same two-wave structure as do Ti and Be single crystals but with a greater visual impact: the difference in the longitudinal stress amplitudes in the two waves reaches a maximum of 5.3% for propagation along a direction that is 73° away from the hexagonal “c-axis.”

Applying Johnson’s analysis to this “hexagonal quartz” material yielded the values listed in Table 3 under “Theoretical” for the waves that result from propagation at 73° from the c-axis. In this table, \( \sigma_{zz1} \) and \( \sigma_{zz0} \) are, respectively, the longitudinal stress behind the faster quasi-longitudinal wave (QL) (called “Region 1”) and behind the quasi-transverse wave (QT) (called “Region 2”). The material frame axes are denoted by a, b, and c, with the c-axis being the unique axis for hexagonal symmetry. Components in the laboratory frame are denoted by subscripts x, y, and z; wave propagation in the sample is along the laboratory z direction with the material b-axis normal to the plane containing the laboratory z-axis and the material c-axis.

The transverse isotropic elasticity model and the periodic boundary conditions were verified against the analytic results by simulating symmetric impact of two identical plates of infinite lateral extent: A mesh of 2x2x120 cubical hexahedral elements was created in a rod of material with dimensions 1x1x3 millimeters. The rod was given an initial velocity of -50 m/s in the z direction and one face of the rod was held with zero z-velocity. Period-
ic boundary conditions were specified in the lateral directions. As explained by Johnson [9], this impact boundary condition is equivalent to the stress boundary condition he used in his calculation. Table 2 gives computed values in each region.

<table>
<thead>
<tr>
<th>Value</th>
<th>Region 0</th>
<th>Region 1</th>
<th>Region 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$ kg/m$^3$</td>
<td>2672.8</td>
<td>2670.6</td>
<td>2650</td>
</tr>
<tr>
<td>$v_x$ km/s</td>
<td>-4.06</td>
<td>10.92</td>
<td>0</td>
</tr>
<tr>
<td>$v_z$ km/s</td>
<td>0.0</td>
<td>-3.56</td>
<td>-50</td>
</tr>
<tr>
<td>$\sigma_{zz}$ Pa</td>
<td>-7.84e8</td>
<td>-7.41e8</td>
<td>0</td>
</tr>
<tr>
<td>$\sigma_{xz}$ Pa</td>
<td>0.0</td>
<td>-1.75e8</td>
<td>0.0</td>
</tr>
<tr>
<td>$\sigma_{yz}$ Pa</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

TABLE 2: Plane Wave Propagation at 73° to c-axis in “Hexagonal Quartz”

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Theoretical</th>
<th>Simulated</th>
<th>Difference(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QL wave speed (km/s) measured at wave midpoints</td>
<td>6.013</td>
<td>6.042</td>
<td>1.0</td>
</tr>
<tr>
<td>QL wave speed (km/s) from $[\rho v_z]^2/[\rho]^2$</td>
<td>6.013</td>
<td>5.971</td>
<td>-0.1</td>
</tr>
<tr>
<td>QT wave speed (km/s) measured at wave midpoints.</td>
<td>4.379</td>
<td>4.37</td>
<td>0.0</td>
</tr>
<tr>
<td>QT wave speed (km/s) from $[\rho v_z]^2/[\rho]^2$</td>
<td>4.379</td>
<td>4.322</td>
<td>-1.3</td>
</tr>
<tr>
<td>QL Particle Velocity Amplitude Ratio $[v_z]^2/[v_x]^2$</td>
<td>4.23</td>
<td>4.25</td>
<td>0.5</td>
</tr>
<tr>
<td>QT Particle Velocity Amplitude Ratio $[v_x]^2/[v_z]^2$</td>
<td>4.23</td>
<td>4.21</td>
<td>-0.5</td>
</tr>
<tr>
<td>$1 - \sigma_{zz1}/\sigma_{zz0}$ (%)</td>
<td>5.293</td>
<td>5.485</td>
<td>3.6</td>
</tr>
<tr>
<td>$\sigma_{xz1}/\sigma_{zz0}$</td>
<td>0.224</td>
<td>0.223</td>
<td>0.4</td>
</tr>
</tbody>
</table>

TABLE 3: Plane Wave Propagation at 73° to c-axis in “Hexagonal Quartz”
Example Calculations

An analysis of the results is shown in Table 3. The wave speeds and amplitude ratios were calculated directly from arrival time differences at nodal points or from the shock jump condition indicated. Numerical broadening of the wave front was evident in the wave profiles. For this reason the speed of the foot of the wave systematically exceeded the speed of the midpoint amplitude of the wave. The amplitude ratios are calculated as described in the table. Figure 8 illustrates the hexqtz test problem.

Figure 8  Two wave structure and transverse motion in the horizontal periodic direction is seen for a wave whose structure depends only on the vertical coordinate. The wave is propagating downward and displacements have been exaggerated.
4.4 Periodic Electric Field (Verification/3D/qsem_physics/ballbox)

Consider a periodic array of spheres embedded in a uniform electric field and assume that the permittivity of the spheres differs from that of the surrounding medium. Using the periodic boundary conditions we can compute the electric field in the medium and the spheres. If each sphere is small with respect to the periodic mesh, then we expect the solution to be close to the classical solution of a sphere in a uniform electric field. For this case the field inside the sphere is then a constant given by

\[ E_{int} = \left( \frac{3}{\varepsilon_r + 2} \right) E_{ext} \]  

(25)

where \( \varepsilon_r \) is the relative permittivity, \( E_{int} \) is the electric field in the sphere and \( E_{ext} \) is the externally imposed electric field.

If \( \varepsilon_r \) is 2 and each component of the electric field is of unit magnitude then the field in a small sphere in a periodic box will be 0.75 of the external field. This is seen in Figure 9.

![Figure 9](image)

Figure 9 Plots of \( \phi \) (solid), \( 10\phi_P \) (dotted) and electric field component (dashed) for dielectric sphere in a periodic medium.

where the total potential, \( \phi \), the periodic particular potential, \( \phi_P \), and the electric field components are shown on a diagonal of a unit cube containing a sphere of diameter 0.25.
Example Calculations

4.5 Periodic Magnetohydrodynamics (Verification/2D/mhd_physics/cnvct45)

This 2D test problem illustrates the effect of a magnetic field on a periodic initial configuration consisting of a 2 by 2 array of rotating vortices. The initial divergence-free velocity field is given by

\[
v = 2(-\sin(\pi x)\cos(\pi y), \cos(\pi x)\sin(\pi y))
\]

and the initial uniform magnetic field is directed along the 45 degree line with magnitude \(2\sqrt{2} \times 10^{-3}\). The material is a \(\gamma = \frac{5}{3}\) ideal gas with unit density and initial internal energy of 0.9. This particular problem is about equally balanced between kinetic and magnetic energies and dissipation plays a fundamental role. The magnetic Reynolds number

\[
R_M = \mu_0\sigma UL = (4\pi 10^{-7})(10^6)(2)(2) = 5.03
\]

is indicative of this. The ratio of kinetic to magnetic energies densities is given by

\[
\beta_v = \frac{\rho v^2/2}{B^2/(2\mu_0)} = \frac{(1)(4)/2}{8 \times 10^{-6}/(8\pi 10^{-7})} = 0.63
\]

which indicates as well that the magnetic energy should play a fundamental role in the solution. The various power terms for the magnetic energy conservation equation are given in Figure 10. It is seen that the magnetic forces works against the initial kinetic energy in the problem since the work power term is negative. The magnetic power is positive at first as the magnetic energy is increasing due to fluid motion then becomes negative as this energy dissipates into heat. The slight dip in the power sum is an open issue relative to im-

![Figure 10 Power rates in cnvct45 test problem.](image_url)
proving the magnetohydrodynamic modeling. The energy histories are shown in Figure 11. Since the mean field $B_0$ is constant, the total magnetic energy must always be greater than the constant mean field energy since

$$\int B^2 dx = \int ((B - B_0) + B_0)^2 dx = \int (B - B_0)^2 dx + \int B_0^2 dx$$

(29)

This feature is seen in Figure 11. The flow undergoes an initial transient phase and then settles down into periodic strips of flow moving in the direction of the mean field lines as illustrated in Figure 12. Since there are no true boundaries, the Poynting powers and energies are zero.

Figure 11 Energy histories from cnvct45 test problem.

Figure 12 Temperature distribution at early time and late times.
5. Unsupported Features and Unresolved Problems

Currently it is possible for a mesh with extreme variations in element size to reach the limits of the periodic ghost node numbering scheme and the ghost numbers calculated will not be unique. This is because the ghosts node numbers are arrived at by a Cartesian space subdivision with limited resolution due to the 32 bit integer limit. For most meshes this will not be a problem.

ALEGRA still needs to be upgraded to support periodic boundary conditions with face and edge centered variables. Required actions include debugging the setup code so that faces and edges will be created properly for all meshes with periodic boundary conditions as well as calculating face and edges global numbers correctly for proper usage of AZTEC with periodic boundary conditions. In particular, the radiation transport and conduction capabilities do not run with periodic boundary conditions. Although most meshes with periodic boundary conditions pass through the face and edge creation code without mishap, there is at least one mesh which currently results in an exception.

No ALE (Arbitrary Lagrangian-Eulerian) problems have been attempted with periodic boundary conditions.

Initial refinement does not work with periodic boundary conditions.

An incompatibility between pressure boundary conditions and periodic boundary conditions has been reported.

There appears to a problem when running periodic boundary conditions in parallel in Eulerian mode. The ALEGRA Verification/3D/periodic/eul problem does not give the same answer in parallel to the precision believed to be possible. Visually the answers appear to be correct.

There have been unresolved reported problems with MHD Rayleigh-Taylor growth being erroneously seeded at periodic boundaries. Subsequently, problems with respect to the use of correctly updated boundary information in parallel in the current MHD coding have been discovered. Generally, periodic boundary condition errors can be either in 1) a general parallel algorithm or 2) the particular periodic boundary condition coding. Thus the MHD RT bug may be corrected now but must be reinvestigated.

The user must not attempt to inconsistently apply periodic boundary conditions and other boundary conditions on the same surfaces. There are currently no checks in the code to try to catch these types of user errors.
6. Conclusions

The current implementation of periodic boundary conditions in ALEGRA is a non-inva-
sive approach. The new capability in ALEGRA provides for improved opportunities to
study anisotropic material models and magnetohydrodynamic test problems. Periodic
boundary conditions are particularly useful for setting up verification problems in which
no mass, momentum or energy can enter or leave the problem. The basic approach taken
may be applied to any finite element code which has been written for distributed memory
parallel machines. The upgrades required are limited to the message passing support cod-
ing and other modifications to support the representation of mean component of periodic
fields given by a gradient or curl of a periodic potential.
Conclusions
References


[7] R. M. Brannon, Personal Communication, September 1999, Unpublished joint work with Steve Montgomery. The implementation of the transversely isotropic model as discussed in the text is known as the Fe Afe Ceramic model in ALEGRA and is also known via the FORTRAN subroutine name EMECH5.


APPENDIX A

We document here the details of the calculation of the anisotropic material wave speed calculations. Relative to the material axes the wave propagation direction is \( \mathbf{a} = (0, \sin \theta, \cos \theta) \) with \( \theta = 73^\circ \). The material properties are (See Johnson [9]):

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho_0 )</td>
<td>2650 kg/m(^3)</td>
</tr>
<tr>
<td>( C_{11} )</td>
<td>86.8 GPa</td>
</tr>
<tr>
<td>( C_{12} )</td>
<td>7.0 GPa</td>
</tr>
<tr>
<td>( C_{13} )</td>
<td>11.9 GPa</td>
</tr>
<tr>
<td>( C_{33} )</td>
<td>105.8 GPa</td>
</tr>
<tr>
<td>( C_{44} = C_{2323} )</td>
<td>73 GPa</td>
</tr>
</tbody>
</table>

Table 4: “Quartz” input parameters

The wave propagation is in the material \( x_2 - x_3 \) plane and in the laboratory \( x'_1 - x'_3 \) plane. For hexagonal materials the relevant eigenvectors and eigenvalues are

\[
U^{(2)} = (0, \chi_-, 1); \quad \lambda^{(2)} = \rho V_-^2
\]

\[
U^{(3)} = (0, \chi_+, 1); \quad \lambda^{(3)} = \rho V_+^2
\]

where the eigenvectors are given relative to the material axes. Johnson gives

\[
2\rho V_\pm = C_{11}(\sin \theta)^2 + C_{33}(\cos \theta)^2 + C_{44} \pm S
\]

\[
S = (C_{11}(\sin \theta)^2 - C_{33}(\cos \theta)^2 + C_{44}\cos(2\theta))^2 + (C_{13} + C_{44})^2(\sin(2\theta))^2
\]

\[
\chi_\pm = \frac{C_{11}(\sin \theta)^2 - C_{33}(\cos \theta)^2 + C_{44}\cos(2\theta) \pm S}{(C_{13} + C_{44})\sin(2\theta)}
\]

The predicted wave speeds are the values of \( V_- \) (quasi-transverse (QT) wave) and \( V_+ \) (quasi-longitudinal (QL) wave) calculated from Equation (32). The ratio of particle velocity amplitudes is obtained from the eigenvectors expressed in the laboratory coordinates (primed). These are
\[ U^{(2)} = (\chi_{\gamma} \sin \theta + \cos \theta, 0, \chi_{\gamma} \cos \theta - \sin \theta) \quad (35) \]

\[ U^{(3)} = (\chi_{\gamma} \sin \theta + \cos \theta, 0, \chi_{\gamma} \cos \theta - \sin \theta) \quad (36) \]

The particle velocity amplitude ratio is for the QL wave

\[ \left[ v'_{1} \right]^{2} / \left[ v'_{3} \right]^{2} = \frac{\chi_{\gamma} \sin (\theta + \gamma) + \cos (\theta + \gamma)}{\chi_{\gamma} \cos (\theta + \gamma) - \sin (\theta + \gamma)} \quad (37) \]

or for the QT wave

\[ \left[ v'_{3} \right]^{1} / \left[ v'_{1} \right]^{1} = \frac{\chi_{\gamma} \cos (\theta + \gamma) - \sin (\theta + \gamma)}{\chi_{\gamma} \sin (\theta + \gamma) + \cos (\theta + \gamma)} \quad (38) \]

Using Johnson's notation the stress is

\[ \sigma'_{11} = \epsilon'_{11} C'_{11} + 2 \epsilon'_{13} C'_{15} \quad (39) \]

\[ \sigma'_{12} = 0 \quad (40) \]

\[ \sigma'_{13} = \epsilon'_{11} C'_{15} + 2 \epsilon'_{13} C'_{55} \quad (41) \]

Taking the derivative of the displacement with respect to \( x'_{1} \) yields the strains:

\[ u'_{1} = \frac{\sigma_{0}}{D} \left[ B_{22} V_{1} U_{1}^{(2)} \left( t - \frac{x'_{1}}{V_{1}} \right) H \left( t - \frac{x'_{1}}{V_{1}} \right) - B_{21} V_{1} U_{1}^{(3)} \left( t - \frac{x'_{1}}{V_{1}} \right) H \left( t - \frac{x'_{1}}{V_{1}} \right) \right] \quad (42) \]

\[ u'_{3} = \frac{\sigma_{0}}{D} \left[ B_{22} V_{3} U_{3}^{(2)} \left( t - \frac{x'_{1}}{V_{3}} \right) H \left( t - \frac{x'_{1}}{V_{3}} \right) - B_{21} V_{3} U_{3}^{(3)} \left( t - \frac{x'_{1}}{V_{3}} \right) H \left( t - \frac{x'_{1}}{V_{3}} \right) \right] \quad (43) \]

\[ \epsilon'_{11} = \frac{\sigma_{0}}{D} \left[ -B_{22} (\chi_{\gamma} \sin \theta + \cos \theta) H \left( t - \frac{x'_{1}}{V_{1}} \right) + B_{21} (\chi_{\gamma} \sin \theta + \cos \theta) H \left( t - \frac{x'_{1}}{V_{1}} \right) \right] \quad (44) \]

\[ 2 \epsilon'_{13} = \frac{\sigma_{0}}{D} \left[ -B_{22} (\chi_{\gamma} \cos \theta - \sin \theta) H \left( t - \frac{x'_{1}}{V_{3}} \right) + B_{21} (\chi_{\gamma} \cos \theta - \sin \theta) H \left( t - \frac{x'_{1}}{V_{3}} \right) \right] \quad (45) \]

Substituting Equation (44) and Equation (45) into Equation (39) and Equation (41) yields
\[ \sigma'_{11} = \frac{\sigma_0}{D} \left[ B_{11} B_{22} H \left( t - \frac{x'_1}{V_+} \right) - B_{12} B_{21} H \left( t - \frac{x'_1}{V_+} \right) \right] \]  
(46)

\[ \sigma'_{13} = \frac{-\sigma_0}{D} B_{21} B_{22} \left[ H \left( t - \frac{x'_1}{V_+} \right) - H \left( t - \frac{x'_1}{V_-} \right) \right] \]  
(47)

The ratio of longitudinal stress amplitudes behind the QL and the QT waves is thus,

\[ 1 - \frac{\sigma'_{11}}{\sigma_0} = 1 + \frac{B_{12} B_{21}}{D}, \text{ for } \frac{x'_1}{V_+} < t < \frac{x'_1}{V_-} \]  
(48)

Similarly, the shear stress behind the QL wave normalized by the input stress is

\[ \frac{\sigma'_{13}}{\sigma_0} = \frac{-B_{22} B_{21}}{D}, \text{ for } \frac{x'_1}{V_+} < t < \frac{x'_1}{V_-}. \]  
(49)

These are the theoretical values given in Table 3 as \( \frac{\sigma_{zz1}}{\sigma_{zz0}} \) and \( \frac{\sigma_{xz1}}{\sigma_{zz0}} \), respectively.
Distribution

External:

Mark A. Christon
Livermore Software Technology Corp.
7209 Aztec Rd., NE
Albuquerque, NM 87110

Andrea Malagoli
The University of Chicago
Enrico Fermi Institute
933 E. 56th St.
LASR 132b
Chicago, IL 60637

Daniel I. Meiron
California Inst. of Technology
Mail Code 217-50
Pasadena, CA 91125

Bryan Oliver
Mission Research Corporation
1720 Randolph Rd.
Albuquerque, NM 87106

P. W. Rambo, MS L-015
University of California
Lawrence Livermore National Laboratory
7000 East Ave.
P.O. Box 808
Livermore, CA 94550

Sandia Internal:

MS 0521 S. T. Montgomery, 2567
MS 1202 J. M. McGlaun, 5903
MS 0841 P. Hommert, 9100
MS 0834 M. R. Baer, 9100
MS 0826 D. K. Gartling, 9100
MS 0828 R. K. Thomas, 9104
MS 0865 J. Moya, 9105
MS 0865 J. Garcia, 9106
MS 0826 W. Hermina, 9111
MS 0834 A. Ratzel, 9112
MS 0835 S. N. Kempka, 9113
MS 0825 W. H. Rutledge, 9115
MS 0825 W. L. Oberkampf, 9115
MS 0836 C. Peterson, 9116
MS 0847 H. Morgan, 9123
MS 0847 J. B. Aidun, 9123
MS 0847 M. Blanford, 9121
MS 0847 M. Heinstein, 9121
MS 0847 K. Brown, 9117
MS 0321 W. J. Camp, 9200
MS 0318 G. S. Davidson, 9201
MS 0316 P. F. Chavez, 9204
MS 0318 P. Heermann, 9215
MS 1111 S. S. Dosanjh, 9221
MS 1111 S. Plimpton, 9221
MS 1111 A. Salinger, 9221
MS 1111 J. N. Shadid, 9221
MS 1111 R. Schmidt, 9221
MS 1111 T. M. Smith, 9221
MS 1110 D. Wamble, 9222
MS 1110 R. Lehoucq, 9222
MS 1110 L. A. Romero, 9222
MS 0321 A. L. Hale, 9224
MS 1111 G. Heffelfinger, 9225
MS 1111 H. Hjalmarson, 9225
MS 1111 M. D. Rintoul, 9225
MS 0847 R. Leland, 9226
MS 1111 B. Hendrickson, 9226
MS 0835 J. S. Peery, 9121
MS 0819 E. Boucheron, 9231
MS 0819 K. G. Budge, 9231
MS 0819 D. Carroll, 9231
MS 0819 R. Drake, 9231
MS 0819 C. J. Garasi, 9231
MS 0819 E. S. Hertel, 9231
MS 0819 S. V. Petney, 9231
MS 0819 A. C. Robinson, 9231 (15)
MS 0819 R. Summers, 9231
MS 0819 T. G. Trucano, 9231
MS 0819 R. Weatherby, 9231
MS 0819 M. Wong, 9231
MS 0820 P. Yarrington, 9232
MS 0820 M. Boslough, 9232
MS 0820 R. Brannon, 9232
MS 0820 D. Crawford, 9232
MS 0820 E. Fang, 9232
MS 0820 A. Farnsworth, 9232
MS 0820 M. E. Kipp, 9232
MS 0820 S. A. Silling, 9232
MS 0820 P. A. Taylor, 9232
MS 0847 D. Martinez, 9234
MS 1187 K. Matzen, 1670
MS 1194 R. B. Spielman, 1644
MS 1186 C. Deeney, 1644
MS 1186 M. R. Douglas, 1644
MS 1186 T. Mehlhorn, 1674
MS 1186 T. Haill, 1674
MS 1186 J. S. Lash, 1674
MS 1186 R. J. Lawrence, 1674
MS 1186 B. M. Marder, 1674
MS 1186 S. A. Slutz, 1674
MS 1186 P. H. Stoltz, 1674

39
| MS 1186  | R. E. Vesey, 1674 |
| MS 1010  | D. Hensinger, 15222 |
| MS 9011  | T. H. Jefferson Jr., 8920 |
| MS 0899  | Technical Library, 4916 (2) |
| MS 0612  | Review and Approval Desk, 04912 (1) for DOE/OSTI |
| MS 9019  | Central Technical Files, 8940-2 |