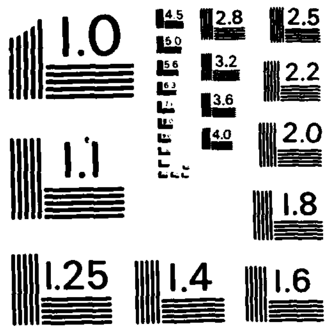
AD-A169 660 IMPOSED-SOLUTION BOUNDARIES FOR THREE-DIMENSIONAL HULL 1/2
(U) ARMY BALLISTIC RESEARCH LAB ABERDEEN PROVING GROUND
MD J D WORTMAN MAR 86 BRL-MR-3499

UNCLASSIFIED F/G 19/4 NL



1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

AD-A169 660

(12)

AD

US ARMY MATERIEL COMMAND

MEMORANDUM REPORT BRL-MR-3499

IMPOSED-SOLUTION BOUNDARIES FOR THREE-DIMENSIONAL HULL

John D. Wortman

March 1986

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER Memorandum Report BRL-MR-3499	2. GOVT ACCESSION NO. AD-A169660	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Imposed-Solution Boundaries for Three-Dimensional Hull		5. TYPE OF REPORT & PERIOD COVERED Final, April 84 - Jan 85
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John Wortman		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: SLCBR-TB Aberdeen Proving Ground, MD 21005-5066		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS RDTE 1L162618AH80
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: SLCBR-DD-T Aberdeen Proving Ground, MD 21005-5066		12. REPORT DATE March 1986
		13. NUMBER OF PAGES 120
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Hydrocode
HULL
Boundary Conditions

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a new imposed-solution transmissive boundary capability that has been made operational in the BRL version of the airblast HULL hydrodynamics code, which can be used for 3-D blast loading computations on rigid structures. Until this time, 3-D blast loading computations have been done by either mapping a 2-D blast wave into the 3-D grid and assigning simple transmissive boundaries, or by imposing a planar constant value (non-decaying) wave, or by imposing a decaying wave as defined by the LAMB coding. The

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) boundary values for these input waves could only be imposed at two or three of the boundaries. The new imposed-solution boundaries for 3-D Cartesian coordinates allow the time-dependent definition of flow field conditions to be imposed at all six boundary planes. (The flow field conditions for these planes may be obtained from a previously-run 2-D cylindrical HULL computation through the use of a new program HULLUP). This new capability permits better definition of free-field flow conditions at boundaries, and thus extends the time that a given grid will produce results on a target that are uncontaminated by artificial waves from improperly defined boundaries. Listings of the changes to HULL, the HULLUP program, and sample runstreams for HULLUP, KEEL, and HULL are included in the Appendices.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

Estimating the blast loading on various targets is an ongoing effort at the US Army Ballistic Research Laboratory (BRL). Such loading may be estimated through tests in shock tubes, through high-explosives (HE) testing in the field, or through computation with hydrocodes. One such hydrocode, which has been used with success at BRL and elsewhere, is an airblast version of the HULL[1,2] hydrodynamic computer code. This code solves the inviscid Euler equations in two dimensions (2-D) or three dimensions (3-D) using an explicit time-stepping method. The 2-D version has a number of options that are not available in 3-D, but the main disadvantages of the 3-D version are the much longer run time and the memory requirement. The mesh spacing for 3-D runs is too frequently set much coarser than desirable in order to fit the storage for the required computational field into the computer or to complete the run in a reasonable time.

For some studies (simulation of an HE field test, for example) it would be advantageous to use the 2-D cylindrical symmetry version of HULL from initiation to near impingement of the blast wave on a target position, then continue the calculation with the 3-D version to study loading on a 3-D target (2-D cylindrical HULL runs can be preceded by, and initiated from, one-dimensional runs of the SAP program). The 2-D cylindrical HULL could be used as long as the problem was cylindrically symmetric. It would be much faster than the 3-D version, even with a finer mesh, and could incorporate some useful options that are not present in 3-D such as dust or multiple materials. However, the 2-D HULL would not be appropriate for examining the loading on a 3-D target such as a building or a vehicle. The hydrodynamic data in a portion of the 2-D cylindrical HULL computational field (actually three-dimensional) could be transferred to a smaller 3-D HULL computational field containing the target, and the computation continued with the 3-D HULL. (Some optional properties, multiple materials for example, do not exist in 3-D and cannot be transferred.)

Such a procedure was at least partially available in our HULL. The grid generation program, called KEEL, will transfer results from a 2-D cylindrical problem to a new 2-D or 3-D problem. The user would have had to assign some existing, but unrealistic, boundary conditions. The results from the new run would be meaningful until the effect of whatever boundary conditions were imposed reached the target area. The most likely boundary conditions would be transmissive or reflective. If realistic boundary conditions from the 2-D run could be imposed, the results would be useful until the reflected waves went from the target to a nonreflective boundary and back to the target. Using such boundary conditions would approximately double the error free time, from program initiation to the arrival of false signals at the target, for a given 3-D computational grid. To double the uncontaminated run time using the wrong boundaries, the distance from the target sides to the edges of the computational region would have to be doubled; for a relatively small 3-D target, the computational region would have to be about 8 times as large.

This report documents the coding for this new type of boundary for HULL, and presents the results of some limited tests. Section II summarizes the older HULL boundaries; the new boundary type is discussed in Section III. The program to produce boundary input for a 3-D HULL run from the restart output from a 2-D cylindrical HULL run is discussed in Section IV. The results from a series of test runs is discussed in Section V. Appendix A contains a tabulation of a sample runstream and the HULLUP program which will produce boundary input for a 3-D run. Appendix B includes listings of a runstream to use KEEL for a 3-D restart, a runstream for the corresponding HULL run, the general UPDATE corrections to HULL, and the HULL changes for the new boundary conditions.

II. HULL COMPUTATIONAL GRID BOUNDARY CONDITIONS

In order to understand the computational grid boundaries used in HULL, one needs an overview of the HULL method for solving the governing equations. The basic equations used in HULL to describe inviscid fluid flow are:

$$\frac{d\rho}{dt} + \rho(\nabla\cdot\vec{u}) = 0 \tag{1}$$

$$\rho\frac{d\vec{u}}{dt} + \nabla p = -\rho\,\vec{g} \tag{2}$$

$$\rho\frac{dE}{dt} + \nabla\cdot(p\vec{u}) = -\rho\vec{u}\cdot\vec{g} \tag{3}$$

$$p = F(\rho, I) \tag{4}$$

$$E = I + \tfrac{1}{2}(\vec{u}\cdot\vec{u}) \tag{5}$$

$$\frac{dp}{dt} + p\gamma_{eff}(\nabla\cdot\vec{u}) = 0 \tag{6}$$

where,

$\nabla\cdot$ = the divergence operator

∇ = the gradient operator

ρ = material density (g/cm^3)

p = pressure ($dynes/cm^2$)

$\vec{u}$ = the vector fluid velocity (cm/sec)

I = specific internal energy (ergs/g)

E = specific total energy (ergs/g)

$\vec{g}$ = the vector acceleration of gravity (cm/sec^2)

t = time (sec)

$\gamma_{eff} = 1 + p/ I$ is the "effective gamma"

Equations (1), (2), and (3) are the conservation equations for mass, momentum, and energy, respectively. Equation (4) is the equation of state (for this study, a gamma law equation of state, $p = (\gamma - 1)\rho I$, was assumed with $\gamma = 1.4$). Equation (6) can be derived by substituting equations (5) and (3) into equation (2) and assuming $\gamma = \gamma_{eff}$ is constant.

The solution is found by two phases for each time step. The first phase is a two step Lagrangian solution in the manner of Lax-Wendroff. The second phase is a transport phase which is also carried through in two steps. (A description of one cycle of the HULL solution at a cell for a 2-D cylindrically symmetric case is contained in the SAIL comments at the beginning of the HULL program. In fact, there are two descriptions; the original HULL authors were in mild disagreement. These comments have equal signs in column 1 and are removed by the preprocessor POST. They are in the output of UPDATE or of program LIST which is part of the HULL file. A description of the phase 1 solution for 3-D HULL, with comments, is contained in Section 2 of Reference 3.)

At the beginning of phase 1 (at time t) the velocity components, the specific internal energy, and the mass are known for every interior cell of the computational field. Routines and data are available to compute other values such as volume, density, and pressure.

When the phase 1 solution is to be advanced for a particular cell, it is assumed that the pressure and normal velocity components are known for each low index side of the cell at time $t + \Delta t/2$. (It is convenient to think of these intermediate time values being on the cell boundaries. They are more closely Lagrange solutions for points that were between cell centers at time t.) From the values in the cell and in the adjacent higher index cells, the pressure and normal velocity components for time $t + \Delta t/2$ are computed at the high index sides of the cell. This completes step 1 of phase 1 for the cell. The second step of phase 1 uses these pressures and velocities "at the cell boundaries" to compute the Lagrange solution for specific energy and velocity at time $t + \Delta t$ for the old cell center point.

For most cells the values at the low index sides of the cell for time $t + \Delta t/2$ are retained from the previous calculations. If the cell is a low index boundary cell (I=1, J=1, or K=1) these boundary values must be supplied. The method of supplying these values differs with the condition, the boundary, the dimension, and the programmer. Four fairly distinct methods are used at low index boundaries:

1. A virtual external cell with the desired properties, at time t, is assumed and the governing equations solved for pressure and normal velocity at time $t + \Delta t/2$ on the boundary.

2. The pressure and normal velocity on the boundary at time $t + \Delta t/2$ are assigned. (Some inconsistency in time was discovered during this study. Changes to make them consistent were added to the correction file.)

3. Hydro values are assigned to the boundary cell and computation for that cell is skipped except for computing the velocity and pressure at $t + \Delta t/2$ on the high index side. This is used only with 2-D at the bottom boundary. The bottom layer of cells is considered to be inside the computation region, hence the mass and energy changes in the boundary row of cells is added to the system mass and energy.

4. Hydro values are assigned to the cell and approximate boundary values are imposed on the low index side. The governing equations are solved for the boundary cell, but the results are in error because of the approximate boundary values. The error is not propagated since new values are assigned to the cell for the next step. Again, the program must account for system energy and mass.

Boundary values at high index boundaries are all imposed in the same manner: The cell with the highest index (I=IMAX, J=JMAX, or K=KMAX) is a virtual external cell; appropriate values are assigned to these virtual cells for the desired boundary condition and these values are used to compute the pressure and normal velocity component at time $t + \Delta t/2$ on the high index side of the adjacent internal cell.

The controls for the phase 2 calculations, the transport phase, are similar. The first step in phase 2 computes the fluxing of mass, momenta, and energy through the cell boundaries. The second step computes the result of this fluxing. Initially, for any internal cell, the Lagrangian solution for velocity and specific energy (at the point that was the cell center at time t) are known at time $t + \Delta t$; the mass (hence density) of the cell is known for time t. The average of the normal velocity components of two contiguous cells is used as the rate of mass transport between them. From this, the mass flux, the momentum fluxes, and the energy flux between the two cells are computed.

Just as for phase 1, the values on the low index sides of the cell are assumed to be known, either from computation for the cell on that side or from boundary condition assignment. Computing the fluxing for the high index sides of the cell completes step 1 of phase 2. Step 2 of phase 2 consists of summing all the fluxes and computing the resulting mass, velocity components, and specific internal energy for the cell at time $t + \Delta t$.

Similar boundary control methods to those described for the first phase are used for phase 2. In addition, any transfer of mass or energy through the boundaries must be accounted for. (In 2-D, some of the options such as multimaterial or radiation require additional computation.)

The 2-D and 3-D codes are really separate programs although they share some coding. They even differ slightly in the solution of the governing equations. (For example, 2-D simply averages for values between cell centers, the 3-D interpolates for some of them.)

Reading the comment lines with the HULL coding could lead one to believe there were eight types of boundary conditions available at every boundary. This is not true. Of the eight options listed, only one type, reflective, is available at all the boundaries. Table 1 is a summary of the author's assessment of the availability and reliability of the boundary conditions in our airblast HULL. The eight types numbered 0 through 7 are called reflective, transmissive, rezone, square wave, LAMB, SAP, HULL, and oblique square wave, respectively.

The next section will describe the new imposed-solution boundaries developed at the BRL.

Table 1. Summary of HULL Boundaries.

		BOUNDARY TYPE							
BOUNDARY		0	1	2	3	4	5	6	7
AFT	3-D	YES	A,C	A,C	A	A	-	-	YES
BOTTOM	3-D	YES	A,C	A,C	-	-	-	-	A
BOTTOM	2-D	YES	A,C	A,C	B	B	A	-	YES
LEFT	3-D	YES	YES	A,C	YES	YES	-	-	YES
LEFT	2-D	YES	A,C	A,C	YES	A	A	A	YES
FORE	3-D	YES	YES	-	-	-	-	-	-
TOP	3-D	YES	YES	-	-	-	-	-	-
TOP	2-D	YES	YES	YES	-	-	-	-	-
RIGHT	3-D	YES	YES	-	-	-	-	-	-
RIGHT	2-D	YES	YES	YES	-	-	-	-	-

(YES) Has been used locally. Looks good.

(-) No implementation.

(A) Has not been used locally. Appears to be fully implemented. May have errors.

(B) Only partially implemented or has known errors.

(C) Types 1 and 2 alike at each low index boundary. Believe they are intended to remain unchanged ambient.

III. BOUNDARY TYPE 9, IMPOSED-SOLUTION BOUNDARY

The primary motivation for this project was to transform a section of a 2-D cylindrical symmetry HULL run into a 3-D Cartesian HULL run. Presumably, the 2-D HULL program would be run with a blast wave expanding from some central source. The 3-D run would be for some subsection of the cylindrical space with a target inside. The 3-D run could be initiated just before the blast wave reached the target and would continue with input from the 2-D solution through the boundaries of the 3-D grid.

The KEEL program has coding for copying a portion of a 2-D cylindrical HULL problem into a 3-D space but the boundary conditions from the 2-D run could not be satisfactorily fed in after the 3-D run was initiated. The result from such a restart would be good until a signal from whatever boundaries were used reached the region of interest. With the addition of input boundaries, with data from the 2-D donor run, the results should be good until reflections from the target reach a boundary and the resulting erroneous signal reaches the region of interest. Imposing a solution on the boundaries of the 3-D grid would essentially double the real time that could be simulated by the same grid with simple transmissive boundaries. For comparison, the same time doubling of simulated time could be achieved by doubling the distances from the target to the simple transmissive boundaries in the 3-D problem, but this could increase the space and computer time required for such a run by a factor of eight. Thus, this new capability offers the opportunity to run problems which might otherwise be too large or expensive.

In keeping with this scenario, the coding for these new boundaries allows imposed values on any side. (Previous programming had not added boundary conditions at the high index boundaries even though the existence of the layer of external, virtual cells make this relatively easy.) These "imposed-solution" boundaries will be called BOUND9 boundaries in this report. They will be activated in HULL as type 9. (Type 8 has been assigned at the BRL to an undocumented, partially successful exhaust type boundary that may be reactivated.)

After some preliminary study, it was decided that the simplest way to introduce these new HULL boundary conditions into the existing code was to supply input values on a plane for each boundary, with a new subroutine in HULL for each of these boundaries. These subroutines check that the point for which data is requested is on the plane. Then they interpolate in time, and in space on the plane, for five hydro values: three velocity components, specific internal energy, and density. At the low index sides these planes are on the boundaries: the "left" boundary at X0, the "aft" boundary at Y0, and the "bottom" boundary at Z0. For these boundaries, the coding computes and stores the pressure and the normal velocity component at time t + t/2, like method (2) in the previous section. At the high index boundaries, the input planes for the imposed-solution values go through the center of the external virtual cells on that side: the "right" boundary for maximum X, the "fore" boundary for maximum Y, and the "top" boundary for maximum Z. The coding inserts the "imposed" hydro data into the external cell (mass is stored instead of density).

These new BOUND9 boundaries are designed to continue a 2-D HULL run in 3-D, but any sort of values could be imposed on a side if a file of data in the proper form were prepared for that side. (This assumes that the boundary values are compatible with whatever is inside the computation region.)

The original plan was to supply two planes of hydro data at successive times for each BOUND9 side. An evenly spaced grid with extreme values at the BOUND9 planes was also imposed. This worked well for small 3-D HULL grids with evenly spaced meshes. The examples cited later were run with this coding, but some memory requirements were excessive even with a coarser mesh on the input data planes than was wanted. Therefore, a new code was prepared which presents the imposed-boundary data to the 3-D HULL one row at a time. To avoid excessive searching for data, the hydro values for two successive times are stored together for each row.

A description of this BOUND9 input for HULL is included in Appendix B along with a listing of the BOUND9 changes for HULL and listings of the runstreams for KEEL and HULL for one problem.

IV. HULLUP

The program to prepare the files for the 3-D HULL BOUND9 input boundaries from a 2-D donor run is a FORTRAN 5 program called HULLUP. This program is tabulated in Appendix A. It includes COMMENT lines that hopefully adequately describe the input and output. The main output is the files of data for the BOUND9 input to HULL described in Appendix B.

The original intent was to allow for 2-D and 3-D input and output at some later date. There are some remnants of this remaining, but the coding is now strictly for producing HULL input on 3-D boundary planes from a 2-D cylindrical HULL run.

The user must plan the 3-D run thoroughly before knowing the input for HULLUP. If there is a target, it is one or more orthogonal parallelepipeds (boxes). If a target is to have flat sides, the 3-D grid must have boundaries parallel to the sides of the target. These boundaries, except for reflective boundaries, must be far enough from the target area to allow HULL to run without sending a false signal to the target area during the time of interest. (For BOUND9 boundaries, the signal from the boundary will be approximately correct until reflection from the target reaches them.) Any BOUND9 boundary plane must be entirely inside the cylindrical region of the 2-D donor. The angle at which the shock front strikes a target is determined by the positioning of the target. The transfer from the 2-D donor space, which HULL denotes as (X,Y) is clearer if one thinks of the 2-D cylindrical space as (R,Z), or the 3-D cylindrical space (R,θ,Z) at arbitrary θ.

Points in the Cartesian 3-D space are defined as $X = R\cos(\theta)$, $Y = R\sin(\theta)$, and $Z = Z$ (no rotation of Z plane). The user does not need to know θ. To set the boundaries of the 3-D Cartesian space the user must envision, or draw, the X and Y boundaries of the 3-D computational field on a constant Z plane. A line through the origin parallel to the X sides of the computational field is $\theta = 0$, and $\theta = 90^{\circ}$ is parallel to the Y sides. The choice of quadrant and rotation angle of θ is arbitrary. The only restrictions are that the left boundary, XO in HULL, is the minimum X, and the aft boundary, YO in HULL, is the minimum Y. For the KEEL run, the initiation run for HULL, the user needs to supply XO, YO, ZO, and the grid spacing for the computational field. For HULLUP, a more complete knowledge of the computational grid is needed.

The input for HULLUP is through four NAMELIST lines and some other data on the INPUT file, and from the donor restart file. A user must have access to file NUHULUP and should get specific instructions for program HULLUP from it. The donor restart file must be ATTACHed as TAPE 9 and the output files must be CATALOGed in the runstream. HULLUP and an example runstream are tabulated in Appendix A.

V. TEST RUNS

A limited number of test problem runs have been made, most of them with the original coding that supplied input data for the entire plane. The first three problems were simply designed to get the program running. First a 2-D cylindrical donor run was made. It had a sphere of high pressure gas (E in Figure 1) with radius of 1000 cm expanding into an (IMAX,JMAX) = (32,16) mesh of toroidal cells, 80x80 cm in cross section. In the 2-D run (for the cylindrical space A in Figure 1 represented by plane B), the elevation, Y, went from 0 to 1200 cm with one additional layer of external cells and the radius, X, from 0 to 2480 cm with an external column of cells. The sphere of high pressure gas was centered at (0,1200). The left, bottom, and top boundaries were declared reflective and the right boundary transmissive. Setting the top boundary reflective (i.e., symmetric) is all right for the assumed constant atmosphere with no gravity until the reflected shock from the ground reaches the top boundary.

The sphere of high pressure gas was given a density of 0.0381204 g/cm^3. The ambient density was set at 0.00120412 g/cm^3. The specific internal energy was 2.10374 x 10^9 ergs/g everywhere and a gamma law gas was assumed with gamma = 1.4. Such a driver gas would produce a 344.7 kPa (50 psi) shock in a straight shock tube. With spherical expansion, the shock front should be about 103 kPa (15 psi) at 1800 cm radius. The 2-D HULL run was initiated at time T = 0.02 seconds and was run to 0.05 seconds.

Two 3-D problems with cubic cells 80 cm on a side were based on this 2-D donor computation. The first of these had an internal region (C in Figure 1) with both X and Y from 0 to 1760 cm and Z from 0 to 1200 cm. This was run with the left, aft, top, and bottom boundaries reflective and the right and fore boundaries transmissive. The initiation for this problem was from the 2-D donor through KEEL at time T = 0.02 sec. This was run to check the starting of a 3-D run from a 2-D run and to give some idea of the difference in 3-D and 2-D results for this problem. (Starting the 3-D run from a 2-D run will tend to further distort the boundary of the high density sphere.) As expected, the results were bilaterally symmetric about X = Y. Radial symmetry was reasonable considering the coarseness of the grids.

The other 3-D run based on the 80 cm grid donor also had 80 cm cubic cells, but it was for a smaller subspace (D in Figure 1) with BOUND9 input from the donor on the left, right, fore, and top boundaries. The bottom and aft boundaries were reflective. A 9 by 7 by 8 mesh of cells was selected with the inner region from X = 880 to 1520 cm, Y = 0 to 480 cm, and Z from 0 to 560 cm. This was also initiated at time T = 0.02 sec from the donor restart file. This region was entirely outside the high pressure sphere defined for the 2-D run, so the initiation was actually for an ambient region.

BOUND9 input planes were supplied through HULLUP at X = 880, the left boundary, X = 1800, the center of the external plane of boundary cells on the right, Z = 600, the center of the top plane of external cells, and Y = 500, the center of the external plane of cells on the fore boundary. The grid on these planes was such that values were available to HULL without further smearing due to interpolation in the boundary planes (i.e., there were data points at the cell centers).

Figures 2-7 show the HULL records of overpressure at 6 stations (See Figure 1) for the three runs. Stations 1, 2, and 3 were assigned at X = 990, 1250, and 1500 cm, respectively, Z = 30 cm, and Y = 30 cm for the 3-D runs (θ= 0 for 2-D). Stations 4, 5, and 6 were assigned to X = 990, 1250, and 1500 cm, respectively, Z = 500 cm, and Y = 30 cm. Although assignment is to a point in space, the recorded values are for a cell containing the point. The results are best thought of as the average values in the cell, or alternatively, as the values at the cell center. For example, station 1 in the 3-D runs is in the cell between X = 960 and 1040, Y = 0 and 80, and Z = 0 and 80, with center at (X,Y,Z) = (1000,40,40).

In Figures 2-7, and the others in this report, time 0 corresponds to the HULL initiation time of 0.02 seconds, and of course, 30 ms corresponds to 0.05 seconds in HULL time. The 3 curves are labeled DONOR, 3DTRAN, and 3DBND9 to identify results from the 2-D donor, the 3-D simulation of the 2-D run, and the 3-D run in the smaller subspace using BOUND9 input, respectively.

There is a shift in time between the 3-D and 2-D curves, most noticeable in the shock arrival time and rise. Most of this is from the different treatment of overpressure recording for 2-D and 3-D. In 2-D runs, the pressure at the beginning of a time step is recorded with the station data. In 3-D, the pressure is computed from energy and density at the end of the time step. For this coarse mesh, the time steps varied from about 0.05 to 0.9 ms after 5 ms. The time shift appears smaller for Figure 4 since the time step is less than 0.3 ms during the initial pressure increase. (Incidentally, the time recorded with a time step for both 2-D and 3-D is the time at the beginning of the time step. The time step is not recorded, and not all time steps have output so the correct time is not known.) Considering the coarseness of the grid, the agreement among these results is reasonable. Three similar runs were made with cells approximately 40 cm on a side. (Halved cell sizes were desired with the same boundaries as before). Since HULL demands an odd number of cells inside the boundary in the vertical direction, there was a minor problem. In the 2-D run and the 3-D full simulation, the top boundary was located at 1240 cm and the layer of cells between 1200 and 1240 cm was declared to be ISLAND cells. This created a reflective (symmetry) boundary at 1200 cm. For the BOUND9 top boundary in 3-D an odd number of cells between Z = 0 and 560 cm was required. The 7 internal cells with DZ = 80 cm were replaced with 15 cells with DZ = 38, 39, or 40 cm.)

Figures 8-13 are overpressure records for these "40 cm grid" runs at the same stations as for the the "80 cm grid" runs. As was expected, the time shifts are less, the peaks are sharper and higher, and the curves have more detail. The peak pressures for the full 3-D run is always somewhat higher than the corresponding 2-D peaks. This is a consistent feature of HULL[4] probably due to minor differences in the finite difference algorithms.

The improvement was so encouraging that a 2-D donor run and the corresponding 3-D run with BOUND9 input boundaries were run with cells approximately 20 cm on a side (overpressure vs time plots, Figures 14-19) and for cells approximately 10 cm on a side (Figures 20-25). Except for station 1, the 2-D donor runs and the BOUND9 3-D runs are more alike for the 20 cm grids than for the 40 or 80 cm grids. The time shift due to the time step has been reduced to about 0.2 ms so the "smearing" of the shock front by interpolation for and from BOUND9 input boundary data is more evident at those stations near the BOUND9 boundaries, stations 1, 4, 5, and 6.

One would expect further improvement with the 10 cm grid runs. This is not evident. There are several reasons, the principal one being additional smearing due to interpolation in the supplied BOUND9 data in HULL. For all the coarser grids, the HULLUP coding used an evenly spaced mesh on every side between the BOUND9 values at their edges. By choosing the appropriate number of mesh divisions there were points in the center of the cells on the evenly spaced sides and near the centers in the Z direction. Such a fine grid was not possible for the 10 cm grid. A much coarser mesh was forced by computer space limitations. Hence, there was smearing in both the HULLUP interpolation and the HULL interpolation. This difficulty led to a realization that a revision was necessary.

Another probable source of differences at stations 1, 2, and 3 is that they are closer to the reflective bottom boundary in 2-D than in 3-D. For 2-D they are in the third layer up between 20 and 30 cm, and for the 3-D they are in the fourth layer between 28.5 and 38 cm.

The notches in the initial pressure rise in Figures 20 and 21 are probably due to an initial decay of the incident shock followed by the reflected shock. (The stations are 3 or 4 cells from the reflective bottom.) An alternative explanation is to assume a Mach reflection which needs time to develop in the 3-D restart run.

Station 1 is near the left BOUND9 input boundary in the 3-D runs. Apparently it is too close for a sharp reflected shock to develop.

Figures 26-31 show a comparison of the overpressure from the four donor runs at each of the 6 stations. The plots for stations 4, 5, and 6 (Figures 29, 30, and 31) are much as one would expect: the peak pressure increases with decreasing cell size, the rise and fall from peak pressure is much sharper, and the curves are more detailed as cell size decreases. The same results are not so true at stations 1, 2, and 3 (Figures 26, 27, and 28) because of the offset in position and the rapid decay of the peak reflected pressure near the reflective bottom boundary. The results seem to converge and the 20 cm and 10 cm grid results agree fairly well. Better agreement would be welcomed at stations 3 and 6 near the end of these short runs.

Figures 32-37 show a comparison of the overpressures for the four BOUND9 driven 3-D runs at the same 6 stations. Overpressure for the donor run with the 10 cm grid is included as an assumed "correct" curve. Here again the convergence looks good with the 20 cm grid results and 10 cm grid results being reasonably alike.

As was stated earlier, the presented results were all from an early version of HULLUP and UPDATE changes to HULL for BOUND9 boundaries. The HULLUP program was revised to supply line by line data on the input planes at grid points that may be specified. The BOUND9 coding was revised to use the line by line input. Spatial interpolation was retained, but cell centered input is desirable. The results for the BOUND9 run with the 80 cm grid were duplicated exactly with the new coding.

The BOUND9 run with the 10 cm grid was also rerun with the new coding. Figures 38-41 show the overpressure records at the stations 1 through 4, respectively, for the 2-D donor run and the 3-D BOUND9 driven runs with the early coding, and with the revised coding. The revised coding did not make as much difference as the author expected. There is some separation of the two 3-D generated curves at station 1, and the peak values at stations 2 and 3 are noticeably different. The results at stations 4, 5, and 6 are nearly identical.

There is no reason to doubt that the HULLUP coding will properly prepare BOUND9 input for a 3-D HULL run from a cylindrical 2-D run, or to doubt that the BOUND9 changes to HULL will give a satisfactory 3-D continuation. However, there are a number of things that were not tested: there was only one computation field location, no test with a target, no test with the bottom or aft boundary a BOUND9 type, no 3-D start with a region partially filled with non-ambient gas to check for conflict between the KEEL set up and the BOUND9 boundary, no testing of the effect of changing grid size, no coordinate shifting, and only accidental testing of one or two abort situations.

VI. CONCLUSIONS

The HULL BOUND9 coding for 3-D HULL and the HULLUP program to prepare the BOUND9 input from a cylindrically symmetric 2-D HULL run seem to be working properly. Any use of them should be carefully monitored; more testing is in order before making any permanent changes to our HULL code.

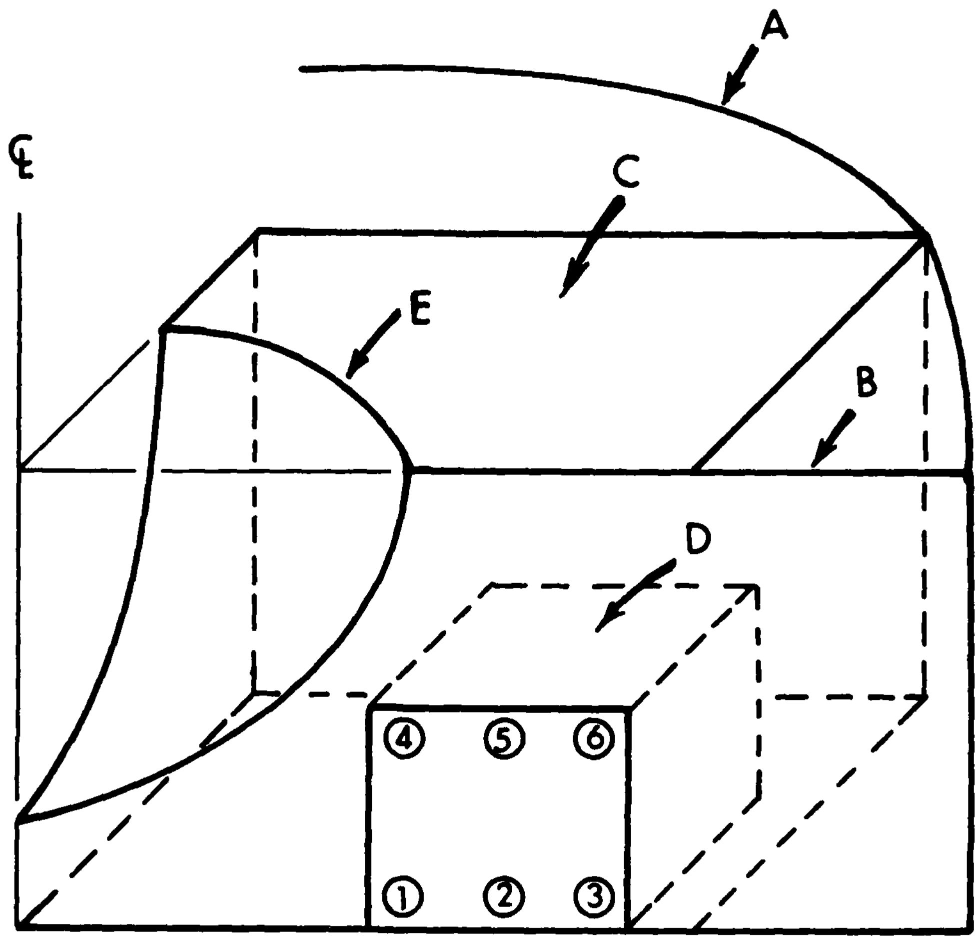
The 3-D run to be made must be thoroughly planned. The locations of the BOUND9 input planes must be exact, on the low index boundaries and in the centers of the external cells on the high index boundaries. A grid of mesh points on the BOUND9 input planes matching the centers of the 3-D HULL cells is desirable since it reduces smearing due to interpolation. A careful analysis of expected velocities and desired run time is needed to determine the placement of boundaries for the 3-D (and the 2-D) run.

The HULL BOUND9 coding will accept input on the boundaries from any source as long as it is in the proper format in the input files. For example, input on a side that is constant in space and varies with time might be useful and would be fairly easy to construct. By coding a program like HULLUP, input from any hydrocode could be used.

BOUND9 type input for 2-D codes has not been coded. A 2-D Cartesian, or new cylindrical, run may be initiated in KEEL from a 2-D cylindrical donor. A large enough region would be needed to prevent false signals from necessarily incorrect boundaries and such a region may not be possible. A 3-D run could be made in the same way but the region needed would have to be much larger than one with BOUND9 boundaries for the same uncontaminated run time.

ACKNOWLEDGEMENT

The author would like to thank Richard Lottero for suggesting this modification and for his helpful advice and support.



(A) CYLINDRICAL SPACE FOR 2-D DONOR COMPUTATION
(B) 2-D CUT IN CYLINDRICAL SPACE
(C) 3-D REGION WITH TRANSMISSIVE BOUNDARIES
(D) 3-D SUBREGION WITH IMPOSED-SOLUTION BOUNDARIES
(E) INITIAL EXTENT OF HIGH PRESSURE SPHERE ①,②···,⑥ LOCATION OF STATIONS FOR RECORDING PRESSURE

Figure 1. Computation Regions.

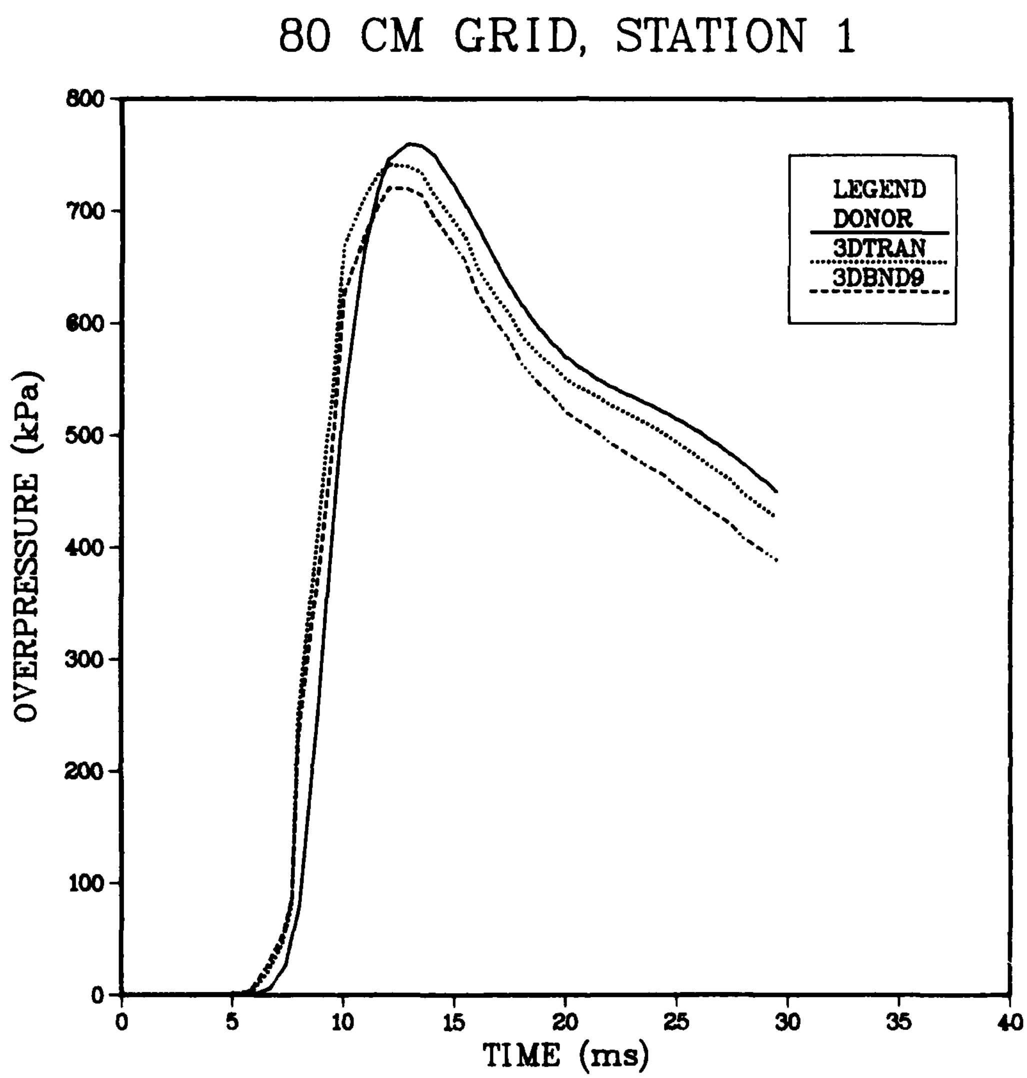


Figure 2. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for an 80 cm Mesh at Station 1.

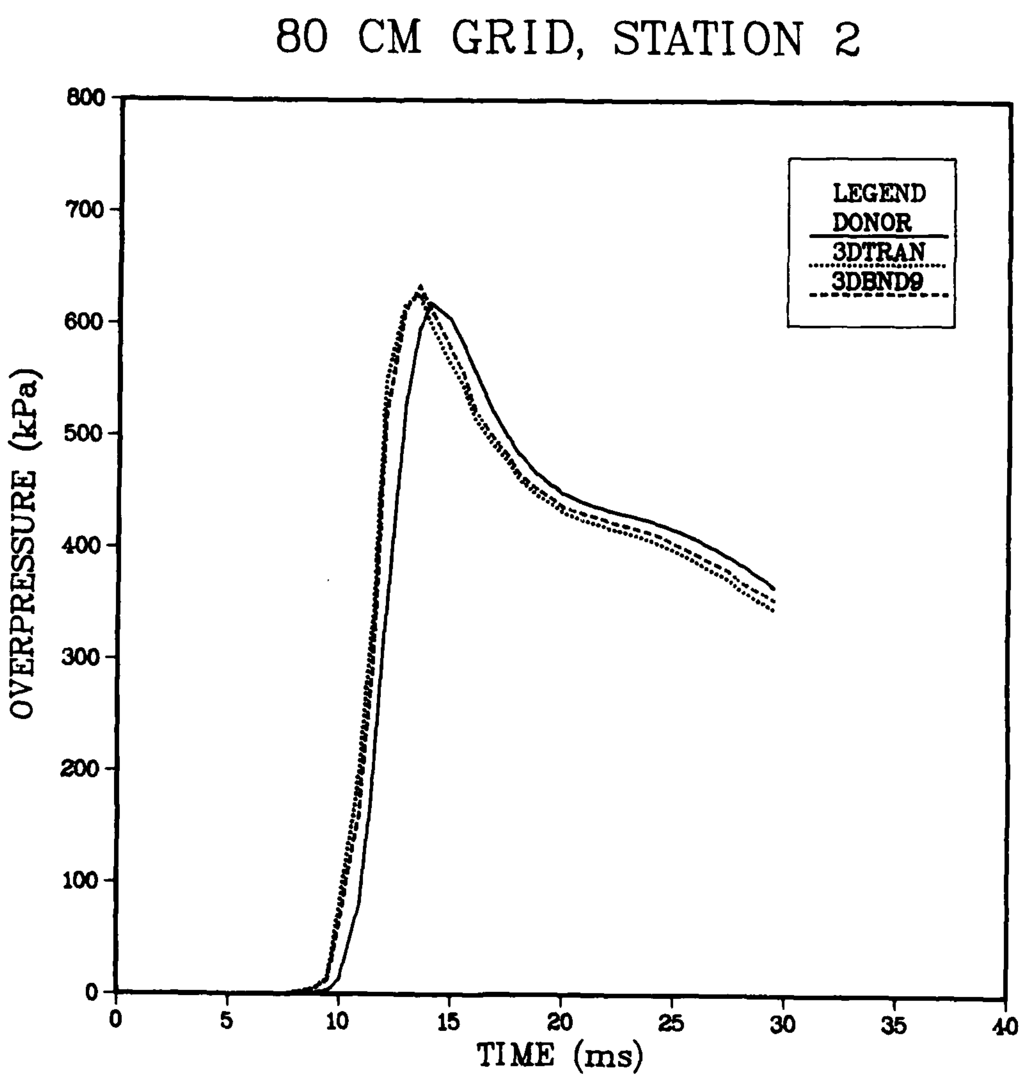


Figure 3. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for an 80 cm Mesh at Station 2.

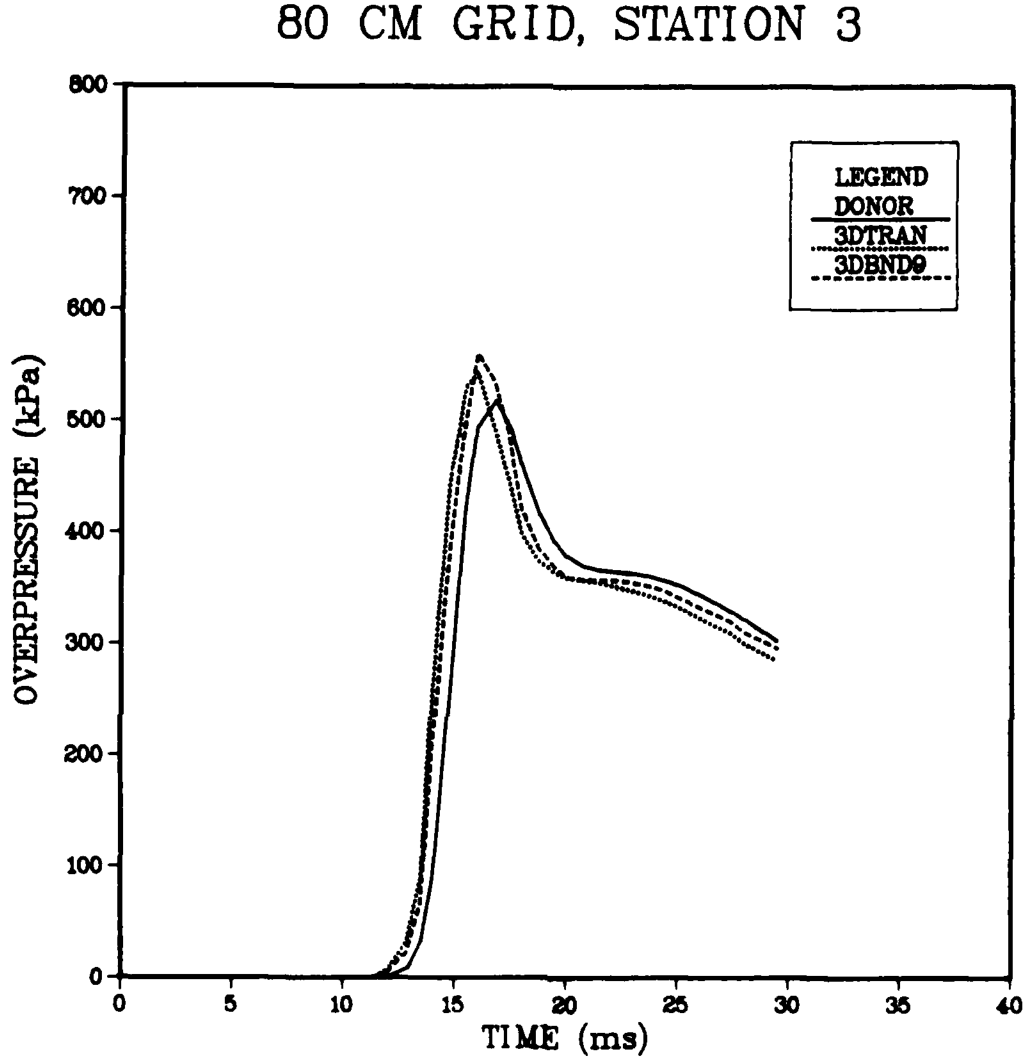


Figure 4. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for an 80 cm Mesh at Station 3.

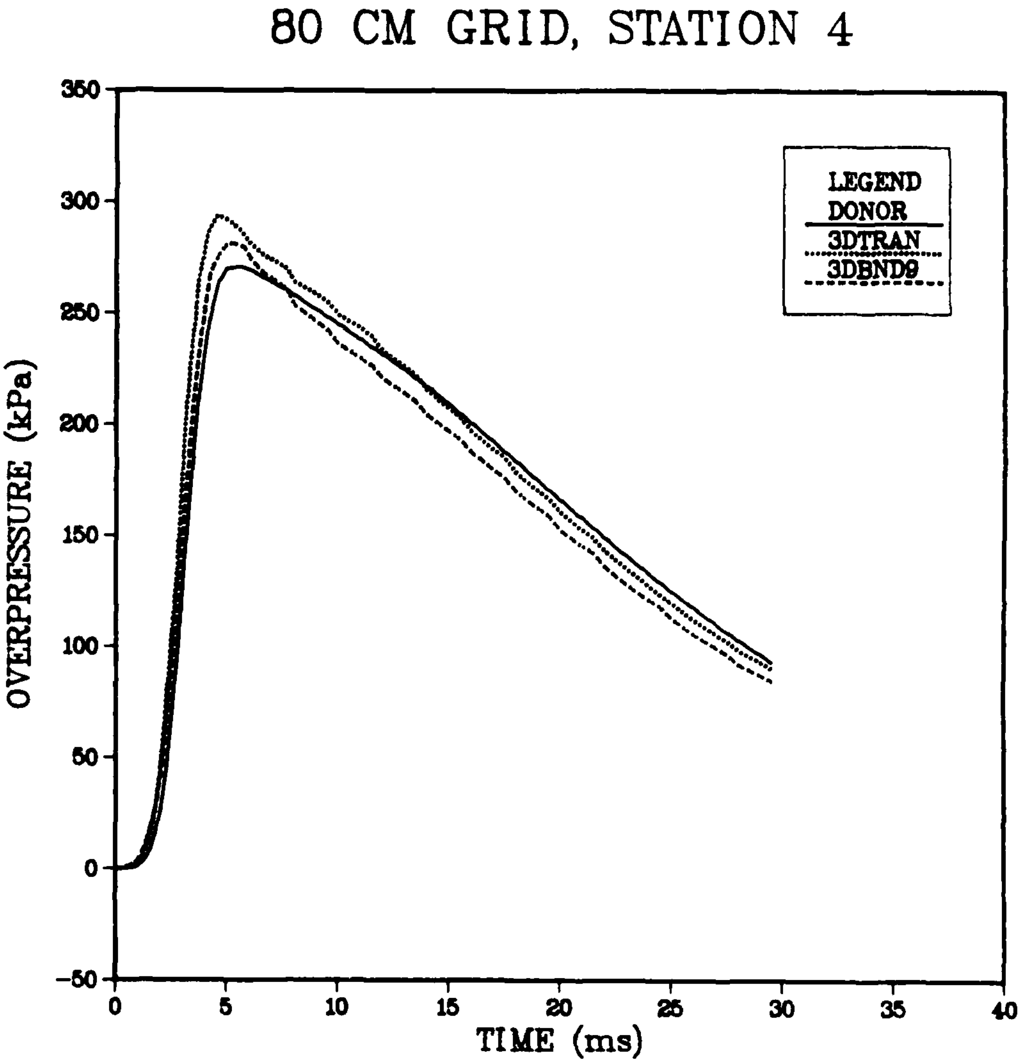


Figure 5. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for an 80 cm Mesh at Station 4.

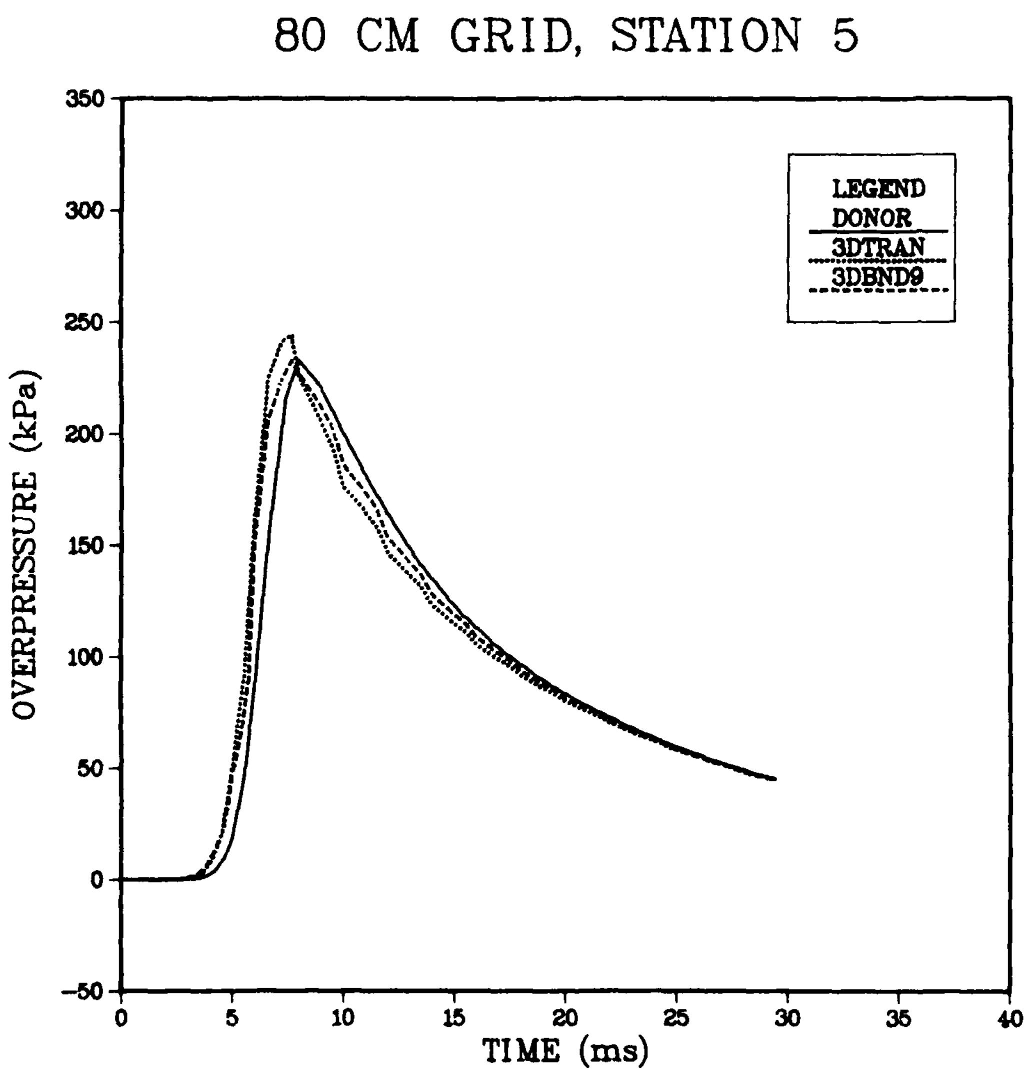


Figure 6. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for an 80 cm Mesh at Station 5.

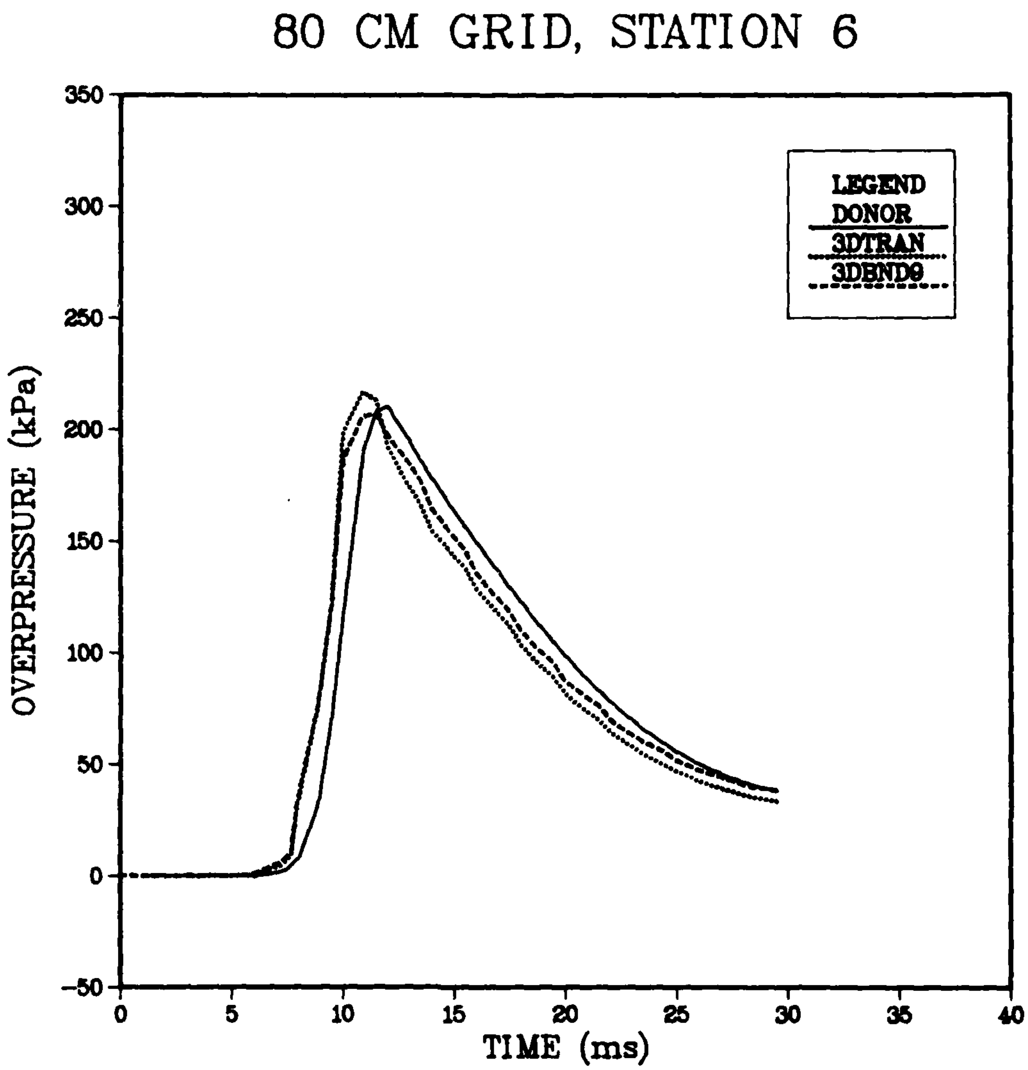


Figure 7. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for an 80 cm Mesh at Station 6.

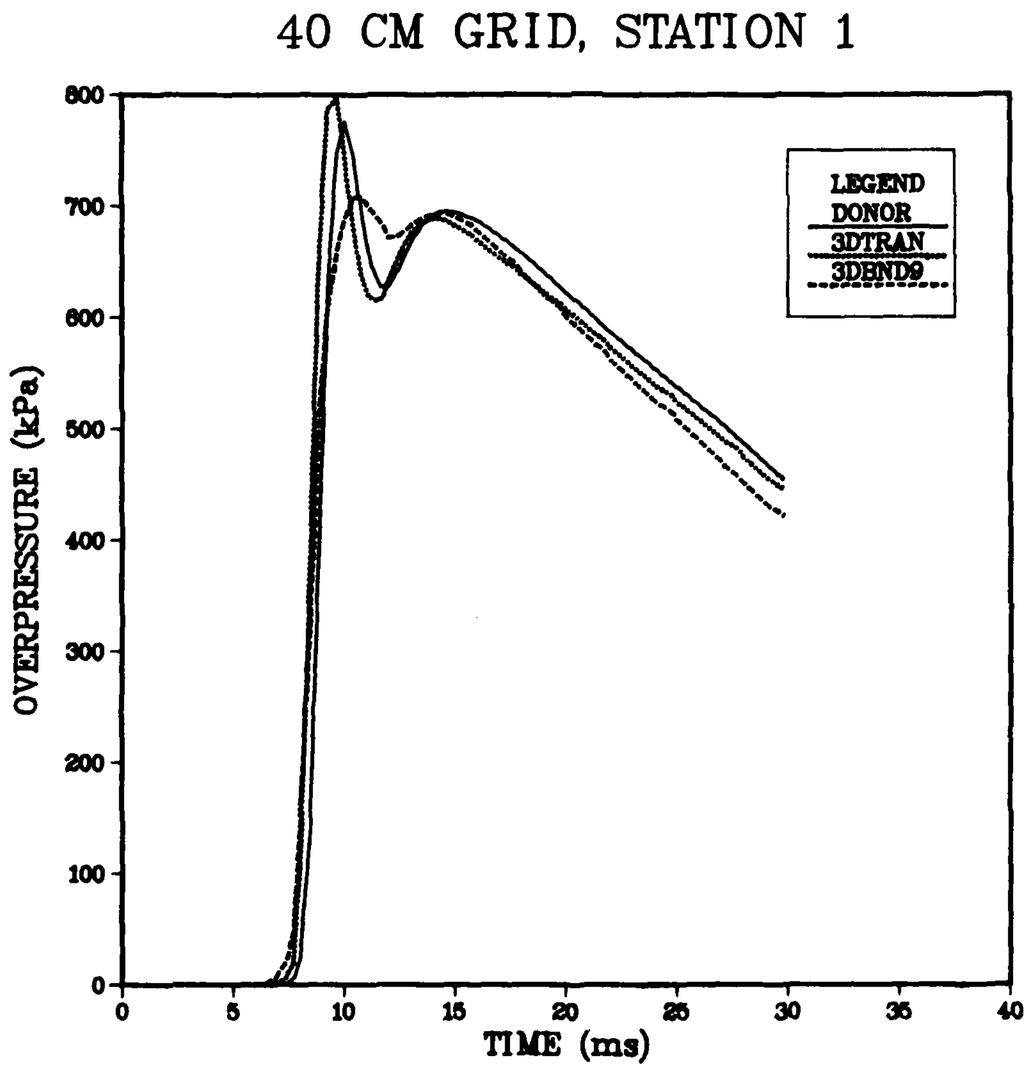


Figure 8. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for a 40 cm Mesh at Station 1.

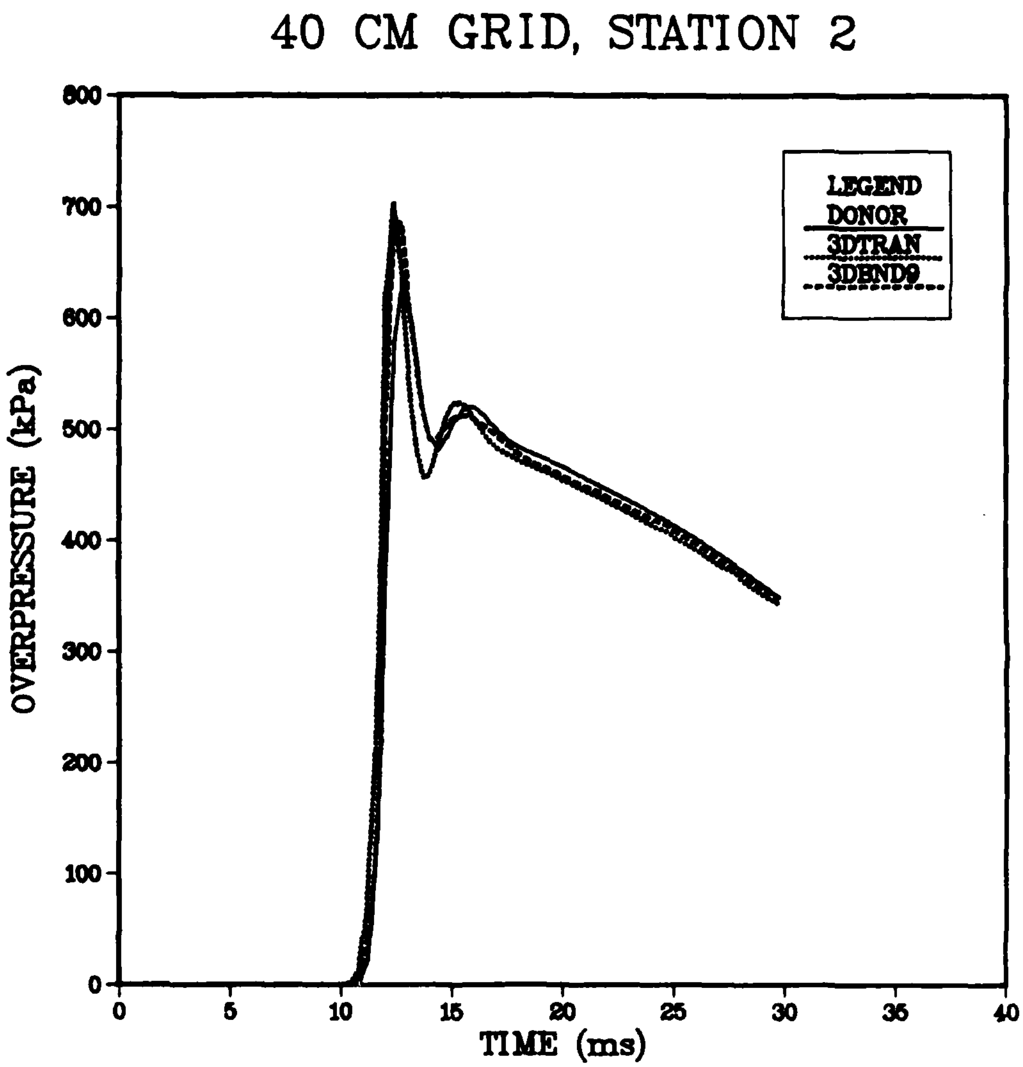


Figure 9. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for a 40 cm Mesh at Station 2.

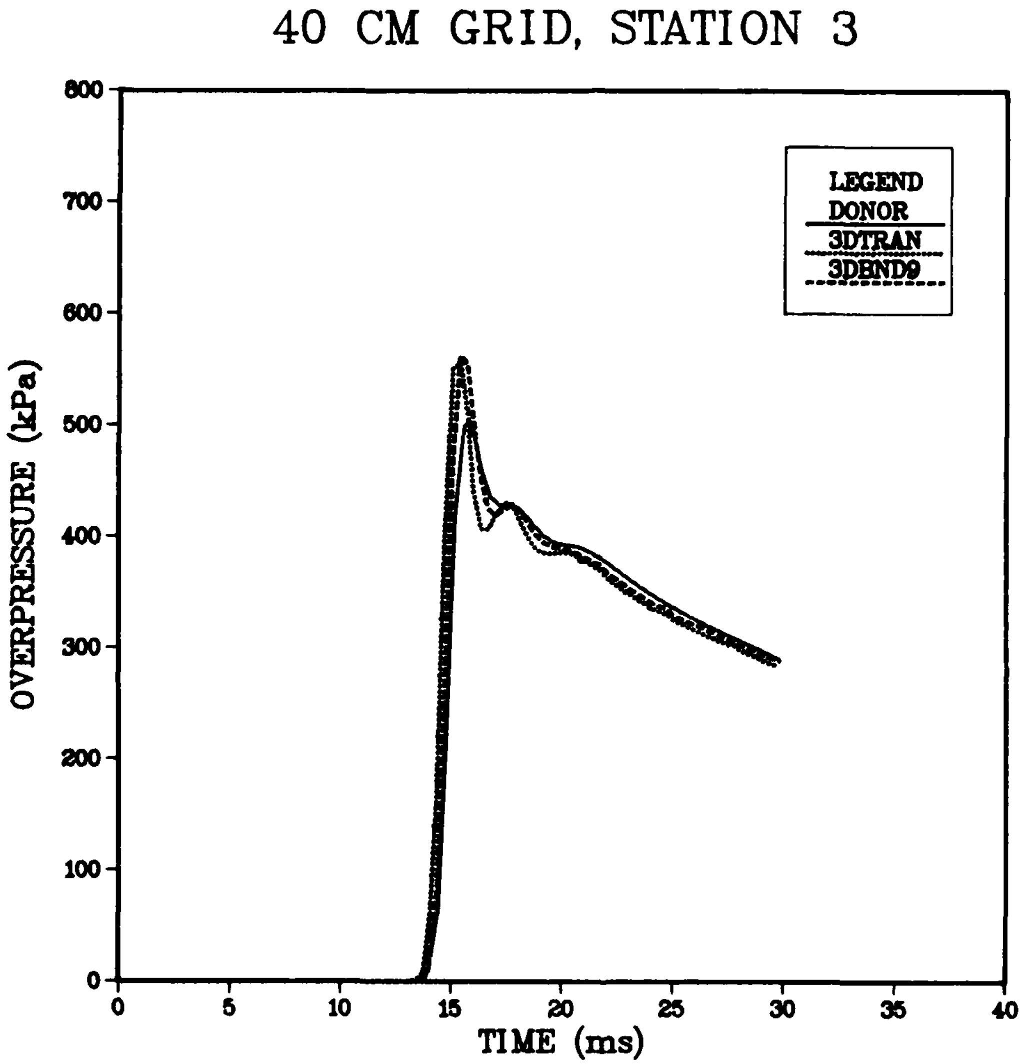


Figure 10. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for a 40 cm Mesh at Station 3.

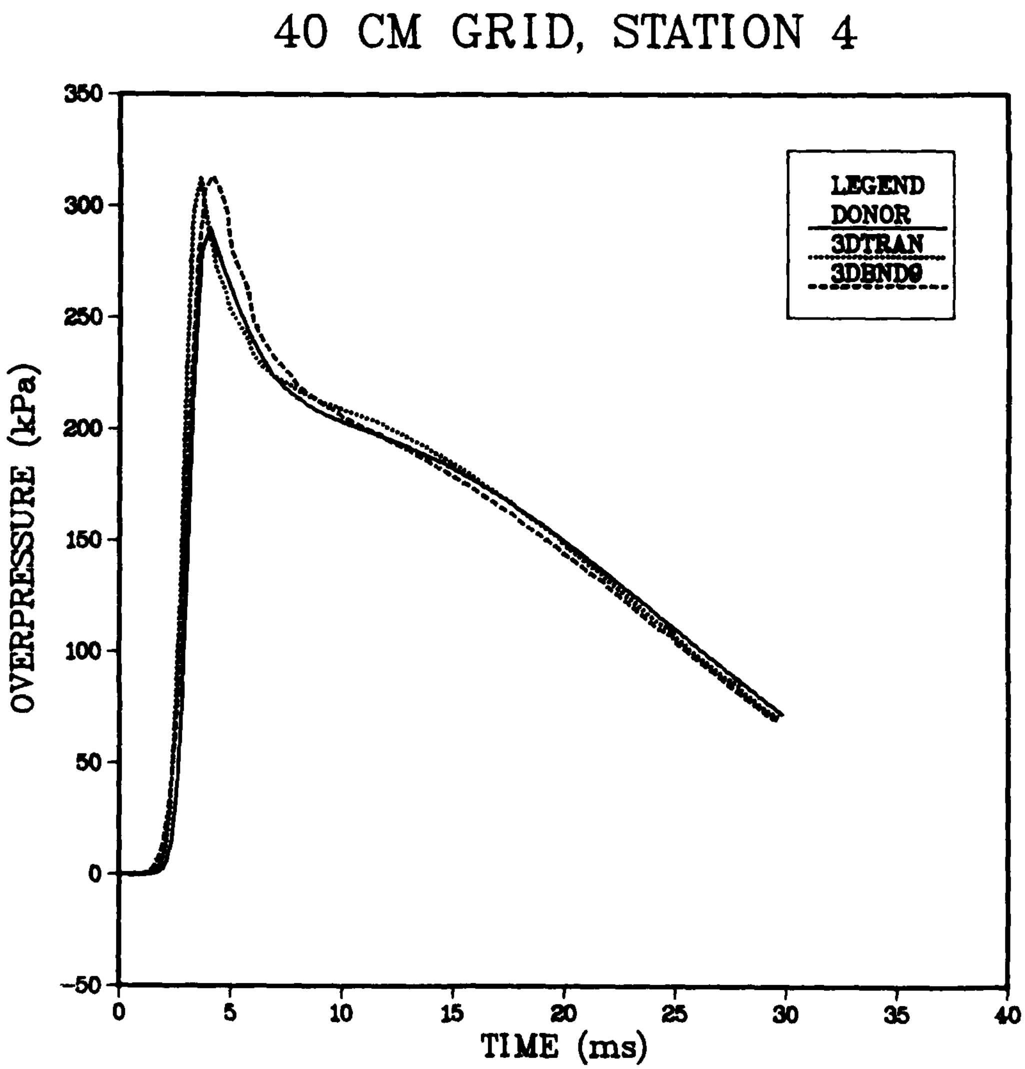


Figure 11. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for a 40 cm Mesh at Station 4.

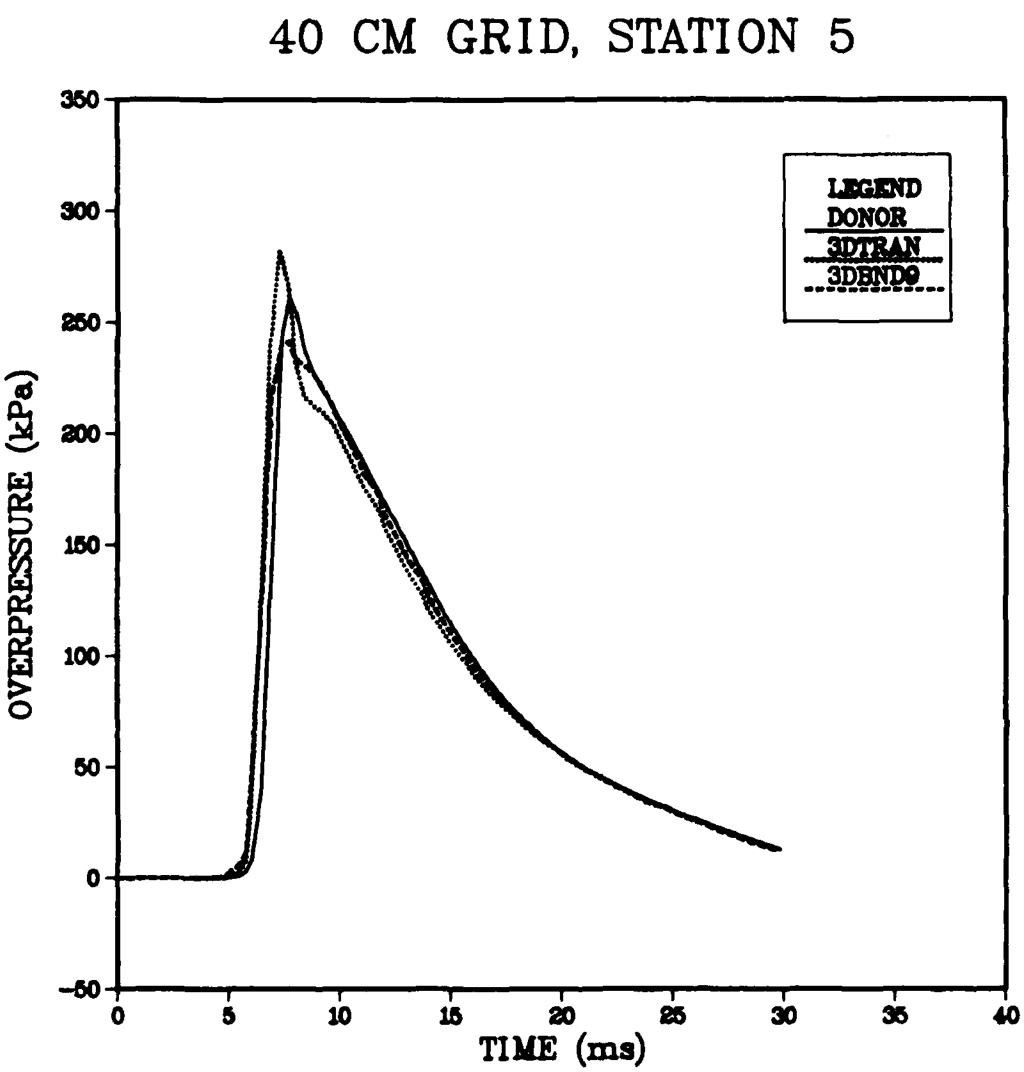


Figure 12. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for a 40 cm Mesh at Station 5.

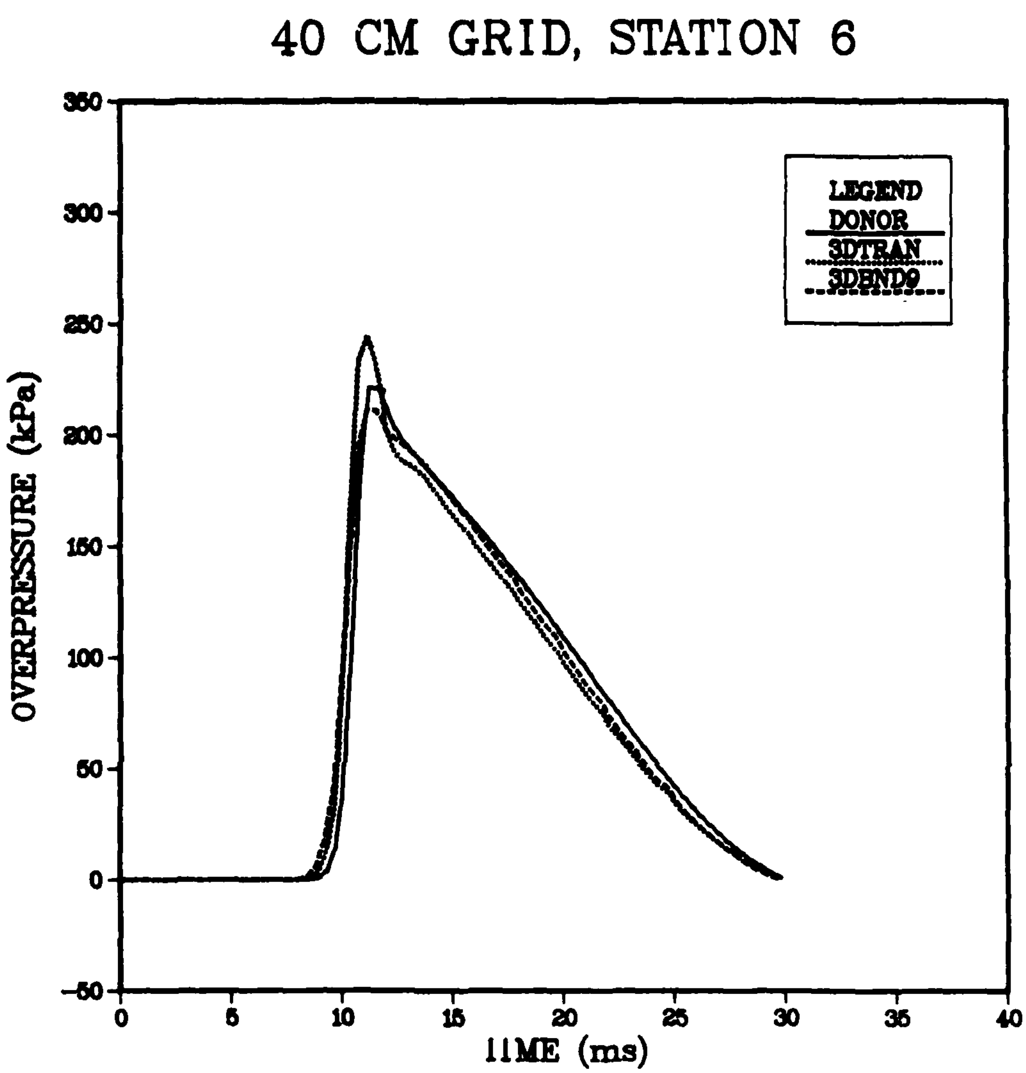


Figure 13. Comparison of Overpressure From a 2-D Donor Run, a Comparative 3-D Run, and a 3-D Imposed-Boundary Run for a 40 cm Mesh at Station 6.

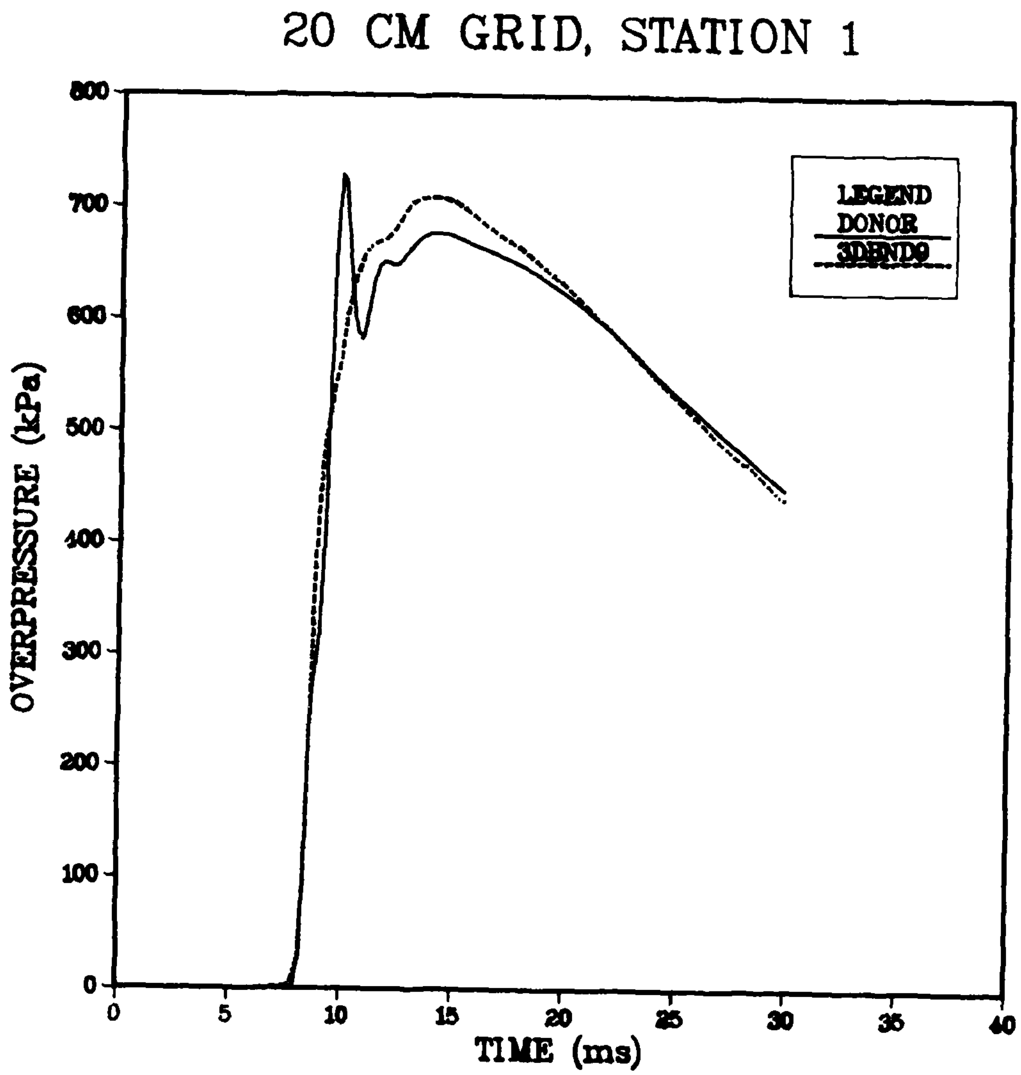


Figure 14. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 20 cm Mesh at Station 1.

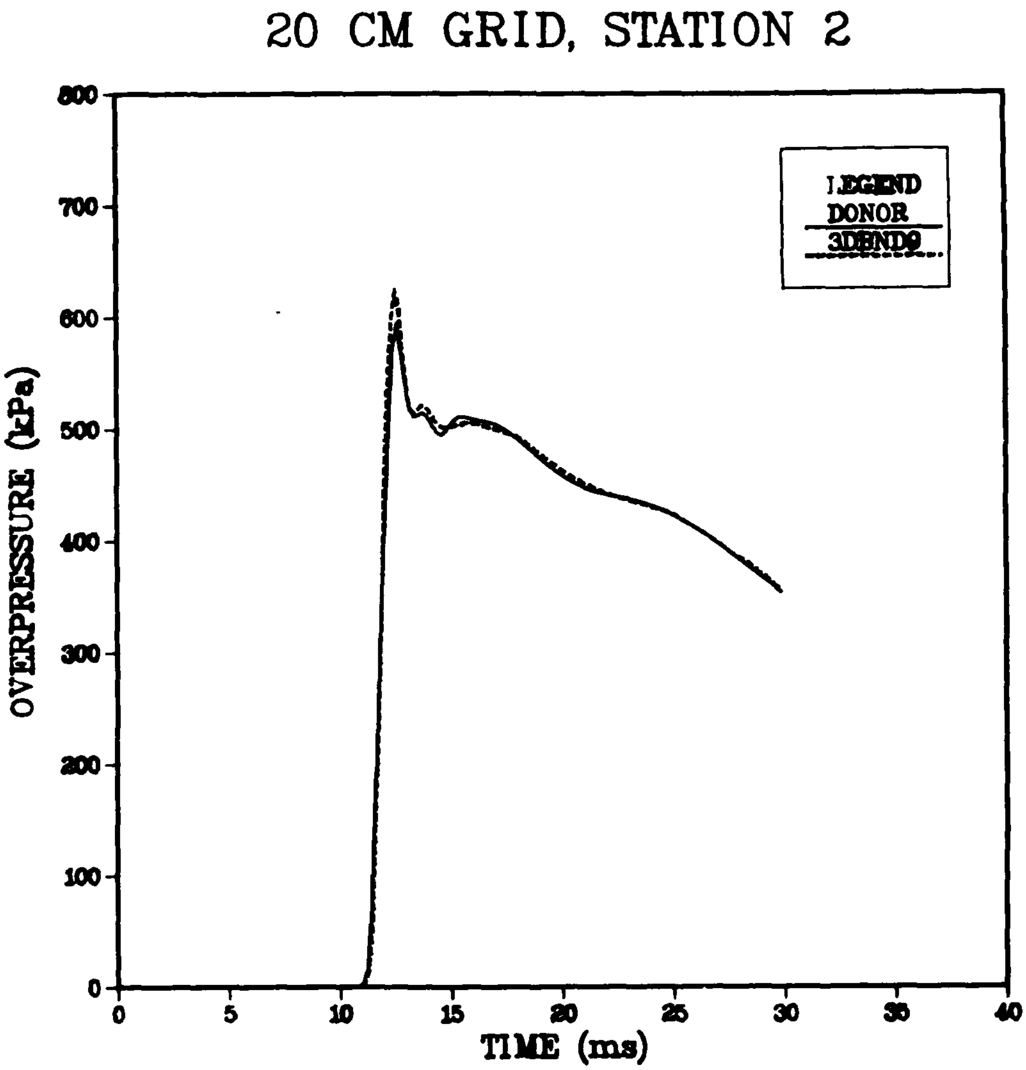


Figure 15. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 20 cm Mesh at Station 2.

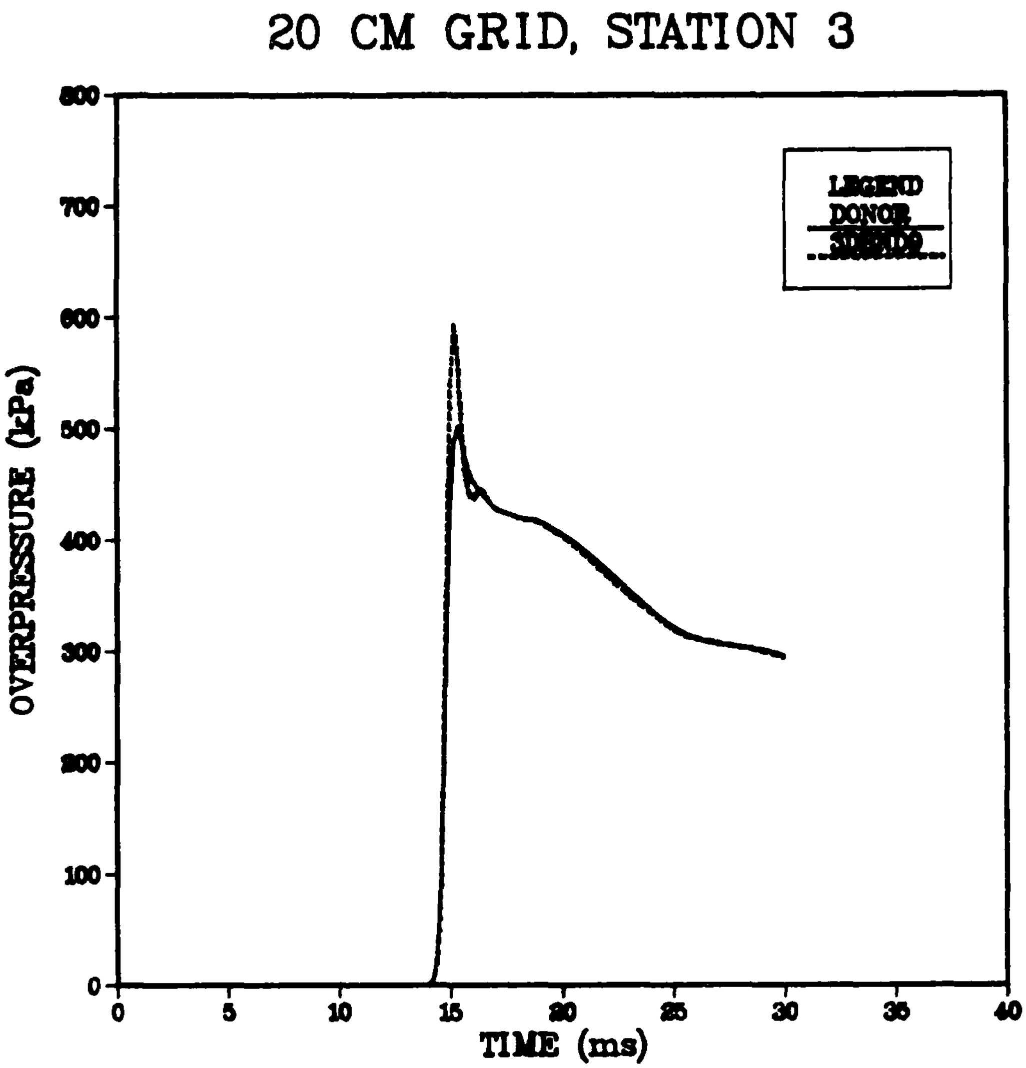


Figure 16. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 20 cm Mesh at Station 3.

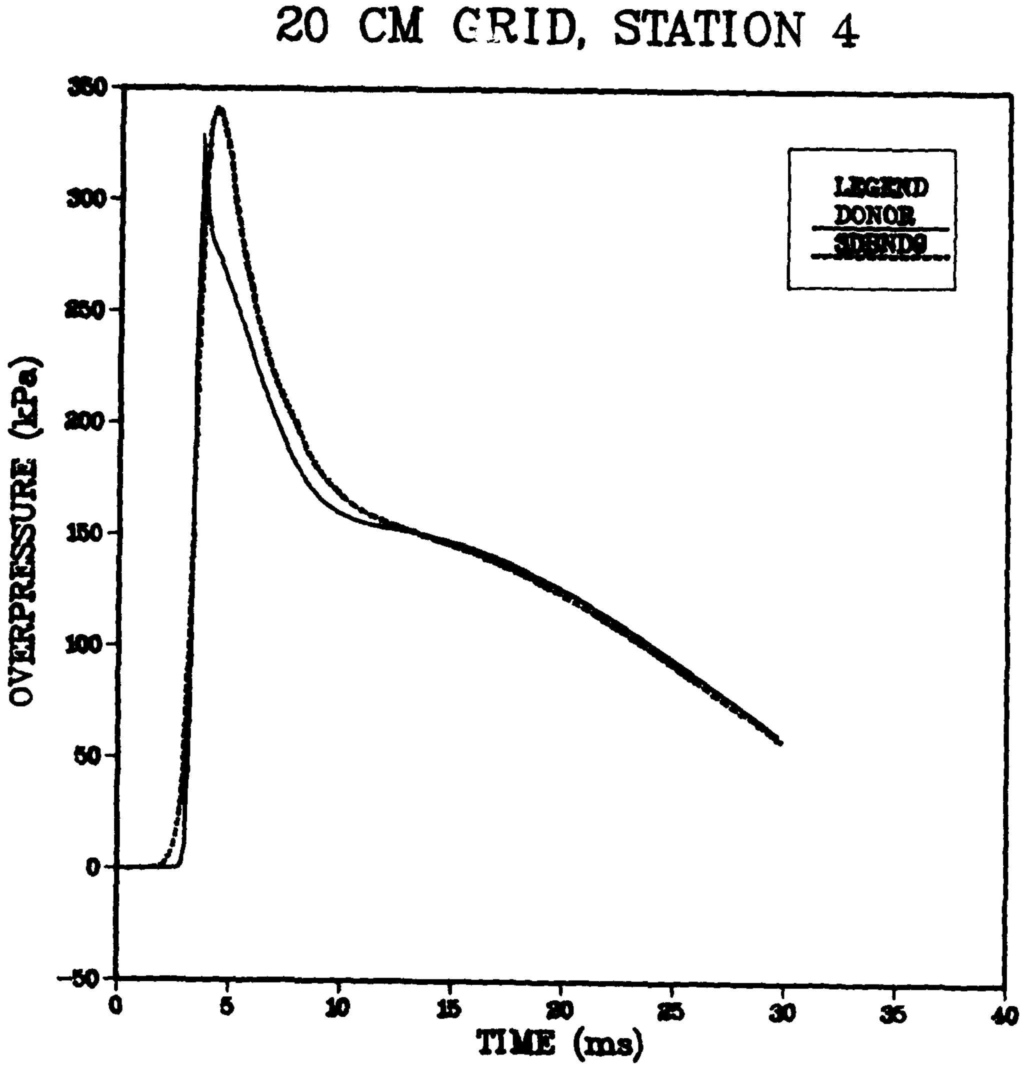


Figure 17. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 20 cm Mesh at Station 4.

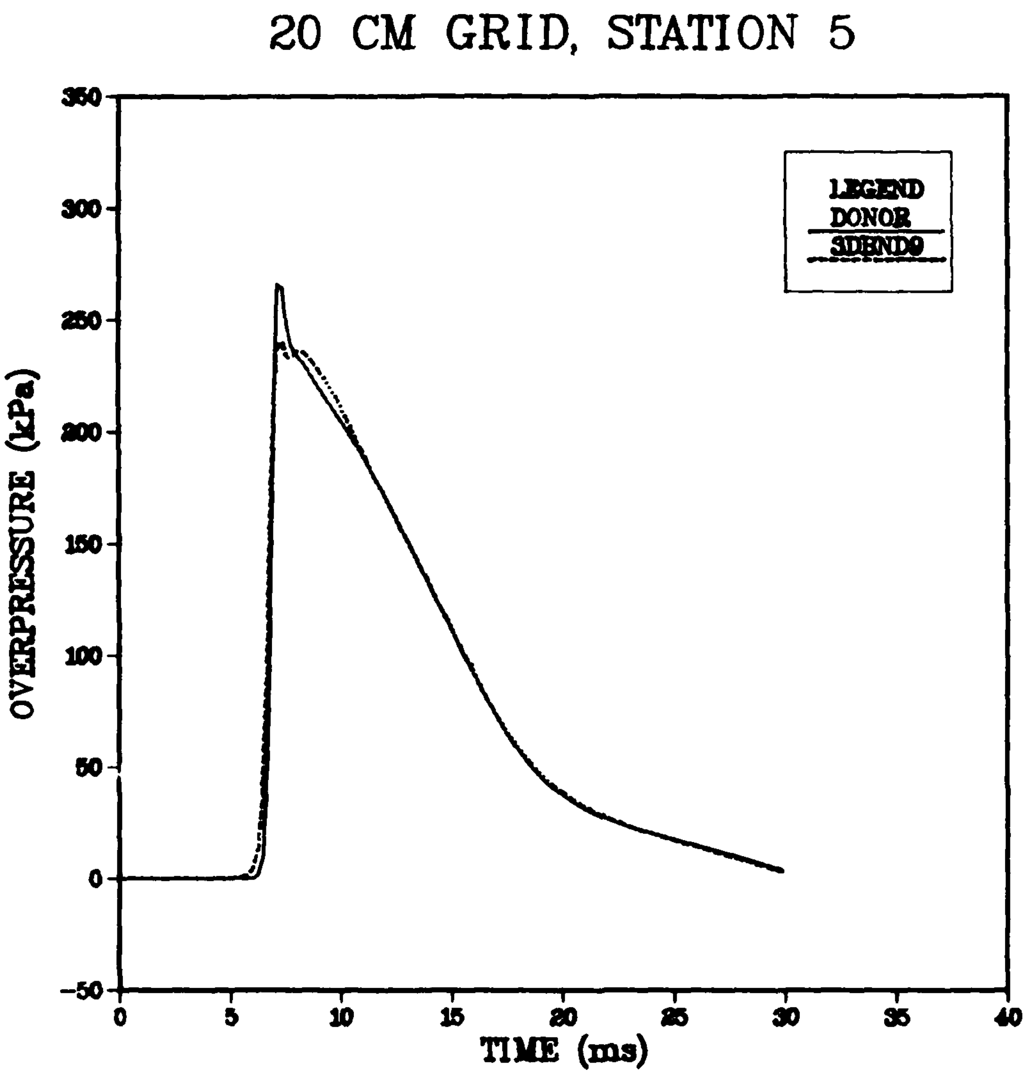


Figure 18. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 20 cm Mesh at Station 5.

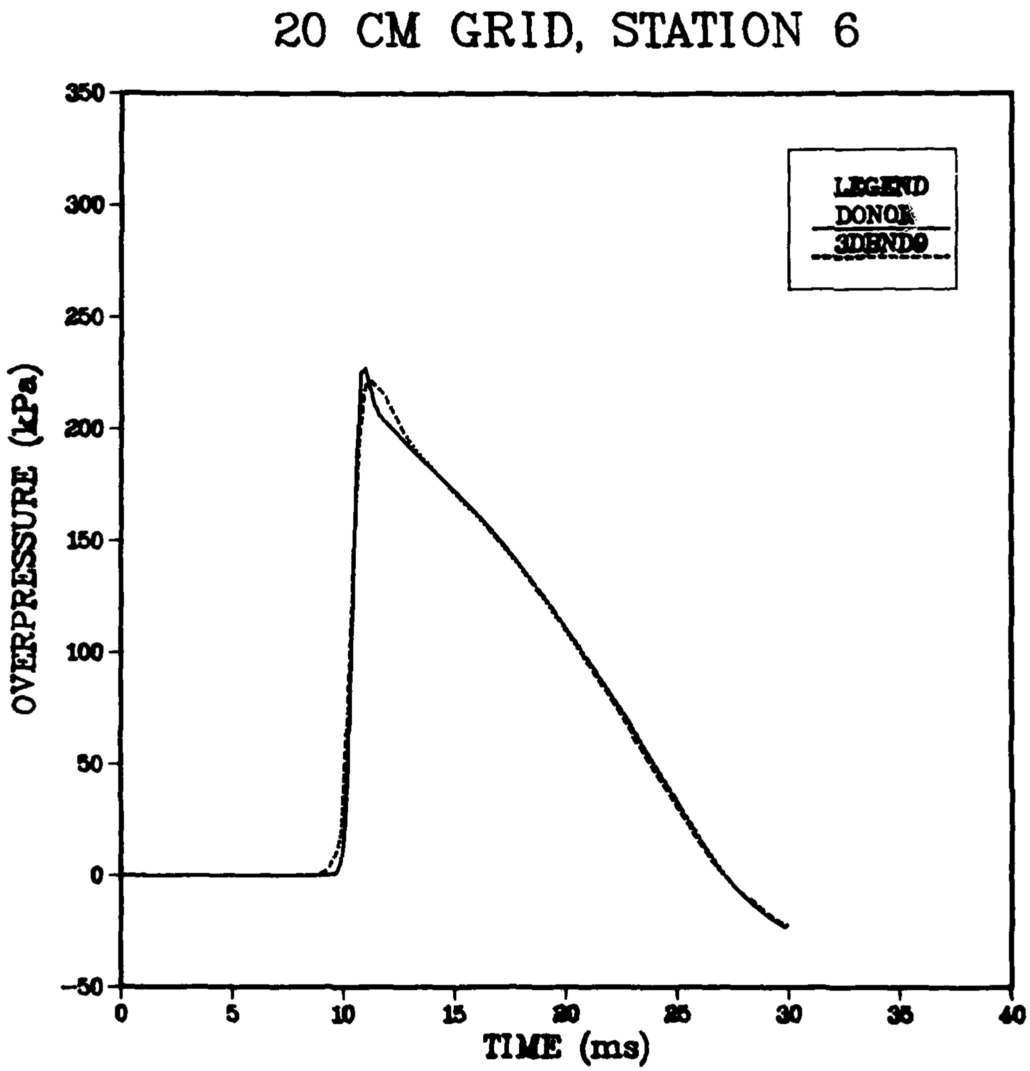


Figure 19. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 20 cm Mesh at Station 6.

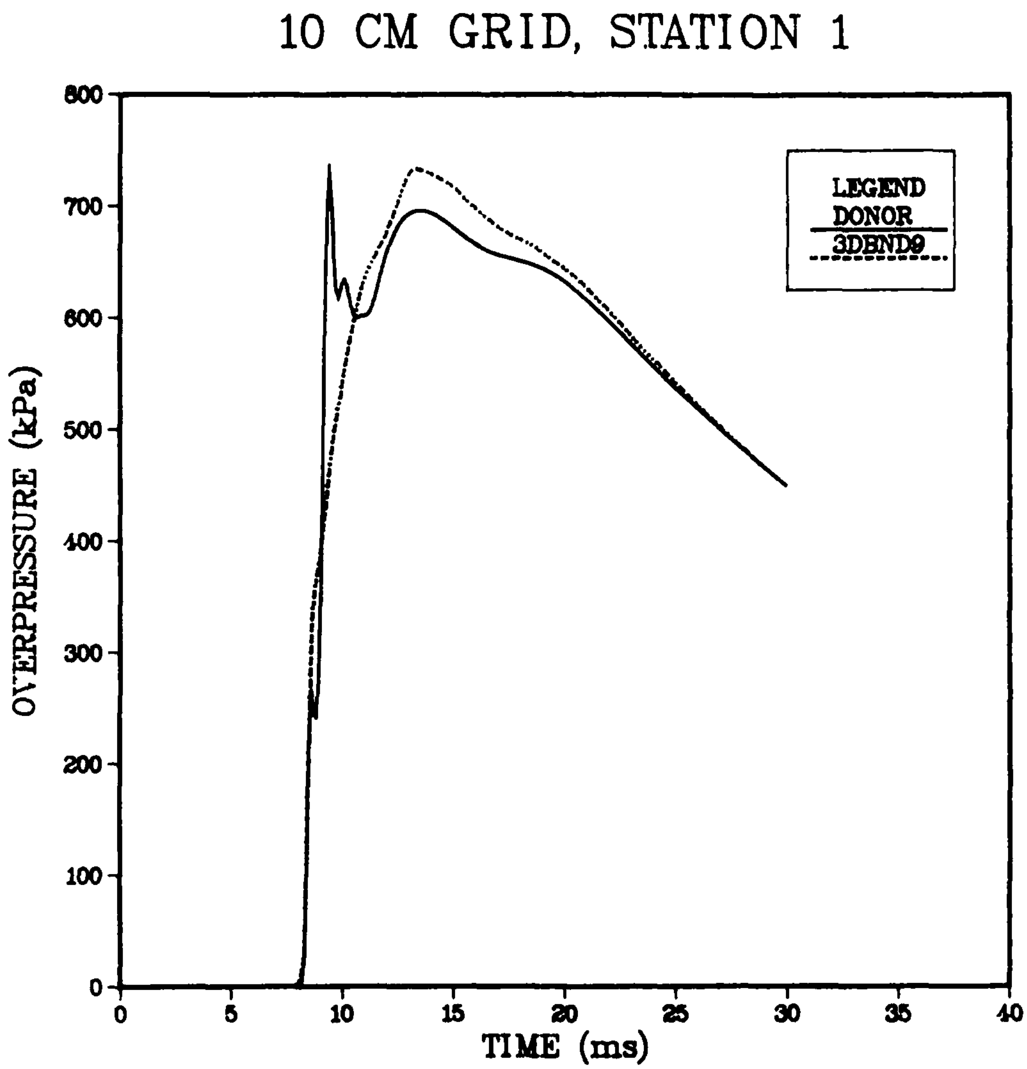


Figure 20. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 10 cm Mesh at Station 1.

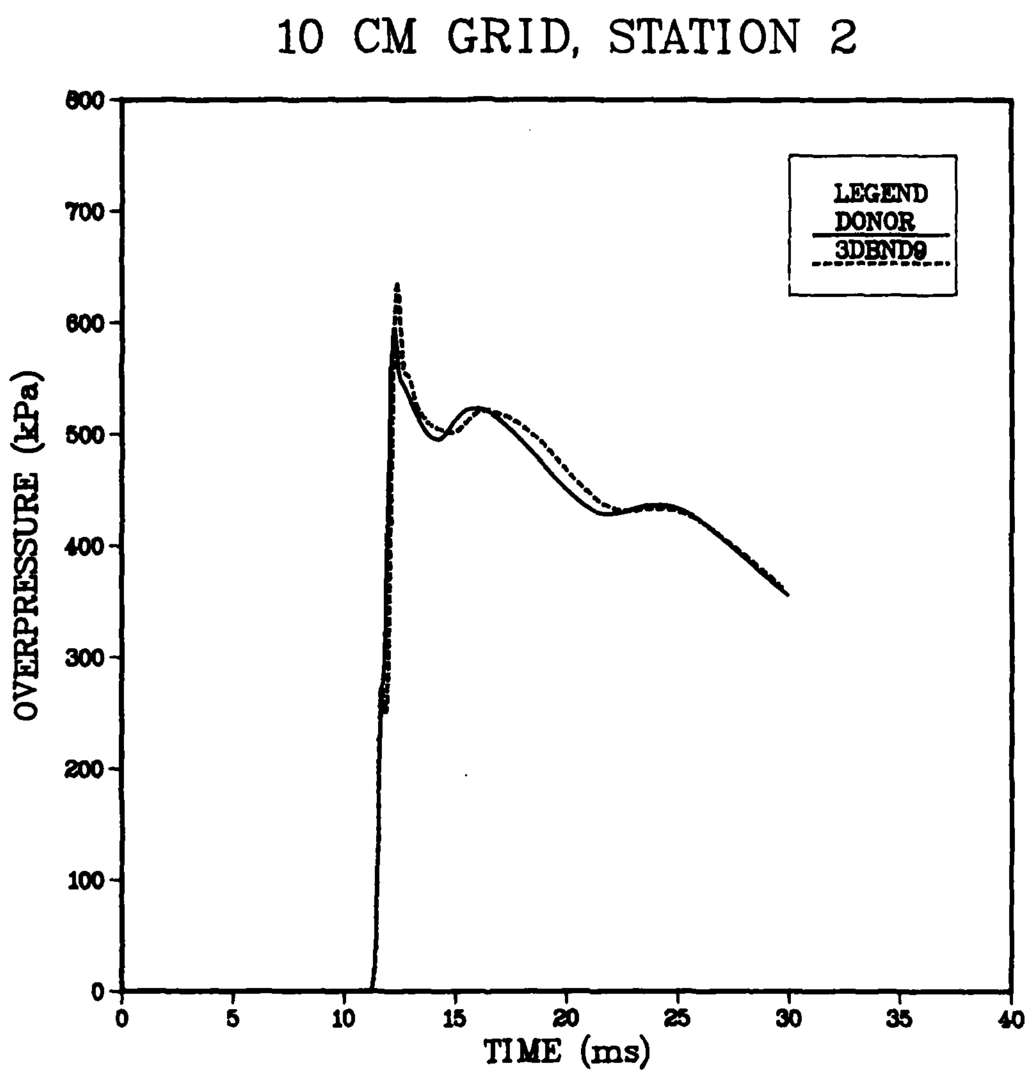


Figure 21. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 10 cm Mesh at Station 2.

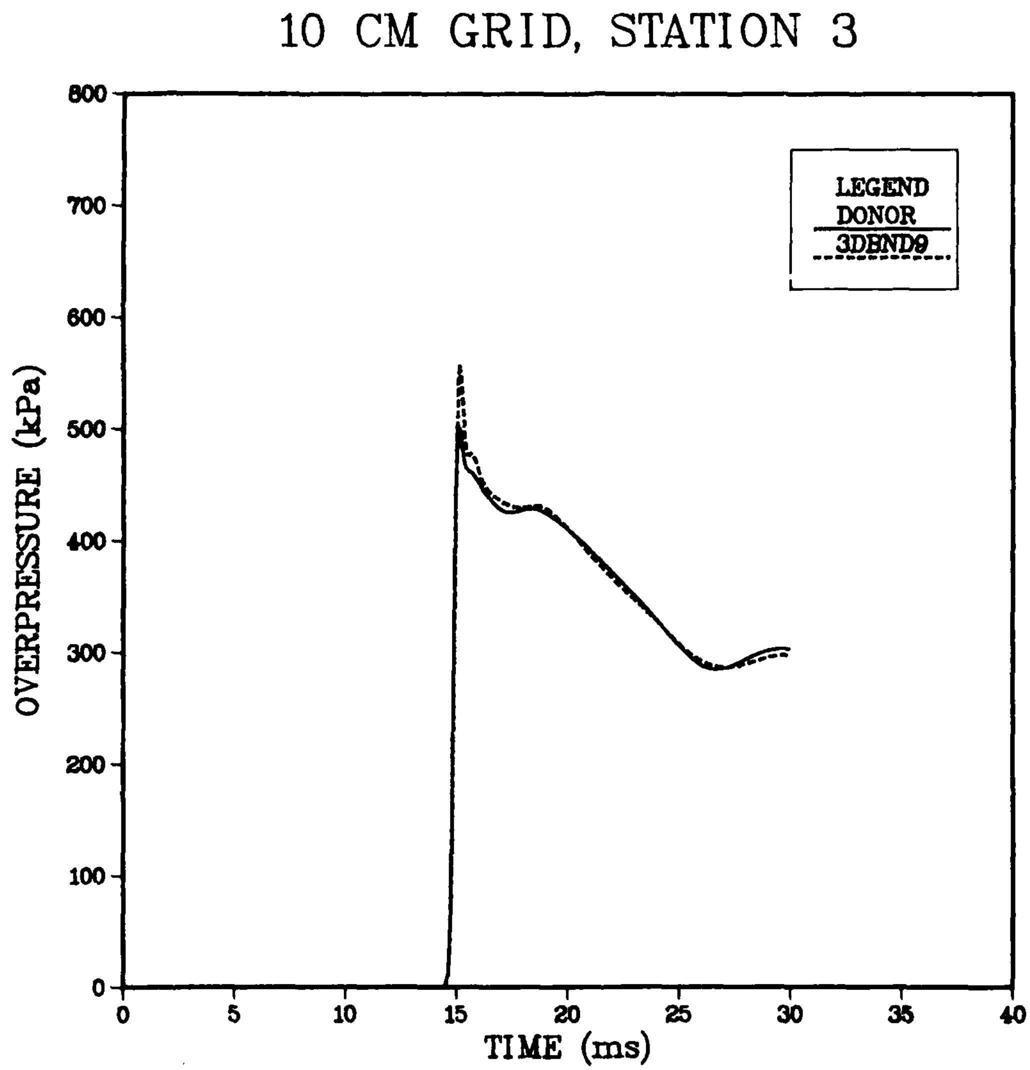


Figure 22. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 10 cm Mesh at Station 3.

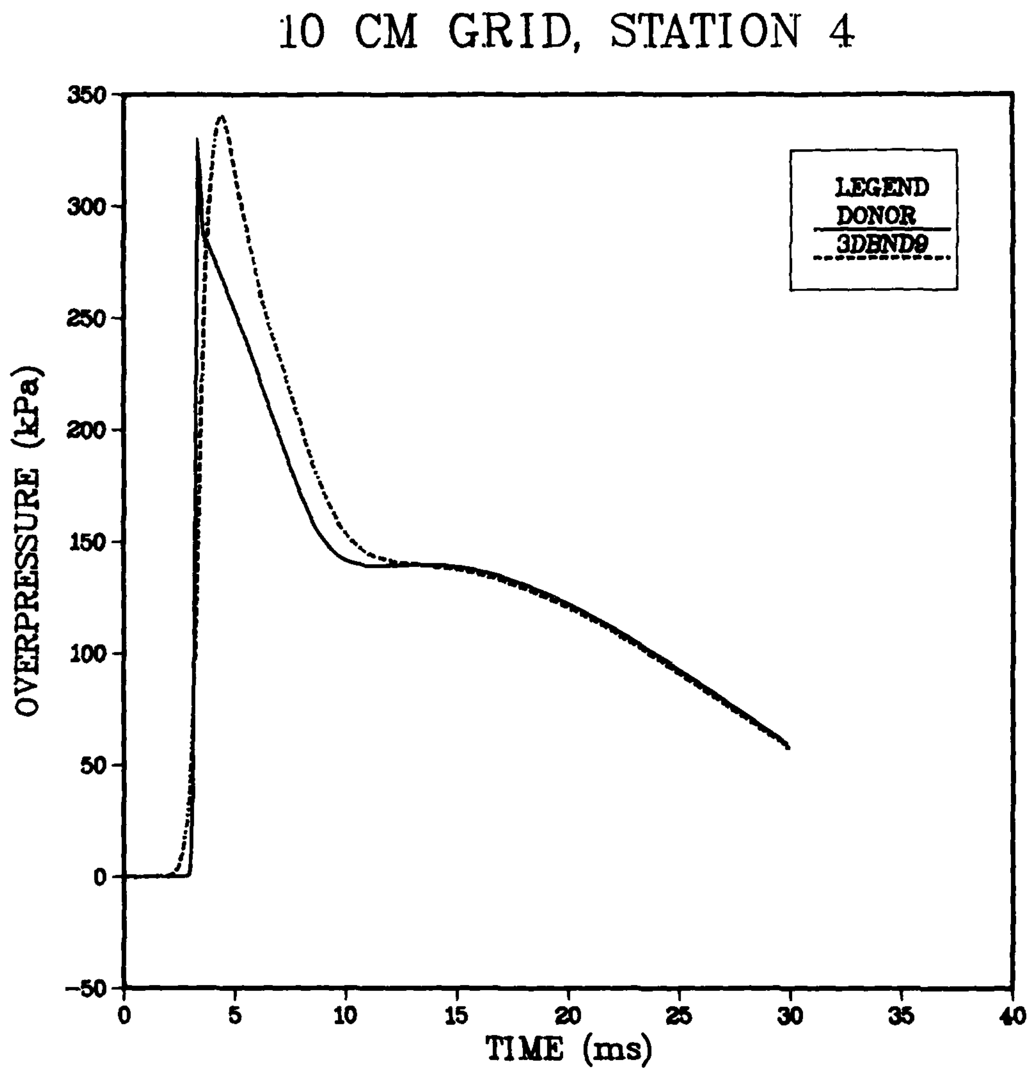


Figure 23. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 10 cm Mesh at Station 4.

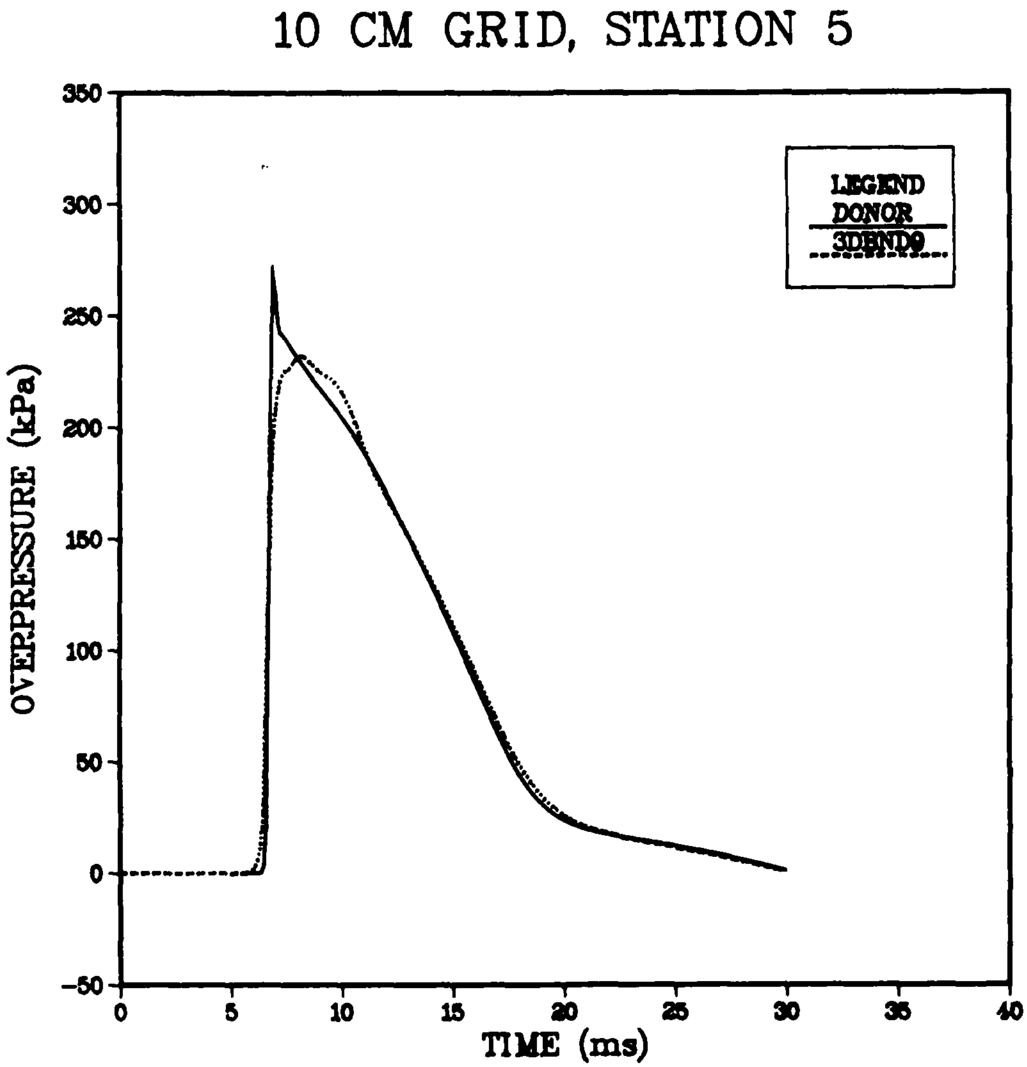


Figure 24. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 10 cm Mesh at Station 5 (Donor runs at Station 1).

10 CM GRID, STATION 6

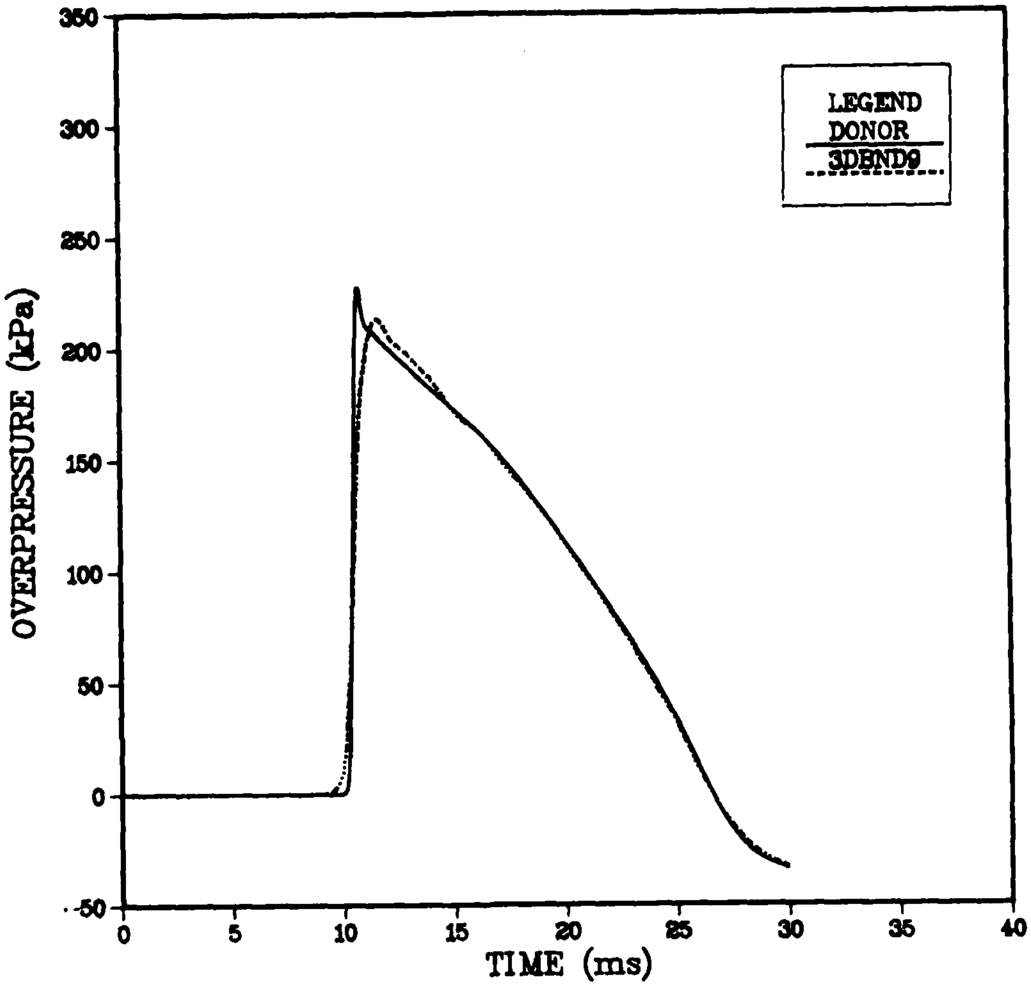


Figure 25. Comparison of Overpressure From a 2-D Donor Run and a 3-D Imposed-Boundary Run for a 10 cm Mesh at Station 6.

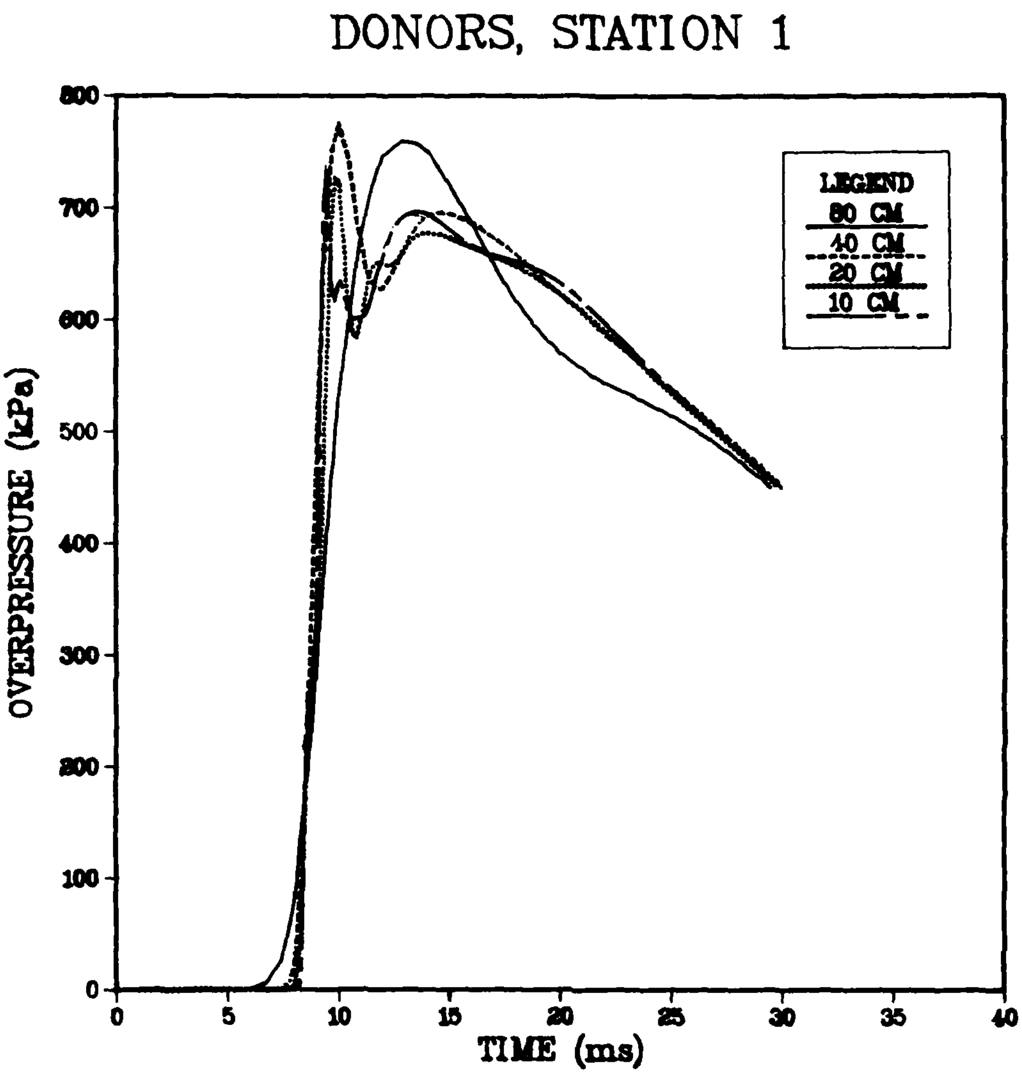


Figure 26. Overpressure From 2-D Donor Runs at Station 1.

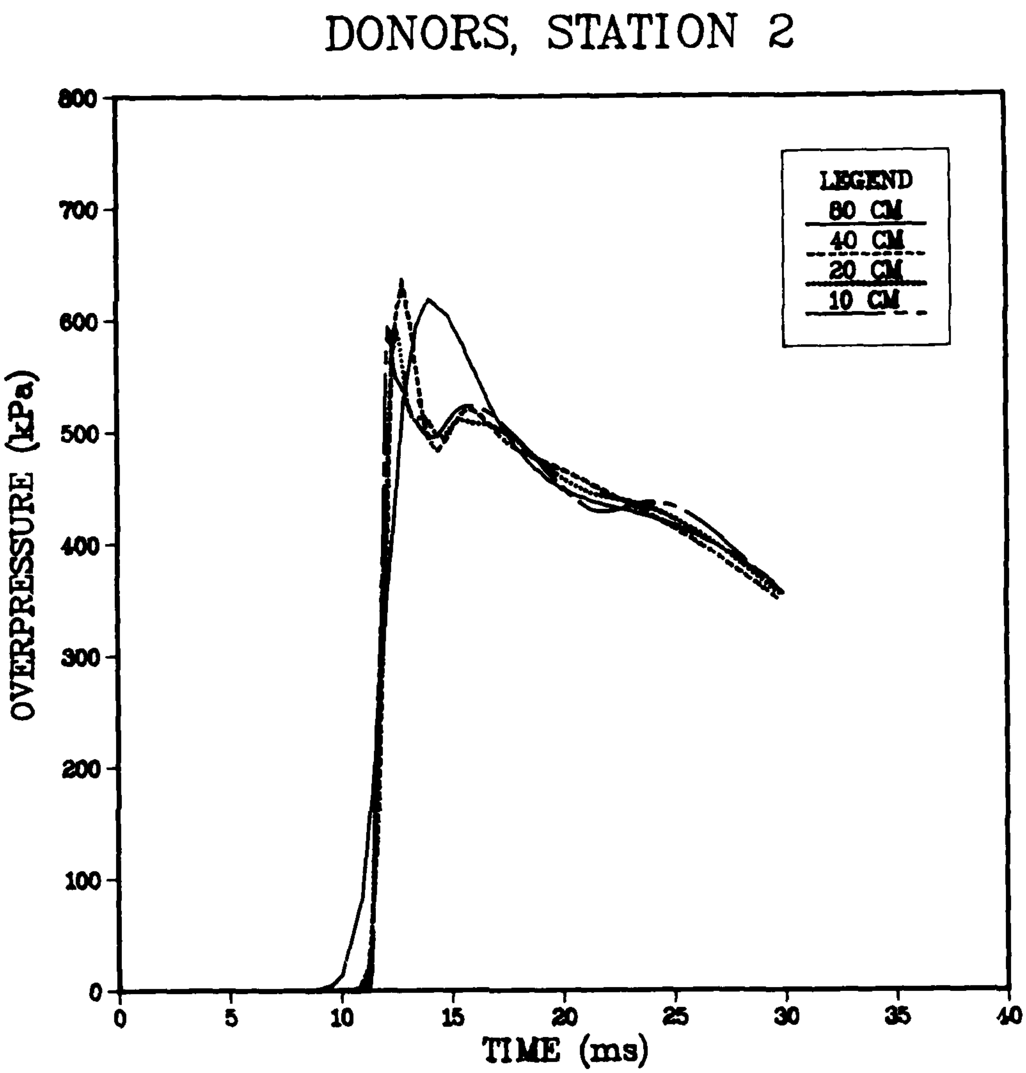


Figure 27. Overpressure From 2-D Donor Runs at Station 2.

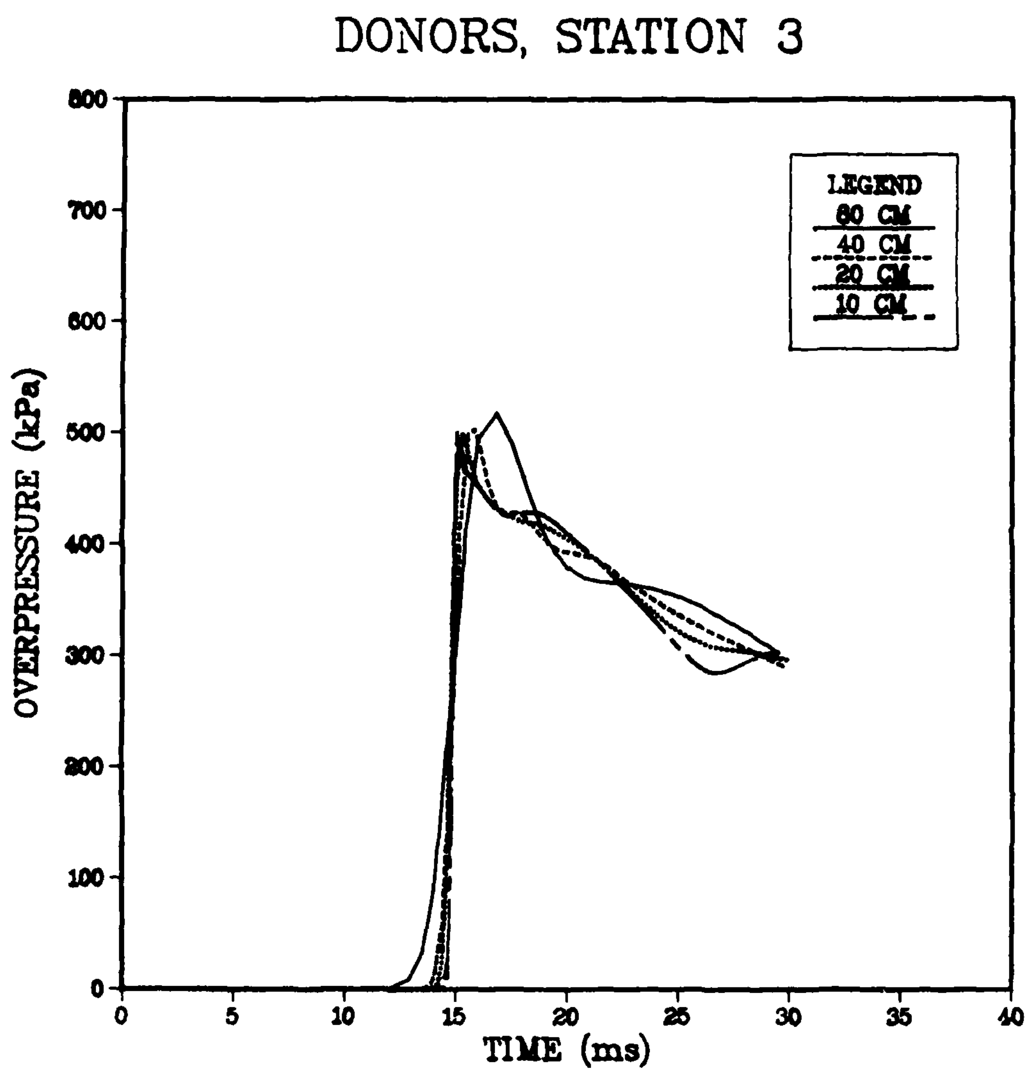


Figure 28. Overpressure From 2-D Donor Runs at Station 3.

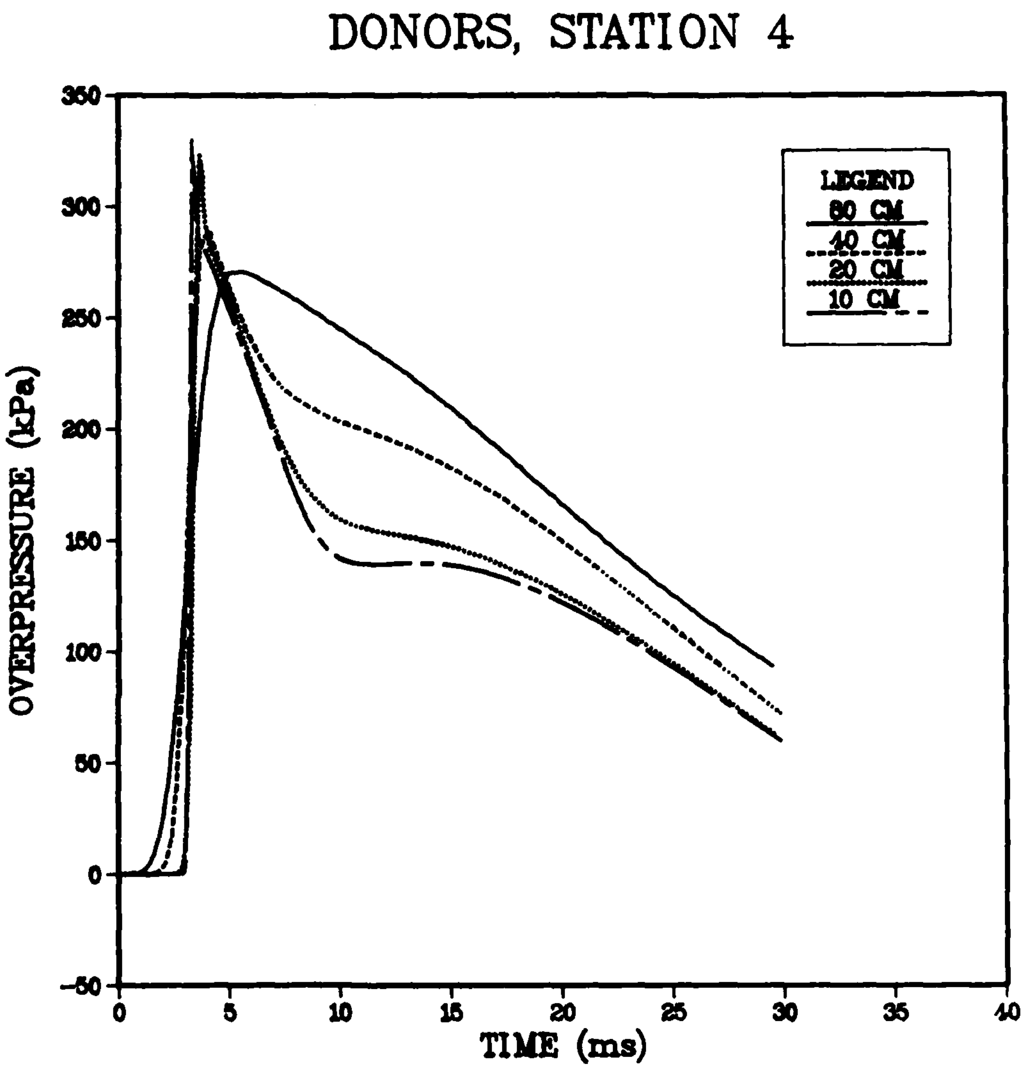


Figure 29. Overpressure From 2-D Donor Runs at Station 4.

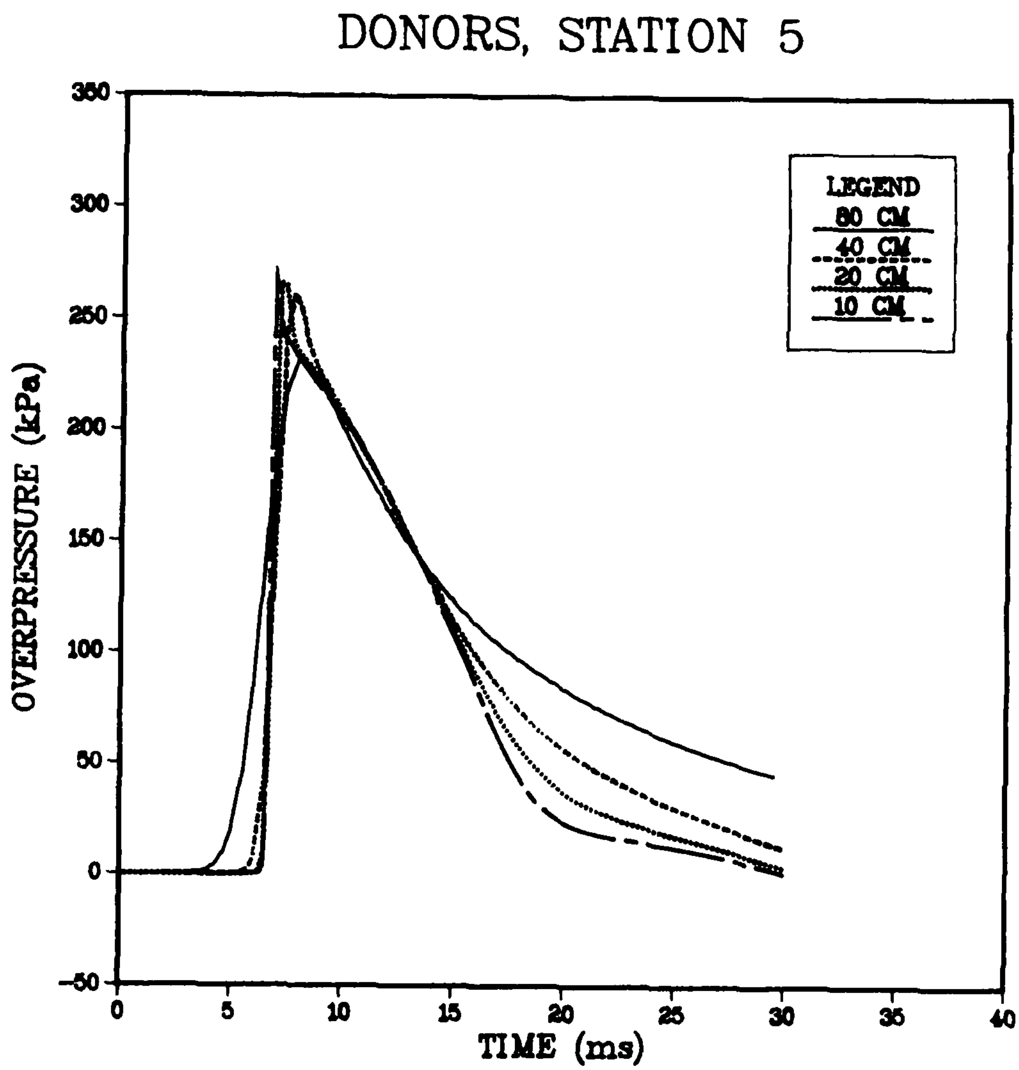


Figure 30. Overpressure From 2-D Donor Runs at Station 5.

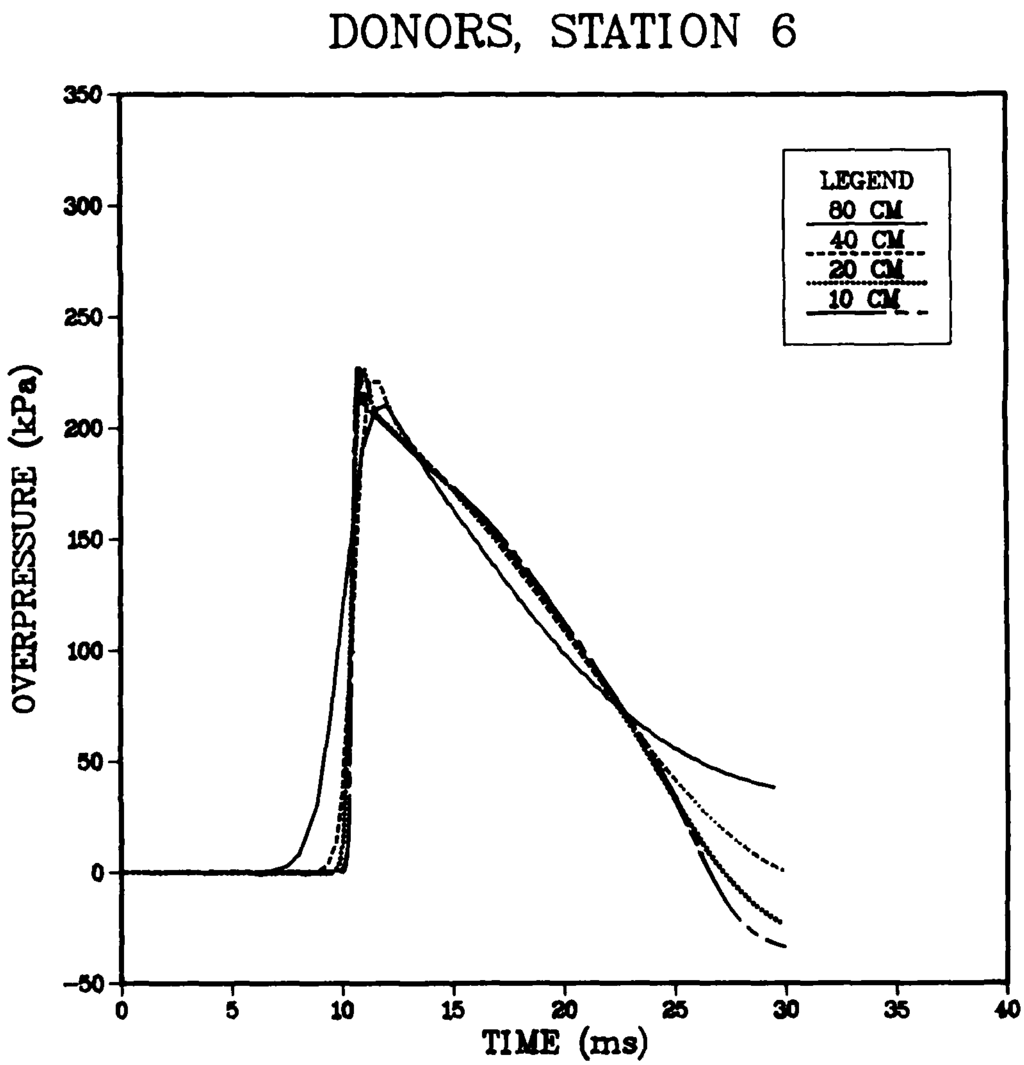


Figure 31. Overpressure From 2-D Donor Runs at Station 6.

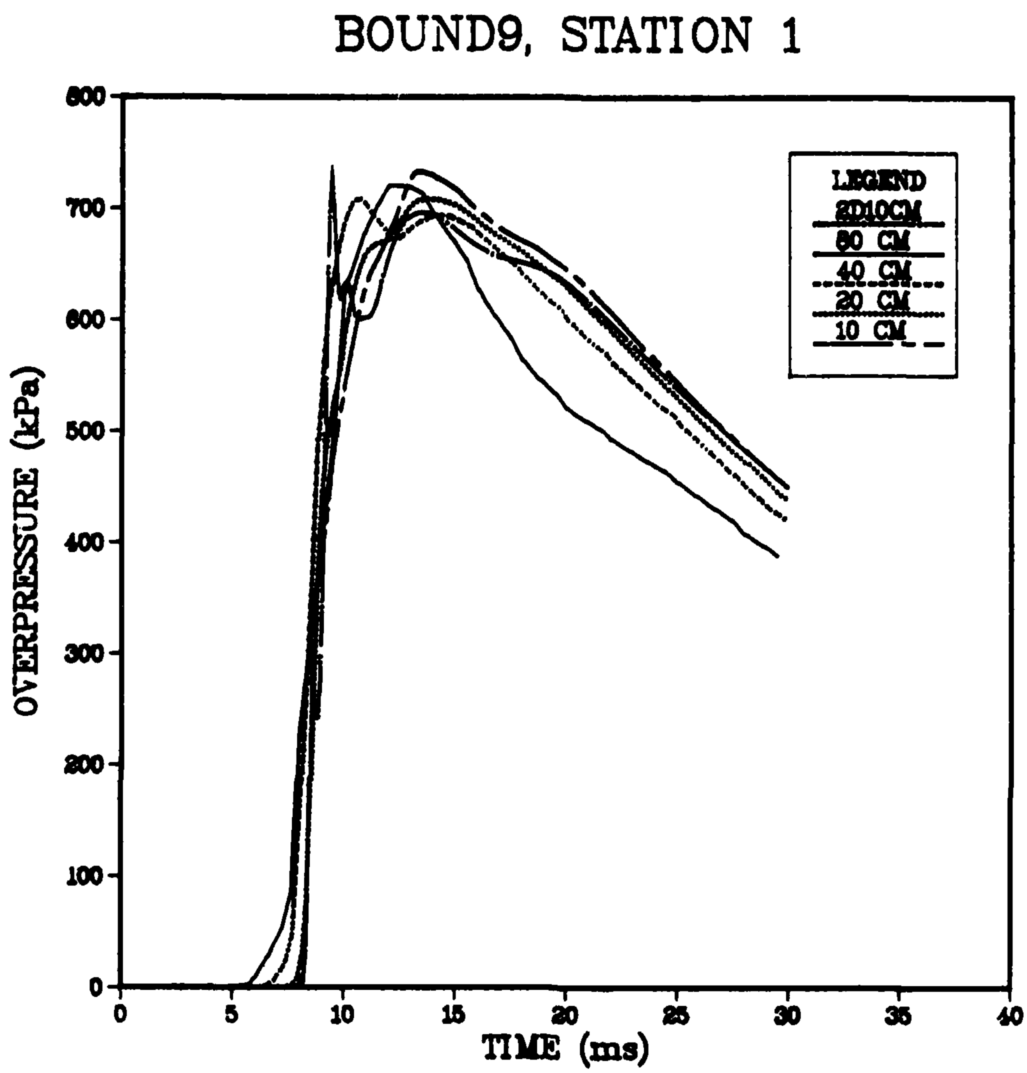


Figure 32. Overpressure From Imposed-Boundary Runs at Station 1.

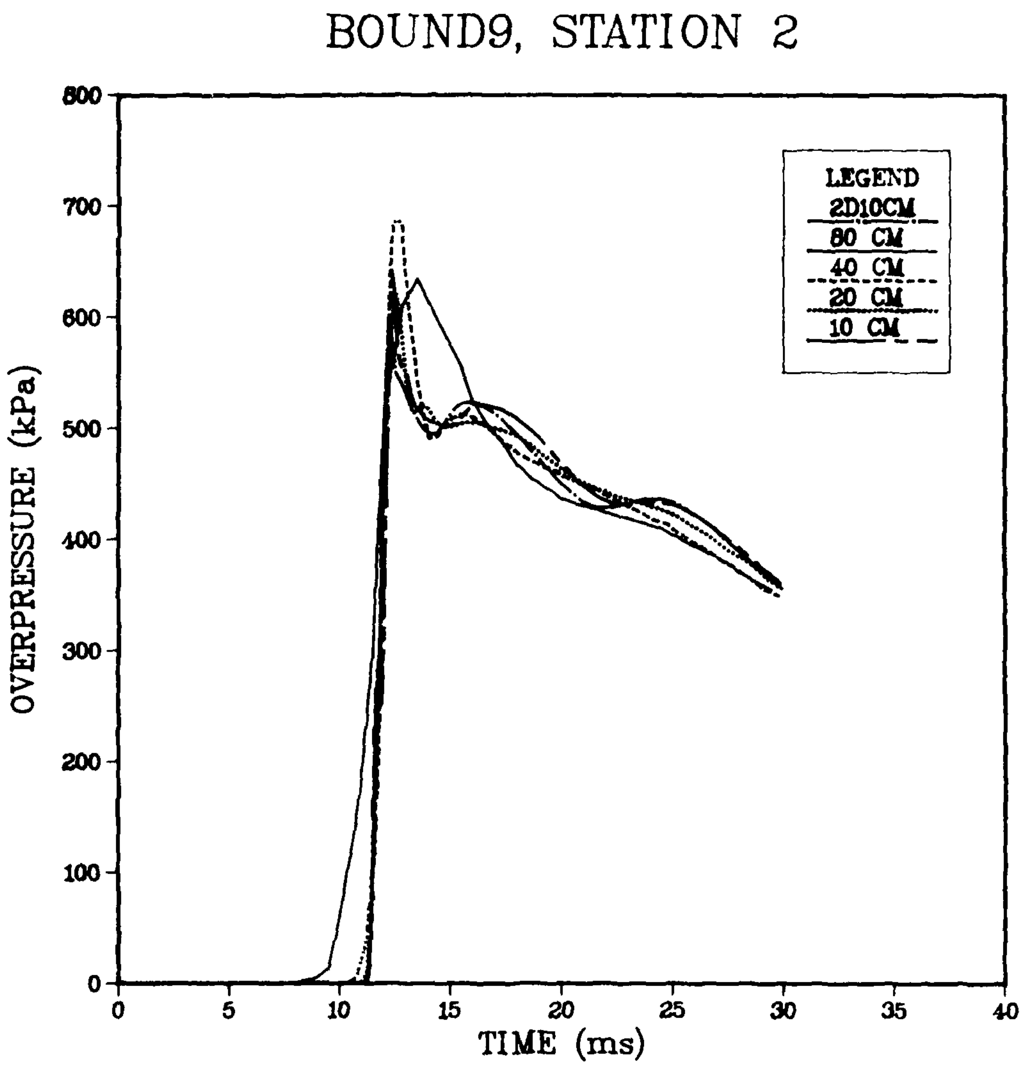


Figure 33. Overpressure From Imposed-Boundary Runs at Station 2.

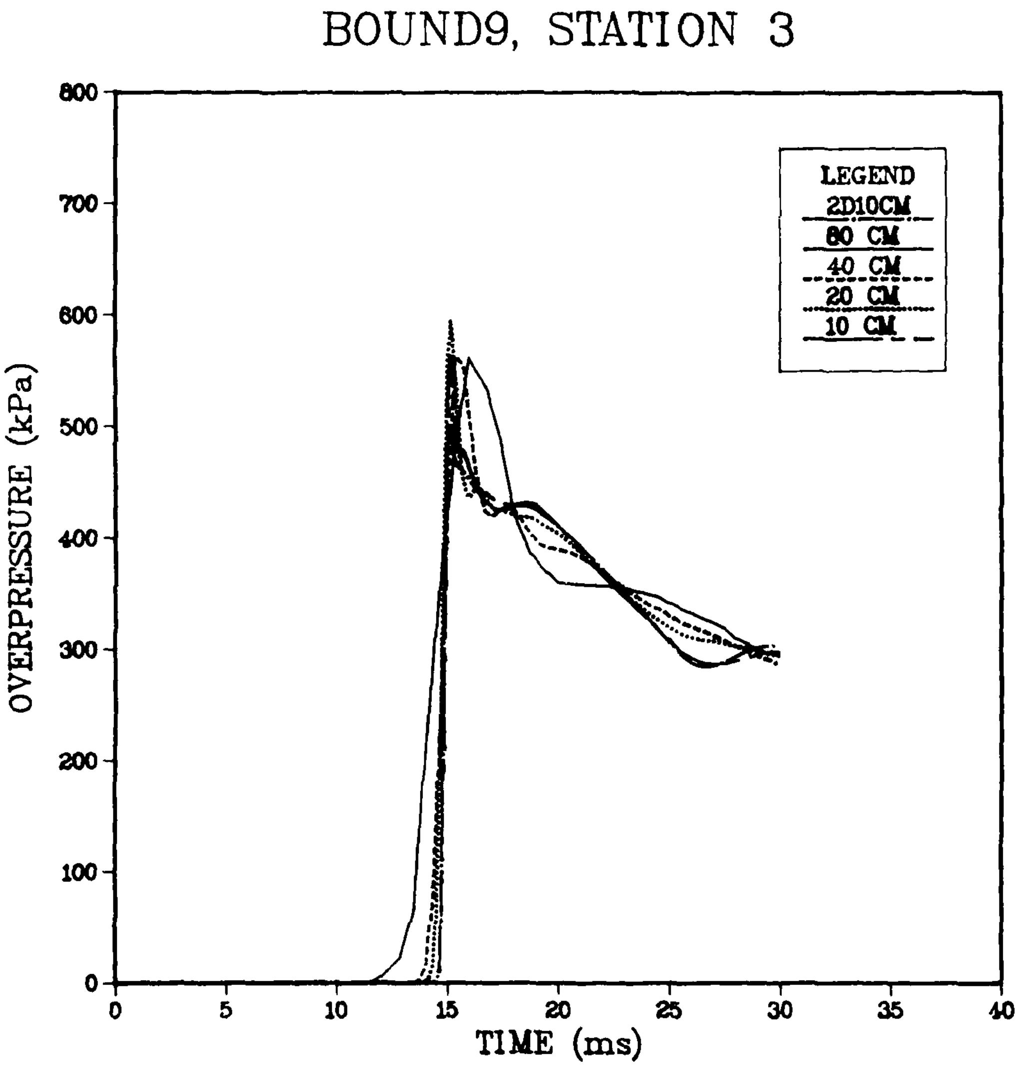


Figure 34. Overpressure From Imposed-Boundary Runs at Station 3.

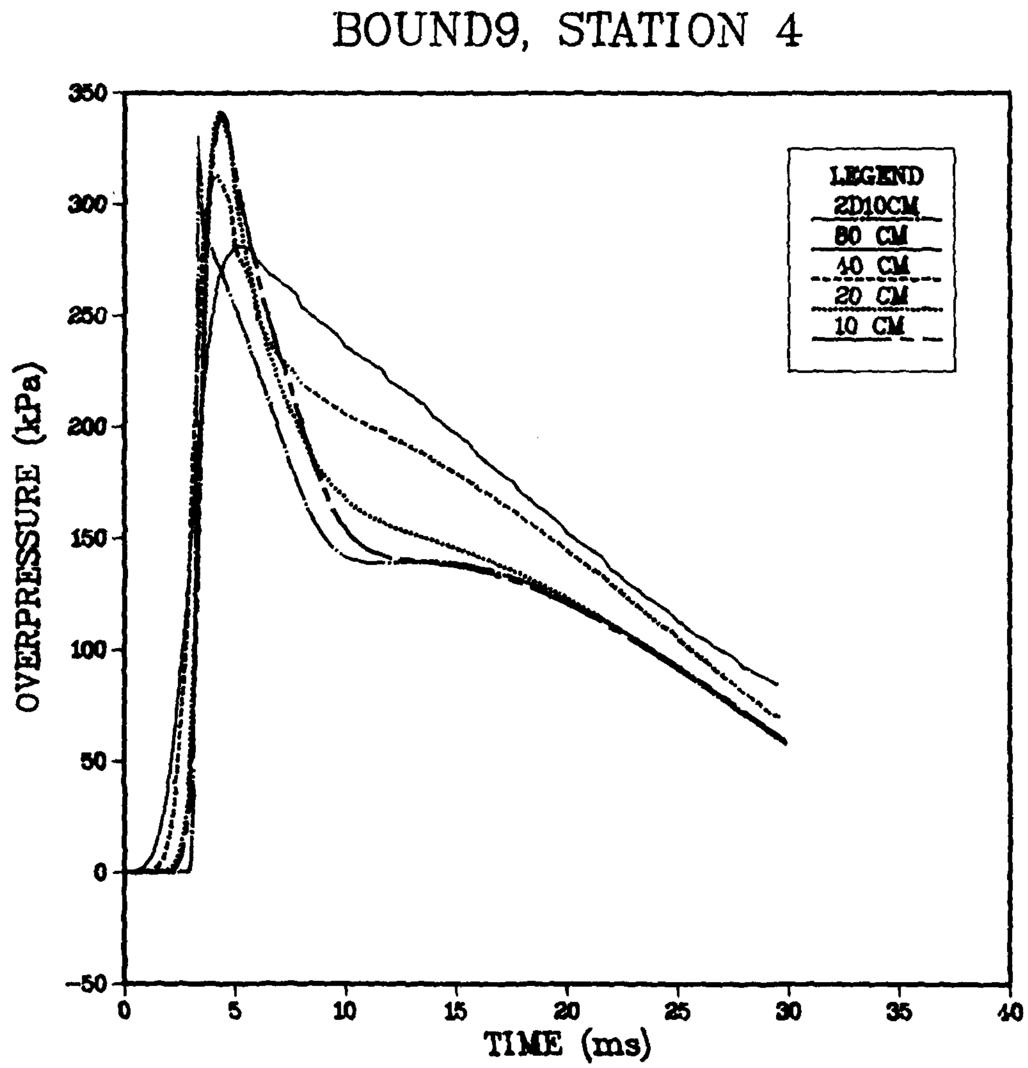


Figure 35. Overpressure From Imposed-Boundary Runs at Station 4.

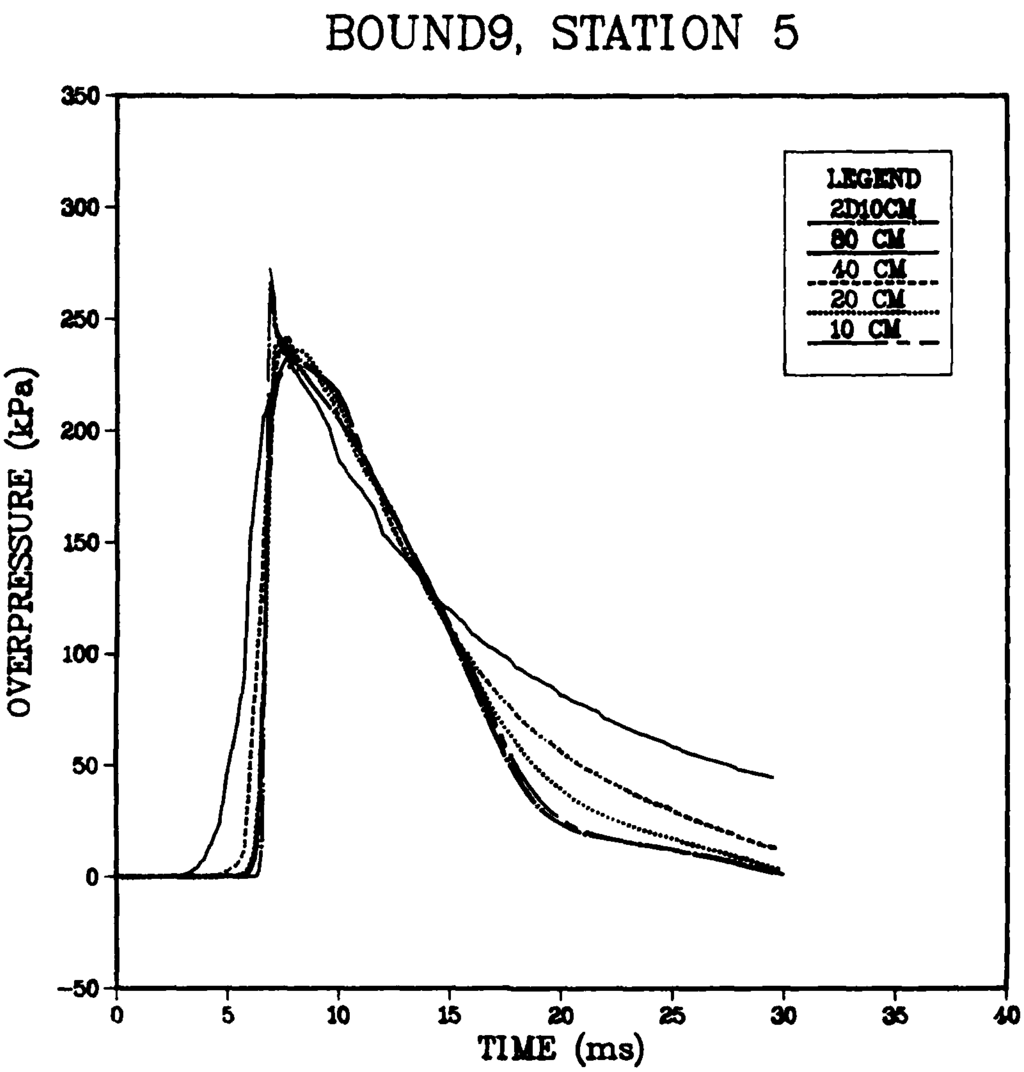


Figure 36. Overpressure From Imposed-Boundary Runs at Station 5.

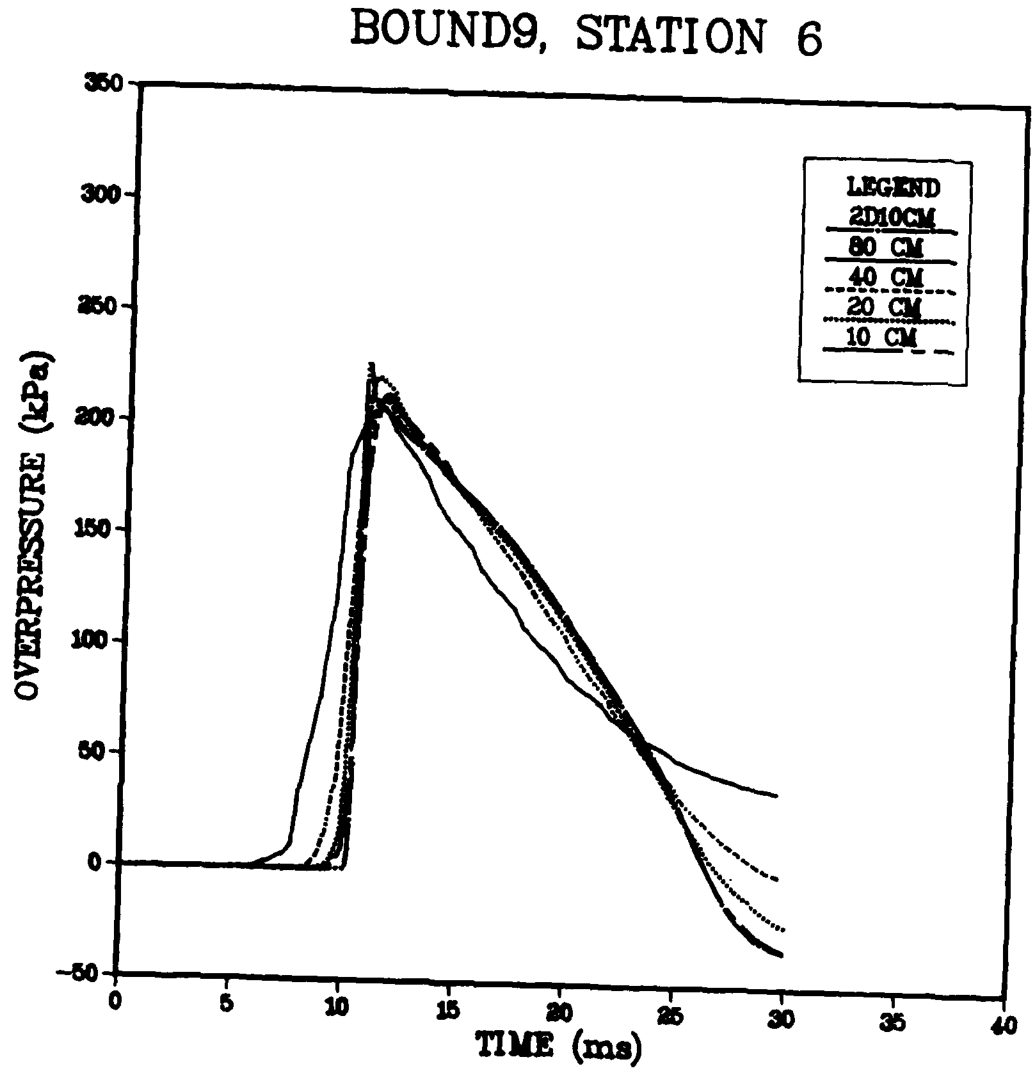


Figure 37. Overpressure From Imposed-Boundary Runs at Station 6.

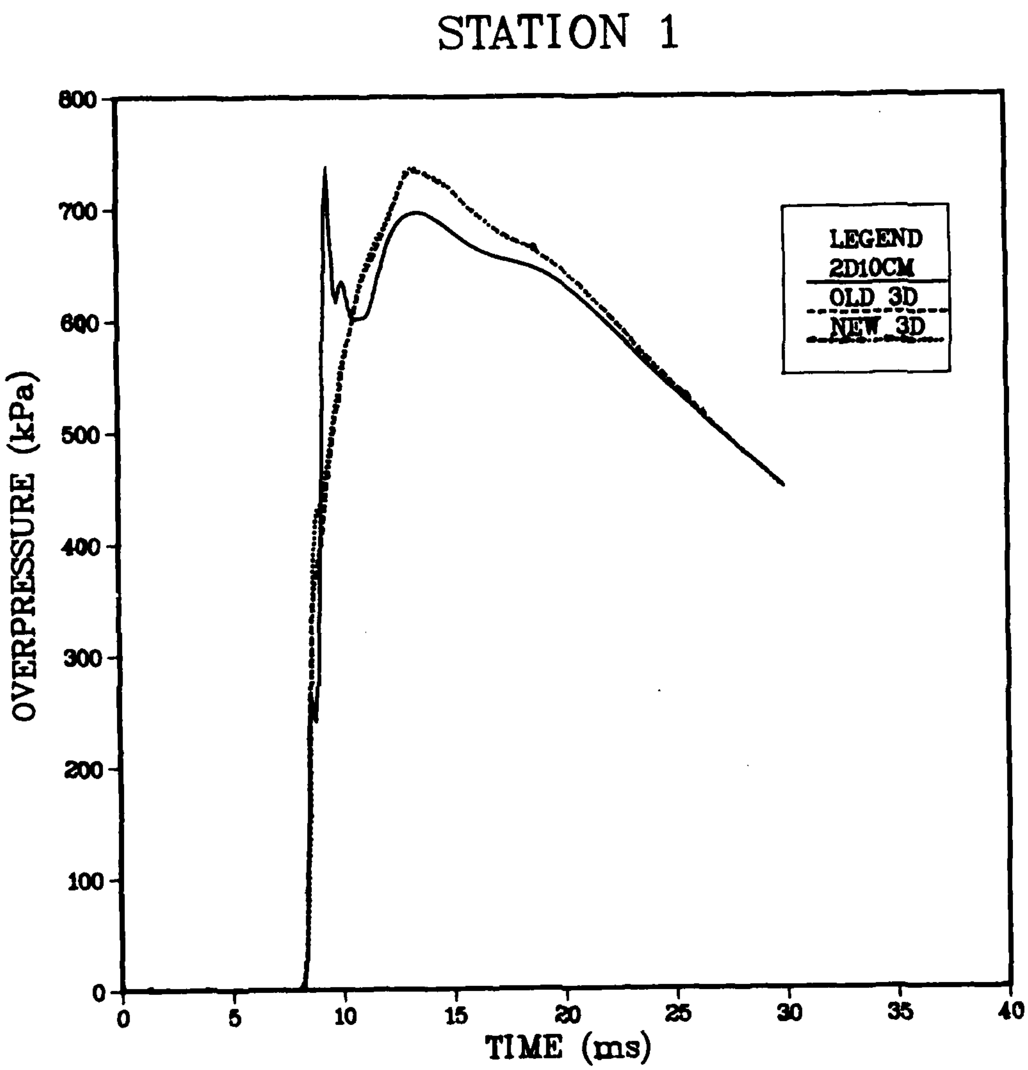


Figure 38. *Comparison of Overpressure From the Original and* the Revised BOUND9 Coding at Station 1.

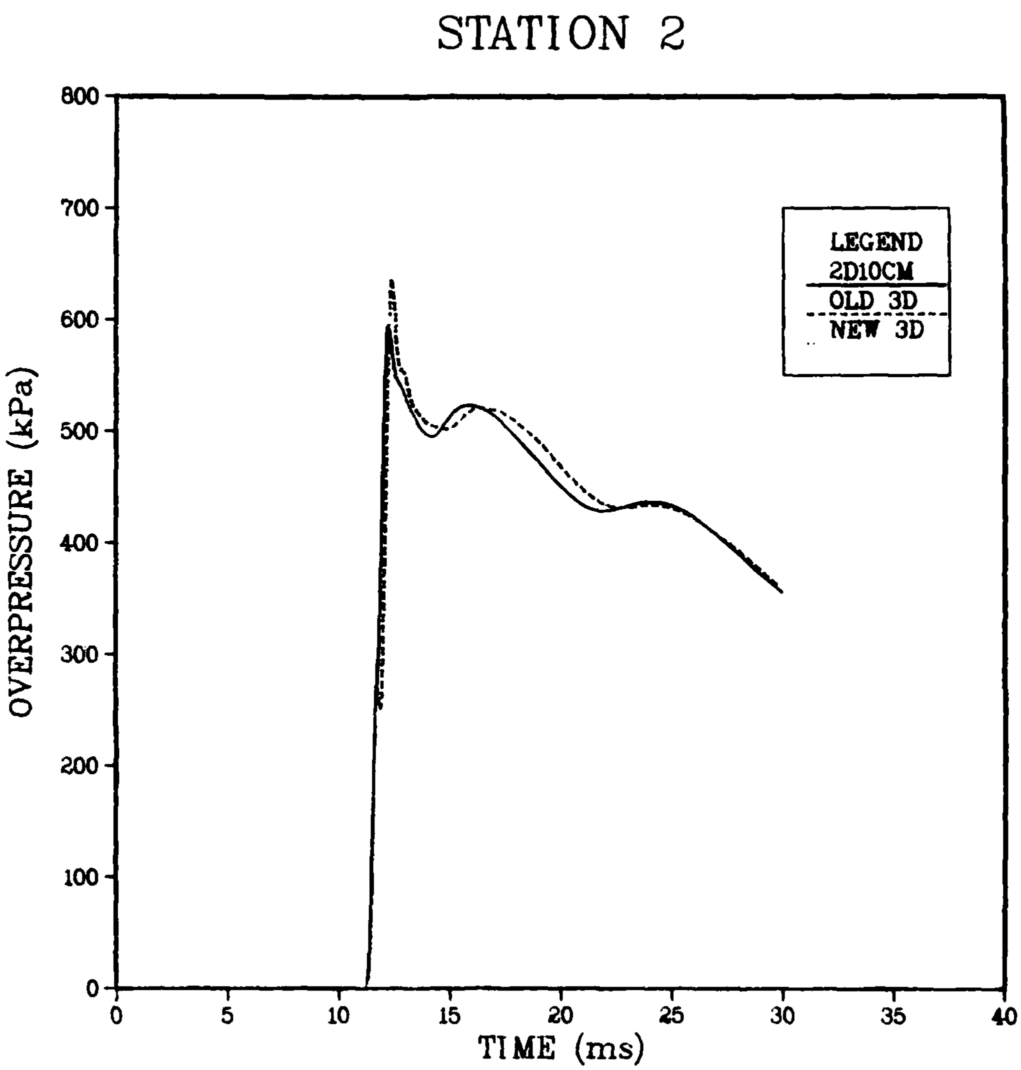


Figure 39. Comparison of Overpressure From the Original and the Revised BOUND9 Coding at Station 2.

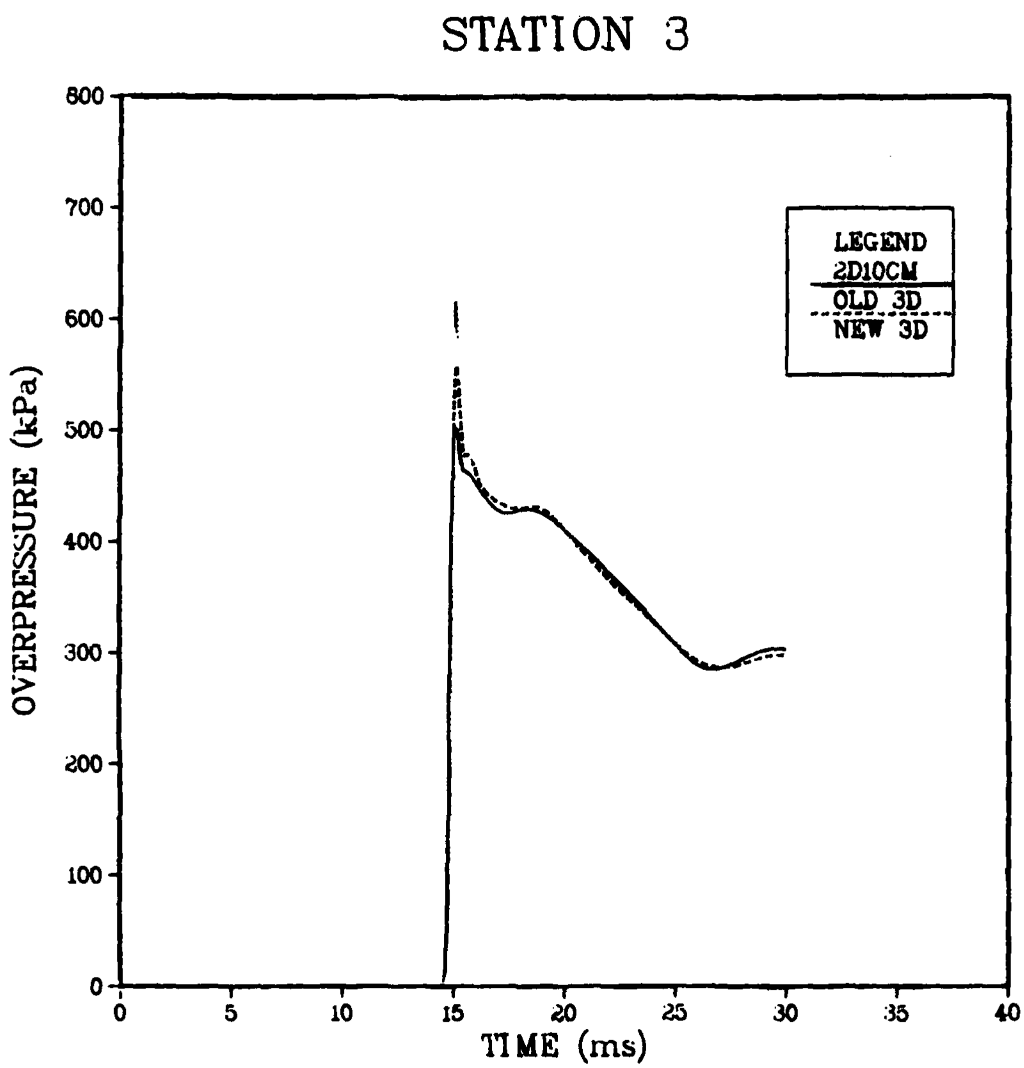


Figure 40. Comparison of Overpressure From the Original and the Revised BOUND9 Coding at Station 3.

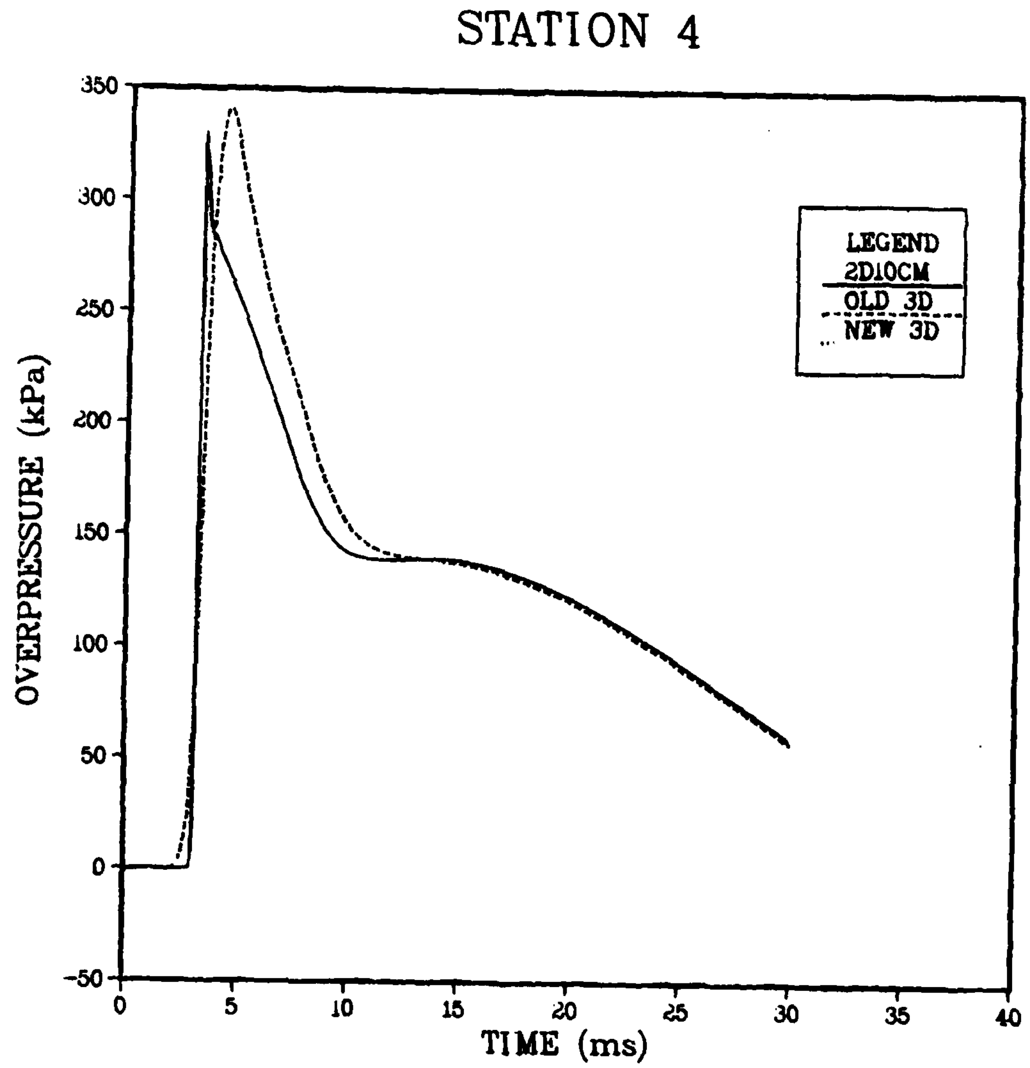


Figure 41. Comparison of Overpressure From the Original and the Revised BOUND9 Coding at Station 4.

REFERENCES

1. M. A. Fry, R. E. Durrett, G. P. Ganong, D. A. Matuska, M. D. Stucker, B. S. Chambers, C. E. Needham, and C. D. Westmoreland, "The HULL Hydrodynamics Computer Code," AFWL-TR-76-183, US Air Force Weapons Laboratory, Kirtland Air Force Base, NM, September 1976. (AD #B014070L)

2. J. A. Hasdal, B. S. Chambers, and R. W. Clemens, "Support to BRL: HULL Code Implementation on a CDC 7600," SAI-80-701-AQ, Science Applications Inc., McLean, VA, August 1979.

3. B. S. Chambers, and J. D. Wortman, "Two-Dimensional Shore (Partial Island) Cells for BRL HULL," ARBRL-CR-00497, US Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, December 1982. (AD #A123357)

4. R. E. Lottero, J. D. Wortman, B. P. Bertrand, and C. W. Kitchens, Jr., "Three-Dimensional Oblique Shock Diffraction Over a Rectangular Parallelepiped: Computational/Experimental Comparison," ARBRL-TR-02443, US Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, November 1982. (AD#A122254)

APPENDIX A

TABULATION OF PROGRAM HULLUP

Appendix A is a listing of a SCOPE 2 runstream to use HULLUP to prepare BOUND9 boundary input with a listing of file NUHULUP which contains the HULLUP program.

Problem 8405.09 is the "80 cm grid" donor file described in section V. Problem 8406.20 (or 8416.20) is the corresponding BOUND9 controlled 3-D test run. Restart output from problem 8405.09 is found in the MFZ file HULL8405P09.

Since data was wanted at all the available times on this restart file, the only input for namelist INDAT1 was the donor problem number.

Namelist NDAT1 and the 10 lines following it describe the locations of the boundaries for the 3-D run and the number of and locations of the data points on these sides. Since KNM was negative, the mesh points were specified (the 10 lines after namelist NDAT1). Equivalent output data, for this case, could have been formed by setting INM = 18, YNM = 14, and KNM = 16 in namelist NDAT1, with no additional input. This would give more mesh points, but the only effective ones would be those at the cell centers. This is possible only because the 3-D run has equal cell sizes.

Namelist NDAT2 informs HULLUP that BOUND9 boundary data is to be saved for X = XN1 and XN2 (the left and right boundaries), Y = YN2 (the fore boundary), and Z = ZN2 (the top boundary). Namelist TABDAT tells HULLUP the number of times that output is wanted for each of these boundaries.

** RUN STREAM FOR HULLUP **

```
WORTMAN(STMFZ,T350,P5)
ACCOUNT(******)  WORTMAN B309 X6028
COMMENT.   PROGRAM TO PRODUCE BOUNDARY INPUT FOR HULL FROM AN OLD HULL.
COMMENT.    SET FOR PROB 8416.20 FROM PROB 8405.09.  8/10/84
COMMENT.    THIS SHOULD REPRODUCE 8406.20 WITH THE NEW HULLUP
COMMENT.    THAT PRODUCES OUTPUT FILES WITH ROW BY ROW RECORDS.
COMMENT.
COMMENT.                  CHECKLIST FOR A NEW PROGRAM
COMMENT.     CHANGE IDENTIFICATION LINES, THE INPUT FILE FOR TAPE9,
COMMENT.        AND SET CATALOGGING OF OUTPUT FILES.
COMMENT.     ADJUST NAMELIST AND OTHER INPUT BELOW.
COMMENT.     CHECK PARAMETER STATEMENT VALUES, FRONT OF FILE NUHULUP.
COMMENT.
ATTACH(TAPE9,HULL8405P09,ID=JDW)
COMMENT.   REQUEST PERMANENT FILES FOR POSSIBLE BOUND9 OUTPUT.
REQUEST(TAPE11,*PF)
REQUEST(TAPE12,*PF)
REQUEST(TAPE13,*PF)
REQUEST(TAPE14,*PF)
REQUEST(TAPE15,*PF)
REQUEST(TAPE16,*PF)
COMMENT.   TEMPORARY SET UP FOR COMPILING HULLUP.
BEGIN(GETMFA,FILE,LF=HULLUP,PF=NUHULUP,UN=JDW)
UPDATE(N,I=HULLUP)
COMMENT.  FTN5(I,L=0,OPT=2)
FTN5(I,LO=S/A/R/M,OPT=2)
LGO(*PL=10000)
COMMENT.   CATALOG FILES OF BOUNDARY DATA HERE.
CATALOG(TAPE11,P841620LB,ID=JDW)
CATALOG(TAPE12,P841620RB,ID=JDW)
COMMENT.  CATALOG(TAPE13,P841620BB,ID=JDW)
CATALOG(TAPE14,P841620TB,ID=JDW)
COMMENT.  CATALOG(TAPE15,P841620AB,ID=JDW)
CATALOG(TAPE16,P841620FB,ID=JDW)
*EOR
 $INDAT1 PROBIN=8405.09  $
 $NDAT1 XN1=880.,XNM=1560.,INM=9,YN1=0.0,YNM=520.0,JNM=7,
  ZN1=0.0,ZNM=600.,KNM=-8$
 920.0
  80.0            8
                 -1
  40.0
  80.0            4
  80.0            2
                 -1
  40.0
  80.0            7
                 -1
 $NDAT2 NBND = 1,1,0,1,0,1$
 $TABDAT LASTPR = 5,4,0,4,0,3$
```

```
*/     FILE NUHULUP.  HULLUP FOR 3D BOUND9 INPUT FROM 2D DONOR.
*/
*/            FIRST,  THE 3 COMMON DECKS FOR SUBPROGRAMS IN HULLUP.
*/         THE FIRST 2 HAVE DONOR AND RECIPIENT ARRAY DIMENSIONS.
*/           PARAMETERS SET FOR PROB 8407.24 FROM 8407.22.
*/           THEY MAY BE CHANGED IF NECESSARY.
*/
*COMDECK CDIN1
C                 VARIABLE DIMENSIONS FOR DONOR ARRAYS.
      PARAMETER(IIBIG=257,IJBIG=123,IKBIG=2,ILBIG=5000)
      COMMON /CDIN1/ XI(IIBIG),DELXI(IIBIG),YI(IJBIG),DELYI(IJBIG)
     +      ,HYDROI(ILBIG)            ,ZI(IKBIG),DELZI(IKBIG)
*COMDECK CDNU1
C                 VARIABLE DIMENSIONS FOR RECIPIENT ARRAYS
      PARAMETER(NIJKMAX=200,NLBIG=1000)
      COMMON /CDNU1/ XN(NIJKMAX),YN(NIJKMAX),ZN(NIJKMAX),BND(NLBIG),
     +   BND2(NLBIG)
*COMDECK CDCOM
C                   OTHER COMMON TERMS FOR HULLUP.
      COMMON/CDIN3/ IN9,PROBIN,CSTART,IIMAX,IIMAX1,IJMAX,IJMAX1,
     +   IKMAX,IKMAX1,IGEOM,INH,IROWPB,INHPB,
     +   ITYPE,ITIME,CYCLEI,TIMEIN
      COMMON/CDNU3/ XN1,XNM,YN1,YNM,ZN1,ZNM,INM,JNM,KNM,TSH,XSH,YSH,ZSH,
     +   NNH,NBND(6),NFOUT(6),XYZB(6),ICONV,NXPL,NYPL,NZPL
*DECK  HULLUP
      PROGRAM HULLUP(INPUT,OUTPUT,TAPE9,TAPE11,TAPE12,TAPE13,TAPE14,
     +  TAPE15,TAPE16,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2)
C                               * * * * *
C        HULLUP EXTRACTS BOUNDARY INPUT VALUES FOR A NEW HULL RUN FROM
C        A HULL RESTART FILE.  THE OLD HULL RUN IS CALLED DONOR OR OLD.
C        PARAMETERS FOR OR FROM IT MAY HAVE PREFIX OR SUFFIX I OR IN .
C        THE NEW HULL RUN WILL BE CALLED RECIPIENT OR NEW.  PARAMETERS
C        FOR IT MAY HAVE PREFIX OR SUFFIX N OR NU.
C
C        AN ATTEMPT HAS BEEN MADE TO ALLOW FOR FUTURE CHANGES.  THIS
C        FIRST CODING WILL BE FOR A CYLINDRICAL SYMMETRIC DONOR AND
C        PREPARE BOUNDARY DATA FOR A 3D CARTESIAN RECIPIENT.
C                 WE HAVE MADE OTHER ASSUMPTIONS.
C        THE RESTART DATA IS ON ONE FILE, TAPE9.  (WE WILL SET IN9=9
C          AND USE IN9 TO MAKE POSSIBLE CHANGE SIMPLER.)
C        THE COORDINATE POINTS OF DATA ON IN9 MAY CHANGE BETWEEN DUMPS.
C        WE ASSUME NO ISLAND OR SHORE CELLS ON NEW BOUNDARY PLANES.
C        NNH=5 FOR 3D.  (IE 5 HYDRO VARIABLES OUTPUT)
C        FOR CYLINDRICAL DONOR, RADIAL VELOCITY AT R = 0.0 SET TO 0.0.
C                               * * * * *
C               WE HAVE VARIABLE PARAMETERS AND ARRAYS IN COMDECKS.
C           VARIABLE DONOR PARAMETERS IN CDIN1.
C           VARIABLE RECIPIENT PARAMETERS IN CDNU1.
C                               * * * * *
C                                 INPUT
C                               * * * * *
C        INPUT IS THRU NAMELIST INPUT, OR DATA, ON TAPE5 AND FROM IN9.
C
C** 1    NAMELIST /INDAT1/       (THIS IS READ IN HULLUP, I.E. MAIN)
C
C     IN9 -    FILE FOR OLD HULL RUN RESTART DUMPS.  DEFAULT = 9.
C                 THIRD FILE ON PROGRAM CARD. MUST ATTACH IN RUN STREAM.
C
C     PROBIN - OLD PROBLEM NUMBER ON INPUT FILE.  REQUIRED.
C
C     CSTART - STARTING CYCLE NUMBER OLD DATA.  DEFAULT = 0.0.
C
```

```
C      CSTOP -  UPPER BOUND FOR INPUT CYCLES.  WILL PROCESS INPUT
C               CYCLES UNTIL CYCLEI GT CSTOP OR INPUT FILE ENDS.
C               DEFAULT = 9999999.
C
C    IN9 HAS A VARIABLE NUMBER OF HULL RESTART DUMPS.  EACH
C    DUMP IS A HEADER, A Z-BLOCK, A VERTEX RECORD, AND HYDRO DATA.
C       THE HEADER RECORD IS 4 NUMBERS.
C     1 - SIGNAL.  555.0 FOR A RESTART DUMP,  666.0 FOR END OF FILE.
C     2 - DONOR PROGRAM NUMBER.  THIS MUST BE THE SAME AS PROBIN.
C     3 - CYCLE NUMBER.
C     4 - TIME.
C       THE Z-BLOCK IS A 200 WORD RECORD, 92 NAMES AND VALUES, AND ID.
C       WE WILL EXTRACT 7 VALUES.
C     IIMAX  - THE NUMBER OF CELLS IN THE X DIRECTION.
C     IJMAX  - THE NUMBER OF CELLS IN THE Y DIRECTION.
C     IKMAX  - THE NUMBER OF CELLS IN THE Z DIRECTION.
C     INH    - NUMBER OF HYDRO VARIABLES/CELL.  WE USE ONLY 4.
C     IDIMEN - INPUT PROBLEM DIMENSION.  (WE ASSUME ONLY 2D NOW)
C     IGEOM  - INPUT GEOMETRY.  1-CARTESIAN, 2-CYLINDRICAL ASSUMED.
C     IROWPB - ROWS STORED/BLOCK.  (EACH BLOCK IS A RECORD).
C       VERTEX RECORD
C     2D  - (XI(I),I=2,IIMAX+1),(YI(J),J=1,IJMAX+1).  XI(1)=0.0 ASSUMED.
C     3D  - (XI(I),I=1,IIMAX+1),(YI(J),J=1,IJMAX+1),(ZI(K),K=1,IKMAX+1).
C       HYDRO VALUES.  INH*IIMAX*IROWPB VALUES PER RECORD.
C       FOR EACH CELL IN 2D, THE FIRST 5 HYDRO VALUES ARE.
C    1 - PRESSURE THAT IS NOT USEFUL.  IT IS FOR ANOTHER TIME.
C    2 - X VELOCITY COMPONENT (RADIAL VELOCITY FOR CYLINDRICAL). (CM/SEC)
C    3 - Y VELOCITY COMPONENT (AXIAL VELOCITY FOR CYLINDRICAL). (CM/SEC)
C    4 - SPECIFIC ENERGY.  (ERGS/GM)
C    5 - MASS IN THE CELL.  (GM)
C
C** 2        NAMELIST /NDAT1/           (READ IN STRTNU)
C
C     THIS DESCRIBES THE NEW BOUND9 PLANES.  SOME EXPLANATION IS NEEDED.
C     FOR LOW INDEX BOUND9 BOUNDARIES, HULL NEEDS VALUES CENTERED ON
C     THE OUTSIDE EDGE OF EACH BOUNDARY CELL.  FOR HIGH INDEX BOUND9
C     BOUNDARIES, VALUES ARE NEEDED IN THE CENTER OF THE EXTERNAL CELLS.
C     HULL WILL INTERPOLATE IN THE SPACE DEFINED ON A PLANE.  IT WILL
C     NOT EXTRAPOLATE.  ACCURACY WILL BE BEST IF THE CELL CENTERS FOR
C     THE NEW HULL GRID ARE SPECIFICALLY INCLUDED IN THE BOUND9 MESH.
C     XN1,XNM,YN1,YNM,ZN1,ZNM ARE THE LOCATIONS OF BOUNDARY PLANES.
C     (XN(I),I=1,INM),(YN(J),J=1,JNM),(ZN(K),K=1,KNM) ARE THE MESHES
C     ON THE BOUNDARY PLANES.
C       2D DONOR CELL VERTICES ARE (XI(I),I=1,IMAX+1),(YI(J),J=1,JMAX+1)
C         X0 = XI(1) IS 0.0.  XI(IMAX+1) AND YI(IMAX+1) EXTERNAL.
C       THE HYDRO IN EXTERNAL CELLS IS NOT USEABLE.
C            WE HAVE THE FOLLOWING RESTRICTIONS FOR A 3D RECIPIENT.
C       FOR XMAX=XI(IMAX), YMAX=YI(JMAX), X0=XI(1), AND Y0=YI(1),
C       (SQRT(XN(I)**2+YN(J)**2) LT XMAX AND Y0 LE ZN(K) LE YMAX.
C       ALL POINTS ON AN OUTPUT PLANE MUST BE INCLUDED INSIDE THE SPACE
C       DEFINED BY THE 2D DONOR GRID.
C         MAKING 2 OR 3 SEPERATE RUNS MAY BE MORE CONVENIENT.
C
C                   - - NAMELIST TERMS  - -
C      XN1 - LEFT BOUNDARY.  DEFAULT =0.0.
C      XNM - RIGHT EXTERNAL BOUNDARY CELL CENTER.  REQUIRED.
C      YN1 - 2D BOTTOM BOUNDARY, 3D AFT BOUNDARY.  DEFAULT = 0.0.
C      YNM - 2D TOP, 3D FORE, EXTERNAL BOUNDARY CELL CENTER.  REQUIRED.
C      ZN1 - 3D BOTTOM BOUNDARY.  DEFAULT = 0.0.
C      ZNM - 3D TOP EXTERNAL BOUNDARY CELL CENTER.  REQUIRED FOR 3D.
C
C      INM - NUMBER OF X POINTS IN OUTPUT ARRAYS.  DEFAULT = 200.
C      JNM - NUMBER OF Y POINTS IN OUTPUT ARRAYS.  DEFAULT = 200.
C      KNM - NUMBER OF Z POINTS IN OUTPUT ARRAYS.  DEFAULT = 200.
```

```
C
C     TSH - SHIFT IN TIME.  TIMENU = TIMEIN - TSH.  DEFAULT = 0.0.
C     XSH - X SHIFT.  XNU = XIN - XSH.  DEFAULT = 0.0
C     YSH - Y SHIFT.  YNU = YIN - ZSH.  DEFAULT = 0.0
C     ZSH - Z SHIFT.  ZNU = XIN - XSH.  DEFAULT = 0.0
C                   - - - - - - -
C
C           THE OUTPUT MESH CAN BE DEFINED IN 2 MUTUALLY EXCLUSIVE WAYS.
C  (1) IF INM>0, JNM>0, AND KNM>0, THEN
C      XN(I) = XN1 + (I-1)*(XNM-XN1)/(INM-1)  FOR I=1,INM
C      YN(J) = YN1 + (J-1)*(YNM-YN1)/(JNM-1)  FOR J=1,JNM
C      ZN(K) = ZN1 + (K-1)*(ZNM-ZN1)/(KNM-1)  FOR K=1,KNM
C
C  (2) IF ANY OF INM, JNM, OR KNM < 0, THEY ARE SET POSITIVE AND
C      THE FOLLOWING LINES ARE READ WITH FORMAT (E15.8,I5):
C      XN(1)
C      DX1,N1 --     FROM WHICH XN(I) = XN(I-1)+DX1 FOR I=2 TO N1+1
C      DX2,N2 --     FROM WHICH XN(I) = XN(I-1)+DX2 FOR I=N1+1 TO N1+N2+1.
C      . . . . . . .
C      DXL,NL --     FOR XN(I) UP TO I=1+N1+N2+...NL (=? INM).
C      ANY,-1 --      TERMINATOR FOR X GRID.
C           MUST HAVE 1+N1+N2+...+NL = INM.
C
C       FOLLOWED BY SIMILAR INPUT FOR YN(J) AND THEN ZN(K).
C
C             THEN,
C
C** 3  NAMELIST /NDAT2/   MORE INPUT FOR NEW PROBLEM.  (READ IN STRTNU)
C
C     NGEOM  - NEW GEOMETRY.  2 IS CYLINDRICAL, 1 IS CARTESIAN.
C                    DEFAULT IS 1 (CARTESIAN)
C
C     NDIMEN - OUTPUT DIMENSIONS.  2 OR 3.  DEFAULT = 3.
C
C     NNH    - NUMBER OF OUTPUT HYDRO VARIABLES PER POINT.
C              DEFAULT = 5.
C
C     (NBND(I),I=1,6)  -  SIGNAL FOR BOUNDARY I.
C              FIND OUTPUT FOR BOUNDARY I IF NBND(I) = 1.  DEFAULT = 0.
C              BOUNDARIES IN ORDER ARE LEFT,RIGHT,BOTTOM,TOP,AFT,FORE.
C                (AT LEAST ONE NBND(I) = 1, OR A NULL REQUEST.)
C
C     (NFOUT(I),I=1,6) - FILE TO STORE NBND(I) RESULTS IN.
C              DEFAULT = I + 10.
C            THESE ARE THE 4TH THRU 9TH FILE ON PROGRAM LINE.
C            THOSE WITH NBND(I) = 1 TO BE CATALOGED IN THE RUNSTREAM.
C
C** 4  NAMELIST/TABDAT/ CONTROLS TAB OF BOUND9 FILES.  (READ IN REWRIT)
C
C     (INITPR(I),I=1,6) - FIRST DUMP OF NFOUT(I) TO TAB.  DEFAULT=1.
C
C     (LASTPR(I),I=1,6) - LAST TIME DUMP OF NFOUT(I) TO TAB.  DEFAULT=0.
C
C     (INITRO(I),I=1,6) - FIRST ROW OF NFOUT(I) TO TAB.  DEFAULT = 1.
C
C     (LASTRO(I),I=1,6) - LAST ROW OF NFOUT(I) TO TAB.  DEFAULT = 10.
C          - - - - - - - - - - - - - - - - - - - - - - - - -
C
C              * * *  SUBROUTINES  * * *
C      MOST OF THE PROGRAM IS IN THE MAIN PROGRAM CALLED HULLUP.
C      THERE ARE NOW (8/08/84) SIX SUBROUTINES.
C
C     STRTIN - INITIATION FOR READING THE INPUT FILE (IN9).
C         READ HEADER RECORD FROM IN9.  CHECK THAT OLD PROGRAM NUMBER
```

```
C           IS PROBIN.  READ THE FIRST Z-BLOCK, EXTRACT AND COMPUTE TERMS.
C           CHECK THAT INPUT ARRAY MAXIMA, (IIBIG,IJBIG,IKBIG,ILBIG), ARE
C           LARGE ENOUGH.
C
C       STRTNU - INITIATION FOR THE NEW HULL.
C           READ NAMELIST INPUTS NDAT1 AND NDAT2.  SET UP OUTPUT POINTS
C           AND PARAMETERS.  CHECK THAT OUTPUT ARRAY MAXIMA, (NIJKMAX
C           AND NLBIG), ARE BIG ENOUGH.
C
C       NXTIM(IEND) - LOCATES NEXT TIME DUMP ON IN9.  (2D ONLY)
C           IF ITIME = 0, SEARCH FOR CYCLEI=CSTART.  READ DUMP HEADER.
C           BYPASS THE Z-BLOCK.  READ 2D VERTICES AND SET UP VOLUME TERMS
C           AND MIDCELL VALUES.  SET IEND TO 0, 1, OR -1.  0 FOR HYDRO
C           DATA AVAILABLE, 1 FOR NORMAL FILE END, -1 FOR READ TROUBLE.
C
C       FINDI(I,Z,Y,IM)    FIND I FOR LINEAR INTERPOLATION IN Y.
C           I=1 IF Z < Y(2), I=IM-1 IF Z > Y(IM-1), OR Y(I) < Z <= Y(I+1).
C
C       NXTBLK(JP,IHB) - GET HYDRO INPUT DATA.  (2D INPUT ONLY)
C           IHB IS THE LAST HYDRO BLOCK READ, JP IS ROW OF DATA NEEDED.
C           THE PROGRAM CHECKS FOR BLOCK NEEDED.  IT IS PUT IN HYDROI(--).
C
C       WRITPL(IBND,NF,X,XS,Y,NY,YS,Z,NZ,ZS,B,NL) - TEMPORARY DUMP TO NF.
C            ROW DUMP FOR A BOUNDARY PLANE.  IF FIRST ENTRANCE FOR THIS
C            TIME, DUMP HEADER AND GRID.  AFTER NZ ENTRANCES FOR FILE NF
C            THERE ARE 2+NZ RECORDS ON FILE NF.
C           1 - HEADER OF 7 TERMS.
C               555.,PROBIN,TIMIN,X,NNH,NY,NZ
C           2 - LOCATION OF POINTS IN THE PLANE.
C               (Y(J),J=1,NY),(Z(K),K=1,NZ)
C           2+NZ - HYDRO DATA FOR THE FOR KTH ROW IN PLANE, (K=1,NZ).
C               (BND(L),L=1,NL)  WHERE NL = NNH*NY.
C          ***   AFTER THE FINAL TIME DUMP A HEADER IS DUMPED, ALL 666.
C
C      REWRIT(IBND,NF)  -   REPLACES CONTENT OF FILE NF WITH FINAL OUTPUT.
C             THE CONTENTS OF NF ARE PUT INTO 2 FILES THEN COLLATED
C           AND PUT BACK ON NF WITH 2 SUCCESSIVE TIMES AT EACH DUMP.
C           AGAIN THERE ARE 2 + NZ RECORDS PER TIME DUMP.
C           EXCEPT FOR 2 SUCCESSIVE TIMES, TI1,TI2, TERMS AS IN WRITPL.
C         1 - HEADER OF 8 TERMS.
C             555.0,PROBIN,TI1,TI2,X,NNH,NY,NZ
C         2 - LOCATION OF POINTS IN THE PLANE
C             (YN(J),J=1,NY),(ZN(K),K=1,NZ)
C         2+NZ - HYDRO DATA AT TI1 AND THEN TI2 FOR ROW K, (K=1,NZ).
C             (BND(L),L=1,NL),(BND2(L),L=1,NL)   WHERE NL=NNH*NY.
C               BND(L) IS HYDRO AT TIME TI1, BND2(L) HYDRO AT TI2.
C          ***   AFTER THE FINAL TIME DUMP A HEADER IS DUMPED, ALL 666.
C
C         - - - - - - - - - - - - - - - - - - - - - - - - - - -
C            * * * * *  GLOSSARY OF TERMS  * * * * *
C         SYMBOLS - * NAME IN A COMMON.
C                   D NAME IN A COMDECK (CDIN1, CDNU1, OR CDCOM).
C                   N IN A NAMELIST INPUT.
C                   Z VALUE IS FROM ZBLOCK ON (IN9), DIRECT OR COMPUTED.
C                   _XXXXX_ DENOTES PARAMETER IN COMDECK CDIN1 OR CDNU1.
C
C *D    BND(_NLBIG_) - ARRAY FOR ROW OF NEW HYDRO DATA.
C *D    BND2(_NLBIG_) - ARRAY FOR ROW OF NEW HYDRO DATA.
C *N    CSTART - STARTING CYCLE TO BE MATCHED ON FILE IN9.
C  N    CSTOP  - STOP PROCESSING CYCLES AFTER CYCLEI >= CSTOP.
C *     CYCLEI - PRESENT CYCLE.
C *DZ   DELXI(_IIBIG_) - VOLUME CONTRIBUTION IN X DIRECTION FOR CELL.
C *DZ   DELYI(_IJBIG_) - VOLUME CONTRIBUTION IN Y DIRECTION FOR CELL.
C *DZ   DELZI(_IKBIG_) - VOLUME CONTRIBUTION IN Z DIRECTION FOR CELL.
```

```
C       CDCOM  - COMDECK, COMMONS /CDIN3/ AND /CDNU3/.
C       CDIN1  - COMDECK, COMMON/CDIN1/, PARAMETERS AND DONOR ARRAYS.
C       CDNU1  - COMDECK, COMMON/CDNU1/, PARAMETERS & RECIPIENT ARRAYS.
C       FINDI  - SUBROUTINE TO LOCATE INTERPOLATION INDEX.
C *DZ   HYDROI(_ILBIG_) - ARRAY FOR 1 BLOCK OF INPUT HYDRO VALUES.
C       IBND   - COUNT ON POSSIBLE OUTPUT FILES.
C *     ICONV  - CONVERSION TYPE.  2 IS CYLINDRICAL TO 2D CARTESIAN,
C                  3 IS CYLINDRICAL TO 3D CARTESIAN.  (3 ONLY 8/08/84)
C                COMBINES IDIMEN, IGEOM, NDIMEN, AND NGEOM.
C  Z    IDIMEN - DIMENSIONS OF OLD HULL RUN.
C       IEND   - SIGNAL FROM NXTIM.  0 MEANS NEW TIME ON FILE (IN9).
C                  1 MEANS NORMAL END OF IN9, -1 DENOTES READING ERROR.
C *Z    IGEOM  - GEOMETRY OF OLD HULL RUN. 1 CARTESIAN, 2 CYLINDRICAL.
C  D   _IIBIG_ - ARRAY SIZE.  NEED IIBIG > IIMAX.
C *Z    IIMAX  - IMAX FROM DONOR Z-BLOCK.
C *Z    IIMAX1 - IIMAX + 1.
C  D   _IJBIG_ - ARRAY SIZE.  NEED IJBIG > IJMAX.
C *Z    IJMAX  - JMAX FROM DONOR Z-BLOCK.
C *Z    IJMAX1 - IJMAX + 1.
C  D   _IKBIG_ - ARRAY SIZE.  NEED IKBIG > IKMAX.
C *Z    IKMAX  - KMAX FROM DONOR Z-BLOCK.
C *Z    IKMAX1 - IKMAX + 1.
C  D   _ILBIG_ - ARRAY SIZE.  NEED ILBIG >= INHPB = IIMAX*IROWPB*INH
C *N    IN9    - FILE NUMBER OF OLD HULL RESTART FILE.
C *Z    INH    - NUMBER OF HYDRO VARIABLES/POINT IN OLD HULL.
C *Z    INHPB  - NUMBER OF HYDRO VARIABLES/BLOCK IN OLD HULL.
C  N    INITPR(6) - TIME DUMP TO START TABULATION OF NFOUT(I),I=1,6.
C  N    INITRO(6) - INITIAL ROW TO START TABULATION IN NFOUT(I),I=1,6.
C *N    INM    - NUMBER OF OUTPUT POINT IN X DIRECTION.
C       IP     - POINTER FOR INTERPOLATION IN X.
C *Z    IROWPB - ROWS/BLOCK IN HYDROI.
C *     ITIME  - TIME DUMPS PROCESSED.  (REPEATED FOR EACH BOUNDARY).
C *     ITYPE  - INPUT TYPE.  1=CYLINDRICAL, 2=2D CARTESIAN, 3=3D.
C *N    JNM    - NUMBER OF Y DIRECTION POINTS FOR OUTPUT.
C       JP     - POINTER FOR INTERPOLATION IN Y.
C *N    KNM    - NUMBER OF Z DIRECTION OUTPUT POINTS.
C  N    LASTPR(6) - TIME DUMP TO STOP TABULATION OF NFOUT(I),I=1,6.
C  N    LASTRO(6) - LAST ROW TO TABULATE IN NFOUT(I),I=1,6.
C *N    NBND(6) - SIGNAL FOR BOUNDARY OUTPUT.  0 IS NO, 1 IS YES.
C  N    NDIMEN - DIMENSION FOR NEW HULL.
C *N    NFOUT(6) - FILES FOR BOUNDARY OUTPUT FOR NEW HULL.
C  N    NGEOM  - GEOMETRY OF NEW HULL RUN.  ONLY 1 (CARTESIAN) NOW.
C                        MAY ADD 2 FOR CYLINDRICAL LATER.
C  D   _NIJKMAX_ -ARRAY SIZE.  NEED NIJKMAX >= MAX(INM,JNM,KNM).
C  D   _NLBIG_ - ARRAY SIZE.  NEED NLBIG >= MAX(NXPL,NYPL,NZPL).
C *N    NNH    - HYDRO VARIABLES/POINT FOR NEW  HULL PROBLEM (5 FOR 3D).
C       NTYPE  - NEW HULL TYPE.  1=CYLINDRICAL, 2=2D CARTESIAN, 3=3D.
C *     NXPL   - NUMBER OF HYDROS/LINE FOR OUTPUT X-PLANES (NNH*JNM).
C       NXTBLK - SUBROUTINE TO FIND NEXT BLOCK OF HYDRO DATA ON (IN9).
C       NXTIM  - SUBROUTINE TO FIND NEXT TIME DUMP ON FILE IN9,
C                  BYPASS Z-BLOCK, READ AND PROCESS VERTICES.
C *     NYPL   - NUMBER OF HYDROS/LINE FOR OUTPUT Y-PLANES (NNH*INM).
C *     NZPL   - NUMBER OF HYDROS/LINE FOR OUTPUT Z-PLANES (NNH*INM).
C *N    PROBIN - PROBLEM NUMBER OF DONOR HULL PROBLEM.
C       STRTIN - SUBROUTINE. CHECKS PROBLEM NUMBER, GETS Z-BLOCK DATA.
C                   CHECKS ARRAYS FOR INPUT.
C       STRTNU - SUBROUTINE.  READS NAMELIST INPUT FOR NEW HULL.
C                  SETS UP POINTS. CHECKS OUTPUT ARRAYS SIZES.
C *     TIMEIN - TIME OF HULL DUMP BEING PROCESSED.
C       TI1,TI2 - TIMES FROM 2 SUCCESSIVE HULL DUMPS.
C *N    TSH    - TIME SHIFT FROM OLD TO NEW HULL.
C *DZ   XI(_IIBIG_) - VERTICES, THEN MID-CELL VALUES, FROM OLD HULL.
C *D    XN(_NIJKMAX_) - OUTPUT POINTS IN X DIRECTION.
C *N    XN1    - LEFT OUTPUT BOUNDARY.  MAY BE MINIMUM X FOR OUTPUT GRIDS
```

```
C *N   XNM    - RIGHT EXTERNAL CELL CENTER.  MAY BE MAX X FOR GRIDS.
C *N   XSH    - SHIFT IN X FROM OLD TO NEW HULL.
C *    XYZB(6) - OUTPUT BOUNDARY LOCATIONS.  (XN1,XNM,YN1,YNM,ZN1,ZNM)
C *DZ  YI(_IJBIG_) - VERTICES, THEN MIDCELL VALUES FROM OLD HULL.
C *D   YN(_NIJKMAX_) - OUTPUT POINTS IN Y DIRECTION.
C *N   YN1    - AFT OUTPUT BOUNDARY (3D).  BOTTOM BOUNDARY (2D).
C               MAY BE MINIMUM X FOR OUTPUT GRIDS FOR Y AND Z PLANES.
C *N   YNM    - EXTERNAL CELL CENTER, FORE BOUNDARY 3D, TOP BOUNDARY 2D.
C               MAY BE MAXIMUM Y FOR OUTPUT GRIDS FOR X OR Z PLANES.
C *N   YSH    - SHIFT IN Y FROM OLD TO NEW HULL.
C *DZ  ZI(_IKBIG_) - VERTICES, THEN MIDCELL VALUES, FROM OLD HULL.
C *D   ZN(_NIJKMAX_) - POINTS FOR OUTPUT IN Z DIRECTION.
C *N   ZN1    - BOTTOM BOUNDARY.  MAY BE MINIMUM Z FOR OUTPUT GRIDS.
C *N   ZNM    - TOP EXTERNAL CELL CENTER.  MAY BE MAX Z FOR OUTPUT GRIDS
C *N   ZSH    - SHIFT IN Z FROM OLD TO NEW HULL.
C          - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C               - - - - - - - - - - - - - - - - - - - - - - - - -
C
*CALL CDIN1
*CALL CDNU1
*CALL CDCOM
C             SET UP PARAMETERS FOR INPUT FILE
      NAMELIST /INDAT1/ IN9,PROBIN,CSTART,CSTOP
C                  INPUT DEFAULTS
      IN9 = 9
      CSTART = 0
      CSTOP = 9999999.0
      READ(5,INDAT1)
      WRITE(6,INDAT1)
      CALL STRTIN
C             SET UP PARAMETERS FOR THE NEW HULL DATA.
      CALL STRTNU
C          READY.  GO TO CODING FOR SELECTED CONVERSION.
      IF(ICONV .EQ. 2)GOTO 2000
      IF(ICONV .EQ. 3)GOTO 3000
      WRITE(6,25)ICONV
      STOP  '  ABORT HULLUP 25.  NO ICONV CODING.'
   25 FORMAT(//'  *** ABORT.  HULLUP 25.  NO CODING FOR ICONV = ',I2)
C        - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C             CODING FOR ICONV =2 NOT YET INSERTED.
 2000 CONTINUE
      WRITE(6,2005)ICONV
      STOP '  ABORT HULLUP 2005.  NO ICONV CODING.'
 2005 FORMAT(//'  ** ABORT.  HULLUP 2005.  NO CODING FOR ICONV = ',I2)
C        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C          ICONV = 3.  INPUT 2D CYLINDRICAL, OUTPUT 3D CARTESIAN.
C            START LOOP ON BOUNDARIES FOR IBND=1,6.
 3000 IBND = 0
 3010 IBND = IBND + 1
      IF(IBND .EQ. 7)GOTO 10000
      IF(NBND(IBND) .GT. 0)GOTO 3020
      GOTO 3010
C          START LOOP ON TIME.  ITIME IS A COUNTER.
 3020 ITIME = 0
      REWIND(IN9)
C          FIND NEXT TIME DUMP (INITIALLY FIND CSTART) AND VERTICES.
 3030 CALL NXTIM(IEND)
C               IF END OF READABLE FILE, GOTO END OF TIME LOOP.
      IF(IEND .NE. 0)GOTO 3900
C             THE INPUT TIME DUMP HAS BEEN FOUND.
C             VERTICES READ, CELL CENTERS AND VOLUME VALUES SET.
C            BRANCH ON BOUNDARY.  1,2 X-PLANE  3,4 Z-PLANE, 5,6 Y-PLANE.
      GOTO(3050,3050,3500,3500,3750,3750),IBND
C
```

```
C          XPLANE BOUNDARY.  IBND = 1 OR 2.
 3050 CONTINUE
C                    SET UP SOME PARAMETERS.
C                 NN = (JN-1)*NNH  IS LOCATION FOR OUTPUT HYDROS.
C                 KN IS ROW, JN IS COLUMN IN THE OUTPUT PLANE.
C            IHB IS INPUT HYDRO BLOCK NUMBER.
C            JP AND JP2 ARE THE ROW POINTERS IN THE INPUT PLANE.
C               YI(JP) .LE. ZN(KN) .LT. YI(JP2).   JP2 = JP + 1.
      IHB = 0
      JP = 1
      JP2 = JP+1
C              START LOOP ON ZN.
      DO 3140 KN=1,KNM
      NN=0
      ZP = ZN(KN)
      CALL FINDI(JP,ZP,YI,IJMAX)
C               ZR IS THE INTERPOLATION RATIO FOR Z IN YI(JP) TO YI(JP+1).
      ZR = (ZP - YI(JP))/(YI(JP+1)-YI(JP))
      IF(JP .LT. (IHB-1)*IROWPB +1 .OR. JP .GT. IHB*IROWPB)
     +  CALL NXTBLK(JP,IHB)
C           GET READY TO INTERPOLATE IN XI  (FOR F(R,YI(JP)).
      IP = 1
C               X-PLANE.  START LOOP ON JN.
      XP = XYZB(IBND)
      XPSQ = XP*XP
      DO 3090 JN=1,JNM
      YP = YN(JN)
      YPSQ = YP*YP
      RIN = SQRT(XPSQ + YPSQ)
      CALL FINDI(IP,RIN,XI,IIMAX)
      LI = ((JP-(IHB-1)*IROWPB -1)*IIMAX + IP-1)*INH
      RR = (RIN - XI(IP))/(XI(IP+1) - XI(IP))
C              FIND F(RIN,XI(IP)) FOR F =  VR,VZ,E,RHO.
C              RADIAL VELOCITY, AXIAL VELOCITY, SPECIFIC ENERGY, AND
C              DENSITY,RESPECTIVELY.  DENSITY IS MASS/VOLUME.
      BND(NN+2) = HYDROI(LI+2) + (HYDROI(LI+2+INH) - HYDROI(LI+2))*RR
      BND(NN+3) = HYDROI(LI+3) + (HYDROI(LI+3+INH) - HYDROI(LI+3))*RR
      BND(NN+4) = HYDROI(LI+4) + (HYDROI(LI+4+INH) - HYDROI(LI+4))*RR
      RHOIP = HYDROI(LI+5)/(DELXI(IP)*DELYI(JP))
      RHOIP1 = HYDROI(LI+5+INH)/(DELXI(IP+1)*DELYI(JP))
      BND(NN+5) = RHOIP + (RHOIP1 - RHOIP)*RR
      NN = NN + NNH
 3090 CONTINUE
C                FINISHED INTERPOLATION FOR F(R,YI(JP)) FOR A ROW.
 3100 NN=0
      IP = 1
C            YI(JP2) IS NOW THE INPUT Y.  Y GT ZN.  (JP2=JP+1)
C                 IS A NEW INPUT HYDRO BLOCK NEEDED?
      IF (JP2 .GT. IHB*IROWPB)CALL NXTBLK(JP2,IHB)
C                 READY TO INTERPOLATE FOR F(R,YI(JP2), THEN FOR F(R,Z).
C               X-PLANE.   IBND = 1 OR 2.   LOOP ON YN.
      DO 3130 JN=1,JNM
      YP = YN(JN)
      YPSQ = YP*YP
      RIN = SQRT(XPSQ + YPSQ)
      CALL FINDI(IP,RIN,XI,IIMAX)
      LI = ((JP2 - (IHB-1)*IROWPB -1)*IIMAX + IP-1)*INH
      RR = (RIN - XI(IP))/(XI(IP+1) - XI(IP))
C            FIND F(RIN,YI(+)), THEN F(RIN,ZN) FOR VX,VY,VZ,E,RHO.
      HYDROP = HYDROI(LI+2) + (HYDROI(LI+2+INH) - HYDROI(LI+2))*RR
      HYDROP = BND(NN+2) + (HYDROP - BND(NN+2))*ZR
C            HYDROP = VR.  FIND AND SAVE VX AND VY VELOCITY COMPONENTS.
      IF(RIN .GT. 0.0)GOTO 3103
C            ZERO RADIUS.  SET VX = VY = 0.0.
```

```
      BND(NN+1) = 0.0
      BND(NN+2) = 0.0
      GOTO 3104
 3103 BND(NN+1) = HYDROP*XP/RIN
      BND(NN+2) = HYDROP*YP/RIN
 3104 HYDROP = HYDROI(LI+3) + (HYDROI(LI+3+INH) - HYDROI(LI+3))*RR
      BND(NN+3) = BND(NN+3) + (HYDROP - BND(NN+3))*ZR
      HYDROP = HYDROI(LI+4) + (HYDROI(LI+4+INH) - HYDROI(LI+4))*RR
      BND(NN+4) = BND(NN+4) + (HYDROP - BND(NN+4))*ZR
      RHOIP = HYDROI(LI+5)/(DELXI(IP)*DELYI(JP2))
      RHOIP1 = HYDROI(LI+5+INH)/(DELXI(IP+1)*DELYI(JP2))
      HYDROP = RHOIP + (RHOIP1 - RHOIP)*RR
      BND(NN+5) = BND(NN+5) + (HYDROP - BND(NN+5))*ZR
      NN = NN + NNH
 3130 CONTINUE
      CALL WRITPL(IBND,
     +  NFOUT(IBND),XP,XSH,YN,JNM,YSH,ZN,KNM,ZSH,BND,NXPL)
 3140 CONTINUE
      GOTO 3850
C
C          YPLANE BOUNDARY.  IBND = 5 OR 6.
 3750 CONTINUE
C                   SET UP SOME PARAMETERS.
C          NN = (IN-1)*NNH IS LOCATION FOR OUTPUT HYDROS.
C          KN IS ROW, IN IS COLUMN IN THE OUTPUT PLANE.
C          IHB IS INPUT HYDRO BLOCK NUMBER.
C          JP AND JP2 ARE THE ROW POINTERS IN THE INPUT PLANE.
C              YI(JP) .LE. ZN(KN) .LT. YI(JP2).   JP2 = JP + 1.
      IHB = 0
      JP2 = JP+1
      JP = 1
C            START LOOP ON ZN.
      DO 3840 KN=1,KNM
      NN = 0
      ZP = ZN(KN)
      CALL FINDI(JP,ZP,YI,IJMAX)
C              ZR IS THE INTERPOLATION RATIO FOR Z IN YI(JP) TO YI(JP+1).
      ZR = (ZP - YI(JP))/(YI(JP+1)-YI(JP))
      IF(JP .LT. (IHB-1)*IROWPB +1 .OR. JP .GT. IHB*IROWPB)
     +  CALL NXTBLK(JP,IHB)
C          GET READY TO INTERPOLATE IN XI  (FOR F(R,YI(JP)).
      IP = 1
C              Y-PLANE.  START LOOP ON IN.
 3770 YP = XYZB(IBND)
      YPSQ = YP*YP
      DO 3791 IN=1,INM
      XP = XN(IN)
      XPSQ = XP*XP
      RIN = SQRT(XPSQ + YPSQ)
      CALL FINDI(IP,RIN,XI,IIMAX)
      LI = ((JP-(IHB-1)*IROWPB -1)*IIMAX + IP-1)*INH
      RR = (RIN - XI(IP))/(XI(IP+1) - XI(IP))
C            FIND F(RIN,XI(IP)) FOR F =  VR,VZ,E,RHO.
C            RADIAL VELOCITY, AXIAL VELOCITY, SPECIFIC ENERGY, AND
C            DENSITY,RESPECTIVELY.  DENSITY IS MASS/VOLUME.
      BND(NN+2) = HYDROI(LI+2) + (HYDROI(LI+2+INH) - HYDROI(LI+2))*RR
      BND(NN+3) = HYDROI(LI+3) + (HYDROI(LI+3+INH) - HYDROI(LI+3))*RR
      BND(NN+4) = HYDROI(LI+4) + (HYDROI(LI+4+INH) - HYDROI(LI+4))*RR
      RHOIP = HYDROI(LI+5)/(DELXI(IP)*DELYI(JP))
      RHOIP1 = HYDROI(LI+5+INH)/(DELXI(IP+1)*DELYI(JP))
      BND(NN+5) = RHOIP + (RHOIP1 - RHOIP)*RR
      NN = NN + NNH
 3791 CONTINUE
C               FINISHED INTERPOLATION FOR F(R,YI(JP)) FOR A ROW.
```

```
 3800 NN=0
      IP = 1
C           YI(JP2) IS NOW THE INPUT Y.  Y GT ZN.  (JP2=JP+1)
C                IS A NEW INPUT HYDRO BLOCK NEEDED?
      IF (JP2 .GT. IHB*IROWPB)CALL NXTBLK(JP2,IHB)
C                READY TO INTERPOLATE FOR F(R,YI(JP2), THEN FOR F(R,Z).
C                 LOOP FOR X-PLANE OR Y-PLANE?
C               Y-PLANE.   IBND = 5 OR 6.   LOOP ON XN.
 3810 DO 3831 IN=1,INM
      XP = XN(IN)
      XPSQ = XP*XP
      RIN = SQRT(XPSQ + YPSQ)
      CALL FINDI(IP,RIN,XI,IIMAX)
      LI = ((JP2 - (IHB-1)*IROWPB -1)*IIMAX + IP-1)*INH
      RR = (RIN - XI(IP))/(XI(IP+1) - XI(IP))
C           FIND F(RIN,YI(+)), THEN F(RIN,ZN) FOR VX,VY,VZ,E,RHO.
      HYDROP = HYDROI(LI+2) + (HYDROI(LI+2+INH) - HYDROI(LI+2))*RR
      HYDROP = BND(NN+2) + (HYDROP - BND(NN+2))*ZR
C           HYDROP = VR.  FIND AND SAVE VX AND VY VELOCITY COMPONENTS.
      IF(RIN .GT. 0.0)GOTO 3813
C           ZERO RADIUS.  SET VX = VY = 0.0.
      BND(NN+1) = 0.0
      BND(NN+2) = 0.0
      GOTO 3814
 3813 BND(NN+1) = HYDROP*XP/RIN
      BND(NN+2) = HYDROP*YP/RIN
 3814 HYDROP = HYDROI(LI+3) + (HYDROI(LI+3+INH) - HYDROI(LI+3))*RR
      BND(NN+3) = BND(NN+3) + (HYDROP - BND(NN+3))*ZR
      HYDROP = HYDROI(LI+4) + (HYDROI(LI+4+INH) - HYDROI(LI+4))*RR
      BND(NN+4) = BND(NN+4) + (HYDROP - BND(NN+4))*ZR
      RHOIP = HYDROI(LI+5)/(DELXI(IP)*DELYI(JP2))
      RHOIP1 = HYDROI(LI+5+INH)/(DELXI(IP+1)*DELYI(JP2))
      HYDROP = RHOIP + (RHOIP1 - RHOIP)*RR
      BND(NN+5) = BND(NN+5) + (HYDROP - BND(NN+5))*ZR
      NN = NN + NNH
 3831 CONTINUE
      CALL WRITPL(IBND,
     + NFOUT(IBND),YP,YSH,XN,INM,XSH,ZN,KNM,ZSH,BND,NYPL)
 3840 CONTINUE
C             TIME DUMP FOR AN X-PLANE OR Y-PLANE BOUNDARY NOW COMPLETE.
 3850 CONTINUE
      ITIME = ITIME + 1
C            DO WE WANT ANOTHER TIME DUMP?
      IF(CYCLEI .LT. CSTOP)GOTO 3030
      GOTO 3900
C              Z-PLANE BOUNDARY.  IBND = 3 OR 4.  (2D CYLINDRICAL TO 3D)
 3500 ZP = XYZB(IBND)
C             FIRST, FIND HYDRO BLOCK WITH ROW OF DATA FOR YI(JP) < ZP.
      IHB = 0
      JP = 1
      CALL FINDI(JP,ZP,YI,IJMAX)
C             FIND INTERPOLATED VALUES AT Z = ZP ROW BY ROW.
C            START LOOP ON ROWS.
C        FIRST, FIND VALUES AT Z = YI(JP).
      DO 3560 JN=1,JNM
      CALL NXTBLK(JP,IHB)
      NN=0
      YP = YN(JN)
      YPSQ = YP*YP
C            START LOOP ON COLUMNS.
      IP = 1
      DO 3510 IN=1,INM
      XP = XN(IN)
      XPSQ = XP*XP
```

```
      RIN = SQRT(XPSQ + YPSQ)
      CALL FINDI(IP,RIN,XI,IIMAX)
      LI = ((JP-(IHB-1)*IROWPB -1)*IIMAX + IP-1)*INH
      RR = (RIN - XI(IP))/(XI(IP+1) - XI(IP))
C           FIND F(RIN,XI(IP)) FOR F =  VR,VZ,E,RHO.
      BND(NN+2) = HYDROI(LI+2) + (HYDROI(LI+2+INH) - HYDROI(LI+2))*RR
      BND(NN+3) = HYDROI(LI+3) + (HYDROI(LI+3+INH) - HYDROI(LI+3))*RR
      BND(NN+4) = HYDROI(LI+4) + (HYDROI(LI+4+INH) - HYDROI(LI+4))*RR
      RHOIP = HYDROI(LI+5)/(DELXI(IP)*DELYI(JP))
      RHOIP1 = HYDROI(LI+5+INH)/(DELXI(IP+1)*DELYI(JP))
      BND(NN+5) = RHOIP + (RHOIP1 - RHOIP)*RR
      NN = NN + NNH
 3510 CONTINUE
C             NOW, FIND INTERPOLATED VALUES AT Z = YI(JP+1), THEN AT ZP.
      JP2 = JP + 1
      IF(JP2 .GT. IHB*IROWPB) CALL NXTBLK(JP2,IHB)
C           COMPUTE INTERPOLATION RATIO IN Z.
      ZR = (ZP - YI(JP2-1))/(YI(JP2) - YI(JP2-1))
C           LOOP ON ROW.
      NN = 0
C          LOOP ON COLUMN.
      IP = 1
      DO 3550 IN=1,INM
      XP=XN(IN)
      XPSQ = XP*XP
      RIN = SQRT(XPSQ + YPSQ)
      CALL FINDI(IP,RIN,XI,IIMAX)
      LI = ((JP2 - (IHB-1)*IROWPB -1)*IIMAX + IP-1)*INH
      RR = (RIN - XI(IP))/(XI(IP+1) - XI(IP))
C          FIND F(RIN,YI(+)), THEN F(RIN,ZN) FOR VX,VY,VZ,E,RHO.
      HYDROP = HYDROI(LI+2) + (HYDROI(LI+2+INH) - HYDROI(LI+2))*RR
      HYDROP = BND(NN+2) + (HYDROP - BND(NN+2))*ZR
C          HYDROP = VR.  FIND AND SAVE VX AND VY VELOCITY COMPONENTS.
      IF(RIN .GT. 0.0)GOTO 3543
C          ZERO RADIUS.  SET VX = VY = 0.0.
      BND(NN+1) = 0.0
      BND(NN+2) = 0.0
      GOTO 3544
 3543 BND(NN+1) = HYDROP*XP/RIN
      BND(NN+2) = HYDROP*YP/RIN
 3544 HYDROP = HYDROI(LI+3) + (HYDROI(LI+3+INH) - HYDROI(LI+3))*RR
      BND(NN+3) = BND(NN+3) + (HYDROP - BND(NN+3))*ZR
      HYDROP = HYDROI(LI+4) + (HYDROI(LI+4+INH) - HYDROI(LI+4))*RR
      BND(NN+4) = BND(NN+4) + (HYDROP - BND(NN+4))*ZR
      RHOIP = HYDROI(LI+5)/(DELXI(IP)*DELYI(JP2))
      RHOIP1 = HYDROI(LI+5+INH)/(DELXI(IP+1)*DELYI(JP2))
      HYDROP = RHOIP + (RHOIP1 - RHOIP)*RR
      BND(NN+5) = BND(NN+5) + (HYDROP - BND(NN+5))*ZR
      NN = NN+ NNH
 3550 CONTINUE
      CALL WRITPL(IBND,
     +  NFOUT(IBND),ZP,ZSH,XN,INM,XSH,YN,JNM,YSH,BND,NZPL)
 3560 CONTINUE
      ITIME = ITIME + 1
C           DO WE WANT ANOTHER TIME DUMP?
      IF(CYCLEI .LT. CSTOP)GOTO 3030
      GOTO 3900
C            END OF THE TIME LOOP FOR ICONV = 3
 3900 GOTO 3010
C          - - - - - - - - - - - - - - - - - - - - - - - - - - -
C            END OF HULLUP.  PUT TERMINAL SIGNAL ON FILES.
10000 DO 10010 I=1,6
      NF = NFOUT(I)
      IF(NBND(I) .GT. 0)WRITE (NF) 666.,666.,666.,666.,666,666,666
```

```
C          TABULATE A PORTION OF EACH BOUND9 FILE.
      IF(NBND(I) .GT. 0)CALL REWRIT(I,NF)
10010 CONTINUE
      WRITE(6,10025) ICONV
      STOP'  NORMAL END OF HULLUP'
10025 FORMAT('  NORMAL END OF HULLUP.  ICONV =',I2)
      END
      SUBROUTINE STRTIN
C      CHECK PROBLEM NUMBER ON FILE IN9 IS PROBIN.
C      READ Z-BLOCK AND EXTRACT INPUT PARAMETERS.
C      CHECK THAT INPUT ARRAYS ARE LARGE ENOUGH.
C
*CALL CDIN1
*CALL CDCOM
C
      DIMENSION DIDI(4), IZ(200),ZBL(200)
      EQUIVALENCE (IZ,ZBL)
C
C         CHECK THAT INPUT PROBLEM NUMBER IS PROBIN.
      REWIND (IN9)
      NTRY = 0
   10 IF(NTRY .LT. 10)GOTO 20
      WRITE(6,15) IN9
      STOP'  ABORT PROBLEM AT STRTIN 10.'
   15 FORMAT (//'  ** ABORT. INPUT PROBLEM AT STRTIN 10. FILE TAPE',I4)
   20 NTRY = NTRY + 1
      READ(IN9)(DIDI(I),I=1,4)
      IF(EOF(IN9) .NE. 0)GOTO 10
      IF(ABS(DIDI(2) - PROBIN) .LT. 0.00001)GOTO 40
      WRITE(6,35) IN9,DIDI(2),PROBIN
   35 FORMAT('  ** ABORT.  PROBLEM NUMBERS DIFFER.'/
     +      '  PROBLEM NUMBER ON FILE',I3,' IS ',E15.6/
     +      '  INPUT PROBLEM NUMBER, PROBIN, IS ',E15.6/)
      STOP'  ** ABORT.  STRTIN 30.  PROBLEM NOS. DISAGREE.'
C          PROBLEM NUMBERS CHECK.  READ Z-BLOCK.  SET UP IN-PARAMETERS.
   40 READ(IN9)(ZBL(I),I=1,200)
      DO 50 I=1,92
      IF(IZ(I+100) .EQ. 6HDIMEN ) IDIMEN = ZBL(I) + 0.5
      IF(IZ(I+100) .EQ. 6HGEOM  ) IGEOM  = ZBL(I) + 0.5
      IF(IZ(I+100) .EQ. 6HIMAX  ) IIMAX  = ZBL(I) + 0.5
      IF(IZ(I+100) .EQ. 6HJMAX  ) IJMAX  = ZBL(I) + 0.5
      IF(IZ(I+100) .EQ. 6HKMAX  ) IKMAX  = ZBL(I) + 0.5
      IF(IZ(I+100) .EQ. 6HNH    ) INH    = ZBL(I) + 0.5
      IF(IZ(I+100) .EQ. 6HNROWPB) IROWPB = ZBL(I) + 0.5
   50 CONTINUE
      INHPR = IIMAX*INH
      INHPB = INHPR*IROWPB
      ITYPE = IDIMEN
      IF(IGEOM .EQ. 2)ITYPE = 1
      IIMAX1 = IIMAX + 1
      IJMAX1 = IJMAX + 1
      IKMAX1 = IKMAX + 1
C           CHECK ON THE INPUT ARRAY LENGTHS
      IF(IIBIG.GE.IIMAX1 .AND. IJBIG.GE.IJMAX1 .AND. IKBIG.GE.IKMAX1
     + .AND. ILBIG.GE.INHPB)GOTO 100
      WRITE(6,55)IIBIG,IJBIG,IKBIG,ILBIG,IIMAX1,IJMAX1,IKMAX1,INHPB
      STOP'  ** ABORT. STRTIN 55.  DONOR ARRAY'
   55 FORMAT(//' ** ABORT. STRTIN 55.  DONOR ARRAY TOO SMALL.',
     +    /'  IIBIG, IJBIG, IKBIG, ILBIG  = ',4I10,
     +    /'  IIMAX1,IJMAX1,IKMAX1,INHPB  = ',4I10)
  100 WRITE(6,105)PROBIN
      RETURN
  105 FORMAT(/' READY TO PROCESS PROB',F10.4)
      END
```

```
      SUBROUTINE STRTNU
C         SET UP OUTPUT CONTROL PARAMETERS FROM DEFAULT AND INPUT.
C
*CALL CDNU1
*CALL CDCOM
C
C
      NAMELIST/NDAT1/XN1,XNM,YN1,YNM,ZN1,ZNM,INM,JNM,KNM,TSH,XSH,YSH,ZSH
      NAMELIST /NDAT2/NGEOM,NDIMEN,NNH,NBND,NFOUT
C               DEFAULT VALUES
C         NO ACTIVE BOUNDARIES.  OUTPUT FILE FOR BOUNDARY I IS 10+I.
      DO 10 I=1,6
      NBND(I) = 0
      NFOUT(I) = 10+I
   10 CONTINUE
C           GEOMETRY IS CARTESIAN.   DIMENSION IS 3.
      NGEOM = 1
      NDIMEN = 3
C          DEFAULT NUMBER OF NEW HYDRO VARIABLES/POINT IS 5.
      NNH = 5
C          DEFAULT BOUNDARIES SET TO 0.0.  DEFAULT SHIFTS SET TO 0.0.
      XN1 = 0.0
      XNM = 0.0
      YN1 = 0.0
      YNM = 0.0
      ZN1 = 0.0
      ZNM = 0.0
      TSH = 0.0
      XSH = 0.0
      YSH = 0.0
      ZSH = 0.0
C         DEFAULT OUTPUT GRID DIMENSIONS SET TO 200.
      INM = 200
      JNM = 200
      KNM = 200
C         READ NDAT1 PARAMETER CHANGES AND PRINT PARAMETERS
      READ(5,NDAT1)
      WRITE(6,NDAT1)
      IF(INM.LT.0 .OR. JNM.LT.0 .OR. KNM.LT.0)GOTO 200
C           GRID INDICES POSITIVE.  SET EVENLY SPACED GRID.
C         SET GRID VALUES.    FIRST, FORCE AT LEAST 2 POINTS.
      IF(INM .LT. 2) INM = 2
      IF(JNM .LT. 2) JNM = 2
      IF(KNM .LT. 2) KNM = 2
      DELD = (XNM - XN1)/(INM - 1)
      XN(1) = XN1
      DO 20 I=2,INM
      XN(I) = XN(I-1) + DELD
   20 CONTINUE
      DELD = (YNM - YN1)/(JNM - 1)
      YN(1) = YN1
      DO 30 J=2,JNM
      YN(J) = YN(J-1) + DELD
   30 CONTINUE
      DELD = (ZNM - ZN1)/(KNM - 1)
      ZN(1) = ZN1
      DO 40 K=2,KNM
      ZN(K) = ZN(K-1) + DELD
   40 CONTINUE
  245 IF(ABS(XN(INM)-XNM) .LT. 0.00001*XNM) XN(INM) = XNM
      IF(ABS(YN(JNM)-YNM) .LT. 0.00001*YNM) YN(JNM) = YNM
      IF(ABS(ZN(KNM)-ZNM) .LT. 0.00001*ZNM) ZN(KNM) = ZNM
C           PRINT THE GRID.
      WRITE(6,246) (XN(I),I=1,INM)
```

```
      WRITE(6,247) (YN(J),J=1,JNM)
      WRITE(6,248) (ZN(K),K=1,KNM)
      GOTO 50
  246 FORMAT(' MESH FOR NEW BOUNDARY PLANES.  SHOULD BE CHECKED.'/,
     + ' NEW CELL CENTERS MUST BE BETWEEN EXTREMES.  RESULTS BEST IF'/,
     +  ' CELL CENTERS ARE AT MESH POINTS (REMOVES INTERPOLATION)'//,
     +    10X,'X GRID'/,(1P6E15.7))
  247 FORMAT(10X,'Y GRID'/,(1P6E15.7))
  248 FORMAT(10X,'Z GRID'/,(1P6E15.7))
   50 READ(5,NDAT2)
      WRITE(6,NDAT2)
C         NTYPE = 1 FOR 2D CYINDRICAL, = 2 FOR 2D CARTESIAN, = 3 FOR 3D.
      NTYPE = NDIMEN + 1 - NGEOM
C           SET THE CONVERSION TYPE, ICONV.
C            ICONV = 3 FOR 2D CYLINDRICAL TO 3D CARTESIAN.
      ICONV = 3*(ITYPE-1) + NTYPE
C           SET THE VALUES FOR BOUNDARIES.
      XYZB(1) = XN1
      XYZB(2) = XNM
      XYZB(3) = ZN1
      XYZB(4) = ZNM
      XYZB(5) = YN1
      XYZB(6) = YNM
C            3D OUTPUT.   COMPUTE SIZE OF HYDRO DATA/PLANE.
      NXPL = JNM*NNH
      NYPL = INM*NNH
      NZPL = INM*NNH
C           CHECK THAT ARRAYS FOR OUTPUT ARE LARGE ENOUGH.
      IF(NIJKMAX.GE.INM .AND. NIJKMAX.GE.JNM .AND. NIJKMAX.GE.KNM
     +  .AND. NLBIG.GE.MAX0(NXPL,NYPL,NZPL))GOTO 100
      MAXPL = MAX0(NXPL,NYPL,NZPL)
      WRITE(6,85)NIJKMAX,NLBIG,INM,JNM,KNM,MAXPL
      STOP' **ABORT. STRTNU 85.  RECIPIENT ARRAY TOO SMALL.'
   85 FORMAT(//'  ** ABORT.  STRTNU 85.  RECIPIENT ARRAY TOO SMALL.',
     +  /'  NIJKMAX,INM,JNM,KNM,MAXPL = ',5I10)
  100 WRITE(6,105)
      RETURN
  105 FORMAT(' OUTPUT PARAMETERS OK. READY TO COMPUTE BOUNDARY VALUES.')
C         WILL READ DATA AND FORM THE RECIPIENT GRID.
  200 INM = ABS(INM)
      JNM = ABS(JNM)
      KNM = ABS(KNM)
C         FORM XN(I),I=1,INM
      READ(5,201) XN(1)
  201 FORMAT(E15.8,I5)
      N1 = 1
  212 READ(5,201) DXN,NNX
      IF(NNX .LT. 0)GOTO 220
      NX = N1+NNX
      N1 = N1+1
      DO 215 L=N1,NX
      XN(L) = XN(L-1) +DXN
  215 CONTINUE
      N1 = NX
      GOTO 212
C        FORM YN(J),J=1,JNM
  220 READ(5,201) YN(1)
      N1 = 1
  222 READ(5,201) DYN,NNY
      IF(NNY .LT. 0)GOTO 230
      NY = N1 + NNY
      N1 = N1+1
      DO 225 L=N1,NY
      YN(L) = YN(L-1) + DYN
```

```
  225 CONTINUE
      N1 = NY
      GOTO 222
C           FORM ZN(K),K=1,KNM
  230 READ(5,201) ZN(1)
      N1 = 1
  232 READ(5,201) DZN,NNZ
      IF(NNZ .LT. 0)GOTO 240
      NZ = N1+NNZ
      N1 = N1+1
      DO 235 L=N1,NZ
      ZN(L) = ZN(L-1) + DZN
  235 CONTINUE
      N1 = NZ
      GOTO 232
C           RECIPIENT GRID FORMED.  CHECK NX, NY, AND NZ.
  240 IF(NX.EQ.INM .AND. NY.EQ.JNM .AND. NZ.EQ.KNM)GOTO 245
      WRITE(6,241)NX,INM,NY,JNM,NZ,KNM
      STOP' ABORT.  STRTNU 240.  GRID INPUT COUNTS DISAGREE'
  241 FORMAT(' ABORT. STRTNU 240. NX.NE.INM OR NY.NE.JNM OR NZ.NE.KNM'
     +  /,'NX,INM,NY,JNM,NZ,KNM=',3(2I5,5X))
      END
      SUBROUTINE NXTIM(IEND)
C           FIND NEXT INPUT TIME DUMP.
C           IF ITIME = 0, FIND CYCLE WITH DIDI(3) = CSTART.
C           SET IEND,CYCLEI,TIMEIN.
C             IEND = 0, DATA AVAILABLE.
C             IEND = 1, NORMAL END OF DATA.  DIDI(I) = 666.0
C             IEND = -1, COULD NOT FIND NEXT TIME DUMP.
C
C            BYPASS THE Z-BLOCK
C            READ VERTEX RECORDS, FIND VOLUME CONTRIBUTIONS AND CELL
C            CENTERS.
C
*CALL CDIN1
*CALL COCOM
C
      DIMENSION DIDI(4)
C
      IEND = 0
    1 NTRY = 0
   10 READ(IN9)(DIDI(I),I=1,4)
      IF(EOF(IN9) .NE. 0)GOTO 20
C            CHECK FOR NORMAL FILE TERMINATION WITH DIDI(I)=666.0.
      IF(ABS(DIDI(1) - 666.0) .LT. 0.001
     +  .AND. ABS(DIDI(2) - 666.0) .LT. 0.001) GOTO 30
C           CHECK FOR START OF TIME DUMP.
      IF(ABS(DIDI(1) - 555.0) .GT. 0.001
     +  .OR. ABS(DIDI(2) - PROBIN) .GT. 0.00001)GOTO 10
C           WE HAVE READ A HEADER RECORD FOR A HULL TIME DUMP.
      IF(ITIME .GT. 0)GOTO 40
C           INITIATION.  FIND STARTING CYCLE.
      IF(ABS(DIDI(3) - CSTART) .LT. 0.1)GOTO 40
C            NOT STARTING CYCLE.
      IF(DIDI(3) -CSTART .LT. 0.0)GOTO 1
C           REQUESTED STARTING CYCLE NOT PRESENT.
      WRITE (6,15)DIDI(3),CSTART
      IEND = -1
      STOP'  ** ABORT IN NXTIM.  CANNOT FIND CSTART.'
   15 FORMAT(//' ** HAVE READ CYCLE',I5,' WITHOUT FINDING CSTART =',I5)
   20 NTRY = NTRY + 1
      IF(NTRY .LE. 10)GOTO 10
C           CANNOT READ ANY MORE INPUT.
      IEND = -1
```

```
      RETURN
C             NORMAL END OF HULL RESTART FILE.
   30 IEND = 1
      RETURN
C              HEADER RECORD READ.  START PROCESSING.
   40 CYCLEI = DIDI(3)
      TIMEIN = DIDI(4)
C              BYPASS THE Z-BLOCK
      READ(IN9)DUMMY
C              READ VERTEX RECORDS.
      IF(IGEOM .EQ. 2)GOTO 100
C             3D VERTEX INPUT PURPOSELY OMITTED UNTIL LATER.
      WRITE(6,45)IGEOM
      STOP'  ** ABORT.  STRTIM 45.  WRONG IGEOM.'
   45 FORMAT(//' ** ABORT.  STRTIM 45.  NO CODING FOR IGEOM =',I2)
C            READ 2D VERTICES.  CHECK SUITABILITY.
  100 READ(IN9)(XI(I),I=2,IIMAX1),(YI(J),J=1,IJMAX1)
      XI(1) = 0.0
C            CHECK THAT INPUT SPACE INCLUDES OUTPUT SPACE.
      IF(ICONV .EQ. 3)GOTO 110
      IF(ICONV .EQ. 6)GOTO 120
C            INPUT AND OUTPUT BOTH 2D.
      IF(XN1 .GE. XI(1) .AND. XNM .LE. XI(IIMAX)
     +   .AND. YN1 .GE. YI(1) .AND. YNM .LE. YI(IJMAX))GOTO 140
      GOTO 130
C            ICONV = 3.  2D CYLINDRICAL INPUT, 3D CARTESIAN OUTPUT.
  110 DSQMAX = AMAX1(XN1*XN1 + YN1*YN1,XN1*XN1 + YNM*YNM,
     +        XNM*XNM + YN1*YN1,XNM*XNM +YNM*YNM)
      IF(DSQMAX .LE. XI(IIMAX)**2
     +   .AND. ZN1 .GE. YI(1) .AND. ZNM .LE. YI(IJMAX))GOTO 140
      GOTO 130
C            ICONV = 6.  2D CARTESIAN INPUT AND 3D CARTESIAN OUTPUT.
  120 IF(XN1 .GE. XI(1) .AND. XNM .LE. XI(IIMAX)
     +   .AND. ZN1 .GE. YI(1) .AND. ZNM .LE. YI(IJMAX))GOTO 140
      GOTO 130
C             INPUT GRID DOES NOT INCLUDE OUTPUT BOUNDARIES, ABORT.
  130 WRITE(6,135)
      STOP'  ** ABORT. NXTIM 135.  GRID INADEQUATE.'
  135 FORMAT ('  ** ABORT.  NXTIM 135.  OUTPUT OUTSIDE DONOR BOUNDARY.')
C             GRID OK.  SET DELYI(J) = Y(J+1)-Y(J).
C                       SET DELXI(I) = VOLUME CELL(I,J)/DELYI(J).
C                       SET X(I) AND Y(J) TO CELL CENTERS.
  140 DO 150 I=1,IIMAX
      DELXI(I) = XI(I+1) - XI(I)
      XI(I) = 0.5*(XI(I+1) + XI(I))
      IF(IGEOM .EQ. 1)GOTO 150
C           INPUT GROMETRY IS 2D CYLINDRICAL.
      DELXI(I) = 6.2831853*XI(I)*DELXI(I)
  150 CONTINUE
      DO 160 J=1,IJMAX
      DELYI(J) = YI(J+1) - YI(J)
      YI(J) = 0.5*(YI(J) + YI(J+1))
  160 CONTINUE
      RETURN
      END
      SUBROUTINE NXTBLK(JP,IHB)
C         THIS MOVES THE DESIRED BLOCK OF INPUT HYDRO DATA INTO HYDROI.
C                 THIS SUBPROGRAM ASSUMES DONOR HULL IS 2D.
C         IHB IS THE NUMBER OF THE PRESENT BLOCK.
C         JP IS THE DESIRED ROW.
C
*CALL CDIN1
*CALL CDCOM
C
```

```
C          IROWPB IS INPUT ROWS PER BLOCK.
C          HYDROI IS THE ARRAY FOR THE INPUT HYDRO DATA.
C          INHPB IS THE HYDRO VARIABLES IN A BLOCK.
C             JLAST IS THE LAST ROW IN THE PRESENT BLOCK.
C
      JLAST = IROWPB*IHB
C             IS JP IN A HIGHER BLOCK?
      IF(JP .GT. JLAST)GOTO 30
C             IS JP IN THE PRESENT BLOCK?
      IF(JP .GT. JLAST-IROWPB)RETURN
C            MOVE TO BEGINNING OF PRESENT BLOCK.
      BACKSPACE (IN9)
C            MOVE TO BEGINNING OF BLOCK IHB-1.
   10 BACKSPACE(IN9)
      IHB = IHB-1
      JLAST = JLAST - IROWPB
C            IS ROW JP IN A LOWER BLOCK?
      IF(JP .LE. JLAST-IROWPB)GOTO 10
C             JP IS IN THIS BLOCK.  READ HYDRO DATA.
   20 READ(IN9)(HYDROI(L),L=1,INHPB)
      RETURN
C             CHECK FOR ROW JP IN THE NEXT HIGHER BLOCK.
   30 IHB=IHB+1
      JLAST = JLAST + IROWPB
C           IS JP IN THIS BLOCK?
      IF(JP .LE. JLAST)GOTO 20
C          JP GT JLAST, SO BYPASS HYDRO BLOCK.
      READ(IN9)DUMMY
      GOTO 30
      END
      SUBROUTINE WRITPL(IBND,NF,X,XS,Y,NY,YS,Z,NZ,ZS,BOUND,NL)
C          THIS ROUTINE STORES THE OUTPUT DATA FOR A PLANE IN A FILE.
C            STORE 1 ROW AT EACH ENTRANCE.
C            TEMPORARY FORMAT.  FINAL STORAGE IN REWRIT.  (8/08/84)
C          THE NOTATION IS LIKE AN X-PLANE, BUT ROUTINE IS FOR ALL 3.
C               X-PLANE ORDER (X,Y,Z)
C               Y-PLANE ORDER (Y,X,Z)
C               Z-PLANE ORDER (Z,X,Y)
C         IBND IS FILE COUNT USED HERE TO TELL INITIAL ENTRY.
C         NF IS THE FILE NUMBER
C         X IS LOCATION OF THE PLANE.
C         XS IS THE SHIFT IN THE PLANE LOCATION.
C         Y IS THE ARRAY OF COLUMN POSITIONS IN THE PLANE.
C         NY IS THE NUMBER OF COLUMNS.
C         YS IS THE SHIFT IN COLUMN POSITIONS.
C         Z IS THE ARRAY OF ROW POSITIONS.
C         NZ IS THE NUMBER OF ROWS.
C         ZS IS THE SHIFT IN ROW POSITIONS.
C         BOUND IS THE LOCATION OF THE HYDRO DATA FOR THE PLANE.
C         NL IS THE AMOUNT OF HYDRO DATA FOR THE PLANE.
C
*CALL CDCOM
      DIMENSION Y(1),Z(1),BOUND(1)
      DIMENSION TBND(6)
      DATA TBND/6*-999999.999/
C
C           SHIFT DATA FOR OUTPUT
      TIMEIN = TIMEIN - TSH
      X = X - XS
      IF(YS .EQ. 0.0)GOTO 30
      DO 20 J=1,NY
      Y(J) = Y(J) - YS
   20 CONTINUE
   30 IF(ZS .EQ. 0.0)GOTO 45
```

```
      DO 40 K=1,NZ
      Z(K) = Z(K) - ZS
   40 CONTINUE
   45 IF(TIMEIN .EQ. TBND(IBND))GOTO 50
      TBND(IBND) = TIMEIN
      WRITE(NF) 555.0,PROBIN,TIMEIN,X,NNH,NY,NZ
      WRITE(NF) (Y(J),J=1,NY),(Z(K),K=1,NZ)
   50 WRITE(NF) (BOUND(L),L=1,NL)
C       SHIFT DATA BACK.
      TIMEIN = TIMEIN + TSH
      X = X + XS
      IF(YS .EQ. 0.0)GOTO 70
      DO 60 J=1,NY
      Y(J) = Y(J) + YS
   60 CONTINUE
   70 IF(ZS .EQ. 0.0)GOTO 90
      DO 80 K=1,NZ
      Z(K) = Z(K) + ZS
   80 CONTINUE
   90 RETURN
      END
      SUBROUTINE FINDI(I,ZP,YI,IM)
C                FIND I SUCH THAT      I = 1     IF ZP LT YI(2)
C                                      I = IM-1  IF ZP GE YI(IM-1)
C             OR        YI(I) < ZP <= YI(I+1).
C
C             I IS THE CURRENT POINTER.
C            ZP IS A VARIABLE (FOR 3D OUTPUT, THE LOCATION OF A PLANE).
C            YI IS A TABLE (AN ARRAY OF CELL CENTERS FROM OLD HULL RUN).
C            IM IS THE NUMBER OF ENTRIES IN THE TABLE.
C
      DIMENSION YI(1)
C
C                 FIRST, CHECK THAT 0 < I < IM.
      IF(I .LT. 1 .OR. I .GE. IM-1)I=1
C                IS ZP > YI(I)?
      IF(ZP .GT. YI(I))GOTO 20
C                 ZP <= YI(I).  CHECK FOR I = 1, THEN REDUCE I.
   10 IF(I .EQ. 1)GOTO 30
      I = I - 1
C                 IS ZP <= YI(I)?
      IF(ZP .LE. YI(I))GOTO 10
      GOTO 30
C                 ZP > YI(I).  CHECK FOR I = IM-1.
   20 IF(I .EQ. IM-1)GOTO 30
C                  CHECK FOR ZP <= YI(I+1).
      IF(ZP .LE. YI(I+1))GOTO 30
      I = I + 1
      GOTO 20
C                 DESIRED I FOUND.  RETURN
   30 RETURN
      END
      SUBROUTINE REWRIT(IBND,NF)
C          SUBROUTINE TO COMBINE TIMES AND PRODUCE FINAL BOUND9 FILES.
*CALL CONU1
C
C          WRITTEN AS IF FOR AN X PLANE, BUT FOR ANY 3D BOUND9 PLANE.
      DIMENSION INITPR(6),LASTPR(6),INITRD(6),LASTRD(6)
      NAMELIST/TABDAT/INITPR,LASTPR,INITRD,LASTRD
      DATA ISW,INITPR,LASTPR,INITRD,LASTRD/0,6*1,6*0,6*1,6*10/
      IF(ISW .GT.0)GOTO 10
C          INITIAL ENTRY.  READ TABULATION CONTROLS.
      ISW = 1
      READ(5,TABDAT)
```

```
      WRITE(6,TABDAT)
C          READ FILE NF.  PUT IN FILE 1 AND, SKIPPING FIRST DUMP, IN 2.
   10 REWIND NF
      REWIND 1
      REWIND 2
      SW = 0.0
      WRITE(6,110)NF
  110 FORMAT('1  BEGIN COMBINING TIMES FOR FILE',I5)
   20 READ(NF)HEAD1,PROBIN,TI1,X,NNH,NY,NZ
      IF(EOF(NF) .NE. 0)GOTO 90
      IF(ABS(HEAD1-666.0) .LT. 0.001)GOTO 40
      IF(ABS(HEAD1-555.0) .GT. 0.001)GOTO 91
      WRITE(1) HEAD1,PROBIN,TI1,X,NNH,NY,NZ
      IF(SW .GT. 0.0)WRITE(2) HEAD1,PROBIN,TI1
      READ(NF)(YN(J),J=1,NY),(ZN(K),K=1,NZ)
      NL = NNH*NY
      DO 30 K=1,NZ
      READ(NF) (BND(L),L=1,NL)
      WRITE(1)(BND(L),L=1,NL)
      IF(SW .GT. 0.0) WRITE(2) (BND(L),L=1,NL)
   30 CONTINUE
      SW = 1.0
      GOTO 20
C         FILE NF HAS BEEN READ AND STORED IN FILES 1 AND 2.
C         NOW WE COLLATE DUMPS WITH PROPER HEADINGS.
   40 WRITE(1)666.0,666.0,666.0,666.0,666,666,666
      WRITE(2)666.0,666.0,666.0
      REWIND 1
      REWIND 2
      REWIND NF
      NOUT = 1
   50 READ(1) HEAD1,PROBIN,TI1,X,NNH,NY,NZ
      IF(ABS(HEAD1-666.0) .LT. 0.001)GOTO 92
      IF(ABS(HEAD1 - 555.0) .GT. 0.001)GOTO 92
      READ(2) HEAD2,DUM,TI2
      IF(ABS(HEAD2-666.0) .LT. 0.001)GOTO 70
      IF(HEAD1 .NE. HEAD2 .OR. DUM .NE. PROBIN)GOTO 93
      WRITE(NF)HEAD1,PROBIN,TI1,TI2,X,NNH,NY,NZ
      IF(INITPR(IBND) .LE. NOUT .AND. LASTPR(IBND) .GE. NOUT)
     + WRITE(6,900) HEAD1,PROBIN,TI1,TI2,X,NNH,NY,NZ
      WRITE(NF) (YN(J),J=1,NY),(ZN(K),K=1,NZ)
      IF(INITPR(IBND) .LE. NOUT .AND. LASTPR(IBND) .GE. NOUT)
     + WRITE(6,901) (YN(J),J=1,NY),(ZN(K),K=1,NZ)
      DO 60 K=1,NZ
      READ(1) (BND(L),L=1,NL)
      READ(2) (BND2(L),L=1,NL)
      WRITE(NF)(BND(L),L=1,NL),(BND2(L),L=1,NL)
      IF(INITPR(IBND) .LE. NOUT .AND. LASTPR(IBND) .GE. NOUT .AND.
     +  INITRO(IBND) .LE. K .AND. LASTRO(IBND) .GE. K)
     +  WRITE(6,902) K,(BND(L),L=1,NL),(BND2(L),L=1,NL)
   60 CONTINUE
      NOUT = NOUT + 1
      GOTO 50
C          FILE NF NOW STORED IN FINAL FORM EXCEPT FOR TERMINAL RECORD.
   70 WRITE(NF) 666.0,666.0,666.0,666.0,666.0,666,666,666
      RETURN
C              ABORT PRINTS
   90 WRITE(6,190)NF
      STOP' ABORT.  REWRIT 90.  NO DATA ON FILE.'
  190 FORMAT(//' ABORT.  REWRIT 90.  NO DATA ON FILE',I5)
   91 WRITE(6,191)NF,HEAD1,PROBIN,TI1,X,NNH,NY,NZ
      STOP' ABORT.  REWRIT 91.  MIX UP IN FILE NF.'
  191 FORMAT(//' ABORT. REWRIT 91. MIXUP IN FILE NF=',I5/
     +    ' HEAD1,PROBIN,TI1,X,NNH,NY,NZ =',/,1P4E15.7,3I10)
```

```
 92 IT = 1
    GOTO 94
 93 IT = 2
 94 WRITE(6,194)IT,HEAD1,PROBIN,TI1,X,NNH,NY,NZ,HEAD2,DUM,TI2
    STOP' ABORT.  REWRIT 94.  MIXUP IN FILE 1 OR 2.'
194 FORMAT('ABORT.REWRIT 94.  MIXUP IN FILES 1 OR 2.  IT=',I5/,
   +  ' HEAD1,PROBIN,TI1,X,NNH,NY,NZ ='/,1P4E15.7,3I10/,
   +  ' HEAD2,DUM,TI2 =',1P3E15.7)
900 FORMAT(20X,'HEADER'/,1P5E15.7,3I10)
901 FORMAT(10X,'VERTICES'/,(1P6E15.7))
902 FORMAT(10X,'PART OF HYDRO DATA STARTING AT ROW ',I5/,(1P5E15.7))
    END
```

APPENDIX B

LISTINGS FOR KEEL AND HULL

Appendix B has listings of the runstreams for KEEL and HULL for problem 8416.20. The listing of the HULL runstream is followed by listings of the correction file for HULL, file HULCORR, and the changes for BOUND9 input, file NEWBND9. These two files are in MFA.

Note that file HULL8405P09, the same file that was used in HULLUP, is attached for initial data in the KEEL run. (This is actually only an illustrative gesture for this problem. The entire 3-D region was ambient initially.) Note also in the HULL runstream, that files for the four BOUND9 input sides are attached to appropriate local files and the MFA files NEWBND9 and HULCORR are included for UPDATE. These attaches and the declaring of BOUND9 boundaries and the GENERATE AIR FIREIN HULL sequence in KEEL are the only additional input at this stage. A lot of planning is needed to set up the HULLUP run to produce the desired input in the files for the BOUND9 boundary input.

KEEL Changes for BOUND9 Boundaries

There were no changes needed to input for KEEL for the new type 9 boundaries. A few minor changes were made in subroutine HULLIN to add some problem aborts with explanatory printing and to use the ambient atmosphere for normalizing. If a region from a 2-D cylindrical run is to be mapped into a 3-D (or new 2-D) region, the restart file from the donor run must be attached as TAPE 44 and the GENERATE AIR FIREIN HULL input sequence must be used for a region that includes all cells in the new space that overlap the non-ambient part of the 2-D donor space at the initiation time. The initiation time is the first time on the donor restart file greater than the input time T. T is replaced by this time. The user may set the boundary conditions in KEEL, but they have no affect until the HULL run.

If one must, the reference location for the new run can be shifted through input of XOB, YOB, and HOB in KEEL. Time cannot be shifted in KEEL without some recoding.

HULL Changes for BOUND9 Boundaries

The coding changes to use the BOUND9 boundaries for 3-D HULL runs is in the MFA file NEWBND9, ID=JDW. This coding may be made a permanent part of the HULL file when the check-out for these boundaries is completed. If one wishes to use this coding with a 3-D HULL run, file NEWBND9 must be picked up in the SCOPE2 runstream and READ as part of the UPDATE input. The user must also have available, and attach, an input file for each BOUND9 boundary. Any of the 6 boundary planes may be declared a type 9 boundary. The NEWBND9 coding assumes:

TAPE31 - Left boundary at XO

TAPE32 - Right boundary at $X_{IMAX} - 1/2\ DX_{IMAX}$

TAPE33 - Bottom boundary at ZO

TAPE34 - Top boundary at Z_{KMAX} - 1/2 DZ_{KMAX}

TAPE35 - Aft boundary at Y0

TAPE36 - Fore boundary at Y_{JMAX} - 1/2 DY_{JMAX}

Notice that the planes of imposed data for the low index boundaries are on the boundaries and the planes of imposed data for the high index planes are through the centers of the external plane of cells.

The data on these files is a series of "dumps" of unformatted data for successive time intervals. Each dump is a header record, a grid record, and the hydro data at both ends of the time interval one row per record.

The header record for an X-plane is 8 numbers, 5 reals and 3 integers:

HEAD1, OLPROB, TI1, TI2, XIN, NNH, JN, KN

where,

HEAD1 = 555.0
OLPROB = The donor problem number
TI1 = The first time
TI2 = The last time
XIN = X value for the plane
NNH = 5, the number of hydro values per point
JN = The number of columns of data
KN = The number of rows of data.

The grid record for an X-plane is:

(YIN(J), J=1,JN), (ZIN(K), K=1,KN).

This defines a JN x KN grid of points on the X-plane at which hydro data will be defined.

This is followed by KN records of hydro data for K=1,2,...,KN:

(((HB(L,J,N),N=1,NNH),J=1,JN),L=1,2).

The 5 hydro values are in order

U - The X velocity component (cm/sec)
V - The Y velocity component (cm/sec)
W - The Z velocity component (cm/sec)
E - Internal specific energy (ergs/g)
ρ - Density (g/cm^3)

The 5 hydro values are given for J=1,2,...,JN for time TI1 (L=1) and then for time TI2.

For a Y-plane, the boundary plane location is YIN and there are IN columns of data at XIN(I), I=1, IN on KN rows at ZIN(K), K=1, KN.

For a Z-plane, the boundary plane is at ZIN and there are IN columns of data at XIN(I), I=1, IN on JN rows at YIN(J), J=1, JN.

The program interpolates in this input data in time and space on the boundary plane. Any attempt to extrapolate in either time or space causes a program abort. The program also aborts if the plane's location does not agree with the requested position.

There is no further new input other than declaring the type 9 boundaries. There are a few coded in values that might be changed: Arrays are set for a maximum of 200 words for IN, JN, and KN, and 4000 words for the hydro data. This uses 400 words of SCM and 4000 words of LCM for each BOUND9 boundary. It might be necessary to increase, or decrease, these arrays. Also, the programming allows shifting in time and position by TSHIFT, XSHIFT, YSHIFT, and ZSHIFT, all of which are set to zero. If needed, these could be changed.

There is a separate subroutine to obtain the data for each of the 6 boundaries. Each is compiled, and called in HULL, if the corresponding boundary is declared type 9 and the dimension is 3 (e.g., LBOUND=9 and DIMENSION=3).

** RUNSTREAM FOR KEEL **

```
WORTMAN(STMFZ,T1000,P2) KEEL RUN,     PROB 8416.20      8/15/84
ACCOUNT(******)     WORTMAN  B309  X6028
COPYSBF(INPUT,OUTPUT)
REWIND(INPUT)
COMMENT.  SPHERICAL RESERVOIR (50 PSI SHOCK IN STINF).  PROB 8416.20.
COMMENT.  START 3D RUN FROM FROM CYCLE 0 OF PROB 8405.09.
COMMENT.   THIS IS PROBLEM 8406.20 BEING RERUN WITH NEW HULLUP AND BND9.
ATTACH(TAPE44,HULL8405P09,ID=JDW)
DISPOSE,OUTPUT,ST=MFAIEI,UN=RJE1401,*PR.
ATTACH(PL,HULL,ID=JDW)
ATTACH(HULLIB,HULLIB,ID=JDW)
LIBRARY(HULLIB)
REQUEST(TAPE4,*PF)
REQUEST(TAPE9,*PF)
UPDATE(P=PL,Q,L=1)
PLANK.
POST(COMPILE)
MAP(OFF)
FTN,I=SAIL,B=KEEL,PL=50000,OPT=2,SL=0,R=0,EL=F)
RETURN(PL,SAIL)
KEEL.
CATALOG(TAPE4,HULL8416P20, ID=JDW)
CATALOG(TAPE9, STA8416P20, ID=JDW)
*EOR
*LIMIT 10000
*COMPILE HULL
*IDENT  KCHNG
*NOABBREV
*DELETE   KEEL.4249,KEEL.4251
C               THE OLD HULL INPUT FILE AS UNIT 44.
C                        THE DATA ARE READ INTO ECS AND THE UNIT
*INSERT   KEEL.4255
      COMMON/RDWR/EOFF,ERR
*INSERT   KEEL.4272
      IF(EOFF .NE. 0)STOP' HULLIN.  NO DATA ON NFILOT (=44).'
*INSERT   KEEL.4281
      IF(EOFF .NE. 0)STOP' HULLIN.  NO MORE DATA ON NFILOT (=44).'
*INSERT   KEEL.4305
      IF(EOFF .NE. 0)STOP' HULLIN.  NO MORE DATA ON NFILOT (=44).'
*INSERT   KEEL.4327
      IF(EOFF .NE. 0)STOP' HULLIN.  NO MORE DATA ON NFILOT (=44).'
*DELETE  KEEL.4341
C          NORMALIZE USING VALUES FROM ATMOSP.
*COMPILE KEEL
*EOR
KEEL
PROB 8416.20
LBOUND=9
RBOUND=9
BBOUND=0
TBOUND=9
ABOUND=0
FBOUND=9
ATMOS=5
DIMEN=3
EOS=2
GEOM=2
HEADER
80 CM GRID.  START FROM 8405.09.  8416.20.
RELGAM=1.4
RELRHO=0.00120412
RELSIE=2.10374E9
```

```
RELPO=1.01325E6
IMAX=9
JMAX=7
KMAX=8
T=0.02
PTSTOP=0.05
NSTN=33
MESH
X0=880.0
NX=9 DX=80.0
NY=7 DY=80.0
NZ=8 DZ=80.0
GENERATE   AIR
FIREIN       HULL
SPHERE  RADIUS=1140.0 XCENTER=0.0 YCENTER=0.0 ZCENTER=1200.0
STATIONS
XS=990    YS=30    ZS=30
XS=1250   YS=30    ZS=30
XS=1500   YS=30    ZS=30
XS=990    YS=30    ZS=500
XS=1250   YS=30    ZS=500
XS=1500   YS=30    ZS=500
XS=990    YS=30    ZS=590
XS=1250   YS=30    ZS=590
XS=1500   YS=30    ZS=590
XS=990    YS=280   ZS=30
XS=1250   YS=280   ZS=30
XS=1500   YS=280   ZS=30
XS=990    YS=280   ZS=500
XS=1250   YS=280   ZS=500
XS=1500   YS=280   ZS=500
XS=990    YS=280   ZS=590
XS=1250   YS=280   ZS=590
XS=1500   YS=280   ZS=590
XS=990    YS=520   ZS=30
XS=1250   YS=520   ZS=30
XS=1500   YS=520   ZS=30
XS=990    YS=520   ZS=500
XS=1250   YS=520   ZS=500
XS=1500   YS=520   ZS=500
XS=990    YS=520   ZS=590
XS=1250   YS=520   ZS=590
XS=1500   YS=520   ZS=590
XS=1184   YS=500   ZS=30
XS=1184   YS=500   ZS=500
XS=1184   YS=500   ZS=590
XS=1560   YS=30    ZS=500
XS=1560   YS=280   ZS=500
XS=1560   YS=520   ZS=500
END
```

** RUNSTREAM FOR HULL **

```
WORTMAN(STMFZ,T50,P1) HULL,     PROB 8416.20    8/15/84
ACCOUNT(******)  B309 X6028
COPYSBF(INPUT,OUTPUT)
REWIND(INPUT)
COMMENT.      RERUN  OF 8406.20 TO TEST NEW BOUND9 CODING.
COMMENT.  SPHERICAL RESERVOIR (50 PSI STINF).  PROB 8416.20.
COMMENT.   3D RUN STARTED FROM PROB 8405.09 THRU FIREIN/HULLIN.
COMMENT.   4 BOUNDARIES SET TO TYPE 9.  INPUT FROM HULLUP.
ATTACH(TAPE31,P841620LB,ID=JDW)
ATTACH(TAPE32,P841620RB,ID=JDW)
ATTACH(TAPE34,P841620TB,ID=JDW)
ATTACH(TAPE36,P841620FB,ID=JDW)
COMMENT.     UPDATE CHANGES (CORRECTIONS) FOR HULL FROM FILE HULCORR.
BEGIN(GETMFA,FILE,LF=HULCORR,PF=HULCORR,UN=JDW)
COMMENT.      UPDATE CHANGES FROM FILE NEWBND9 IN MFA TO BND9 IN MFZ.
BEGIN(GETMFA,FILE,LF=BND9,PF=NEWBND9,UN=JDW)
DISPOSE,OUTPUT,ST=MFAIEI,UN=RJE1401,*PR.
ATTACH(T4,HULL8416P20, ID=JDW)
ATTACH(T9, STA8416P20, ID=JDW)
REQUEST(TAPE4,*PF)
COPYP(T4,TAPE4,1000)
RETURN(T4)
REWIND(TAPE4)
REQUEST(TAPE9,*PF)
COPYP(T9,TAPE9,1000)
RETURN(T9)
REWIND(TAPE9)
ATTACH(HULLIB,HULLIB,ID=JDW)
LIBRARY(HULLIB)
ATTACH(PL,HULL,ID=JDW)
UPDATE(P=PL,Q,L=1)
SWITCH(6,ON)
PLANK.
SWITCH(6,OFF)
POST(COMPILE)
MAP(PART)
FTN(A,I=SAIL,B=HULL,OPT=2,SL=0,R=0,EL=F,LCM=I)
RETURN(PL,SAIL,COMPILE)
HULL.
EXIT(U)
CATALOG(TAPE4,HULL8416P20, ID=JDW)
CATALOG(TAPE9, STA8416P20, ID=JDW)
*EOR
*LIMIT 10000
*READ HULCORR
*READ BND9
*COMPILE HULL
*EOR
HULL
PROB 8416.20
CYCLE=0
INPUT
STABF=0.05
MRELER=1.0E-6
CSTOP=200
END
```

** HULL CORRECTIONS. FILE HULCORR **

```
*IDENT   BNDCORR
*/       CORRECT AN ERROR:  CHANGE LBOUND TO ABOUND
*DELETE   HULL.11059
*KEEPTO   *1   ABOUND4 AND LAMB
*/      AMEND SHKWV CALLS:
*/      MAKE BOUNDARIES MORE GENERAL.  NO Y=0.0 OR X=0.0 FOR 3D.
*/      FIRST PHASE, TIME =T FOR EXTERNAL CELL, =T + 0.5*DT ON BOUNDARY.
*/      SECOND PHASE, TIME = T + DT.
*DELETE   HULL.11062
      CALL SHKWV(T+0.5*DT,DISX,YO,ALT,UAB,VAB,WAB,EAB,RHOAB)
*DELETE   HULL.11389
      CALL SHKWV(T+0.5*DT,DISX,DISY,ALT,UBB,VBB,WBB,EBB,RHOBB)
*DELETE   HULL.11634
      CALL SHKWV(T+0.5*DT,XO,DISY,ALT,ULB,VLB,WLB,ELB,RHOLB)
*DELETE   HULL.12444
      CALL SHKWV(T+DT,DISX,YO,ALT,UAB,VAB,WAB,EAB,RHOAB)
*DELETE   HULL.12600
      CALL SHKWV(T+DT,DISX,DISY,ALT,UBB,VBB,WBB,EBB,RHOBB)
*DELETE   HULL.12815
      CALL SHKWV(T+DT,XO,DISY,ALT,ULB,VLB,WLB,ELB,RHOLB)
*DELETE   HULL.13200
      CALL SHKWV2(T+DT,DISX,DISY,UBB,VBB,EBB,RHOBB)
*DELETE   HULL.13329
      CALL SHKWV2(T+DT,DISX,DISY,ULB,VLB,ELB,RHOLB)
*IDENT   DTFR2D
*/     BETTER 2D TIME STEP.  FIXED GAMMA ONLY.
*INSERT   HULL.4074
C        COMPUTE CS FOR CELL(I,J)
C        THIS IS FOR FIXED GAMMA.   OTHERWISE, USE EQ. OF STATE.
      EFORCS = H(N1+4)-0.5*(H(N1+2)**2 + H(N1+3)**2)
      CS = SQRT(GAMMA*(GAMMA-1.0)*EFORCS)
C      END CS INSERT.
*IDENT   HULLCOR
*NOABBREV
*/       ERRORS IN SHORE CELL OPTION (CORRECTION THANKS TO PAUL LEWIS)
*/       POSSIBILITY OF ALL FLUID CELLS OMITTED.
*INSERT   HULL.4116
C        CHECK FOR RIGHT CELL ALL FLUID
      IF(LSR .EQ. 0) GOTO 105
*DELETE   HULL.4127
  105 RHOR = H(NR1+5)/VOLR
*INSERT   HULL.4145
C        CHECK FOR CELL ABOVE ALL FLUID
      IF(LSA .EQ. 0) GOTO 115
*DELETE   HULL.4157
  115 RHOA = H(NA1+5)/VOLA
*DELETE HULL.11256
      IF(LSI .LE. 0)GOTO 6
      VOL = VOL*0.5
*INSERT HULL.11260
    6 CONTINUE
*INSERT HULL.11517
      IF(LSI .LE. 0)GOTO 5
*INSERT HULL.11521
    5 CONTINUE
*/       END OF THE PAUL LEWIS CORRECTION.
*/         TBR1 HAS DUPLICATE ADDRESS IF NM.GT.1
*DELETE HULL.12336,HULL.12337
      DO 17 IM=1,NM
   17 H(NA+NHM+IM) = H(NTBR+NHM+IM)*DEN
*/       REMOVE AN ANOMALY BELOW HULL.13661
*DELETE HULL.13661,HULL.13662
```

```
*KEEPTO TB352D RAD1 AND LBOUND NEO
*/        OBVIOUS MISSPELLED NAME AT HULL.17495
*DELETE HULL.17495
$    5 AWT(SHALE,5)/47.90/,      FRACN(SHALE,1)/0.0055885/,
*/         *KEEPTO DOES NOT COUNT '='.
*/        NOTE..   THINK BBOUND=1AND2 NOT CHECKED, WORTHLESS IF CHECKED.
*DELETE    HULL.2039
*KEEPTO    *4       BBOUNDO
```

** BOUND9 CODING. FILE NEWBND9 **

```
*IDENT BOUND9
*/    XBOUND=9 IS INPUT FROM A PREVIOUS HULL RUN AT XBOUND.
*/      MAXIMUM INPUT PLANE A 76X76 MESH OF POINTS.     7/24/84.
*INSERT    HULL.704
*KEEPTO    ENDBND9    LBOUND9 OR RBOUND9 OR BBOUND9 OR TBOUND9 OR +
                 ABOUND9 OR FBOUND9
*INCLUDE DEEP-SIX REZONE
*LABEL    ENDBND9
*/       FILES FOR XBOUND=9.
*INSERT    HULL.1146
*KEEPTO    *1    LBOUND9
      +   TAPE31,TAPE61,
*KEEPTO    *1    RBOUND9
      +   TAPE32,TAPE62,
*KEEPTO    *1    BBOUND9
      +   TAPE33,TAPE63,
*KEEPTO    *1    TBOUND9
      +   TAPE34,TAPE64,
*KEEPTO    *1    ABOUND9
      +   TAPE35,TAPE65,
*KEEPTO    *1    FBOUND9
      +   TAPE36,TAPE66,
*INSERT    HULL.8161
*KEEPTO    *1    LBOUND9 OR RBOUND9 OR BBOUND9 OR TBOUND9 OR +
                 ABOUND9 OR FBOUND9
      COMMON/SHIFT/TSHIFT,XSHIFT,YSHIFT,ZSHIFT
*INSERT     HULL.8369
*KEEPTO    *4    LBOUND9 OR RBOUND9 OR BBOUND9 OR TBOUND9 OR +
                 ABOUND9 OR FBOUND9
      TSHIFT = 0.0
      XSHIFT = 0.0
      YSHIFT = 0.0
      ZSHIFT = 0.0
*INSERT     HULL.9250
*KEEPTO    ENDBND9     DIMEN3
*KEEPTO    ENDBND9    LBOUND9 OR RBOUND9 OR BBOUND9 OR TBOUND9 OR  +
                          ABOUND9 OR FBOUND9
C          ADD HB ARRAYS TO LCM.  USE NEXT LARGER MULTIPLE OF 8.
C     !!!!  4000 AS OF 8/21/84.  !!!  ASSUMES MAX(IN,JN,KN) .LE. 200.
      CALL FLECS(IB9ECS)
      IF(ABOUND .EQ. 9) IB9ECS = IB9ECS + 4000
      IF(FBOUND .EQ. 9) IB9ECS = IB9ECS + 4000
      IF(BBOUND .EQ. 9) IB9ECS = IB9ECS + 4000
      IF(TBOUND .EQ. 9) IB9ECS = IB9ECS + 4000
      IF(LBOUND .EQ. 9) IB9ECS = IB9ECS + 4000
      IF(RBOUND .EQ. 9) IB9ECS = IB9ECS + 4000
      CALL SETECS(IB9ECS)
*LABEL     ENDBND9
*INSERT    HULL.11058
*KEEPTO    *1    ABOUND9
      CALL ABND9D3(T+0.5*DT,DISX,YO,ALT,UAB,VAB,WAB,EAB,RHOAB)
*SKIPTO    *4    ABOUND9
*INSERT   HULL.11092
*SKIPTO    STBND9    FBOUND9
*INSERT    HULL.11102
*LABEL     STBND9
*KEEPTO    ENDBND9    FBOUND9
      DISX = X(I) - 0.5*DX(I)
      DISY = Y(JMAX) - 0.5*DY(JMAX)
      ALT = Z(K) - 0.5*DZ(K)
      CALL FBND9D3(T,DISX,DISY,ALT,H(N1+1),H(N1+2),H(N1+3),H(N1+4),
     +                 RHOFB)
```

```
      H(N1+5) = RHOFB*DX(I)*DY(JMAX)*DZ(K)
*LABEL   ENDBND9
*INSERT   HULL.11388
*KEEPTO   *1   BBOUND9
      CALL BBND9D3(T+0.5*DT,DISX,DISY,ALT,UBB,VBB,WBB,EBB,RHOBB)
*SKIPTO   *1   BBOUND9
*INSERT   HULL.11630
*KEEPTO   *1   LBOUND9
      CALL LBND9D3(T+0.5*DT,X0,DISY,ALT,ULB,VLB,WLB,ELB,RHOLB)
*SKIPTO   *4   LBOUND9
*INSERT  HULL.11665
*SKIPTO   STBND9   RBOUND9
*INSERT   HULL.11677
*LABEL    STBND9
*KEEPTO   ENDBND9   RBOUND9
      DISX = X(IMAX) - 0.5*DX(IMAX)
      DISY = Y(J) - 0.5*DY(J)
      ALT = Z(K) - 0.5*DZ(K)
      CALL RBND9D3(T,DISX,DISY,ALT,H(NL+1),H(NL+2),H(NL+3),H(NL+4),
     +        RHORB)
      H(NL+5) = RHORB*DX(IMAX)*DY(J)*DZ(K)
*LABEL   ENDBND9
*INSERT  HULL.12362
*SKIPTO   STBND9   TBOUND9
*INSERT   HULL.12370
*LABEL    STBND9
*KEEPTO   ENDBND9   TBOUND9
      DISX = X(I) - 0.5*DX(I)
      DISY = Y(J) - 0.5*DY(J)
      ALT = Z(KMAX) - 0.5*DZ(KMAX)
      CALL TBND9D3(T,DISX,DISY,ALT,H(NA+1),H(NA+2),H(NA+3),H(NA+4),
     +             RHOTB)
      H(NA+5) = RHOTB*DX(I)*DY(J)*DZ(KMAX)
*LABEL   ENDBND9
*INSERT   HULL.12443
*KEEPTO   *1   ABOUND9
      CALL ABND9D3(T+DT,DISX,Y0,ALT,UAB,VAB,WAB,EAB,RHOAB)
*SKIPTO   *1   ABOUND9
*INSERT   HULL.12455
*SKIPTO   STBND9   FBOUND9
*INSERT   HULL.12467
*LABEL    STBND9
*KEEPTO   ENDBND9   FBOUND9
      DISX = X(I) - 0.5*DX(I)
      DISY = Y(JMAX) - 0.5*DY(JMAX)
      ALT = Z(K) - 0.5*DZ(K)
      CALL FBND9D3(T+DT,DISX,DISY,ALT,H(N1+1),H(N1+2),H(N1+3),H(N1+4),
     +          RHOFB)
      H(N1+4) = H(N1+4) + 0.5*(H(N1+1)**2 + H(N1+2)**2 + H(N1+3)**2)
      H(N1+5) = RHOFB*DX(I)*DY(JMAX)*DZ(K)
*LABEL   ENDBND9
*INSERT   HULL.12599
*KEEPTO   *1   BBOUND9
      CALL BBND9D3(T+DT,DISX,DISY,ALT,UBB,VBB,WBB,EBB,RHOBB)
*SKIPTO   *1   BBOUND9
*INSERT    HULL.12671
*KEEPTO    ENDBND9    FBOUND9
      IF(I .EQ. IMAX)GOTO 5
      DTH = DTH - FEJ(JL)
      DMTH = DMTH - FMJ(JL)
    5 CONTINUE
*LABEL ENDBND9
*INSERT   HULL.12672
*SKIPTO   STBND9   FBOUND9
```

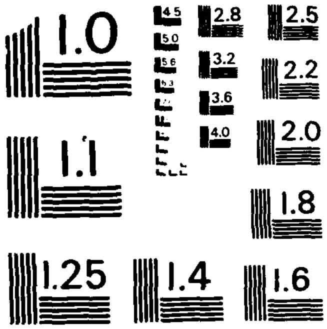
```
*INSERT   HULL.12690
*LABEL    STBND9
*INSERT   HULL.12814
*KEEPTO   *1   LBOUND9
      CALL LBND9D3(T+DT,XO,DISY,ALT,ULB,VLB,WLB,ELB,RHOLB)
*SKIPTO   *1   LBOUND9
*INSERT   HULL.12826
*SKIPTO   STBND9   RBOUND9
*INSERT   HULL.12838
*LABEL    STBND9
*KEEPTO   ENDBND9   RBOUND9
      DISX = X(IMAX) - 0.5*DX(IMAX)
      DISY = Y(J) - 0.5*DY(J)
      ALT = Z(K) - 0.5*DZ(K)
      CALL RBND9D3(T+DT,DISX,DISY,ALT,H(NL+1),H(NL+2),H(NL+3),H(NL+4),
     +             RHORB)
      H(NL+4) = H(NL+4) + 0.5*(H(NL+1)**2 + H(NL+2)**2 + H(NL+3)**2)
      H(NL+5) = RHORB*DX(IMAX)*DY(J)*DZ(K)
*LABEL   ENDBND9
*INSERT   HULL.12890
*KEEPTO   ENDBND9    RBOUND9
      DTH = DTH - FEL
      DMTH = DMTH - FML
*LABEL   ENDBND9
*INSERT   HULL.12898
*SKIPTO   STBND9   RBOUND9
*INSERT   HULL.12911
*LABEL    STBND9
*INSERT   HULL.12980
*KEEPTO   ENDBND9   TBOUND9
      IF(I .EQ. IMAX .OR. J .EQ. JMAX)GOTO 5
      DTH = DTH - FEK(KL)
      DMTH = DMTH - FMK(KL)
    5 CONTINUE
*LABEL   ENDBND9
*INSERT   HULL.12981
*SKIPTO   STBND9   TBOUND9
*INSERT   HULL.12998
*LABEL    STBND9
*INSERT   HULL.13022
*SKIPTO   STBND9   TBOUND9
*INSERT   HULL.13030
*LABEL    STBND9
*KEEPTO   ENDBND9   TBOUND9
      DISX = X(I) - 0.5*DX(I)
      DISY = Y(J) - 0.5*DY(J)
      ALT = Z(KMAX) - 0.5*DZ(KMAX)
      CALL TBND9D3(T+DT,DISX,DISY,ALT,H(NA+1),H(NA+2),H(NA+3),H(NA+4),
     +             RHOTB)
      H(NA+4) = H(NA+4) + 0.5*(H(NA+1)**2 + H(NA+2)**2 + H(NA+3)**2)
      H(NA+5) = RHOTB*DX(I)*DY(J)*DZ(KMAX)
*LABEL   ENDBND9
*INSERT   HULL.11010
*KEEPTO   ENDBND9   LBOUND9 AND DIMEN3
      SUBROUTINE LBND9D3(TH,XH,YH,ZH,U,V,W,E,RHO)
C               BOUNDARY SUBROUTINE FOR LBOUND=9.  INPUT PREPARED BY
C               HULLUP IS ASSUMED ON FILE TAPE31 FOR X = (XIN).
C                  XH SHOULD BE XO.  XH + XSHIFT SHOULD BE XIN.
C
C               KN + 2 RECORDS PER TIME DUMP:
C          HEADER--555.0,OLD PROB,TI1,TI2,XIN,NNH,JN,KN
C          GRID--(YIN(J),J=1,JN),(ZIN(K),K=1,KN)
C             HYDRO DATA NNH/POINT FOR TIMES TI1 AND TI2
C          KN ROWS AT (YIN(J),ZIN(K),J=1,JN) FOR K=1,KN.
```

```
C
      COMMON/SHIFT/TSHIFT,XSHIFT,YSHIFT,ZSHIFT
C             VALUES IN SHIFT SET IN HULLIN.
C
C       FOLLOWING ARRAYS SET TO ACCOMODATE JN .LE. 200.
C          COMMON/HB1/ TO ACCOMODATE LEVEL 2 STATEMENT.
      COMMON/HB1/HB1(4000)
      DIMENSION H(5)
      DIMENSION YIN(200),ZIN(200)
C
      LEVEL 2, HB1
      NAMELIST/HEADIN/HEAD1,OLPROB,TI1,TI2,XIN,NNH,JN,KN
      NAMELIST/GRID/XH,XSHIFT,X,XIN,YH,YSHIFT,Y,YIN,ZH,ZSHIFT,Z,ZIN
C             SET FILE NUMBER.   STATE ARRAY DIMENSIONS.
      DATA IBND,NBND,JMX,KMX,NMX/31,61,200,200,4000/
C        NMX IS ADDED TO LCM IN HULLIN (HULL9250).
C             SET TF1 TO -1.0 FOR INITIAL VALUES.
      DATA TF1/-1.0/
C
C          SHIFT DATA FROM HULL COORDINATES TO BOUNDARY COORDINATES.
      T = TH + TSHIFT
      X = XH + XSHIFT
      Y = YH + YSHIFT
      Z = ZH + ZSHIFT
C          IS THIS AFTER THE INITIAL ENTRY?
      IF(TF1 .GT. -0.9)GOTO 30
      TF1 = 1.0
C            READ FIRST HEADING, GRID, AND HYDRO SET.  CHECK.
      REWIND IBND
      READ(IBND)HEAD1,OLPROB,TI1,TI2,XIN,NNH,JN,KN
      IF(EOF(IBND) .NE. 0 )GOTO 100
      TTEST = 0.00001*(TI2 - TI1)
      XTEST = 0.0000001*XIN
      WRITE(6,5)
    5 FORMAT(//' INITIATE LBOUND9 INPUT.  FIRST HEADER RECORD:')
      WRITE(6,HEADIN)
      IF(ABS(HEAD1 - 555.0) .GT. 0.01)GOTO 110
      NNHYPR = NNH*JN
      NNHYPP = NNHYPR*2
      IF(JN .GT. JMX .OR. KN .GT. KMX .OR. NNHYPP .GT. NMX)GOTO 120
      IF(T+TTEST .LT. TI1)GOTO 130
      OLPRB1 = OLPROB
C          READ GRID FOR INPUT BOUNDARY PLANE.
      READ(IBND) (YIN(J),J=1,JN),(ZIN(K),K=1,KN)
      YTEST = (YIN(2) - YIN(1) )*0.00001
      ZTEST = (ZIN(2) - ZIN(1) )*0.00001
   10 IF(T+TTEST .LE. TI2)GOTO 20
C          FIND THE NEXT HEADER RECORD.
   14 READ(IBND)HEAD1,OLPROB,TI1,TI2,XIN,NNH,JN,KN
      IF(EOF(IBND) .NE. 0)GOTO 160
      IF(OLPROB .NE. OLPRB1 .OR. HEAD1 .NE. 555.0)GOTO 14
C             THIS IS A NEW HEADER RECORD.  BYPASS THE GRID RECORD.
      READ(IBND)DUMMY
      GOTO 10
C             READ HYDRO RECORDS.  STORE IN FILE NBND.
   20 REWIND NBND
      DO 22 K=1,KN
      READ(IBND) (HB1(L),L=1,NNHYPP)
      WRITE(NBND) (HB1(L),L=1,NNHYPP)
   22 CONTINUE
   24 REWIND NBND
C             READ FIRST 2 ROWS FROM FILE NBND.
      READ(NBND) (HB1(L),L=1,NNHYPP)
      READ(NBND) (HB1(NNHYPP+L),L=1,NNHYPP)
```

```
      JIN = 2
      KIN = 2
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
      TDENOM = 1.0/(TI2 - TI1)
   30 IF(T-TTEST .GT. TI2)GOTO 14
      IF(ABS(X - XIN) .GT. XTEST)GOTO 140
      IF(YIN(1) - Y .GT. YTEST .OR. Y - YIN(JN) .GT. YTEST)GOTO 140
      IF(ZIN(1) - Z .GT. ZTEST .OR. Z - ZIN(KN) .GT. ZTEST)GOTO 140
   32 IF(Z-ZTEST .LT. ZIN(KIN) )GOTO 38
      IF(ISW .EQ. 1)GOTO 34
      READ(NBND) (HB1(L),L=1,NNHYPP)
      ISW = 1
      INDH1 = NNHYPP
      INDH2 = 0
      GOTO 36
   34 READ(NBND) (HB1(NNHYPP+L),L=1,NNHYPP)
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
   36 KIN = KIN+1
      ZDENOM = 1.0/(ZIN(KIN) - ZIN(KIN-1))
      JIN = 2
      GOTO 32
   38 IF(Z+ZTEST .LT. ZIN(KIN-1) )GOTO 24
C           LOCATE THE CORRECT COLUMN.
      IF(Y+YTEST .LT. YIN(JIN-1) ) JIN = 2
   40 IF(Y-YTEST .LT. YIN(JIN) )GOTO 42
      JIN = JIN+1
      GOTO 40
C            NOW, SET ADDRESSES AND DO THE INTERPOLATION.
   42 IREF = (JIN - 2)*NNH
      YRATIO = (Y - YIN(J.N-1) )/(YIN(JIN) - YIN(JIN-1) )
      TRATIO = (T - TI1)*TDENOM
      ZRATIO = (Z - ZIN(KIN-1) )*ZDENOM
C            THREE INDICES THAT ARE 1 OR 2 BELOW REPRESENT LOWER OR
C            HIGHER VALUES OF Y, Z, OR T, RESPECTIVELY.
      LF111 = INDH1 + IREF
      LF211 = LF111 + NNH
      LF112 = LF111 + NNHYPR
      LF212 = LF112 + NNH
      LF121 = INDH2 + IREF
      LF221 = LF121 + NNH
      LF122 = LF121 + NNHYPR
      LF222 = LF122 + NNH
      DO 50 L=1,5
      F111 = HB1(LF111 + L)
      FY11 = F111 + (HB1(LF211+L) - F111)*YRATIO
      F121 = HB1(LF121+L)
      FY21 = F121 + (HB1(LF221+L) - F121)*YRATIO
      FYZ1 = FY11 + (FY21 - FY11)*ZRATIO
      F112 = HB1(LF112 + L)
      FY12 = F112 + (HB1(LF212+L) - F112)*YRATIO
      F122 = HB1(LF122+L)
      FY22 = F122 + (HB1(LF222+L) - F122)*YRATIO
      FYZ2 = FY12 + (FY22 - FY12)*ZRATIO
      H(L) = FYZ1 + (FYZ2 - FYZ1)*TRATIO
   50 CONTINUE
      U = H(1)
      V = H(2)
      W = H(3)
      E = H(4)
      RHO = H(5)
```

AD-A169 660 IMPOSED-SOLUTION BOUNDARIES FOR THREE-DIMENSIONAL HULL 2/2
(U) ARMY BALLISTIC RESEARCH LAB ABERDEEN PROVING GROUND
MD J D WORTMAN MAR 86 BRL-MR-3499

UNCLASSIFIED F/G 19/4 NL



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS - 1963 - A

```
      RETURN
C
  100 WRITE(6,105)
      STOP' ABORT, LBND9D3.  NO DATA ON FILE.'
  105 FORMAT(//'  ABORT, LBND9D3.   NO DATA ON FILE.')
  110 WRITE(6,115)
      STOP' ABORT LBND9D3.  FIRST FILE WORD NOT 555.0'
  115 FORMAT(//' ABORT, LBND9D3.  HEAD1 NOT 555.0')
  120 WRITE(6,125)JN,JMX,KN,KMX,NNHYPP,NMX
      STOP' ABORT,LBND9D3.  DIMENSION TOO SMALL.'
  125 FORMAT(//' ABORT, LBND9D3.  DIMENSION TOO SMALL.'/
     +  '   JN,JMX,KN,KMX,NNHYPP,NMX=',6I8)
  130 WRITE(6,135)TH,TSHIFT,T,TI1
      STOP' ABORT,LBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'
  135 FORMAT(' ABORT, LBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'/
     +  '   TH,TSHIFT,T,TI1=   ',1P4E15.7)
  140 WRITE(6,145)
      WRITE(6,GRID)
      STOP' ABORT, LBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.'
  145 FORMAT(//' ABORT, LBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.')
  160 WRITE(6,165)TH,TSHIFT,T,TI1,TI2
      STOP' ABORT, LBND9D3.  END OF FILE BEFORE T.'
  165 FORMAT(' ABORT, LBND9D3.  END OF FILE BEFORE T.'/
     +  '   TH,TSHIFT,T,TI1,TI2=   ',1P5E15.7)
      END
*LABEL ENDBND9
*KEEPTO   ENDBND9   RBOUND9 AND DIMEN3
      SUBROUTINE RBND9D3(TH,XH,YH,ZH,U,V,W,E,RHO)
C              BOUNDARY SUBROUTINE FOR RBOUND=9.  INPUT PREPARED BY
C              HULLUP IS ASSUMED ON FILE TAPE32 FOR X = (XIN).
C         XH SHOULD BE X(IMAX)-0.5*DX(IMAX).  XH + XSHIFT SHOULD BE XIN.
C
C              KN + 2 RECORDS PER TIME DUMP:
C          HEADER--555.0,OLD PROB,TI1,TI2,XIN,NNH,JN,KN
C          GRID--(YIN(J),J=1,JN),(ZIN(K),K=1,KN)
C             HYDRO DATA NNH/POINT FOR TIMES TI1 AND TI2
C          KN ROWS AT (YIN(J),ZIN(K),J=1,JN) FOR K=1,KN.
C
      COMMON/SHIFT/TSHIFT,XSHIFT,YSHIFT,ZSHIFT
C              VALUES IN SHIFT SET IN HULLIN.
C
C       FOLLOWING ARRAYS SET TO ACCOMODATE JN .LE. 200.
C          COMMON/HB2/ TO ACCOMODATE LEVEL 2 STATEMENT.
      COMMON/HB2/HB2(4000)
      DIMENSION H(5)
      DIMENSION YIN(200),ZIN(200)
C
      LEVEL 2, HB2
      NAMELIST/HEADIN/HEAD1,OLPROB,TI1,TI2,XIN,NNH,JN,KN
      NAMELIST/GRID/XH,XSHIFT,X,XIN,YH,YSHIFT,Y,YIN,ZH,ZSHIFT,Z,ZIN
C              SET FILE NUMBER.   STATE ARRAY DIMENSIONS.
      DATA IBND,NBND,JMX,KMX,NMX/32,62,200,200,4000/
C         NMX IS ADDED TO LCM IN HULLIN (HULL9250).
C              SET TF1 TO -1.0 FOR INITIAL VALUES.
      DATA TF1/-1.0/
C
C          SHIFT DATA FROM HULL COORDINATES TO BOUNDARY COORDINATES.
      T = TH + TSHIFT
      X = XH + XSHIFT
      Y = YH + YSHIFT
      Z = ZH + ZSHIFT
C          IS THIS AFTER THE INITIAL ENTRY?
      IF(TF1 .GT. -0.9)GOTO 30
      TF1 = 1.0
```

```
C          READ FIRST HEADING, GRID, AND HYDRO SET.  CHECK.
      REWIND IBND
      READ(IBND)HEAD1,OLPROB,TI1,TI2,XIN,NNH,JN,KN
      IF(EOF(IBND) .NE. 0 )GOTO 100
      TTEST = 0.00001*(TI2 - TI1)
      XTEST = 0.0000001*XIN
      WRITE(6,5)
    5 FORMAT(//' INITIATE RBOUND9 INPUT.  FIRST HEADER RECORD:')
      WRITE(6,HEADIN)
      IF(ABS(HEAD1 - 555.0) .GT. 0.01)GOTO 110
      NNHYPR = NNH*JN
      NNHYPP = NNHYPR*2
      IF(JN .GT. JMX .OR. KN .GT. KMX .OR. NNHYPP .GT. NMX)GOTO 120
      IF(T+TTEST .LT. TI1)GOTO 130
      OLPRB1 = OLPROB
C        READ GRID FOR INPUT BOUNDARY PLANE.
      READ(IBND) (YIN(J),J=1,JN),(ZIN(K),K=1,KN)
      YTEST = (YIN(2) - YIN(1) )*0.00001
      ZTEST = (ZIN(2) - ZIN(1) )*0.00001
   10 IF(T+TTEST .LE. TI2)GOTO 20
C        FIND THE NEXT HEADER RECORD.
   14 READ(IBND)HEAD1,OLPROB,TI1,TI2,XIN,NNH,JN,KN
      IF(EOF(IBND) .NE. 0)GOTO 160
      IF(OLPROB .NE. OLPRB1 .OR. HEAD1 .NE. 555.0)GOTO 14
C           THIS IS A NEW HEADER RECORD.  BYPASS THE GRID RECORD.
      READ(IBND)DUMMY
      GOTO 10
C          READ HYDRO RECORDS.  STORE IN FILE NBND.
   20 REWIND NBND
      DO 22 K=1,KN
      READ(IBND) (HB2(L),L=1,NNHYPP)
      WRITE(NBND) (HB2(L),L=1,NNHYPP)
   22 CONTINUE
   24 REWIND NBND
C          READ FIRST 2 ROWS FROM FILE NBND.
      READ(NBND) (HB2(L),L=1,NNHYPP)
      READ(NBND) (HB2(NNHYPP+L),L=1,NNHYPP)
      JIN = 2
      KIN = 2
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
      TDENOM = 1.0/(TI2 - TI1)
   30 IF(T-TTEST .GT. TI2)GOTO 14
      IF(ABS(X - XIN) .GT. XTEST)GOTO 140
      IF(YIN(1) - Y .GT. YTEST .OR. Y - YIN(JN) .GT. YTEST)GOTO 140
      IF(ZIN(1) - Z .GT. ZTEST .OR. Z - ZIN(KN) .GT. ZTEST)GOTO 140
   32 IF(Z-ZTEST .LT. ZIN(KIN) )GOTO 38
      IF(ISW .EQ. 1)GOTO 34
      READ(NBND) (HB2(L),L=1,NNHYPP)
      ISW = 1
      INDH1 = NNHYPP
      INDH2 = 0
      GOTO 36
   34 READ(NBND) (HB2(NNHYPP+L),L=1,NNHYPP)
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
   36 KIN = KIN+1
      ZDENOM = 1.0/(ZIN(KIN) - ZIN(KIN-1))
      JIN = 2
      GOTO 32
   38 IF(Z+ZTEST .LT. ZIN(KIN-1) )GOTO 24
C        LOCATE THE CORRECT COLUMN.
```

```
      IF(Y+YTEST .LT. YIN(JIN-1) ) JIN = 2
   40 IF(Y-YTEST .LT. YIN(JIN) )GOTO 42
      JIN = JIN+1
      GOTO 40
C          NOW, SET ADDRESSES AND DO THE INTERPOLATION.
   42 IREF = (JIN - 2)*NNH
      YRATIO = (Y - YIN(JIN-1) )/(YIN(JIN) - YIN(JIN-1) )
      TRATIO = (T - TI1)*TDENOM
      ZRATIO = (Z - ZIN(KIN-1) )*ZDENOM
C          THREE INDICES THAT ARE 1 OR 2 BELOW REPRESENT LOWER OR
C          HIGHER VALUES OF Y, Z, OR T, RESPECTIVELY.
      LF111 = INDH1 + IREF
      LF211 = LF111 + NNH
      LF112 = LF111 + NNHYPR
      LF212 = LF112 + NNH
      LF121 = INDH2 + IREF
      LF221 = LF121 + NNH
      LF122 = LF121 + NNHYPR
      LF222 = LF122 + NNH
      DO 50 L=1,5
      F111 = HB2(LF111 + L)
      FY11 = F111 + (HB2(LF211+L) - F111)*YRATIO
      F121 = HB2(LF121+L)
      FY21 = F121 + (HB2(LF221+L) - F121)*YRATIO
      FYZ1 = FY11 + (FY21 - FY11)*ZRATIO
      F112 = HB2(LF112 + L)
      FY12 = F112 + (HB2(LF212+L) - F112)*YRATIO
      F122 = HB2(LF122+L)
      FY22 = F122 + (HB2(LF222+L) - F122)*YRATIO
      FYZ2 = FY12 + (FY22 - FY12)*ZRATIO
      H(L) = FYZ1 + (FYZ2 - FYZ1)*TRATIO
   50 CONTINUE
      U = H(1)
      V = H(2)
      W = H(3)
      E = H(4)
      RHO = H(5)
      RETURN
C
  100 WRITE(6,105)
      STOP' ABORT, RBND9D3.  NO DATA ON FILE.'
  105 FORMAT(//'  ABORT, RBND9D3.   NO DATA ON FILE.')
  110 WRITE(6,115)
      STOP' ABORT RBND9D3.  FIRST FILE WORD NOT 555.0'
  115 FORMAT(//' ABORT, RBND9D3.  HEAD1 NOT 555.0')
  120 WRITE(6,125)JN,JMX,KN,KMX,NNHYPP,NMX
      STOP' ABORT,RBND9D3.  DIMENSION TOO SMALL.'
  125 FORMAT(//' ABORT, RBND9D3.  DIMENSION TOO SMALL.'/
     +  '   JN,JMX,KN,KMX,NNHYPP,NMX=',6I8)
  130 WRITE(6,135)TH,TSHIFT,T,TI1
      STOP' ABORT,RBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'
  135 FORMAT(' ABORT, RBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'/
     +  '   TH,TSHIFT,T,TI1=  ',1P4E15.7)
  140 WRITE(6,145)
      WRITE(6,GRID)
      STOP' ABORT, RBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.'
  145 FORMAT(//' ABORT, RBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.')
  160 WRITE(6,165)TH,TSHIFT,T,TI1,TI2
      STOP' ABORT, RBND9D3.  END OF FILE BEFORE T.'
  165 FORMAT(' ABORT, RBND9D3.  END OF FILE BEFORE T.'/
     +  '   TH,TSHIFT,T,TI1,TI2=  ',1P5E15.7)
      END
*LABEL ENDBND9
*KEEPTO   ENDBND9   BBOUND9 AND DIMEN3
```

```
      SUBROUTINE BBND9D3(TH,XH,YH,ZH,U,V,W,E,RHO)
C              BOUNDARY SUBROUTINE FOR BBOUND=9.  INPUT PREPARED BY
C              HULLUP IS ASSUMED ON FILE TAPE33 FOR Z = (ZIN).
C                 ZH SHOULD BE ZO.  ZH + ZSHIFT SHOULD BE ZIN.
C
C              JN + 2 RECORDS PER TIME DUMP:
C          HEADER--555.0,OLD PROB,TI1,TI2,ZIN,NNH,IN,JN
C          GRID--(XIN(I),I=1,IN),(YIN(J),J=1,JN)
C             HYDRO DATA NNH/POINT FOR TIMES TI1 AND TI2
C          JN ROWS AT (XIN(I),YIN(J),I=1,IN) FOR J=1,JN.
C
      COMMON/SHIFT/TSHIFT,XSHIFT,YSHIFT,ZSHIFT
C              VALUES IN SHIFT SET IN HULLIN.
C
C       FOLLOWING ARRAYS SET TO ACCOMODATE IN .LE. 200.
C           COMMON/HB3/ TO ACCOMODATE LEVEL 2 STATEMENT.
      COMMON/HB3/HB3(4000)
      DIMENSION H(5)
      DIMENSION XIN(200),YIN(200)
C
      LEVEL 2, HB3
      NAMELIST/HEADIN/HEAD1,OLPROB,TI1,TI2,ZIN,NNH,IN,JN
      NAMELIST/GRID/XH,XSHIFT,X,XIN,YH,YSHIFT,Y,YIN,ZH,ZSHIFT,Z,ZIN
C              SET FILE NUMBER.   STATE ARRAY DIMENSIONS.
      DATA IBND,NBND,IMX,JMX,NMX/33,63,200,200,4000/
C        NMX IS ADDED TO LCM IN HULLIN (HULL9250).
C              SET TF1 TO -1.0 FOR INITIAL VALUES.
      DATA TF1/-1.0/
C
C          SHIFT DATA FROM HULL COORDINATES TO BOUNDARY COORDINATES.
      T = TH + TSHIFT
      X = XH + XSHIFT
      Y = YH + YSHIFT
      Z = ZH + ZSHIFT
C          IS THIS AFTER THE INITIAL ENTRY?
      IF(TF1 .GT. -0.9)GOTO 30
      TF1 = 1.0
C             READ FIRST HEADING, GRID, AND HYDRO SET.  CHECJ.
      REWIND IBND
      READ(IBND)HEAD1,OLPROB,TI1,TI2,ZIN,NNH,IN,JN
      IF(EOF(IBND) .NE. 0 )GOTO 100
      TTEST = 0.00001*(TI2 - TI1)
      ZTEST = 0.00000001*ZIN
      WRITE(6,5)
    5 FORMAT(//' INITIATE BBOUND9 INPUT.  FIRST HEADER RECORD:')
      WRITE(6,HEADIN)
      IF(ABS(HEAD1 - 555.0) .GT. 0.01)GOTO 110
      NNHYPR = NNH*IN
      NNHYPP = NNHYPR*2
      IF(IN .GT. IMX .OR. JN .GT. JMX .OR. NNHYPP .GT. NMX)GOTO 120
      IF(T+TTEST .LT. TI1)GOTO 130
      OLPRB1 = OLPROB
C          READ GRID FOR INPUT BOUNDARY PLANE.
      READ(IBND) (XIN(I),I=1,IN),(YIN(J),J=1,JN)
      XTEST = (XIN(2) - XIN(1) )*0.00001
      YTEST = (YIN(2) - YIN(1) )*0.00001
   10 IF(T+TTEST .LE. TI2)GOTO 20
C          FIND THE NEXT HEADER RECORD.
   14 READ(IBND)HEAD1,OLPROB,TI1,TI2,ZIN,NNH,IN,JN
      IF(EOF(IBND) .NE. 0)GOTO 160
      IF(OLPROB .NE. OLPRB1 .OR. HEAD1 .NE. 555.0)GOTO 14
C            THIS IS A NEW HEADER RECORD.  BYPASS THE GRID RECORD.
      READ(IBND)DUMMY
      GOTO 10
```

```
C           READ HYDRO RECORDS.  STORE IN FILE NBND.
   20 REWIND NBND
      DO 22 J=1,JN
      READ(IBND) (HB3(L),L=1,NNHYPP)
      WRITE(NBND) (HB3(L),L=1,NNHYPP)
   22 CONTINUE
   24 REWIND NBND
C           READ FIRST 2 ROWS FROM FILE NBND.
      READ(NBND) (HB3(L),L=1,NNHYPP)
      READ(NBND) (HB3(NNHYPP+L),L=1,NNHYPP)
      IIN = 2
      JIN = 2
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
      TDENOM = 1.0/(TI2 - TI1)
   30 IF(T-TTEST .GT. TI2)GOTO 14
      IF(ABS(Z - ZIN) .GT. ZTEST)GOTO 140
      IF(XIN(1) - X .GT. XTEST .OR. X - XIN(IN) .GT. XTEST)GOTO 140
      IF(YIN(1) - Y .GT. YTEST .OR. Y - YIN(JN) .GT. YTEST)GOTO 140
   32 IF(Y-YTEST .LT. YIN(JIN) )GOTO 38
      IF(ISW .EQ. 1)GOTO 34
      READ(NBND) (HB3(L),L=1,NNHYPP)
      ISW = 1
      INDH1 = NNHYPP
      INDH2 = 0
      GOTO 36
   34 READ(NBND) (HB3(NNHYPP+L),L=1,NNHYPP)
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
   36 JIN = JIN+1
      YDENOM = 1.0/(YIN(JIN) - YIN(JIN-1))
      IIN = 2
      GOTO 32
   38 IF(Y+YTEST .LT. YIN(JIN-1) )GOTO 24
C        LOCATE THE CORRECT COLUMN.
      IF(X+XTEST .LT. XIN(IIN-1) ) IIN = 2
   40 IF(X-XTEST .LT. XIN(IIN) )GOTO 42
      IIN = IIN+1
      GOTO 40
C         NOW, SET ADDRESSES AND DO THE INTERPOLATION.
   42 IREF = (IIN - 2)*NNH
      XRATIO = (X - XIN(IIN-1) )/(XIN(IIN) - XIN(IIN-1) )
      TRATIO = (T - TI1)*TDENOM
      YRATIO = (Y - YIN(JIN-1) )*YDENOM
C         THREE INDICES THAT ARE 1 OR 2 BELOW REPRESENT LOWER O.
C         HIGHER VALUES OF X, Y, OR T, RESPECTIVELY.
      LF111 = INDH1 + IREF
      LF211 = LF111 + NNH
      LF112 = LF111 + NNHYPR
      LF212 = LF112 + NNH
      LF121 = INDH2 + IREF
      LF221 = LF121 + NNH
      LF122 = LF121 + NNHYPR
      LF222 = LF122 + NNH
      DO 50 L=1,5
      F111 = HB3(LF111 + L)
      FX11 = F111 + (HB3(LF211+L) - F111)*XRATIO
      F121 = HB3(LF121+L)
      FX21 = F121 + (HB3(LF221+L) - F121)*XRATIO
      FXY1 = FX11 + (FX21 - FX11)*YRATIO
      F112 = HB3(LF112 + L)
      FX12 = F112 + (HB3(LF212+L) - F112)*XRATIO
```

```
      F122 = HB3(LF122+L)
      FX22 = F122 + (HB3(LF222+L) - F122)*XRATIO
      FXY2 = FX12 + (FX22 - FX12)*YRATIO
      H(L) = FXY1 + (FXY2 - FXY1)*TRATIO
   50 CONTINUE
      U = H(1)
      V = H(2)
      W = H(3)
      E = H(4)
      RHO = H(5)
      RETURN
C
  100 WRITE(6,105)
      STOP' ABORT, BBND9D3.  NO DATA ON FILE.'
  105 FORMAT(//' ABORT, BBND9D3.   NO DATA ON FILE.')
  110 WRITE(6,115)
      STOP' ABORT BBND9D3.  FIRST FILE WORD NOT 555.0'
  115 FORMAT(//' ABORT, BBND9D3.  HEAD1 NOT 555.0')
  120 WRITE(6,125)IN,IMX,JN,JMX,NNHYPP,NMX
      STOP' ABORT,BBND9D3.  DIMENSION TOO SMALL.'
  125 FORMAT(//' ABORT, BBND9D3.  DIMENSION TOO SMALL.'/
     + '  IN,IMX,JN,JMX,NNHYPP,NMX=',6I8)
  130 WRITE(6,135)TH,TSHIFT,T,TI1
      STOP' ABORT,BBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'
  135 FORMAT(' ABORT, BBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'/
     + '  TH,TSHIFT,T,TI1=  ',1P4E15.7)
  140 WRITE(6,145)
      WRITE(6,GRID)
      STOP' ABORT, BBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.'
  145 FORMAT(//' ABORT, BBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.')
  160 WRITE(6,165)TH,TSHIFT,T,TI1,TI2
      STOP' ABORT, BBND9D3.  END OF FILE BEFORE T.'
  165 FORMAT(' ABORT, BBND9D3.  END OF FILE BEFORE T.'/
     + '  TH,TSHIFT,T,TI1,TI2=  ',1P5E15.7)
      END
*LABEL ENDBND9
*KEEPTO   ENDBND9   TBOUND9 AND DIMEN3
      SUBROUTINE TBND9D3(TH,XH,YH,ZH,U,V,W,E,RHO)
C              BOUNDARY SUBROUTINE FOR TBOUND=9.  INPUT PREPARED BY
C              HULLUP IS ASSUMED ON FILE TAPE34 FOR Z = (ZIN).
C        ZH SHOULD BE Z(KMAX)-0.5*DZ(KMAX).  ZH + ZSHIFT SHOULD BE ZIN.
C
C              JN + 2 RECORDS PER TIME DUMP:
C         HEADER--555.0,OLD PROB,TI1,TI2,ZIN,NNH,IN,JN
C         GRID--(XIN(I),I=1,IN),(YIN(J),J=1,JN)
C            HYDRO DATA NNH/POINT FOR TIMES TI1 AND TI2
C         JN ROWS AT (XIN(I),YIN(J),I=1,IN) FOR J=1,JN.
C
      COMMON/SHIFT/TSHIFT,XSHIFT,YSHIFT,ZSHIFT
C              VALUES IN SHIFT SET IN HULLIN.
C
C        FOLLOWING ARRAYS SET TO ACCOMODATE IN .LE. 200.
C           COMMON/HB4/ TO ACCOMODATE LEVEL 2 STATEMENT.
      COMMON/HB4/HB4(4000)
      DIMENSION H(5)
      DIMENSION XIN(200),YIN(200)
C
      LEVEL 2, HB4
      NAMELIST/HEADIN/HEAD1,OLPROB,TI1,TI2,ZIN,NNH,IN,JN
      NAMELIST/GRID/XH,XSHIFT,X,XIN,YH,YSHIFT,Y,YIN,ZH,ZSHIFT,Z,ZIN
C              SET FILE NUMBER.     STATE ARRAY DIMENSIONS.
      DATA IBND,NBND,IMX,JMX,NMX/34,64,200,200,4000/
C        NMX IS ADDED TO LCM IN HULLIN (HULL9250).
C              SET TF1 TO -1.0 FOR INITIAL VALUES.
```

```
      DATA TF1/-1.0/
C
C         SHIFT DATA FROM HULL COORDINATES TO BOUNDARY COORDINATES.
      T = TH + TSHIFT
      X = XH + XSHIFT
      Y = YH + YSHIFT
      Z = ZH + ZSHIFT
C         IS THIS AFTER THE INITIAL ENTRY?
      IF(TF1 .GT. -0.9)GOTO 30
      TF1 = 1.0
C             READ FIRST HEADING, GRID, AND HYDRO SET.  CHECJ.
      REWIND IBND
      READ(IBND)HEAD1,OLPROB,TI1,TI2,ZIN,NNH,IN,JN
      IF(EOF(IBND) .NE. 0 )GOTO 100
      TTEST = 0.00001*(TI2 - TI1)
      ZTEST = 0.00000001*ZIN
      WRITE(6,5)
    5 FORMAT(//' INITIATE TBOUND9 INPUT.  FIRST HEADER RECORD:')
      WRITE(6,HEADIN)
      IF(ABS(HEAD1 - 555.0) .GT. 0.01)GOTO 110
      NNHYPR = NNH*IN
      NNHYPP = NNHYPR*2
      IF(IN .GT. IMX .OR. JN .GT. JMX .OR. NNHYPP .GT. NMX)GOTO 120
      IF(T+TTEST .LT. TI1)GOTO 130
      OLPRB1 = OLPROB
C         READ GRID FOR INPUT BOUNDARY PLANE.
      READ(IBND) (XIN(I),I=1,IN),(YIN(J),J=1,JN)
      XTEST = (XIN(2) - XIN(1) )*0.00001
      YTEST = (YIN(2) - YIN(1) )*0.00001
   10 IF(T+TTEST .LE. TI2)GOTO 20
C         FIND THE NEXT HEADER RECORD.
   14 READ(IBND)HEAD1,OLPROB,TI1,TI2,ZIN,NNH,IN,JN
      IF(EOF(IBND) .NE. 0)GOTO 160
      IF(OLPROB .NE. OLPRB1 .OR. HEAD1 .NE. 555.0)GOTO 14
C             THIS IS A NEW HEADER RECORD.  BYPASS THE GRID RECORD.
      READ(IBND)DUMMY
      GOTO 10
C           READ HYDRO RECORDS.  STORE IN FILE NBND.
   20 REWIND NBND
      DO 22 J=1,JN
      READ(IBND) (HB4(L),L=1,NNHYPP)
      WRITE(NBND) (HB4(L),L=1,NNHYPP)
   22 CONTINUE
   24 REWIND NBND
C           READ FIRST 2 ROWS FROM FILE NBND.
      READ(NBND) (HB4(L),L=1,NNHYPP)
      READ(NBND) (HB4(NNHYPP+L),L=1,NNHYPP)
      IIN = 2
      JIN = 2
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
      TDENOM = 1.0/(TI2 - TI1)
   30 IF(T-TTEST .GT. TI2)GOTO 14
      IF(ABS(Z - ZIN) .GT. ZTEST)GOTO 140
      IF(XIN(1) - X .GT. XTEST .OR. X - XIN(IN) .GT. XTEST)GOTO 140
      IF(YIN(1) - Y .GT. YTEST .OR. Y - YIN(JN) .GT. YTEST)GOTO 140
   32 IF(Y-YTEST .LT. YIN(JIN) )GOTO 38
      IF(ISW .EQ. 1)GOTO 34
      READ(NBND) (HB4(L),L=1,NNHYPP)
      ISW = 1
      INDH1 = NNHYPP
      INDH2 = 0
      GOTO 36
```

```
   34 READ(NBND) (HB4(NNHYPP+L),L=1,NNHYPP)
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
   36 JIN = JIN+1
      YDENOM = 1.0/(YIN(JIN) - YIN(JIN-1))
      IIN = 2
      GOTO 32
   38 IF(Y+YTEST .LT. YIN(JIN-1) )GOTO 24
C          LOCATE THE CORRECT COLUMN.
      IF(X+XTEST .LT. XIN(IIN-1) ) IIN = 2
   40 IF(X-XTEST .LT. XIN(IIN) )GOTO 42
      IIN = IIN+1
      GOTO 40
C           NOW, SET ADDRESSES AND DO THE INTERPOLATION.
   42 IREF = (IIN - 2)*NNH
      XRATIO = (X - XIN(IIN-1) )/(XIN(IIN) - XIN(IIN-1) )
      TRATIO = (T - TI1)*TDENOM
      YRATIO = (Y - YIN(JIN-1) )*YDENOM
C           THREE INDICES THAT ARE 1 OR 2 BELOW REPRESENT LOWER OR
C           HIGHER VALUES OF X, Y, OR T, RESPECTIVELY.
      LF111 = INDH1 + IREF
      LF211 = LF111 + NNH
      LF112 = LF111 + NNHYPR
      LF212 = LF112 + NNH
      LF121 = INDH2 + IREF
      LF221 = LF121 + NNH
      LF122 = LF121 + NNHYPR
      LF222 = LF122 + NNH
      DO 50 L=1,5
      F111 = HB4(LF111 + L)
      FX11 = F111 + (HB4(LF211+L) - F111)*XRATIO
      F121 = HB4(LF121+L)
      FX21 = F121 + (HB4(LF221+L) - F121)*XRATIO
      FXY1 = FX11 + (FX21 - FX11)*YRATIO
      F112 = HB4(LF112 + L)
      FX12 = F112 + (HB4(LF212+L) - F112)*XRATIO
      F122 = HB4(LF122+L)
      FX22 = F122 + (HB4(LF222+L) - F122)*XRATIO
      FXY2 = FX12 + (FX22 - FX12)*YRATIO
      H(L) = FXY1 + (FXY2 - FXY1)*TRATIO
   50 CONTINUE
      U = H(1)
      V = H(2)
      W = H(3)
      E = H(4)
      RHO = H(5)
      RETURN
C
  100 WRITE(6,105)
      STOP' ABORT, TBND9D3.  NO DATA ON FILE.'
  105 FORMAT(//'  ABORT, TBND9D3.   NO DATA ON FILE.')
  110 WRITE(6,115)
      STOP' ABORT TBND9D3.  FIRST FILE WORD NOT 555.0'
  115 FORMAT(//' ABORT, TBND9D3.  HEAD1 NOT 555.0')
  120 WRITE(6,125)IN,IMX,JN,JMX,NNHYPP,NMX
      STOP' ABORT,TBND9D3.  DIMENSION TOO SMALL.'
  125 FORMAT(//' ABORT, TBND9D3.  DIMENSION TOO SMALL.'/
     +  '   IN,IMX,JN,JMX,NNHYPP,NMX=',6I8)
  130 WRITE(6,135)TH,TSHIFT,T,TI1
      STOP' ABORT,TBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'
  135 FORMAT(' ABORT, TBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'/
     +  '   TH,TSHIFT,T,TI1=  ',1P4E15.7)
  140 WRITE(6,145)
```

```
      WRITE(6,GRID)
      STOP' ABORT, TBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.'
  145 FORMAT(//' ABORT, TBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.')
  160 WRITE(6,165)TH,TSHIFT,T,TI1,TI2
      STOP' ABORT, TBND9D3.  END OF FILE BEFORE T.'
  165 FORMAT(' ABORT, TBND9D3.  END OF FILE BEFORE T.'/
     + '   TH,TSHIFT,T,TI1,TI2=   ',1P5E15.7)
      END
*LABEL ENDBND9
*KEEPTO   ENDBND9   ABOUND9 AND DIMEN3
      SUBROUTINE ABND9D3(TH,XH,YH,ZH,U,V,W,E,RHO)
C               BOUNDARY SUBROUTINE FOR ABOUND=9.  INPUT PREPARED BY
C               HULLUP IS ASSUMED ON FILE TAPE35 FOR Y = (YIN).
C                  YH SHOULD BE YO.  YH + YSHIFT SHOULD BE YIN.
C
C               KN + 2 RECORDS PER TIME DUMP:
C          HEADER--555.0,OLD PROB,TI1,TI2,YIN,NNH,IN,KN
C          GRID--(XIN(I),I=1,IN),(ZIN(K),K=1,KN)
C             HYDRO DATA NNH/POINT FOR TIMES TI1 AND TI2
C          KN ROWS AT (XIN(I),ZIN(K),I=1,IN) FOR K=1,KN.
C
      COMMON/SHIFT/TSHIFT,XSHIFT,YSHIFT,ZSHIFT
C               VALUES IN SHIFT SET IN HULLIN.
C
C        FOLLOWING ARRAYS SET TO ACCOMODATE IN .LE. 200.
C           COMMON/HB5/ TO ACCOMODATE LEVEL 2 STATEMENT.
      COMMON/HB5/HB5(4000)
      DIMENSION H(5)
      DIMENSION XIN(200),ZIN(200)
C
      LEVEL 2, HB5
      NAMELIST/HEADIN/HEAD1,OLPROB,TI1,TI2,YIN,NNH,IN,KN
      NAMELIST/GRID/XH,XSHIFT,X,XIN,YH,YSHIFT,Y,YIN,ZH,ZSHIFT,Z,ZIN
C               SET FILE NUMBER.   STATE ARRAY DIMENSIONS.
      DATA IBND,NBND,IMX,KMX,NMX/35,65,200,200,4000/
C         NMX IS ADDED TO LCM IN HULLIN (HULL9250).
C               SET TF1 TO -1.0 FOR INITIAL VALUES.
      DATA TF1/-1.0/
C
C          SHIFT DATA FROM HULL COORDINATES TO BOUNDARY COORDINATES.
      T = TH + TSHIFT
      X = XH + XSHIFT
      Y = YH + YSHIFT
      Z = ZH + ZSHIFT
C          IS THIS AFTER THE INITIAL ENTRY?
      IF(TF1 .GT. -0.9)GOTO 30
      TF1 = 1.0
C              READ FIRST HEADING, GRID, AND HYDRO SET.  CHECK.
      REWIND IBND
      READ(IBND)HEAD1,OLPROB,TI1,TI2,YIN,NNH,IN,KN
      IF(EOF(IBND) .NE. 0 )GOTO 100
      TTEST = 0.00001*(TI2 - TI1)
      YTEST = 0.00000001*YIN
      WRITE(6,5)
    5 FORMAT(//' INITIATE ABOUND9 INPUT.  FIRST HEADER RECORD:')
      WRITE(6,HEADIN)
      IF(ABS(HEAD1 - 555.0) .GT. 0.01)GOTO 110
      NNHYPR = NNH*IN
      NNHYPP = NNHYPR*2
      IF(IN .GT. IMX .OR. KN .GT. KMX .OR. NNHYPP .GT. NMX)GOTO 120
      IF(T+TTEST .LT. TI1)GOTO 130
      OLPRB1 = OLPROB
C          READ GRID FOR INPUT BOUNDARY PLANE.
      READ(IBND) (XIN(I),I=1,IN),(ZIN(K),K=1,KN)
```

```
      XTEST = (XIN(2) - XIN(1) )*0.00001
      ZTEST = (ZIN(2) - ZIN(1) )*0.00001
   10 IF(T+TTEST .LE. TI2)GOTO 20
C         FIND THE NEXT HEADER RECORD.
   14 READ(IBND)HEAD1,OLPROB,TI1,TI2,YIN,NNH,IN,KN
      IF(EOF(IBND) .NE. 0)GOTO 160
      IF(OLPROB .NE. OLPRB1 .OR. HEAD1 .NE. 555.0)GOTO 14
C            THIS IS A NEW HEADER RECORD.  BYPASS THE GRID RECORD.
      READ(IBND)DUMMY
      GOTO 10
C           READ HYDRO RECORDS.  STORE IN FILE NBND.
   20 REWIND NBND
      DO 22 K=1,KN
      READ(IBND) (HB5(L),L=1,NNHYPP)
      WRITE(NBND) (HB5(L),L=1,NNHYPP)
   22 CONTINUE
   24 REWIND NBND
C           READ FIRST 2 ROWS FROM FILE NBND.
      READ(NBND) (HB5(L),L=1,NNHYPP)
      READ(NBND) (HB5(NNHYPP+L),L=1,NNHYPP)
      IIN = 2
      KIN = 2
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
      TDENOM = 1.0/(TI2 - TI1)
   30 IF(T-TTEST .GT. TI2)GOTO 14
      IF(ABS(Y - YIN) .GT. YTEST)GOTO 140
      IF(XIN(1) - X .GT. XTEST .OR. X - XIN(IN) .GT. XTEST)GOTO 140
      IF(ZIN(1) - Z .GT. ZTEST .OR. Z - ZIN(KN) .GT. ZTEST)GOTO 140
   32 IF(Z-ZTEST .LT. ZIN(KIN) )GOTO 38
      IF(ISW .EQ. 1)GOTO 34
      READ(NBND) (HB5(L),L=1,NNHYPP)
      ISW = 1
      INDH1 = NNHYPP
      INDH2 = 0
      GOTO 36
   34 READ(NBND) (HB5(NNHYPP+L),L=1,NNHYPP)
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
   36 KIN = KIN+1
      ZDENOM = 1.0/(ZIN(KIN) - ZIN(KIN-1))
      IIN = 2
      GOTO 32
   38 IF(Z+ZTEST .LT. ZIN(KIN-1) )GOTO 24
C         LOCATE THE CORRECT COLUMN.
      IF(X+XTEST .LT. XIN(IIN-1) ) IIN = 2
   40 IF(X-XTEST .LT. XIN(IIN) )GOTO 42
      IIN = IIN+1
      GOTO 40
C          NOW, SET ADDRESSES AND DO THE INTERPOLATION.
   42 IREF = (IIN - 2)*NNH
      XRATIO = (X - XIN(IIN-1) )/(XIN(IIN) - XIN(IIN-1) )
      TRATIO = (T - TI1)*TDENOM
      ZRATIO = (Z - ZIN(KIN-1) )*ZDENOM
C          THREE INDICES THAT ARE 1 OR 2 BELOW REPRESENT LOWER OR
C          HIGHER VALUES OF X, Z, OR T, RESPECTIVELY.
      LF111 = INDH1 + IREF
      LF211 = LF111 + NNH
      LF112 = LF111 + NNHYPR
      LF212 = LF112 + NNH
      LF121 = INDH2 + IREF
      LF221 = LF121 + NNH
```

```
      LF122 = LF121 + NNHYPR
      LF222 = LF122 + NNH
      DO 50 L=1,5
      F111 = HB5(LF111 + L)
      FX11 = F111 + (HB5(LF211+L) - F111)*XRATIO
      F121 = HB5(LF121+L)
      FX21 = F121 + (HB5(LF221+L) - F121)*XRATIO
      FXZ1 = FX11 + (FX21 - FX11)*ZRATIO
      F112 = HB5(LF112 + L)
      FX12 = F112 + (HB5(LF212+L) - F112)*XRATIO
      F122 = HB5(LF122+L)
      FX22 = F122 + (HB5(LF222+L) - F122)*XRATIO
      FXZ2 = FX12 + (FX22 - FX12)*ZRATIO
      H(L) = FXZ1 + (FXZ2 - FXZ1)*TRATIO
   50 CONTINUE
      U = H(1)
      V = H(2)
      W = H(3)
      E = H(4)
      RHO = H(5)
      RETURN
C
  100 WRITE(6,105)
      STOP' ABORT, ABND9D3.  NO DATA ON FILE.'
  105 FORMAT(//'  ABORT, ABND9D3.   NO DATA ON FILE.')
  110 WRITE(6,115)
      STOP' ABORT ABND9D3.  FIRST FILE WORD NOT 555.0'
  115 FORMAT(//' ABORT, ABND9D3.  HEAD1 NOT 555.0')
  120 WRITE(6,125)IN,IMX,KN,KMX,NNHYPP,NMX
      STOP' ABORT,ABND9D3.  DIMENSION TOO SMALL.'
  125 FORMAT(//' ABORT, ABND9D3.  DIMENSION TOO SMALL.'/
     + '   IN,IMX,KN,KMX,NNHYPP,NMX=',6I8)
  130 WRITE(6,135)TH,TSHIFT,T,TI1
      STOP' ABORT,ABND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'
  135 FORMAT(' ABORT, ABND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'/
     + '   TH,TSHIFT,T,TI1=   ',1P4E15.7)
  140 WRITE(6,145)
      WRITE(6,GRID)
      STOP' ABORT, ABND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.'
  145 FORMAT(//' ABORT, ABND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.')
  160 WRITE(6,165)TH,TSHIFT,T,TI1,TI2
      STOP' ABORT, ABND9D3.  END OF FILE BEFORE T.'
  165 FORMAT(' ABORT, ABND9D3.  END OF FILE BEFORE T.'/
     + '   TH,TSHIFT,T,TI1,TI2=   ',1P5E15.7)
      END
*LABEL ENDBND9
*KEEPTO   ENDBND9    FBOUND9 AND DIMEN3
      SUBROUTINE FBND9D3(TH,XH,YH,ZH,U,V,W,E,RHO)
C                BOUNDARY SUBROUTINE FOR FBOUND=9.  INPUT PREPARED BY
C                HULLUP IS ASSUMED ON FILE TAPE36 FOR Y = (YIN).
C          YH SHOULD BE Y(JMAX)-0.5*DY(JMAX).  YH + YSHIFT SHOULD BE YIN.
C
C                KN + 2 RECORDS PER TIME DUMP:
C            HEADER--555.0,OLD PROB,TI1,TI2,YIN,NNH,IN,KN
C            GRID--(XIN(I),I=1,IN),(ZIN(K),K=1,KN)
C               HYDRO DATA NNH/POINT FOR TIMES TI1 AND TI2
C            KN ROWS AT (XIN(I),ZIN(K),I=1,IN) FOR K=1,KN.
C
      COMMON/SHIFT/TSHIFT,XSHIFT,YSHIFT,ZSHIFT
C                VALUES IN SHIFT SET IN HULLIN.
C
C         FOLLOWING ARRAYS SET TO ACCOMODATE IN .LE. 200.
C            COMMON/HB6/ TO ACCOMODATE LEVEL 2 STATEMENT.
      COMMON/HB6/HB6(4000)
```

```
      DIMENSION H(5)
      DIMENSION XIN(200),ZIN(200)
C
      LEVEL 2, HB6
      NAMELIST/HEADIN/HEAD1,OLPROB,TI1,TI2,YIN,NNH,IN,KN
      NAMELIST/GRID/XH,XSHIFT,X,XIN,YH,YSHIFT,Y,YIN,ZH,ZSHIFT,Z,ZIN
C           SET FILE NUMBER.   STATE ARRAY DIMENSIONS.
      DATA IBND,NBND,IMX,KMX,NMX/36,66,200,200,4000/
C        NMX IS ADDED TO LCM IN HULLIN (HULL9250).
C              SET TF1 TO -1.0 FOR INITIAL VALUES.
      DATA TF1/-1.0/
C
C         SHIFT DATA FROM HULL COORDINATES TO BOUNDARY COORDINATES.
      T = TH + TSHIFT
      X = XH + XSHIFT
      Y = YH + YSHIFT
      Z = ZH + ZSHIFT
C         IS THIS AFTER THE INITIAL ENTRY?
      IF(TF1 .GT. -0.9)GOTO 30
      TF1 = 1.0
C              READ FIRST HEADING, GRID, AND HYDRO SET.  CHECK.
      REWIND IBND
      READ(IBND)HEAD1,OLPROB,TI1,TI2,YIN,NNH,IN,KN
      IF(EOF(IBND) .NE. 0 )GOTO 100
      TTEST = 0.00001*(TI2 - TI1)
      YTEST = 0.00000001*YIN
      WRITE(6,5)
    5 FORMAT(//' INITIATE FBOUND9 INPUT.  FIRST HEADER RECORD:')
      WRITE(6,HEADIN)
      IF(ABS(HEAD1 - 555.0) .GT. 0.01)GOTO 110
      NNHYPR = NNH*IN
      NNHYPP = NNHYPR*2
      IF(IN .GT. IMX .OR. KN .GT. KMX .OR. NNHYPP .GT. NMX)GOTO 120
      IF(T+TTEST .LT. TI1)GOTO 130
      OLPRB1 = OLPROB
C         READ GRID FOR INPUT BOUNDARY PLANE.
      READ(IBND) (XIN(I),I=1,IN),(ZIN(K),K=1,KN)
      XTEST = (XIN(2) - XIN(1) )*0.00001
      ZTEST = (ZIN(2) - ZIN(1) )*0.00001
   10 IF(T+TTEST .LE. TI2)GOTO 20
C         FIND THE NEXT HEADER RECORD.
   14 READ(IBND)HEAD1,OLPROB,TI1,TI2,YIN,NNH,IN,KN
      IF(EOF(IBND) .NE. 0)GOTO 160
      IF(OLPROB .NE. OLPRB1 .OR. HEAD1 .NE. 555.0)GOTO 14
C             THIS IS A NEW HEADER RECORD.  BYPASS THE GRID RECORD.
      READ(IBND)DUMMY
      GOTO 10
C             READ HYDRO RECORDS.  STORE IN FILE NBND.
   20 REWIND NBND
      DO 22 K=1,KN
      READ(IBND) (HB6(L),L=1,NNHYPP)
      WRITE(NBND) (HB6(L),L=1,NNHYPP)
   22 CONTINUE
   24 REWIND NBND
C             READ FIRST 2 ROWS FROM FILE NBND.
      READ(NBND) (HB6(L),L=1,NNHYPP)
      READ(NBND) (HB6(NNHYPP+L),L=1,NNHYPP)
      IIN = 2
      KIN = 2
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
      TDENOM = 1.0/(TI2 - TI1)
   30 IF(T-TTEST .GT. TI2)GOTO 14
```

```
      IF(ABS(Y - YIN) .GT. YTEST)GOTO 140
      IF(XIN(1) - X .GT. XTEST .OR. X - XIN(IN) .GT. XTEST)GOTO 140
      IF(ZIN(1) - Z .GT. ZTEST .OR. Z - ZIN(KN) .GT. ZTEST)GOTO 140
   32 IF(Z-ZTEST .LT. ZIN(KIN) )GOTO 38
      IF(ISW .EQ. 1)GOTO 34
      READ(NBND) (HB6(L),L=1,NNHYPP)
      ISW = 1
      INDH1 = NNHYPP
      INDH2 = 0
      GOTO 36
   34 READ(NBND) (HB6(NNHYPP+L),L=1,NNHYPP)
      ISW = 2
      INDH1 = 0
      INDH2 = NNHYPP
   36 KIN = KIN+1
      ZDENOM = 1.0/(ZIN(KIN) - ZIN(KIN-1))
      IIN = 2
      GOTO 32
   38 IF(Z+ZTEST .LT. ZIN(KIN-1) )GOTO 24
C         LOCATE THE CORRECT COLUMN.
      IF(X+XTEST .LT. XIN(IIN-1) ) IIN = 2
   40 IF(X-XTEST .LT. XIN(IIN) )GOTO 42
      IIN = IIN+1
      GOTO 40
C          NOW, SET ADDRESSES AND DO THE INTERPOLATION.
   42 IREF = (IIN - 2)*NNH
      XRATIO = (X - XIN(IIN-1) )/(XIN(IIN) - XIN(IIN-1) )
      TRATIO = (T - TI1)*TDENOM
      ZRATIO = (Z - ZIN(KIN-1) )*ZDENOM
C          THREE INDICES THAT ARE 1 OR 2 BELOW REPRESENT LOWER OR
C          HIGHER VALUES OF X, Z, OR T, RESPECTIVELY.
      LF111 = INDH1 + IREF
      LF211 = LF111 + NNH
      LF112 = LF111 + NNHYPR
      LF212 = LF112 + NNH
      LF121 = INDH2 + IREF
      LF221 = LF121 + NNH
      LF122 = LF121 + NNHYPR
      LF222 = LF122 + NNH
      DO 50 L=1,5
      F111 = HB6(LF111 + L)
      FX11 = F111 + (HB6(LF211+L) - F111)*XRATIO
      F121 = HB6(LF121+L)
      FX21 = F121 + (HB6(LF221+L) - F121)*XRATIO
      FXZ1 = FX11 + (FX21 - FX11)*ZRATIO
      F112 = HB6(LF112 + L)
      FX12 = F112 + (HB6(LF212+L) - F112)*XRATIO
      F122 = HB6(LF122+L)
      FX22 = F122 + (HB6(LF222+L) - F122)*XRATIO
      FXZ2 = FX12 + (FX22 - FX12)*ZRATIO
      H(L) = FXZ1 + (FXZ2 - FXZ1)*TRATIO
   50 CONTINUE
      U = H(1)
      V = H(2)
      W = H(3)
      E = H(4)
      RHO = H(5)
      RETURN
C
  100 WRITE(6,105)
      STOP' ABORT, FBND9D3.  NO DATA ON FILE.'
  105 FORMAT(//'  ABORT, FBND9D3.   NO DATA ON FILE.')
  110 WRITE(6,115)
      STOP' ABORT FBND9D3.  FIRST FILE WORD NOT 555.0'
```

```
  115 FORMAT(//' ABORT, FBND9D3.  HEAD1 NOT 555.0')
  120 WRITE(6,125)IN,IMX,KN,KMX,NNHYPP,NMX
      STOP' ABORT,FBND9D3.  DIMENSION TOO SMALL.'
  125 FORMAT(//' ABORT, FBND9D3.  DIMENSION TOO SMALL.'/
     + '   IN,IMX,KN,KMX,NNHYPP,NMX=',6I8)
  130 WRITE(6,135)TH,TSHIFT,T,TI1
      STOP' ABORT,FBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'
  135 FORMAT(' ABORT, FBND9D3.  REQUESTED T LT INITIAL TIME IN FILE.'/
     + '   TH,TSHIFT,T,TI1=  ',1P4E15.7)
  140 WRITE(6,145)
      WRITE(6,GRID)
      STOP' ABORT, FBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.'
  145 FORMAT(//' ABORT, FBND9D3.  SPATIAL VARIABLE OUTSIDE RANGE.')
  160 WRITE(6,165)TH,TSHIFT,T,TI1,TI2
      STOP' ABORT, FBND9D3.  END OF FILE BEFORE T.'
  165 FORMAT(' ABORT, FBND9D3.  END OF FILE BEFORE T.'/
     + '   TH,TSHIFT,T,TI1,TI2=  ',1P5E15.7)
      END
*LABEL ENDBND9
```

DISTRIBUTION LIST

No. of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	Director of Defense Research & Engineering ATTN: DD/TWP Washington, DC 20301
1	Asst. to the Secretary of Defense (Atomic Energy) ATTN: Document Control Washington, DC 20301
1	Director Defense Advanced Research Projects Agency ATTN: Tech Lib 1400 Wilson Boulevard Arlington, VA 22209
2	Director Federal Emergency Management Agency ATTN: D. A. Bettge Technical Library Washington, DC 20472
1	Director Defense Intelligence Agency ATTN: DT-2/Wpns & Sys Div Washington, DC 20301
1	Director National Security Agency ATTN: E. F. Butala, R15 Ft. George G. Meade, MD 20755
1	Director Joint Strategic Target Planning Staff JCS Offut AFB Omaha, NB 68113

No. of Copies	Organization
1	Director Defense Communications Agency ATTN: 930 Washington, DC 20305
9	Director Defense Nuclear Agency ATTN: DDST TIPL/Tech Lib SPSS/K. Goering SPTD/T. Kennedy SPAS/P.R. Rohr G. Ullrich STSP/COL Kovel NATD NATA Washington, DC 20305
2	Commander Field Command, DNA ATTN: FCPR FCTMOF Kirtland AFB, NM 87117
1	Commander Field Command, DNA Livermore Branch ATTN: FCPRL P.O. Box 808 Livermore, CA 94550
1	HQDA DAMA-ART-M Washington, DC 20310
10	Central Intelligence Agency Office of Central Reference Dissemination Branch Room GE-47 HQS Washington, DC 20502
1	Program Manager US Army BMD Program Office ATTN: John Shea 5001 Eisenhower Avenue Alexandria, VA 22333

DISTRIBUTION LIST

No. of Copies	Organization
2	Director US Army BMD Advanced Technology Center ATTN: CRDABH-X CRDABH-S Huntsville, AL 35807
1	Commander US Army BMD Command ATTN: BDMSC-TFN/N.J. Hurst P.O. Box 1500 Huntsville, AL 35807
1	Commander US Army Engineer Division ATTN: HNDED-FD P.O. Box 1500 Huntsville, AL 35807
2	Deputy Chief of Staff for Operations and Plans ATTN: Technical Library Director of Chemical & Nuc Operations Department of the Army Washington, DC 20310
2	Office, Chief of Engineers Department of the Army ATTN: DAEN-MCE-D DAEN-RDM 890 South Pickett Street Alexandria, VA 22304
3	Commander US Army Engineer Waterways Experiment Station ATTN: Technical Library Jim Watt Jim Ingram P.O. Box 631 Vicksburg, MS 39180
1	Commander US Army Engineering Center ATTN: ATSEN-SY-L Fort Belvoir, VA 22060

No. of Copies	Organization
1	US Army Belvoir R&D Center ATTN: STRBE-GE, D. Frink Fort Belvoir, VA 22060
1	Commander US Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TDC Dover, NJ 07801
2	Commander Armament R&D Center US Army AMCCOM ATTN: W. Reiner SMCAR-TSS Dover, NJ 07801
1	Commander US Army Armament, Munitions and Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299
1	Director Benet Weapons Laboratory Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Aviation Research and Development Command ATTN: AMSAV-E 4300 Goodfellow Boulevard St. Louis, MO 63120
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Communications - Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703
1	Commander US Army Communications Rsch and Development Command ATTN: DRSEL-ATDD Fort Monmouth, NJ 07703
1	Commander ERADCOM Technical Library ATTN: DELSD-L (Reports Section) Fort Monmouth, NJ 07703-5301
2	Commander US Army Electronics Research and Development Command ATTN: DELEW-E, W. S. McAfee DELSD-EI, J. Roma Fort Monmouth, NJ 07703-5301
7	Director US Army Harry Diamond Labs ATTN: Mr. James Gaul Mr. L. Belliveau Mr. J. Meszaros Mr. J. Gwaltney Mr. Bill Vault Mr. R. J. Bostak Dr. W. J. Schuman, Jr. 2800 Powder Mill Road Adelphi, MD 20783
4	Director US Army Harry Diamond Labs ATTN: DELHD-TA-L DRXDO-TI/002 DRXDO-NP DELHD-RBA/J. Rosado 800 Powder Mill Road Adelphi, MD 20783
1	Commander US Army Missile Command Research, Development and Engineering Center ATTN: AMSMI-RD Redstone Arsenal, AL 35898
1	Director US Army Missile and Space Intelligence Center ATTN: AIAMS-YDL Redstone Arsenal, AL 35898-5500
2	Commander US Army Natick Research and Development Center ATTN: DRXRE/Dr. D. Sieling STRNC-UE/J. Calligeros Natick, MA 01762
1	Commander US Army Tank Automotive Rsch and Development Command ATTN: AMSTA-TSL Warren, MI 48090
1	Commander US Army Foreign Science and Technology Center ATTN: Rsch & Cncepts Br 220 7th Street , NE Charlottesville, VA 22901
1	Commander US Army Logistics Management Center ATTN: ATCL-O, Mr. Robert Cameron Fort Lee, VA 23801
3	Commander US Army Materials Technology Laboratory ATTN: Technical Library DRXMR-ER, Joe Prifti Eugene de Luca Watertown, MA 02172

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Research Office P.O. Box 12211 Research Triangle Park NC 27709
4	Commander US Army Nuclear & Chemical Agency ATTN: ACTA-NAW MONA-WE Technical Library LTC Finno 7500 Backlick Rd, Bldg. 2073 Springfield, VA 22150
1	Commander US Army TRADOC ATTN: DCST&E Fort Monroe, VA 23651
2	Director US Army TRADOC Systems Analysis Activity ATTN: LTC John Hesse ATAA-SL White Sands Missile Range NM 88002
1	Commander US Combined Arms Combat Developments Activity ATTN: ATCA-CO, Mr. L. C. Pleger Fort Leavenworth, KS 66027
2	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905
1	Commander US Army Development & Employment Agency ATTN: MODE-TED-SAB Fort Lewis, WA 98433
1	Commandant Interservice Nuclear Weapons School ATTN: Technical Library Kirtland AFB, NM 87115
1	Chief of Naval Material ATTN: MAT 0323 Department of the Navy Arlington, VA 22217
2	Chief of Naval Operations ATTN: OP-03EG OP-985F Department of the Navy Washington, DC 20350
1	Chief of Naval Research ATTN: N. Perrone Department of the Navy Arlington, VA 22217
1	Director Strategic Systems Projects Ofc ATTN: NSP-43, Tech Library Department of the Navy Washington, DC 20360
1	Commander Naval Electronic Systems Com ATTN: PME 117-21A Washington, DC 20360
1	Commander Naval Facilities Engineering Command Washington, DC 20360
1	Commander Naval Sea Systems Command ATTN: SEA-62R Department of the Navy Washington, DC 20362

DISTRIBUTION LIST

No. of Copies	Organization
3	Officer-in-Charge(Code L31) Civil Engineering Laboratory Naval Constr Btn Center ATTN: Stan Takahashi R. J. Odello Technical Library Port Hueneme, CA 93041
1	Commander David W. Taylor Naval Ship Research & Development Ctr ATTN: Lib Div, Code 522 Bethesda, MD 20084
1	Commander Naval Surface Weapons Center ATTN: DX-21, Library Br. Dahlgren, VA 22448
2	Commander Naval Surface Weapons Center ATTN: Code WA501/Navy Nuclear Programs Office Code WX21/Tech Library Silver Spring, MD 20910
1	Commander Naval Weapons Center ATTN: Code 533, Tech Lib China Lake, CA 93555
1	Commander Naval Weapons Evaluation Fac ATTN: Document Control Kirtland AFB, NM 87117
1	Commander Naval Research Laboratory ATTN: Code 2027, Tech Lib Washington, DC 20375
1	Superintendent Naval Postgraduate School ATTN: Code 2124, Technical Reports Library Monterey, CA 93940

No. of Copies	Organization
1	AFSC/SDOA Andrews Air Force Base MD 20334
1	ADTC/DLODL, Tech Lib Eglin AFB, FL 32542
1	AFWL/SUL Kirtland AFB, NM 87117
1	Air Force Armament Laboratory ATTN: AFATL/DLODL Eglin AFB, FL 32542-5000
1	AFATL (DLYV) Eglin AFB, FL 32542
1	RADC (EMTLD/Docu Libray) Griffiss AFB, NY 13441
1	AFWL/NTES (R. Henny) Kirtland AFB, NM 87117
1	AFWL/NTE, CPT J. Clifford Kirtland AFB, NM 87117
2	Commander-in-Chief Strategic Air Command ATTN: NRI-STINFO Lib Offutt AFB, NB 68113
1	AFIT (Lib Bldg. 640, Area B) Wright-Patterson AFB Ohio 45433
1	FTD/NIIS Wright-Patterson AFB Ohio 45433
1	Director Lawrence Livermore Lab. ATTN: Tech Info Dept L-3 P.O. Box 808 Livermore, CA 94550

DISTRIBUTION LIST

No. of Copies	Organization
2	Director Los Alamos Scientific Lab. ATTN: Doc Control for Rpts Lib P.O. Box 1663 Los Alamos, NM 87544
2	Director Sandia Laboratories ATTN: Doc Control for 3141 Sandia Rpt Collection L. J. Vortman P.O. Box 5800 Albuquerque, NM 87185
1	Director Sandia Laboratories Livermore Laboratory ATTN: Doc Control for Technical Library P.O. Box 969 Livermore, CA 94550
1	Director National Aeronautics and Space Administration Scientific & Tech Info Fac P.O. Box 8757 Baltimore/Washington International Airport MD 21240
1	Director NASA-Ames Research Center Applied Computational Aerodynamics Branch MS 202-14, Dr. T. Holtz Moffett Field, CA 94035
1	Aberdeen Research Center 30 Diamond Street Aberdeen, MD 21001
1	Aerospace Corporation ATTN: Tech Info Services P.O. Box 92957 Los Angeles, CA 90009

No. of Copies	Organization
1	Agbabian Associates ATTN: M. Agbabian 250 North Nash Street El Segundo, CA 90245
1	The BDM Corporation ATTN: Richard Hensley P.O. Box 9274 Albuquerque International Albuquerque, NM 87119
1	The Boeing Company ATTN: Aerospace Library P.O. Box 3707 Seattle, WA 98124
2	California Research and Technology ATTN: M. Rosenblatt F. Sauer Suite B 130 11875 Dublin Blvd Dublin, CA 94568
1	Carpenter Research Corporation ATTN: H. Jerry Carpenter Suite 424 904 Silver Spur Road Rolling Hills Estates, CA 90274
1	Goodyear Aerospace Corp ATTN: R. M. Brown, Bldg 1 Shelter Engineering Litchfield Park, AZ 85340
6	Kaman AviDyne ATTN: Dr. R. Reutenick (4 cys) Mr. S. Criscione Mr. R. Milligan 83 Second Avenue Northwest Industrial Park Burlington, MA 01830

DISTRIBUTION LIST

No. of Copies	Organization
3	Kaman Sciences Corporation ATTN: Library P. A. Ellis F. H. Shelton 1500 Garden of the Gods Road Colorado Springs, CO 80907
1	Kaman Sciences Corporation ATTN: Don Sachs Suite 703 2001 Jefferson Davis Highway Arlington, VA 22202
1	Kaman-TEMPO ATTN: DASIAC P.O. Drawer QQ Santa Barbara, CA 93102
1	Kaman-TEMPO ATTN: E. Bryant, Suite UL-1 715 Shamrock Road Bel Air, MD 21014
1	Lockheed Missiles & Space Co. ATTN: J. J. Murphy, Dept. 81-11, Bldg. 154 P.O. Box 504 Sunnyvale, CA 94086
1	Martin Marietta Aerospace Orlando Division ATTN: G. Fotieo P.O. Box 5837 Orlando, FL 32805
2	McDonnell Douglas Astronautics Corporation ATTN: Robert W. Halprin K.A. Heinly 5301 Bolsa Avenue Huntington Beach, CA 92647
1	New Mexico Engineering Research Institute (CERF) ATTN: J. Leigh P.O. Box 25 UNM Albuquerque, NM 87131
2	Physics International Corp 2700 Merced Street San Leandro, CA 94577
2	R&D Associates ATTN: Technical Library Allan Kuhl P.O. Box 9695 Marina del Rey, CA 90291
2	Science Applications, Inc. ATTN: W. Layson John Cockayne PO BOX 1303 1710 Goodridge Drive McLean, VA 22102
1	Science Applications, Inc. ATTN: Technical Library 1250 Prospect Plaza La Jolla, CA 92037
2	Systems, Science and Software ATTN: C. E. Needham Lynn Kennedy PO Box 8243 Albuquerque, NM 87198
3	Systems, Science and Software ATTN: Technical Library R. Duff K. Pyatt PO Box 1620 La Jolla, CA 92037
1	Texas Engineering Experiment Station ATTN: Dr. D. Anderson 301 Engineering Research Center College Station, TX 77843
2	TRW Systems Group ATTN: Benjamin Sussholtz Stanton Fink One Space Park Redondo Beach, CA 90278

DISTRIBUTION LIST

No. of Copies	Organization
1	Battelle Memorial Institute ATTN: Technical Library 505 King Avenue Columbus, OH 43201
1	California Inst of Tech ATTN: T. J. Ahrens 1201 E. California Blvd. Pasadena, CA 91109
2	Denver Research Institute University of Denver ATTN: Mr. J. Wisotski Technical Library PO Box 10127 Denver, CO 80210
1	TRW Ballistic Missile Division ATTN: H. Korman, Mail Station 526/614 P.O. Box 1310 San Bernadino, CA 92402
1	Massachusetts Institute of Technology Aeroelastic and Structures Research Laboratory ATTN: Dr. E. A. Witmer Cambridge, MA 02139
2	Southwest Research Institute ATTN: Dr. W. E. Baker A. B. Wenzel 8500 Culebra Road San Antonio, TX 78228
1	SRI International ATTN: Dr. G. R. Abrahamson 333 Ravenswood Avenue Menlo Park, CA 94025
1	Stanford University ATTN: Dr. D. Bershader Durand Laboratory Stanford, CA 94305

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: AMXSY-D
AMXSY-MP, H. Cohen
Cdr, USATECOM
ATTN: AMSTE-TO-F
Cdr, CRDC, AMCCOM
ATTN: SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number ______________________ Date of Report ____________

2. Date Report Received ______________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ______________________

____

____

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.) ______________________

____

____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate. ______________________

____

____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

____

____

____

CURRENT ADDRESS

____
Name

____
Organization

____
Address

____
City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

____
Name

____
Organization

____
Address

____
City, State, Zip

(Remove this sheet along the perforation, fold as indicated, staple or tape closed, and mail.)

FOLD HERE

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-9989

FOLD HERE

END

DTIC

8-86