ORNL/TM-10995

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

Recent ORNL Improvements to the HULL Hydrocode System

T. J. Burns

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-10995

Engineering Physics and Mathematics

Recent ORNL Improvements to the HULL Hydrocode System

T. J. Burns

DATE PUBLISHED — May 1989

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Modifications to the HULL hydrocode system are described. The modifications concern the problem generator of the system, KEEL. Revised algorithms for implementing combinatorial geometry, defining reference geometries, material "packaging," and restarting two– and three–dimensional calculations are described. The user input necessary to exercise the added/improved features is outlined and examples illustrating the revised capabilities are provided.

1. INTRODUCTION

The HULL hydrocode system[1] was selected as the principal analysis tool to support ORNL's Fast Track Defensive Shield Program and the Advanced Shield Phenomenology Program (ASP). Fast Track and ASP are programs of the Strategic Defense Initiative Organization's (SDIO) Passive Survivability Technology Program—Spacecraft Armor Subtask. The Fast Track program designed and demonstrated a lightweight combined hypervelocity impact and laser irradiation shield built from off-the-shelf materials. ASP is investigating advanced impact phenomenology to improve lightweight shield design and performance. Over the course of the programs, HULL has been used principally to model hypervelocity impact phenomena in greater and greater detail. As part of this modeling effort, the need to improve and extend the modeling capability of the system became apparent.

Although corrections, modifications, and extensions were made throughout the HULL code system as a result of its use in the ORNL SDI-related programs, this particular report is concerned with modifications/extensions to the problem generator, KEEL, of the HULL code system. The more significant of these former changes are described in Reference 2. They have been communicated to Orlando Technology, Inc. (currently maintaining the HULL system) and have been incorporated into Version 121 of HULL. The KEEL modifications implemented can be grouped into three broad categories:

1. modifications designed to facilitate the specification of the model itself (combinatorial geometry, reference geometries),

2. modifications designed to improve the accuracy of the resulting model (revised "packaging" and 2-D restart algorithms), and

3. modifications designed to extend the capability of the code system itself (implementation of a 3-D restart capability).

Chapter 2 provides the rationale for the modifications implemented into the HULL system as well as brief descriptions of the new algorithms. Chapter 3 is designed to serve as a revised user's manual for the GENERATE section of the KEEL program. It describes the required input which will access the new features. Chapter 4 provides a series of examples to illustrate both the use and the utility of the the additional features.

It should be noted that the changes were implemented in Version 121 of the HULL system, and have only been tested with this version.

2. MODIFICATIONS TO THE HULL HYDROCODE SYSTEM

The modifications to KEEL can be grouped roughly into four areas based on the type of modification made: implementation of combinatorial geometry, provision for the specification of reference geometries, increasing the accuracy of the "packaging" algorithm, and the recoding of the restart option to improve the accuracy and flexibility (as well as the provision for restarting a 3D calculation). The rationale for each of these improvements, as well as a brief description of how each was implemented in the HULL hydrocode, is given below.

2.1 COMBINATORIAL GEOMETRY

The problem generator of the HULL code system (KEEL) has been recoded to implement the three Boolean operators: OR, AND, and NOT, as part of a PACKAGE specification. Inclusion of all three operators permits a greatly simplified and more intuitive geometric description of the actual physical configuration to be used. Use of the OR operator corresponds to a union of two geometric shapes, use of the AND operator the intersection of two geometric shapes, and the NOT operator corresponds to the complement of a given shape.

The use of the Boolean operators however, can best be illustrated by an example. Consider an object composed of a sphere and a cylinder as shown in Figure 2.1a. If both are composed on the same material, then the combined geometry is given by

Zone2 OR Zone3,

which is interpreted to mean that a point is within the composite geometry if it is contained by the sphere or if it is contained by the cylinder or both (Figure 2.1b). [As defined here, the OR operator is an inclusive OR.] If the sphere and the cylinder are to be different materials, then a variety of composite geometries can be utilized, such as

Zone2 AND Zone3,

describing a cylinder with a rounded end (Figure 2.1c), or

Zone2 AND NOT Zone3,

describing a sphere with a cylindrical penetration (Figure 2.1d), or

Zone3 AND NOT Zone2,

describing a cylinder with a dished end (Figure 2.1e).

It should be noted the previous versions of KEEL did provide some limited Boolean capability through the use of the operators DELETE, INSIDE, and OUTSIDE. The correspondence between the two sets of operators is as follows:

ORNL-DWG 88Z-15198

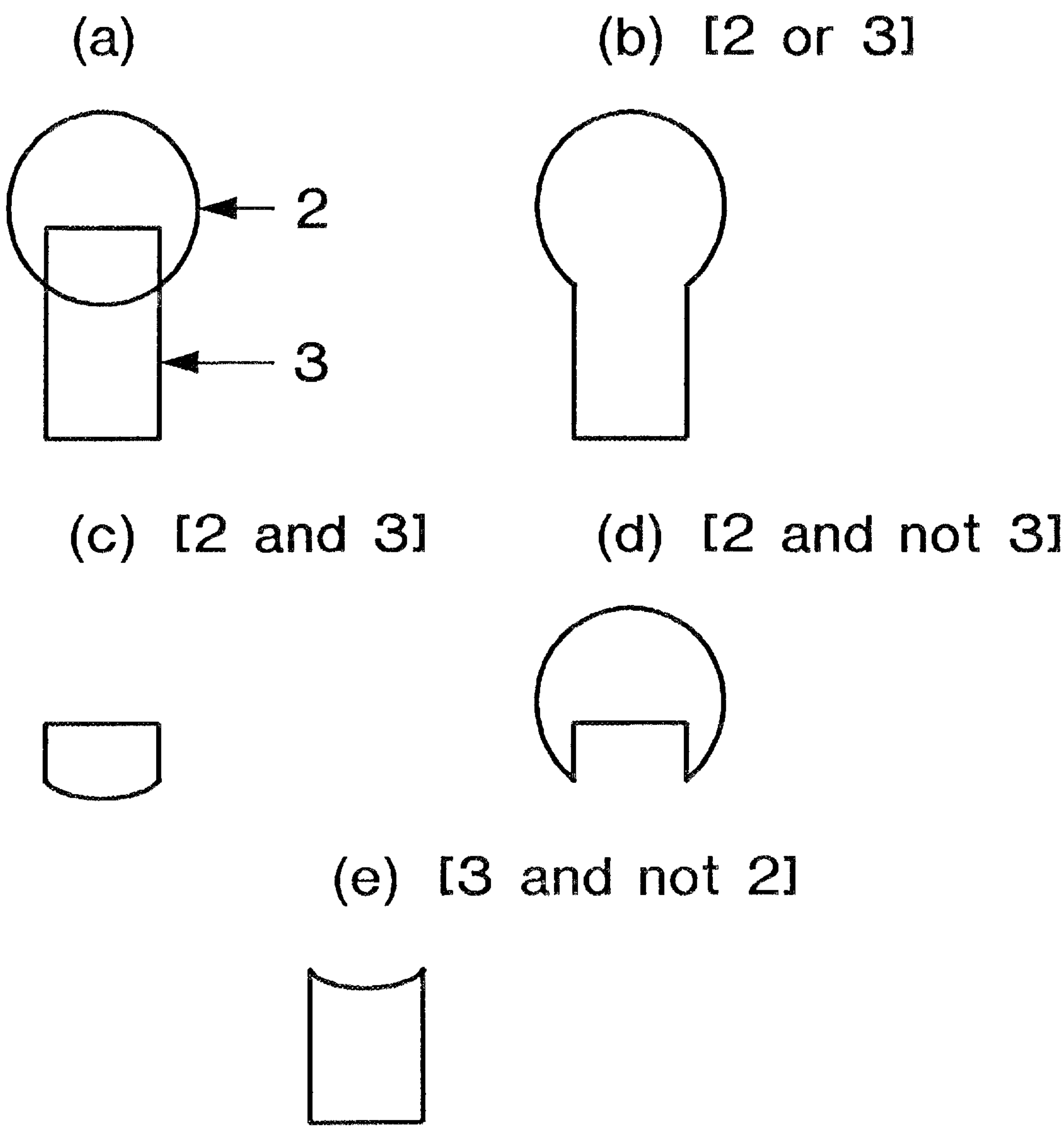

Figure 2.1. Combinatorial Geometry Illustration

DELETE geom = AND NOT geom

INSIDE geom = geom

OUTSIDE geom = NOT geom

DELETE OUTSIDE geom = AND NOT OUTSIDE geom = AND geom.

Thus, by judicious use of the DELETE and OUTSIDE operators, the intersection of two simple bodies could be constructed. However, the representation of an arbitrary shape within a single package is limited by the fact that no equivalent to the OR operator can be constructed.

The addition of the OR operator, as well as a simplified intersection operator, should greatly simplify the modeling of complex physical configurations. Moreover, since it requires a loop through the entire mesh to package a given material, the implementation of the OR operator will allow two disjunct regions consisting of the same material to be packaged in a single pass through the mesh.

Additionally, the previous operators DELETE, INSIDE, and OUTSIDE continue to be supported in the revised version to maintain downward compatibility, and can even be combined with any or all of the Boolean operators, if desired.

2.2 REFERENCE DESCRIPTIONS

The option to declare a geometry for the purposes of reference later in the input stream has also been implemented. Implementation of this feature in the HULL system was based on the observation that certain geometric specifications occur a number of times in setting up complex configurations. Rather than specify the complete geometry (i.e., shape, dimensions, etc.) a number of times, it seems preferable to define the geometry once and then merely refer to it when required. Such a procedure also guarantees that all references to a geometry are consistent since it is specified only once.

As an example of the use of both the Boolean operators and reference descriptions, consider the system shown in Figure 2.2, which consists of three geometric shapes: two rectangles (or boxes) labeled 1 and 3 and the circle (or sphere) labeled 2. The task is to describe the shaded Zone A and the unshaded Zone B. If Zone 1, Zone 2, and Zone 3 are defined (using the reference description options) as a rectangle, circle, and rectangle, respectively, Zone A can then be described as

Zone1 AND Zone2.

Similarly, Zone B can be described as

OR NOT Zone 1

OR NOT Zone 2

AND Zone 3.

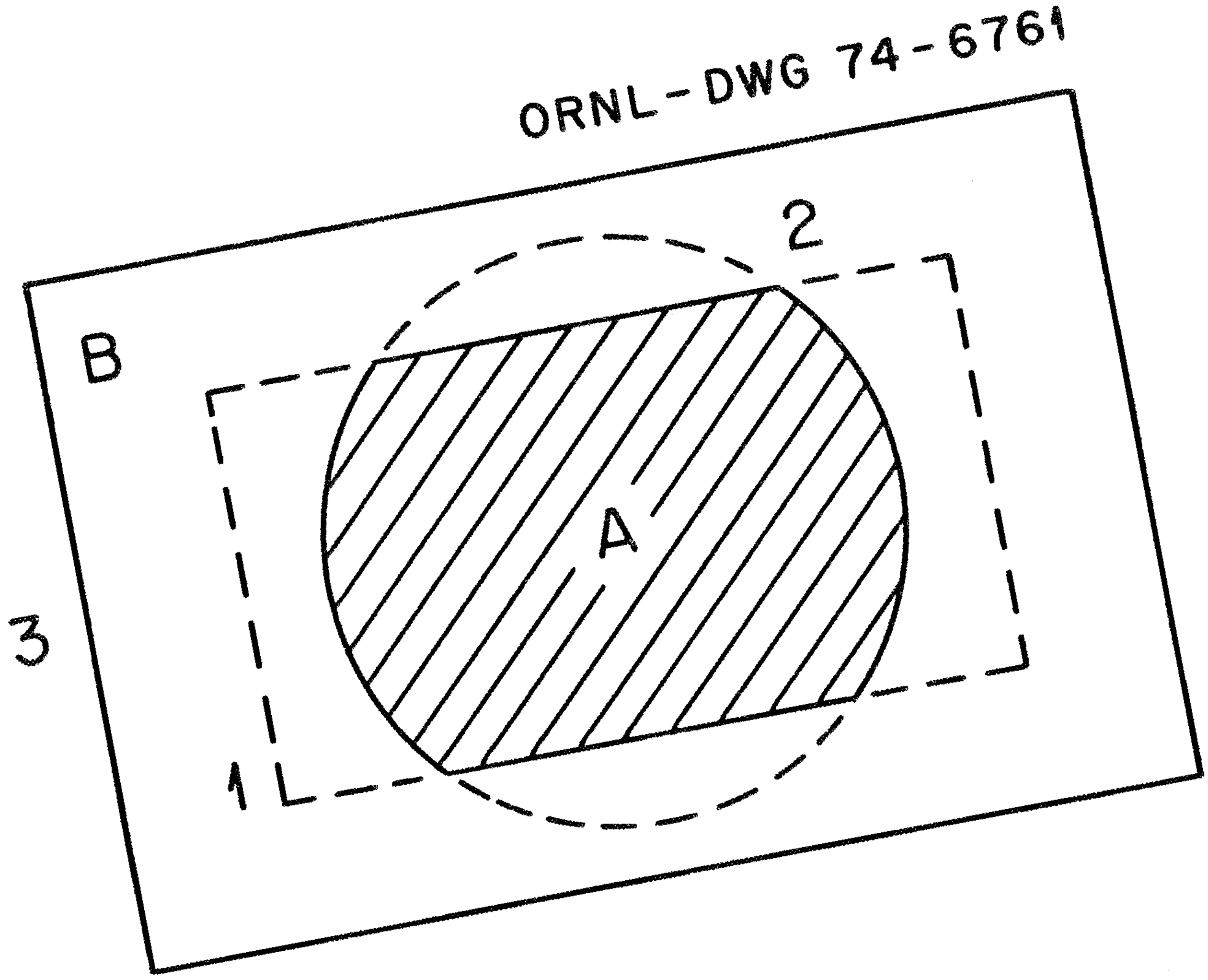

Figure 2.2. Illustration of the use of a reference geometry.

By way of contrast, although the old version of KEEL does provide the capability to describe Zone A as,

Zone1 DELETE Zone2,

it does not permit the specification of Zone B as a single package.

2.3 PACKAGING ALGORITHM

A major modification has been made to the "packaging" algorithm which is used to initialize the material properties of each mesh cell.

In prior versions of the HULL system, the determination of whether a particular material (i.e., a specific combination of material properties such as density, internal energy, etc.) was to be inserted or "packaged" into a given mesh cell, was implemented by dividing each mesh cell into a number of subcells. The default division resulted in 9 subcells per mesh cell (a 3 $\times$ 3 grid) for two-dimensional problems. The midpoint coordinates of the subcells were then checked to determine the presence or absence of the given material (i.e., correspondence with the geometric specification of the package). If the midpoint coordinates of the subcell were contained within the geometry specification, the entire subcell is filled with the material in question. This algorithm requires 9 calls to the geometry routines for each mesh cell and has a potential error for a specific cell of approximately 31%. (For three-dimensional problems, the corresponding numbers are 27 calls with an error potential of 42%.) Figure 2.3 illustrates the procedure for a typical geometry. Four of the 9 subcells are detected as being within the RECTANGLE and a volume fraction of 0.4444 of the original mesh cell filled with the material. The packaging error in this case is approximately 21%.

The potential error for any mesh cell, of course, can be made arbitrarily small by changing the number of subcells utilized, i.e., by modifying the Fortran source code to use a finer grid for the subcell computation. However, increasing the subcell grid increases the number of computations for every mesh cell in the problem, including those which are totally contained within a specific geometry and hence have no "packaging" errors.

In order to increase the accuracy of the "packaging" routines and minimize the amount of computation performed, an alternate algorithm has been developed. For each mesh cell, the corners of the mesh cells are first checked against the geometry specifications. In addition, the midpoint of the mesh cell is also checked (Figure 2.4). Thus, 5 points in two-dimensional problems (9 points in three-dimensional problems) are initially tested for a match with the geometry specified for the material. Only in the case of 1, 2, 3, or 4 matches are further computations necessary to resolve the amount of material to be inserted into the cell. Note that if none of the points correspond, the mesh cell can be skipped, whereas if all five points match, the entire cell can be filled with the material in question.

ORNL-DWG 88Z-15196

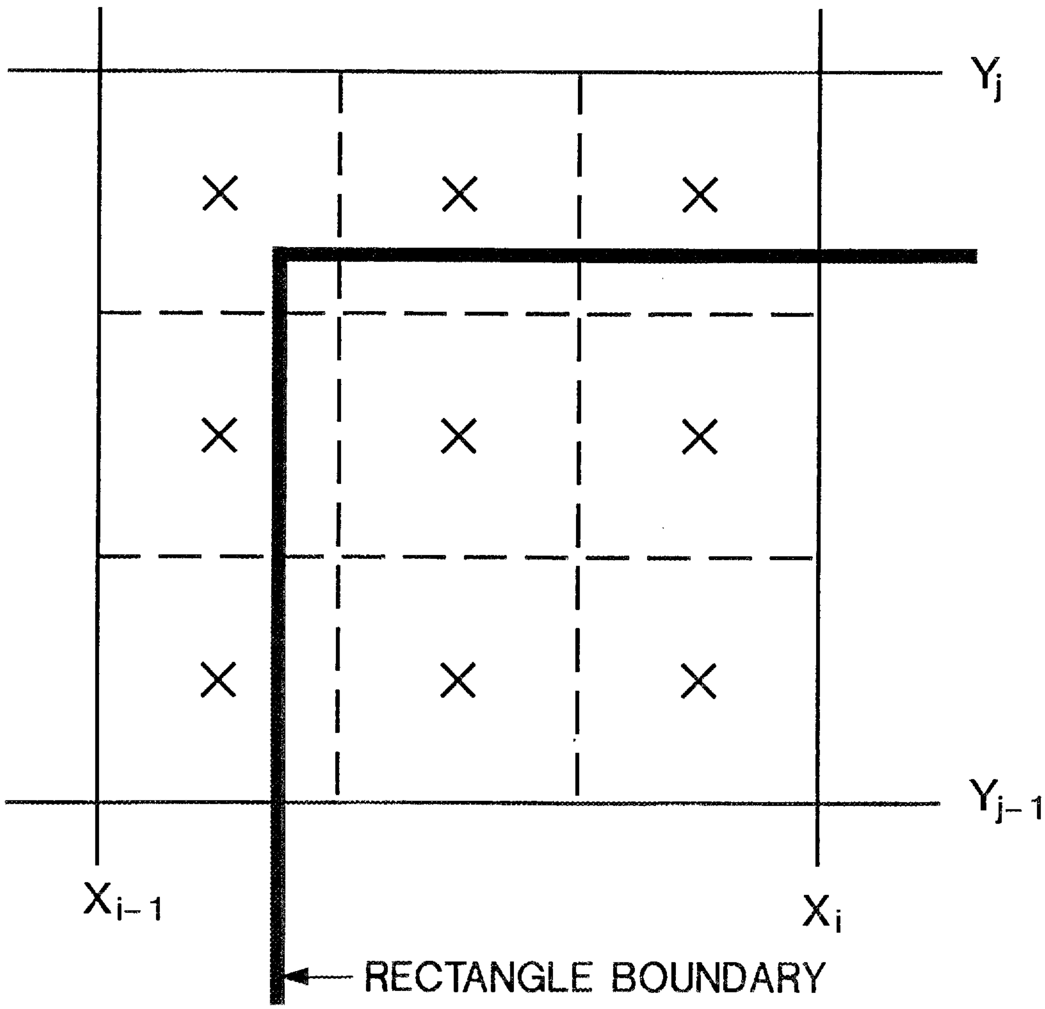

Figure 2.3. Coordinates (X) checked for mesh cell (i,j) using current HULL subcell division algorithm (3 × 3).

ORNL-DWG 88Z-15194

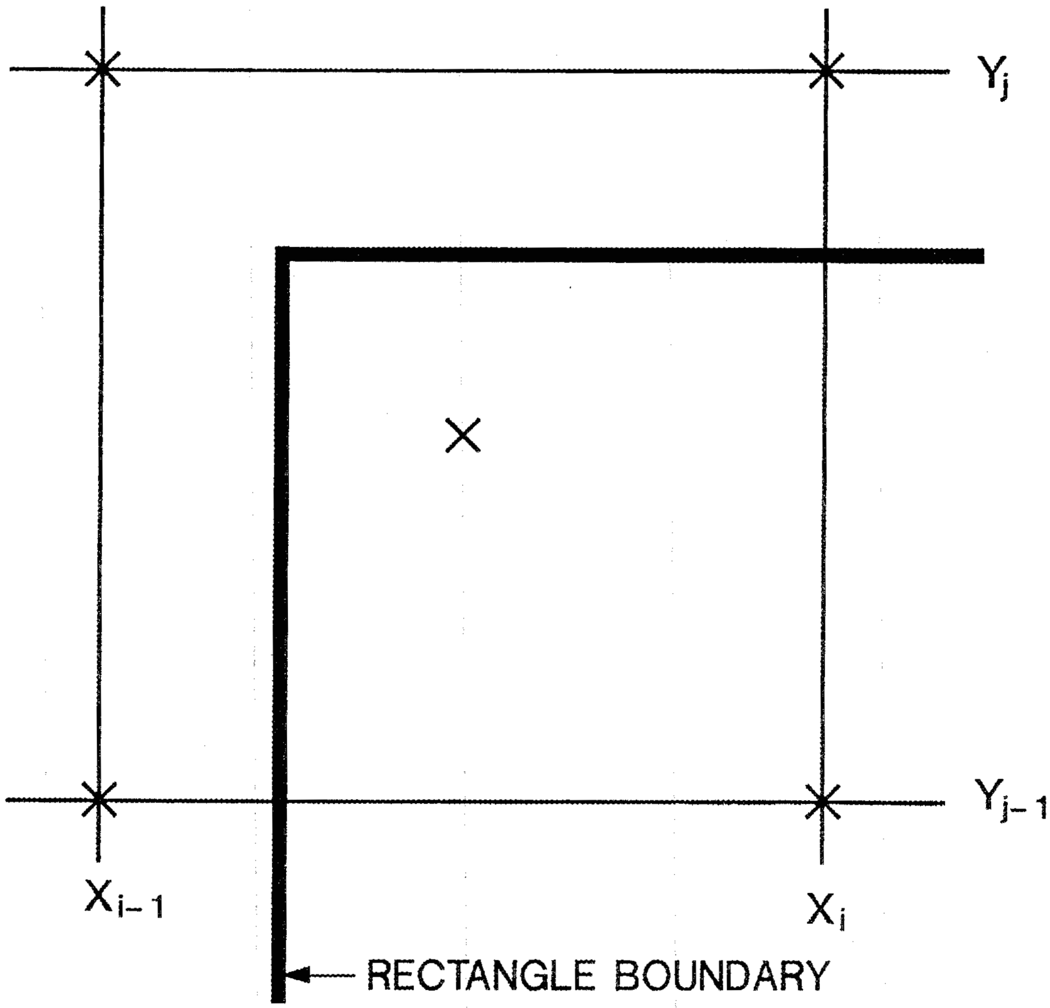

Figure 2.4. Coordinates (X) checked for mesh cell (i,j) using revised "packaging" algorithm—initial press.

For partially occupied cells, the number of subcells required to achieve a given tolerance relative to the volume fraction is determined. Figures 2.5a–c indicate the subdivision of the mesh cell for (admittedly coarse) tolerances of 100%, 25%, and 10%. The midpoint of each cell is checked against the geometry specification, and the subcell volume is included in the volume fraction if the point is within the geometry. For the example given, the volume fractions calculated via the revised algorithm are:

Tolerance	Subcells	Volume Fraction
1.00	4	1.0000
0.25	9	0.4444
0.10	16	0.5625
0.01	121	0.5289

Three points are worth noting. First, a tolerance of 0.25 corresponds to the old default of 3×3 subcell division and produces an identical volume fraction. Second, the exact result produced by a tolerance of 0.10 is an artifact of the example and is not reliable. Finally, the example tolerances are extremely coarse. The default value used by the code is currently 0.0001 (or .01%) resulting in four digit accuracy for the volume fraction regardless of the type of geometric figure contained in the mesh cell.

The revised algorithm is thus reactive to the type of mesh cell being processed, reducing the computations by approximately 50% for the readily identifiable cells, and increasing the accuracy for the partially filled cells. Using the default tolerance results in a subdivision of approximately 100×100 cells, but only for those mesh cells which "fail" the screening criteria. By way of contrast, achieving the same tolerance using the old algorithm (i.e., specifying a subcell division of 100×100) would result in a factor of 1100 increase in calculational time, since all the mesh cells would be treated identically.

It should be noted that due to the 5 point screening criteria utilized for each mesh cell, the algorithm will fail to resolve geometries whose extent is less than half of the mesh spacing. The original algorithm, with appropriate changes to the subcell division parameters, is capable of resolving geometries of any extent. The rationale for this limitation is that, if geometries of such a limited extent are important to the problem, the mesh size should have been chosen small enough to adequately model them in the first place.

An additional option, that of replication, has also been added to the packaging algorithm. Although the KEEL module previously incorporated the capabilities for translation and rotation, the option to replicate an object multiple times was judged to be desirable. The replication option is programmed as an extension of the translation option (i.e., a simple translation being one replication) and can be combined with the rotation option. Thus, a complex object may be defined (complete with rotation) and multiple copies of the object positioned throughout the model.

ORNL-DWG 88Z-15195

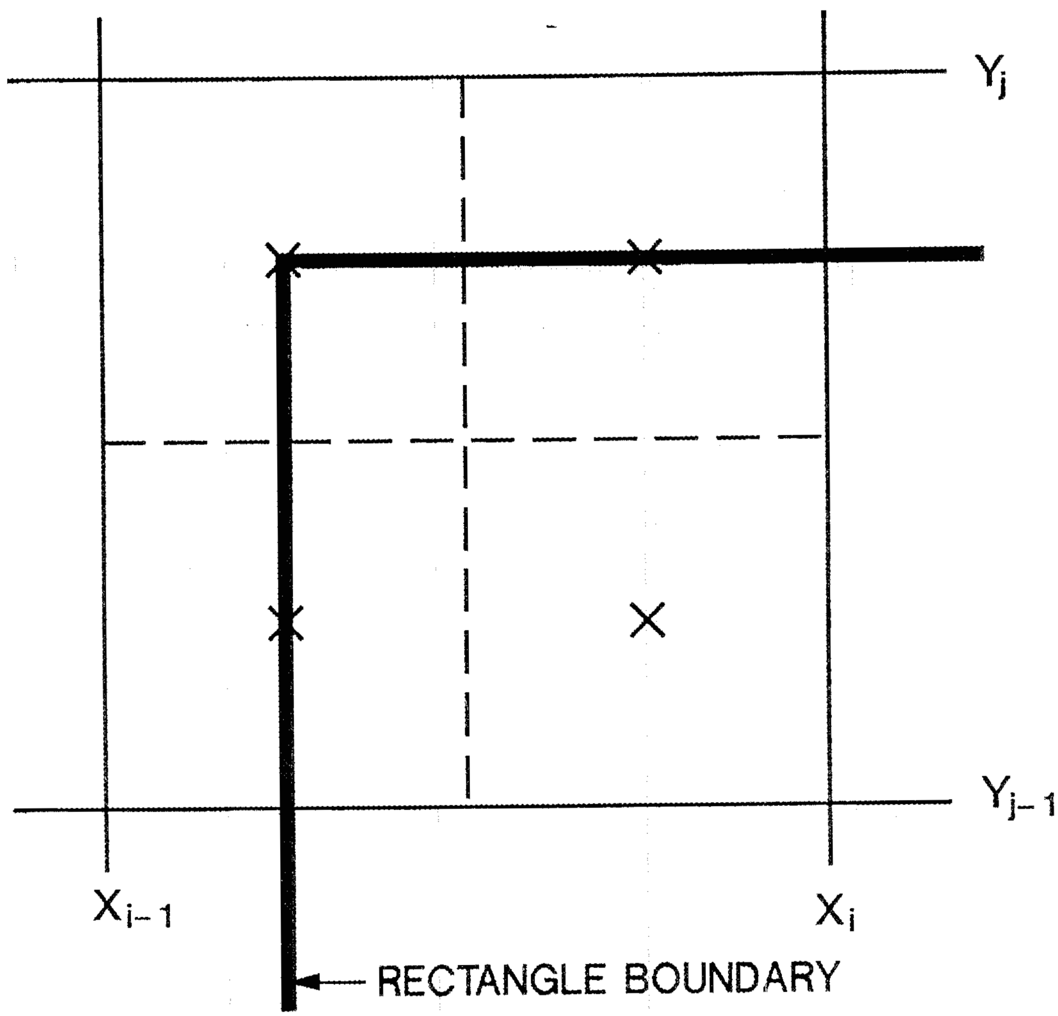

Figure 2.5a. Coordinates (X) checked for mesh cell (i,j) using revised "packaging" algorithm—second pass tolerance = 100%.

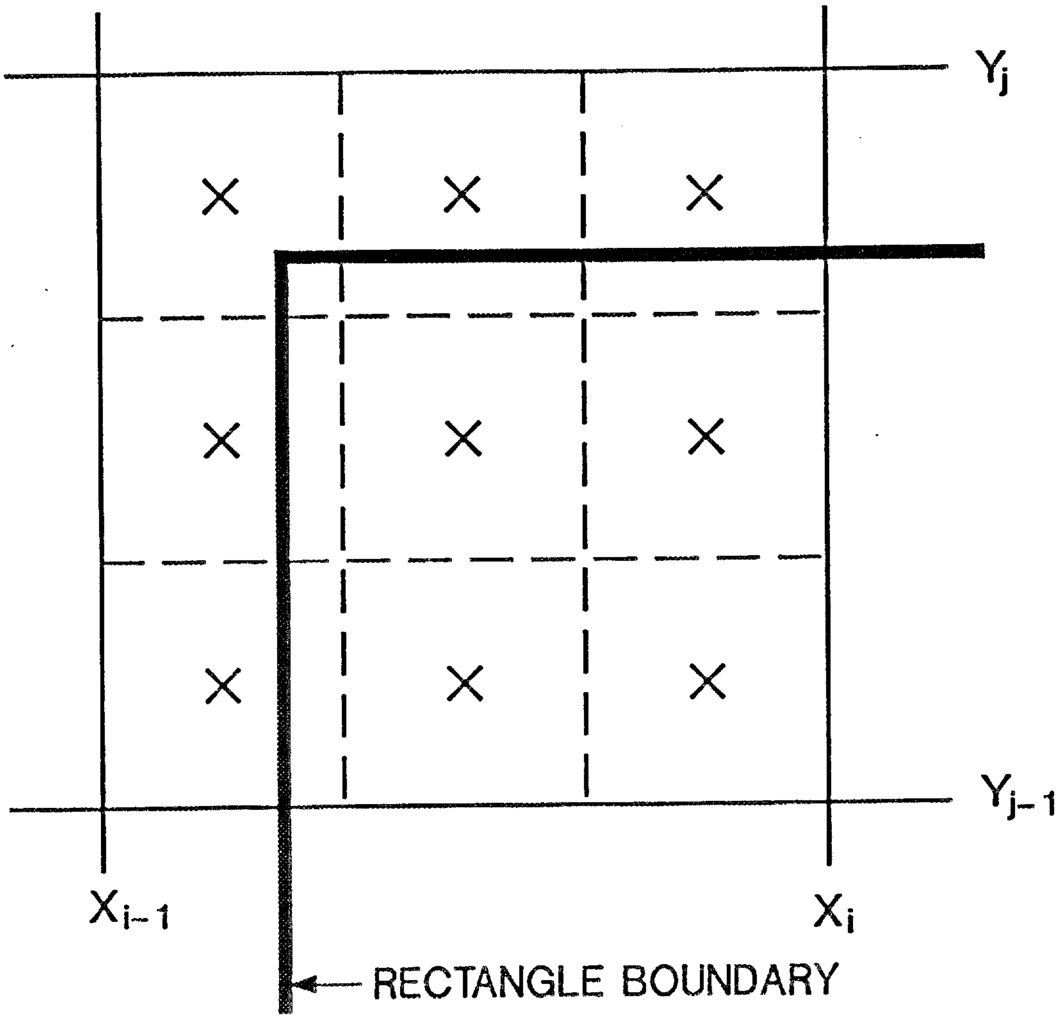

Figure 2.5b. Coordinates (X) checked for mesh cell (i,j) using revised "packaging" algorithm—second pass tolerance = 25%.

ORNL-DWG 88Z-15197

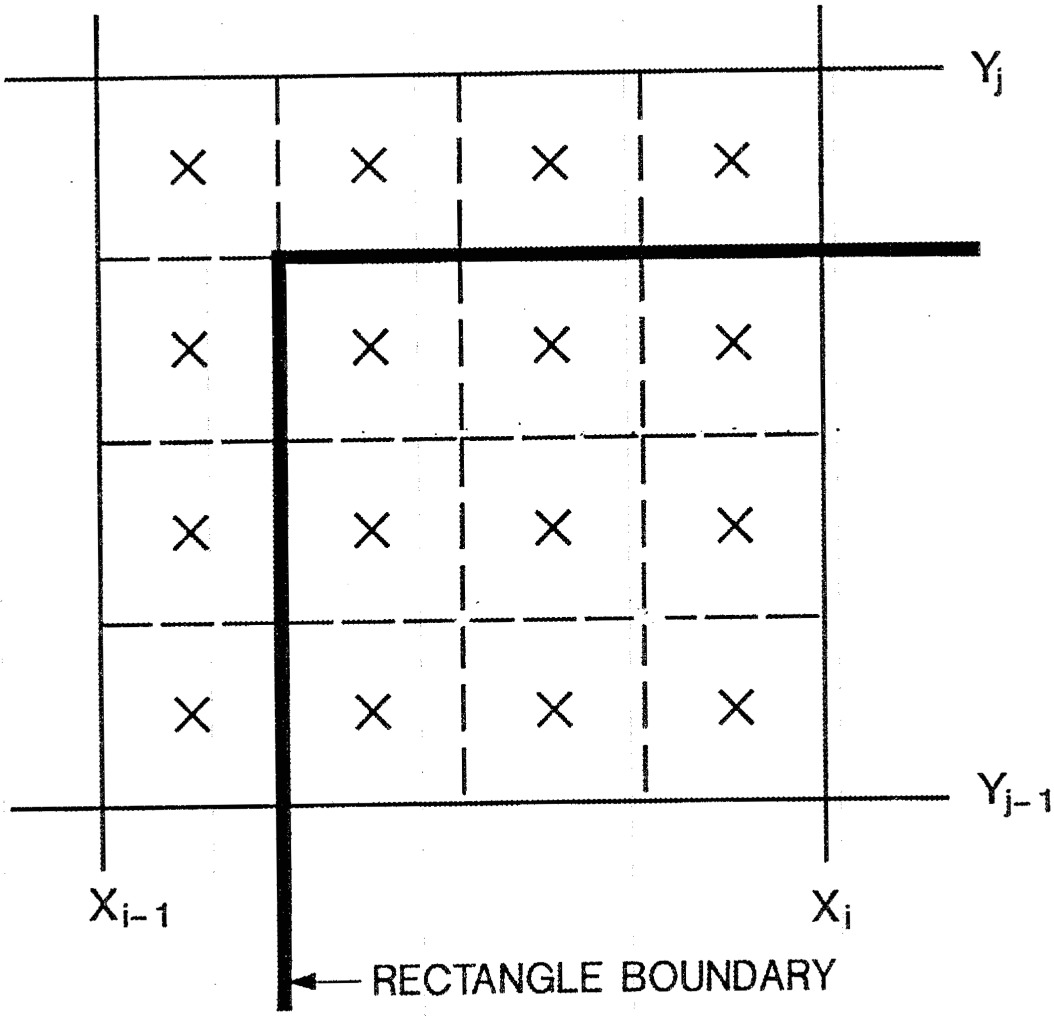

Figure 2.5c. Coordinates (X) checked for mesh cell (i,j) using revised "packaging" algorithm—second pass tolerance = 10%.

2.4 RESTART CAPABILITY

A necessary facility in any hydrocode is the capability of using all or part of the results of an existing calculation to initialize a new model. Such a facility should include the capability of selecting certain portions of the existing data, together with the options for changing various problem parameters. Such a facility could be very useful in a variety of applications such as:

1. treating various parts of a problem at differing levels of detail,
2. minimizing the computational expense of parametric studies, and
3. overcoming operating system limits on problem size.

Modeling a physical configuration in HULL is subject to two different (and sometimes mutually exclusive) goals. In the interests of accuracy, it is advantageous to design the mesh to have cells as nearly square as possible. However, many of the configurations which must be modeled, such as thin plates, lend themselves to being modeled with very elongated cells. Thus, although such detail inherent with the use of square cells may not be necessary in each dimension, the accuracy criterion requires their use. More importantly, since a given mesh spacing extends over the entire problem, the modeling of other regions is influenced by the presence of such structures. One way to minimize this problem is to split the calculation into two or more parts, with each calculation being modeled to a different degree of detail. Thus, one could use a fine mesh spacing to investigate the penetration of an initial thin plate and then create a new model using a coarser mesh (and the aggregated data from the first calculation) to continue the problem.

A second reason for inclusion of a restart facility is to minimize the expense of certain parametric studies. For example, an initial impact problem can be set up and run to completion. Using the restart capability, additional problems can be initialized using the state of the initial calculation at any appropriate time. Materials unaffected by the impact (at the specified time) could be replaced with alternate ones and the calculation continued to completion. Thus, once the initial calculation is done, it can serve as a "source" for a large number of parametric studies.

An additional rationale for a complete restart facility is the desire to split the results of a calculation into successive steps. It should be noted that a restart is distinguished from a continuation in that it essentially begins a new problem. A continuation merely adds further data to an existing problem. In particular, three-dimensional problems tend to produce very large data files. It is useful (and sometimes mandatory) to be able to split the data files into smaller files which are more manageable. The restart facility would allow a "new" problem to be initiated (and hence a new data file started) at any time.

Although the HULL system previously contained a restart facility, this mechanism was limited. It was only applicable to two-dimensional problems, and more importantly, the algorithm for "packaging" the old materials relied on a variant of the subcell algorithm discussed above and hence was subject to the same limitations. That is, each of the new mesh cells was partitioned into a number of subcells, and the midpoint coordinates of each subcell were tested against the old mesh.

The revised algorithm for "packaging" the old mesh into the new mesh specification is based on a distinctly different approach. The old and new mesh are first concatenated to form a revised combined mesh. This insures that each of the revised mesh cells either overlays one of the old mesh cells or is beyond the limits of the old mesh (i.e., it is not part of the old problem). Thus, the only determination which must be made is whether the old mesh cell is contiguous with the revised mesh cell, and if so, whether it contains a material (or materials) which is to be incorporated into the new mesh. It should also be noted, that since both the old and the new mesh are rectangular regions and the extent of each is readily determined, only those regions containing the old mesh need to be processed at all.

3. MATERIAL AND GEOMETRY DEFINITIONS

Overall material and geometry definitions are accomplished in KEEL via the GENERATE keyword. Specific input within the GENERATE block is accomplished by data prefaced by one of six keywords: DEFINE, PACKAGE, FIREIN, DETGEOM, PARTICLES, or STATIONS. The required input for each of these subblocks is described below.

As in the rest of the HULL system, data input usually consists of keyword and data value pairs, separated by a blank, comma, or =. The GENERATE block is terminated by the keyword ENDGENERATE or by the end of the input data stream.

3.1 DEFINE

Reference objects are defined via the following construct in the input stream:

DEFINE refname geom gparms goptions

where	refname	is a user selected name [8 characters on the Cray] by which the description will be referenced,
	geom	is the name of a geometric shape. Valid geometries for X–Y, R–Z, and X–Y–Z are listed in Tables A.1 and A.2
	gparms	is a parameter list of dimensions required to size the particular shape such as the radius of sphere. Any dimensions not specified are assigned the default values listed in Tables A.1 and A.2. Additionally, Table A.3 lists acceptable sets of synonyms for the various parameters.
	goptions	is a list of geometry options which apply to the object defined by geom and gparms, such as translation, rotation, and replication.

The goptions are used to translate, rotate, and replicate the specific shape. Valid translation options (retained from the previous versions) are

XCC = x1 YCC = y1 ZCC = z1.

Use of these options permits the geometric frame of reference to be displaced x1 in the x direction, y1 in the y direction, and z1 in the z direction. If not explicitly input, each of these options defaults to 0.0.

Replication of a defined object can be invoked using

$$\begin{aligned} \mathrm{DXX} &= \mathrm{x} \quad \mathrm{NXX} = l \\ \mathrm{DYY} &= \mathrm{y} \quad \mathrm{NYY} = \mathrm{m} \\ \mathrm{DZZ} &= \mathrm{z} \quad \mathrm{NZZ} = \mathrm{n}\ . \end{aligned}$$

Here, the object is replicated l times in the x direction with a spacing of x starting from the geometric frame of reference (XCC,YCC,ZCC), m times in the y direction with a spacing of y, etc. Negative values for DXX, DYY, and DZZ cause the replication to take place in the negative direction of the respective axis.

Similarly, rotation options (about the geometric frame of reference) are invoked via

$$\begin{aligned} \mathrm{ANGLA} &= \mathrm{a} \quad \mathrm{DANA} = \mathrm{da} \quad \mathrm{NDA} = l \\ \mathrm{ANGLB} &= \mathrm{b} \quad \mathrm{DANB} = \mathrm{da} \quad \mathrm{NDB} = \mathrm{m} \\ \mathrm{ANGLC} &= \mathrm{c} \quad \mathrm{DANC} = \mathrm{da} \quad \mathrm{NDC} = \mathrm{n} \end{aligned}$$

corresponding to rotations about the Z, Y, and X axes respectively. The DANA and NDA fields, if present, permit the replication of the object l times at increments of DANA degrees, beginning with ANGLA. Positive values for ANGLA, ANGLB, and ANGLC denote counter–clockwise rotations. Rotated copies of the geometry are considered as multiple OR geometries, i.e., the resultant figure is the union of all the rotations.

Obviously, the definition itself must occur prior to any reference to it in the input stream, and the number of definitions is currently limited to 50 per GENERATE block. Use of subsequent GENERATE blocks in the input stream will cancel all previous definitions.

3.2 PACKAGE

Materials are inserted into the mesh using the PACKAGE construct. The PACKAGE construct typically consists of a material specification followed by up to 10 geometry specifications. It identifies both the material to be inserted as well as the region of the mesh for which the insertion is to take place. The hydrodynamic data for the material can come from one of two sources, either via a complete specification as part of the input stream (i.e., typically used for an initial problem setup), or from a previous HULL calculation (i.e., via the FIREIN option. For an initial problem definition, the PACKAGE construct is given by:

```
PACKAGE matname U=u V=v W=w I=i RHO=rho

                {Boolean Ops} [ geom1 {gparms1} {goptions1} ]
                              [ refname1 ]
                {Boolean Ops} [ geom2 gparms2 goptions2 ]
                              [ refname2 ]
                .
                .
                .
                {Boolean Ops} [ geom10 gparms10 goptions10 ]
                              [ refname10 ]
```

where	matname	are material names which appear in the Equation of State library
	u	is the x (or r) component of velocity (cm/sec),
	v	is the y (or z) component of velocity (cm/sec),
	w	is the z (3D only) component of velocity (cm/sec),
	i	is the internal energy of the material (ergs/gm),
	rho	is the density of the material (gm/cm**3), and
	Boolean Ops	are the operators AND, OR, and NOT. (The operators DELETE, INSIDE, and OUTSIDE are also supported for compatility with previous versions).

The specification of U, V, W, I, and RHO is optional. Unspecified velocities default to zero, while I and RHO default to the ambient values in the Equation of State library.

The use of the Boolean operators is also optional. However, if they are invoked at least 1 geometry in the description must be characterized as an OR geometry. If they are not invoked, the first geometry encountered in a description is defaulted to OR, while subsequent geometries default to DELETE (AND NOT) to maintain compatibility with previous versions of the HULL system.

The various geometric portions of the PACKAGE specification can be either a

geom gparms goptions

construct as discussed above in the section on reference definitions, or can consist of merely the name (refname) assigned in a previous DEFINE statement.

The keyword PACKAGE is only required if more than one geometry specification is to be used to specify the appropriate region. Otherwise, the use of a material name (matname) followed by a single geometry or reference specification is sufficient.

3.3 FIREIN

Hydrodynamic data from previous calculations is inserted into the mesh using the FIREIN construct. It consists of the keyword FIREIN followed by a description of the source of the data to be utilized, i.e.

FIREIN HULL FILE = olddata.

Here, the word HULL indicates a previous HULL data tape identified by the name olddata. The construct illustrated assumes that the old data file is a local file on the CRAY. Following the specification of the source of the data, the materials and placement of the materials in the new mesh are specified using a modified version of the PACKAGE construct, i.e.,

```
PACKAGE matname  {Boolean Ops} [ geom1 {gparms1} {goptions1} ]
                               [ refname1 ]
                 {Boolean Ops} [ geom2 {gparms2} {goptions2} ]
                               [ refname2 ]
                 .
                 .
                 .
                 {Boolean Ops} [ geom10 {gparms10} {goptions10} ]
                               [ refname10 ],
```

which indicates that the material identified by matname is to be inserted into the mesh as described by the geometry specified. Multiple PACKAGE constructs may be used to insert different materials into different regions of the mesh. Alternatively, the single matname may be replaced by a list of material designators to insert the selected materials into the same region of the mesh. Finally, if all the materials of the old mesh (i.e., those defined for the new mesh) are to be inserted into a common region, the matname designator can be replaced by the keyword ALL, which is the default if no materials are specified.

As in the normal PACKAGE specification, the keyword PACKAGE is only required if more than one geometry specification is to be used to specify the appropriate region. Otherwise, the use of a material name (matname) followed by a single geometry or reference specification is sufficient.

The FIREIN data block is terminated whenever a DEFINE, FIREIN, DETGEOM, PARTICLE, or STATION keyword is encountered in the input stream or the end of the input stream is reached. To allow for the possibility of resuming normal PACKAGE input, the FIREIN data block can also be terminated by the keyword ENDFIRE.

3.4 DETGEOM

The DETGEOM data block is utilized to specify the detonation parameters for explosives to be included in the problem. These parameters are the time of detonation, detonation wave velocity, and the initial location of the detonation.

It should be noted that the combinatorial aspects of the detonation package have not yet been implemented. Thus, the following discussion is tentative at this point.

The location for ignition can be a point, line, or surface depending on the dimensionality of the problem. The detonation parameters are specified using (default values are in brackets),

DETGEOM

TDET = time of ignition [Initial problem time]

VDET = detonation velocity [8.5 km/sec]

{Boolean Ops} [geom1 {gparms1} {goptions1}]
[refname1]
{Boolean Ops} [geom2 {gparms2} {goptions2}]
[refname2]

.
.
.

{Boolean Ops} [geom10 {gparms10} {goptions10}]
[refname10].

In addition to the standard geometries, listed in Tables A.1 and A.2, the following specifications are permitted in a DETGEOM block,

POINT XD=x YD=y ZD=z,

where (x,y,z) is the detonation point, or

LINE XD1=x1 YD1=y1 ZD1=z1,
XD2=x2 YD1=y2 ZD1=z2.

Here, cells lying along the line (x1,y1,z1)–(x2,y2,z2) are detonated at TDET.

As currently programmed, the DETGEOM block is terminated by the keyword ENDDET, and must preceded the PACKAGE specification which describes the explosive.

3.5 PARTICLES

Lagrangian trace particles can be specified via the PARTICLES construct. It is specified similar to the PACKAGE construct, i.e.,

```
PARTICLES   {Boolean Ops} [ geom1 {gparms1} {goptions1} ]
                          [ refname1 ]
            {Boolean Ops} [ geom2 {gparms2} {goptions2} ]
                          [ refname2 ]
            .
            .
            .
            {Boolean Ops} [ geom10 {gparms10} {goptions10} ]
                          [ refname10 ],
```

This construct results in the specification of a Lagrangian trace particle for every mesh cell whose center lies within the geometry specified.

The PARTICLES data block is terminated whenever a DEFINE, PACKAGE, DETGEOM, PARTICLE, or STATION keyword is encountered in the input stream or the end of the input stream is reached.

3.6 STATIONS

The STATION data block is used to specify data collection points in the mesh. Although it is unaffected by the modifications made to the rest of the GENERATE block, the input specifications are included here for completeness. This data block consists of

```
STATION XS = x1 x2 x3 ... xl
        YS = y1 y2 y3 ... ym
        ZS = z1 z2 z3 ... zn
```

The location of data collection points are at each (xi, yj, zk). Thus the total number of data collection locations is l * m * n, which must be less than the global KEEL parameter NSTN. The construct given above utilizes Eulerian stations (i.e. fixed with respect to the Eulerian reference). Lagrangian behavior (i.e. following the flow) can be invoked independently for any (or all) coordinate directions by replacing the XS, YS, and/or ZS by XL, YL, or ZL as desired.

The STATION data block is terminated whenever a DEFINE, PACKAGE, FIREIN, DETGEOM, PARTICLE, or another STATION keyword is encountered in the input stream or the end of the input stream is reached. Multiple STATION blocks are allowed with the restriction that the total number of stations specified are less than NSTN.

4. EXAMPLES

4.1 EXAMPLE 1. SHAPED CHARGE

Figure 4.1 is a simplified two-dimensional example which will be used to illustrate both the use of the DEFINE construct as well as the revisions made to the PACKAGE construct. The GENERATE block input used to model the shaped charge depicted in Figure 4.1 is

```
generate

    define standoff quadrilateral
      x1=0.0 y1=0.0 x2=5.0 y2=0.0 x3=2.5 y3=-4.5 x4=0.0 y4=-8.0

    define liner quadrilateral
      x1=0.0 y1=0.0 x2=5.0 y2=0.0 x3=2.5 y3=-5.0 x4=0.0 y4=-8.5

    define case2 rectangle
      x1=0.0 y1=0.0 x2=5.0 y2=-12.0

    define case1 rectangle
      x1=0.0 y1=0.0 x2=7.0 y2=-14.0

    package air standoff
    package cu liner
          and not standoff
    package octol case2
          and not liner
    package ssteel case1
          and not case2
    package air rectangle
          and not case1
```

The basic model consists of a cylindrical case partially filled with high explosive. A tapered Cu liner, together with the standoff, complete the simple model.

The initial GENERATE specifications consist of 4 DEFINE statements which serve to define the major components of the geometry, mnemonically labeled as standoff, liner, case1 and case2. For the sake of the example, both the standoff and the liner are constructed as quadrilaterals rather than triangles or ellipses. Each of the DEFINE statements serves to associate the assigned name with a geometry specification.

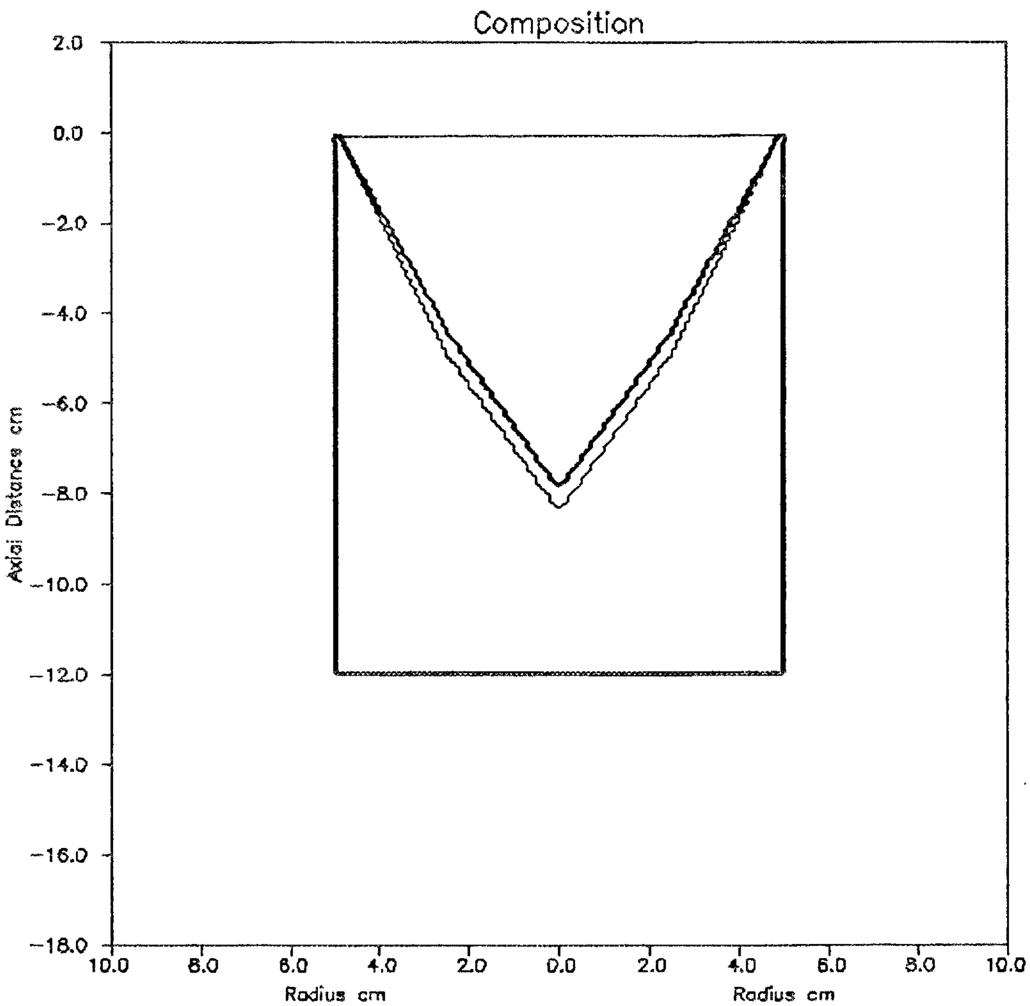

Figure 4.1. Combinatorial Geometry, Example 1—Shaped Charge

Following the definitions, the various materials are packaged as follows:

1. The geometry defined by STANDOFF is filled with air
2. The Cu liner is created by specifying that the material, Cu, is to be inserted into those cells interior to the larger quadrilateral (LINER) but exterior to the smaller (AND NOT STANDOFF)
3. Similarly, the high explosive (Octol in this case) is inserted in those cells interior to the smaller rectangle (CASE2) but exterior to the Cu liner.
4. The case for the shaped charge is packaged as stainless steel using the "difference" between the two rectangles.
5. Finally, the model is completed by surrounding the case by air. The initial specification essentially defaults to an infinite rectangle from which the external boundary of the case and all interior points are excluded.

Although this example is rather simple, it does illustrate two significant points. As can be noticed from the input description, the material air is used twice. To simplify the input even further, it is tempting to combine the two specifications into a single one using the OR operator, i.e.,

```
package air        rectangle
                or standoff
           and not case1.
```

This is, of course, incorrect since the last specification (AND NOT CASE1) would exclude all points in the interior from being filled.

The other point is somewhat more subtle. If the input specified is actually used in a KEEL run, the output listing reports that the single cell where the Cu liner tapers to a point is only 97.46% filled. Closer examination of this cell indicates that the Cu is the missing material since the cell contains both air and octol. This discrepancy is the result of the screening criteria outlined in Chapter 2. The initial check for materials to be packaged in a mesh cell uses the coordinates of the corners and the cell center to determine whether to insert a specific material. The wedge of Cu contained in the cell is small enough so that only one of the 5 points checked indicates that it should be present in the cell. Furthermore, the single point also corresponds to a point in the excluded geometry (STANDOFF), in effect canceling out any insertion of Cu into the cell. This type of result serves to alert the user that the geometry specifications have resulted in objects smaller than the resolution of the mesh cells. In this case, however, it is unlikely that the missing material will significantly affect the results of the calculation. Nevertheless, a mechanism is available to "complete" the model. By using the following specification as the final packaging option,

```
fill cu rectangle
```

any partially filled cells in the problem will be supplemented using Cu. In this case, filling in the missing 2.54% in a single cell.

4.2 EXAMPLE 2. THREE-DIMENSIONAL EXAMPLE

The following example is used to illustrate some of the newly programmed capabilities of the combinatorial geometry. It can be viewed as the construction of a model designed to simulate the oblique impact of a tomato with a screen door at 7 km/sec. The GENERATE commands necessary to construct such a model are as follows:

```
generate

  define wire box x1=-0.5 z1=-1.0 x2=0.5 z2=0.0

  define tomato sphere xc=0.0 yc=0.0 zc=-1.0 r=1.0

  package water u=-3.5e+07 w=6.062e+07 tomato

  package air u=-3.5e+07 w=6.062e+07
                 cylinder xc=0.0 yc=0.0 z2=-1
                         anglb=30.0
          and not tomato

   package ssteel   wire xcc=-4.0 zcc=1.0
                         nxx=5 dxx=2.0
                 or wire ycc= 0.0 zcc=2.0
                         angla=90.0
                         nyy=3 dyy=2.0
                 or wire xcc=-5.0 zcc=3.0
                         nxx=6 dxx=2.0
                         or wire ycc=1.0 zcc=4.0
                         angla=90.0
                         nyy=3 dyy=2.0
```

Two DEFINE commands are used in this specification. The first defines a reference geometry designated as wire which will serve as the basic building block for the screen door. It is a rectangular parallelepiped of square cross-section (XZ) which (since no y coordinates are used) is essentially infinite in the y-direction. The second definition is used to define the tomato itself as a 2-cm diameter sphere. These two definitions are then used to actually construct the model.

The first package specification is for the projectile itself, using water as the material and tomato as the geometry. Since the HULL materials library does not specifically contain material properties for tomatoes, it was felt that water would be a reasonable approximation (particularly at 7 km/sec). Two velocity components are also specified since this is to be an oblique impact.

The second specification is concerned with the slipstream behind the tomato. Using air as the material is constructed as a semi-infinite cylinder beginning at the center of the tomato and extending in the negative z-direction. Since the impact is oblique, the cylinder is rotated 30 degrees about the y-axis and the velocity components specified accordingly. Finally, the region consisting of the tomato is excluded from the specification.

The final specification is for the screen door itself, and consists of 4 geometries OR'ed together, corresponding to 4 layers of wires. The reference geometry labeled wire is translated 4 cm in the negative x direction, and 1 cm in the positive z direction. It is then replicated 5 times in the XY plane at intervals of 2 cm to construct the first layer. Similarly, the wire is translated 2 cm in the positive z direction, rotated 90 degrees about the z-axis, and then replicated to construct the second layer. The remaining two layers are handled similarly except that the initial translations are adjusted to offset layers 3 and 4 from layers 1 and 2. Figures 4.2a-d indicate the results of these operations for the four layers. [Note that Figures 4.2b and 4.2d are displaced by 1y increment due to an inconsistency in the HULL plotting program.]

This example indicates the power of the combinatorial geometry options included in KEEL. Specifying the interlocking mesh alone using the old version of KEEL would have required 17 separate package specifications, each with its own set of geometric parameters. It is anticipated that as the computational models become more complex, the advantage of the revised KEEL input options will become more apparent.

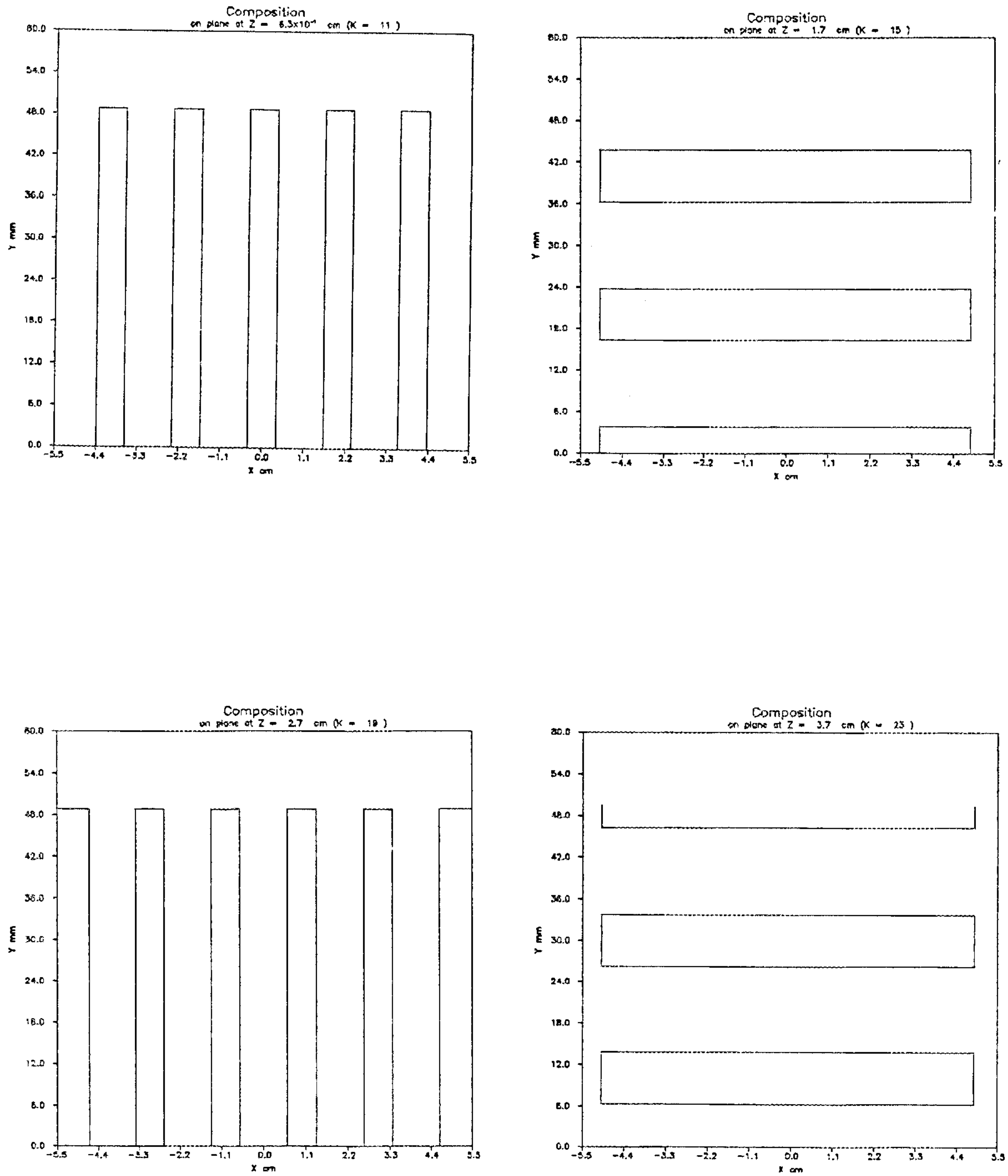

Figure 4.2a-d. Combinatorial Geometry (Z-Planes), Example 2—Tomato

APPENDIX

Table A.1. Two-Dimensional Geometry Options

The following is a list of the various geometry options available for 2 dimensional problems, together with the input parameters required for their specification. Unless, otherwise specified the parameters can be input in any order. Default values are indicated in brackets. Newly added options are prefaced with an "*".

1. RECTANGLE	X1=x1 [-1.0e+06] Y1=y1 [-1.0e+06] X2=x2 [+1.0e+06] Y2=y2 [+1.0e+06]

A rectangle with vertices at (x1,y1),(x1,y2),(x2,y1), and (x2,y2)

2. TRIANGLE	X1=x1 [-1.0e+06] Y1=y1 [-1.0e+06] X2=x2 [+1.0e+06] Y2=y2 [+1.0e+06] X3=x3 [-1.0e+06] Y3=y3 [+1.0e+06]

A triangle with vertices at (x1,y1),(x2,y2), and (x3,y3)

*3. PARALLELOGRAM	X1=x1 [-1.0e+06] Y1=y1 [-1.0e+06] X2=x2 [+1.0e+06] Y2=y2 [+1.0e+06] X3=x3 [-1.0e+06] Y3=y3 [+1.0e+06]

A parallelogram defined by the line segments (x1,y1)-(x2,y2), and (x1,y1)-(x3,y3)

*4. QUADRILATERAL	X1=x1 [-1.0e+06] Y1=y1 [-1.0e+06] X2=x2 [+1.0e+06] Y2=y2 [-1.0e+06] X3=x3 [+1.0e+06] Y3=y3 [+1.0e+06] X4=x4 [-1.0e+06] Y4=y4 [+1.0e+06]

A quadrilateral defined by vertices (x1,y1),(x2,y2),(x3,y3), and (x4,y4). The vertices are assumed to be in clockwise or counter-clockwise order [i.e. (X2,Y2) is clockwise with reference to (X1,Y1), (X3,Y3) is clockwise with reference to (X2,Y2),etc.].

*5. POLYGON   NSIDE=n (X(i)=xi Y(i)=yi) i=1,n

An n-sided convex polygon defined by n vertices (xi,yi). The points representing the vertices must be entered in either clockwise or counter-clockwise order.

6. CIRCLE   XC = xc [0.0]
YC = yc [0.0]
RADIUS= r [+1.0e+06]

A circle of radius r centered at (xc,yc).

7. ELLIPSE   XC=xc [0.0]
YC=yc [0.0]
B = b [1.0]
D = d [1.0]

An ellipse centered at (xc,yc) with semiaxis length d in the x direction and b in the y direction.

8. PARABOLA   X1=x1 [0.0]
Y1=y1 [0.0]
A = a [0.0]
B = b [1.0]

All (x,y); y-y1 <= a + b * (x-x1)**2

9. HYPERBOLA   X1=x1 [0.0]
Y1=y1 [0.0]
A = a [0.0]
B = b [1.0]

All (x,y); y-y1 <= a + b * (x-x1)**2

10. GNRLFIT   A5=a5 [0.0]
A4=a4 [0.0]
A3=a3 [0.0]
A2=a2 [0.0]
A1=a1 [0.0]
A0=a0 [0.0]

All (x,y); y >= a5*(x**5) + a4*(x**4) + a3*(x**3)+ a2*(x**2) + a1*x+ a0

11. CURVE TABLE NPT=n   (X(i)=xi Y(i)=yi) i=1,n

All (x,y); x1 <= x <= xn and
y >= yi + (yi+1-yi) * (xi+1 - x) / (xi+1-xi)

The word TABLE is optional.

Table A.2. Three-Dimensional Geometry Options

The following is a list of the various geometry options available for 3 dimensional problems, together with the input parameters required for their specification. Unless otherwise specified, the parameters can be input in any order. Default values are indicated in brackets. Newly added options are prefaced with an "*".

1. BOX	X1=x1 [-1.0e+06] Y1=y1 [-1.0e+06] Z1=z1 [-1.0e+06] X2=x2 [+1.0e+06] Y2=y2 [+1.0e+06] Z2=z2 [+1.0e+06]

A rectangular parallepiped with vertices at (x1,y1,z1),(x1,y1,z2), (x1,y2,z1),(x1,y2,z2),(x2,y1,z1),(x2,y1,z2),(x2,y2,z1),and (x2,y2,z2)

2. PYRAMID	X1=x1 [-1.0e+06] Y1=y1 [-1.0e+06] Z1=z1 [-1.0e+06] X2=x2 [+1.0e+06] Y2=y2 [+1.0e+06] Z2=z2 [-1.0e+06] X3=x3 [-1.0e+06] Y3=y3 [+1.0e+06] Z3=z3 [-1.0e+06] X4=x4 [-1.0e+06] Y4=y4 [+1.0e+06] Z4=z4 [+1.0e+06]

A tetrahedron with vertices at (x1,y1,z1), (x2,y2,z2), (x3,y3,z3), and (x4,y4,z4)

*3. PARALELEPIPED	X1=x1 [-1.0e+06] Y1=y1 [-1.0e+06] Z1=z1 [-1.0e+06] X2=x2 [+1.0e+06] Y2=y2 [+1.0e+06] Z2=z2 [-1.0e+06] X3=x3 [-1.0e+06] Y3=y3 [+1.0e+06] Z3=z3 [-1.0e+06] X4=x4 [-1.0e+06] Y4=y4 [+1.0e+06] Z4=z4 [+1.0e+06]

A parallelepiped defined by the line segments (x1,y1,z1)-(x2,y2,z2), (x1,y1,z1)-(x3,y3,z3), and (x1,y1,z1)-(x4,y4,z4)

*4. POLYHEDRON

X1=x1 [-1.0e+06]
Y1=y1 [-1.0e+06]
Z1=z1 [-1.0e+06]
X2=x2 [+1.0e+06]
Y2=y2 [-1.0e+06]
Z2=z2 [-1.0e+06]
X3=x3 [+1.0e+06]
Y3=y3 [+1.0e+06]
Z3=z3 [-1.0e+06]
X4=x4 [-1.0e+06]
Y4=y4 [+1.0e+06]
Z4=z4 [-1.0e+06]
X5=x5 [-1.0e+06]
Y5=y5 [-1.0e+06]
Z5=z5 [+1.0e+06]
X6=x6 [+1.0e+06]
Y6=y6 [-1.0e+06]
Z6=z6 [+1.0e+06]
X7=x7 [+1.0e+06]
Y7=y7 [+1.0e+06]
Z7=z7 [+1.0e+06]
X8=x8 [-1.0e+06]
Y8=y8 [+1.0e+06]
Z8=z8 [+1.0e+06]

An arbitrary eight-sided figure whose faces are quadrilaterals. Defined by the eight vertices (xi,yi,zi) i =1,8.

*5. CONE

X1=x1 [0.0]
Y1=y1 [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

The interior of an elliptical cone whose axis is parallel to the z axis with vertex at (x1,y1,0), which lies between the planes z = z1 and z = z2. i.e.

All (x,y,z); z**2 <= [(x-x1)/b]**2 + [(y-y1)/d]**2 for z1 <= z <= z2

Setting b = d produces a cone with a circular cross-section.

6. SPHERE

XC =xc [0.0]
YC =yc [0.0]
ZC =zc [0.0]
RADIUS= r [+1.0e+06]

A sphere of radius r centered at (xc,yc,zc).

*7. ELLIPSOID

XC=xc [0.0]
YC=yc [0.0]
ZC=zc [0.0]
B = b [1.0]
D = d [1.0]
F = f [1.0]

An ellipsoid centered at (xc,yc,zc) with semiaxis length d in the x direction, b in the y direction, and f in the z direction. Note that the defaults produce a sphere of radius 1.

8. WEDGE

X1=x1 [-1.0e+06]
Y1=y1 [-1.0e+06]
X2=x2 [+1.0e+06]
Y2=y2 [+1.0e+06]
X3=x3 [-1.0e+06]
Y3=y3 [+1.0e+06]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

A triangular prism defined by the coordinates (x1,y1,0), (x2,y2,0), and (x3,y3,0) extending parallel to the z axis between the planes z = z1 and z = z2.

9. CYLINDER

XC= xc [0.0]
YC= yc [0.0]
RADIUS= r [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

A right circular cylinder, centered at (xc,yc) with radius r whose axis is parallel to the z axis and which extends between the planes z = z1 and z = z2.

10. ELLIPCYL

XC=xc [0.0]
YC=yc [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

An elliptical cylinder, centered at (xc,yc) with semiaxis length d in the x direction and b in the y direction whose axis is parallel to the z axis and which extends between the planes z = z1 and z = z2.

11. PARACYL

XC=xc [0.0]
YC=yc [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

A parabolic cylinder, centered at (xc,yc) with semiaxis length d in the x direction and b in the y direction whose axis is parallel to the z axis and which extends between the planes z = z1 and z = z2.

12. HYPERCYL

XC=xc [0.0]
YC=yc [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

A hyperbolic cylinder, centered at (xc,yc) with semiaxis length d in the x direction and b in the y direction whose axis is parallel to the z axis and which extends between the planes z = z1 and z = z2.

*13. PARABOLOID

X1=x1 [0.0]
Y1=y1 [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [0.0]
Z2=z2 [+1.0e+06]

The points interior to a paraboloid whose axis is parallel to the z axis which lie between the planes z = z1 and z = z2, i.e.

All (x,y,z); z <= [(x-x1)/b]**2 + [(y-y1)/d]**2 for z1 <= z <= z2

*14. HYPERBOLOID

X1=x1 [0.0]
Y1=y1 [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

The points interior to a hyperboloid of one sheet whose axis is parallel to the z axis which lie between the planes z = z1 and z = z2, i.e.

All (x,y,z); z**2 <= [(x-x1)/b]**2 + [(y-y1)/d]**2 -f**2
for z1 <= z <= z2

*15. HYPERLD2

X1=x1 [0.0]
Y1=y1 [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

The points interior to a hyperboloid of two sheets whose axis is parallel to the z axis which lie between the planes z = z1 and z = z2, i.e.

All (x,y,z); z**2 <= [(x-x1)/b]**2 + [(y-y1)/d]**2 +f**2
for z1 <= z <= z2

*16. HYPERPARA

X1=x1 [0.0]
Y1=y1 [0.0]
B = b [1.0]
D = d [1.0]
Z1=z1 [-1.0e+06]
Z2=z2 [+1.0e+06]

All points above a hyperbolic paraboloid whose axis is parallel to the z axis which lie between the planes z = z1 and z = z2, i.e.

All (x,y,z); z >= -[(x-x1)/b]**2 + [(y-y1)/d]**2 for z1 <= z <= z2

The following figures are described with reference to the X-Y plane and rotated about the z axis. For example, a CIRCLEROT specification will produce a torus.

17. RECTAROT

X1=x1 [-1.0e+06]
Y1=y1 [-1.0e+06]
X2=x2 [+1.0e+06]
Y2=y2 [+1.0e+06]

18. TRIAROT

X1=x1 [-1.0e+06]
Y1=y1 [-1.0e+06]
X2=x2 [+1.0e+06]
Y2=y2 [+1.0e+06]
X3=x3 [-1.0e+06]
Y3=y3 [+1.0e+06]

19. CIRCLEROT

XC =xc [0.0]
YC =yc [0.0]
RADIUS= r [+1.0e+06]

A circle of radius r centered at (xc,yc) rotated about the z axis to produce a torus.

20. ELLIPSROT

XC=xc [0.0]
YC=yc [0.0]
B = b [1.0]
D = d [1.0]

An ellipse centered at (xc,yc) with semiaxis of length d in the x direction and b in the y direction which is rotated about the z-axis.

21. PARALROT

X1=x1 [-1.0e+06]
Y1=y1 [-1.0e+06]
X2=x2 [+1.0e+06]
Y2=y2 [+1.0e+06]
X3=x3 [-1.0e+06]
Y3=y3 [+1.0e+06]

A parallelogram defined by the line segments (x1,y1)-(x2,y2), and (x1,y1)-(x3,y3) in the X-Y plane rotated about the z-axis

*22. QUADRAROT

X1=x1 [-1.0e+06]
Y1=y1 [-1.0e+06]
X2=x2 [+1.0e+06]
Y2=y2 [-1.0e+06]
X3=x3 [+1.0e+06]
Y3=y3 [+1.0e+06]
X4=x4 [-1.0e+06]
Y4=y4 [+1.0e+06]

A quadrilateral defined by vertices (x1,y1),(x2,y2),(x3,y3), and (x4,y4) in the X-Y plane and rotated about the z axis. The vertices are assumed to be in clockwise or counter-clockwise order [i.e. (X2,Y2) is clockwise with reference to (X1,Y1), (X3,Y3) is clockwise with reference to (X2,Y2),etc.] .

23. PARABLROT

X1=x1 [0.0]
Y1=y1 [0.0]
A = a [0.0]
B = b [1.0]

The interior of the region generated by rotating the specified parabola about the z axis.

*24. HYPERROT

X1=x1 [0.0]
Y1=y1 [0.0]
A = a [0.0]
B = b [1.0]

The interior of the region generated by rotating the specified hyperbola about the z axis.

25. GNRLROT A5=a5 [0.0]
A4=a4 [0.0]
A3=a3 [0.0]
A2=a2 [0.0]
A1=a1 [0.0]
A0=a0 [0.0]

Polynomial curve as in 2-D, described in terms of the X-Y plane, rotated about the z axis.

26. CURVEROT TABLE NPT=n (X(i)=xi Y(i)=yi) i=1,n

User specified curve as in 2-D, described in terms of the X-Y plane, rotated about the z axis.

The word TABLE is optional.

Table A.3. Keyword Synonyms

Various synonyms can be used to specify the above geometric figures. Six sets of synonyms are defined and each of the names are treated as equivalent.

Set 1: X0, X1, XL, XLEFT, XC, XCENT, XCNTR, and XCENTER

Set 2: Y0, Y1, YL, YB, YBOT, YBOTTOM, YC, YCENT, YCNTR, and YCENTER

Set 3: Z0, Z1, ZL, ZB, ZBOT, ZBOTTOM, ZC, ZCENT, ZCNTR, and ZCENTER

Set 4: X2, XR, XRIGHT

Set 4: Y2, YR, YF, YT, YTOP

Set 5: Z2, ZR, ZT, ZTOP

REFERENCES

1. Daniel A. Matuska and John J. Osborne, "HULL Documentation" Volume I & II, Orlando Technology, Inc., Shalimar, Florida.
2. R. T. Santoro, Program Manager, *Final Report of the Technical Progress on the Oak Ridge Fast Track Defensive Shield Demonstration Program – April 1986—September 1987 (U)*, Martin Marietta Energy Systems, Inc., Oak Ridge National Laboratory, March 1989 (SECRET RD).

INTERNAL DISTRIBUTION

1. R. G. Alsmiller, Jr.
2. B. R. Appleton
3. D. E. Bartine
4. J. M. Barnes
5–9. T. J. Burns
10. R. M. Davis
11. J. B. Dooley
12. J. D. Drischler
13. T. A. Gabriel
14. C. M. Haaland
15. E. B. Harris
16. W. R. Hendrich
17. D. T. Ingersoll
18. J. O. Johnson
19. R. A. Lillie
20. F. C. Maienschein
21. J. R. Merriman
22. W. L. Roberts
23. R. W. Roussin
24. R. T. Santoro
25. M. S. Smith
26. D. G. Thomas
27. R. C. Ward
28. C. R. Weisbin
29. EPMD Reports Office
30–31. Laboratory Records Department
32. Laboratory Records, ORNL–RC
33. Document Reference Section
34. Central Research Library
35. ORNL Patent Section

EXTERNAL DISTRIBUTION

36. Dr. Charles Aeby, AFWL/NTC, Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico 87117-6008
37. Lt Dale Atkinson, AFWL/NTCA, Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico 87117-6008
38. J. J. Dorning, Department of Nuclear Engineering and Engineering Physics, Thornton Hall, University of Virginia, Charlottesville, Virginia 22901
39. Lt Richard Engstrom, AFWL/NTCA, Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico 87117-6008
40. R. M. Haralick, Department of Electrical Engineering, University of Washington, Seattle, Washington 98195
41. Orlando Technology, Inc., 60 Second Street, Building 5, Shalimar, Florida 32579
42. Office of the Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations, P.O. Box 2001, Oak Ridge, TN 37831
43–52. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, Tennessee 37830