LA-12503-MS

UC-361
Issued: March 1993

KIVA-3: A KIVA Program with Block-Structured Mesh for Complex Geometries

Anthony A. Amsden

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

CONTENTS

KIVA-3: A KIVA PROGRAM WITH BLOCK-STRUCTURED MESH FOR COMPLEX GEOMETRIES

by

Anthony A. Amsden

ABSTRACT

This report describes the KIVA-3 computer program for numerical calculation of transient, two- and three-dimensional chemically reactive fluid flows with sprays. KIVA-3 is an extension of the earlier KIVA-II, uses the same numerical solution procedure, and solves the same set of equations. The full generality of KIVA-II has been retained; thus KIVA-3 is applicable to laminar or turbulent flows, subsonic or supersonic flows, and single-phase or dispersed two-phase flows. KIVA-3 differs from KIVA-II in that it uses a block-structured mesh with connectivity defined through indirect addressing. The departure from a single rectangular structure in (i,j,k) logical space allows complex geometries to be modeled with significantly greater efficiency than was previously possible because large regions of deactivated cells are no longer necessary. Cell-face boundary conditions permit greater flexibility and simplification in the application of boundary conditions. This report discusses those features of KIVA-3 that differ from KIVA-II, the input required from a mesh generation preprocessor, and the output provided to a graphics postprocessor. Basic pre- and post-processors are included in the KIVA-3 package, and are also described.

I. INTRODUCTION AND BACKGROUND

Over the past several years, computational fluid dynamics (CFD) has been increasingly accepted as an adjunct to experimentation in the design and understanding of practical combustion systems. A major goal in the engine modeling effort, the calculation of three-dimensional engine flows with sprays and combustion, was realized in 1987.[1,2] Since that time, three-dimensional studies with sprays and/or combustion have become commonplace in the literature.[3-9] A recent patent for a high-turbulence piston design specifically identifies three-dimensional computer simulation for making the invention possible.[10]

The original KIVA program was publicly released in 1985,[11,12] and was replaced by the improved KIVA-II in 1989.[13,14] These programs are in use worldwide for the numerical calculation of transient two- and three-dimensional chemically reactive fluid flows with sprays. A license has been issued to Cray Research, Inc., to distribute their commercial version of KIVA-II, known as CRI/TurboKiva, for use on Cray platforms.[15]

These earlier versions of KIVA lend themselves well to confined in-cylinder flows and a variety of open combustion systems, but can become quite inefficient to use in complex geometries that contain such features as inlet ports and valves, diesel prechambers, and entire transfer ports. In all these code versions, the computing grid is a tensor-product mesh, in which each cell is identified by an ordered triplet of indices (i,j,k), and each of these indices varies over a fixed range (i = 1, ... , NXP; j = 1, ..., NYP; k = 1, ..., NZP), independent of the values of the other two indices. In modeling the above complex geometries, a significant number of cells must generally be deactivated to create a grid that will encompass the entire region of interest. KIVA has been successfully applied to such geometries in spite of the cost in storage and computer time, as evidenced by a number of studies[16-20] involving complex flow domains.

KIVA-3 has been designed to overcome this inefficiency by employing a block-structured mesh that eliminates the necessity of maintaining large regions of deactivated cells and results in a reduction of both storage and computer time for complex geometries. The method uses indirect addressing, taking advantage of the fact that this construct vectorizes on present Cray systems and allows completely arbitrary connectivity of cells. Each cell is identified by a single index, rather than three, and neighbor identification is maintained in arrays that specify the indices of the six neighboring cells that share common faces with the cell in question. The overall mesh is then surrounded by fictitious cells to simplify the application of inflow/outflow boundary conditions.

The data structure in KIVA-3 allows us to minimize the length of vector loops and eliminate testing on cell and vertex flag arrays, F and FV. This is achieved by a simple sort (Fig. 1) that places all the deactivated or ghost cells (F = 0.0, FV = 0.0) at the beginning of the vector. These are followed by all the active cells (F = 1.0, FV > 0.0), and finally the set of all the active vertices at mesh boundaries (F = 0.0, FV > 0.0). A DO loop over all active cells therefore spans only the second portion of the vector, and a DO loop over all active vertices sweeps the second through third portions of the vector.

Tailored boundary condition data is carried in tables that allow the code to sweep in shorter vectors over only those vertices or cells involved, without having to sweep the entire mesh to locate them every time a boundary condition subroutine is called. At solid wall vertices, the tabular data includes information that zeros the normal velocity component. The data sorting and boundary condition table generation are performed initially in the setup, and thereafter only when planes are activated or deactivated, as for piston motion.

Piston motion in KIVA-3 is Lagrangian, as in earlier versions of KIVA, and uses a novel approach called snapper. In addition to serving as a chopper, the snapper also allows a simple scheme for moving the piston past ports in the cylinder wall.

The KIVA-3 package has a distinct three-part structure in which the grid generator and graphics are separate programs from the hydro program because, in general, users will supply their own pre- and post-processors. The grid generator creates a file TAPE17 to be read by KIVA-3 that contains all the grid coordinates X, Y, Z, flags F and FV, six neighbor connectivity arrays, three cell-face boundary condition arrays, and a region identifier array. KIVA-3 completes the setup by sorting the storage, generating all other arrays required to describe the problem, and creating the tailored boundary condition tables. KIVA-3 writes an output file TAPE9 to be used by a graphics post-processor. The KIVA-3 package includes both a pre-processor (K3PREP) and a post-processor (K3POST). Although these are fairly basic, they are adequate for many applications and serve as models for replacement packages supplied by the user.

KIVA-3 uses the same solution algorithms and solves the same set of equations as KIVA-II. This report should be considered a companion report to the comprehensive KIVA-II report,[14] as it describes only the differences from KIVA-II. It is assumed that the user is familiar with KIVA-II. KIVA-3 is not only

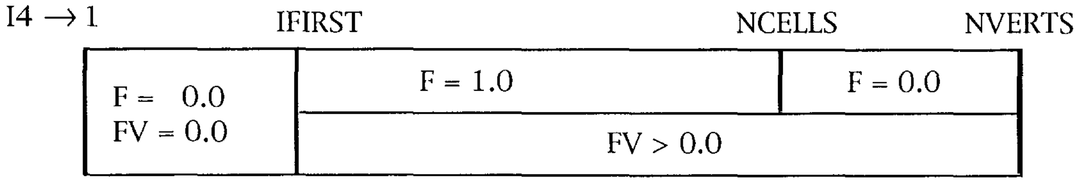


Fig. 1. KIVA-3 storage is sorted by F and FV to maintain the shortest possible vector lengths.

faster than KIVA-II, but can be considerably simpler to use and more efficient when the geometry is complex. Pre-release versions of KIVA-3 have already been applied to studies reported in the literature.[21,22]

Distribution of the KIVA-3 package (source codes for KIVA-3, K3PREP, and K3POST, along with data files for four sample calculations) is through the Energy Science and Technology Software Center (ESTSC), which became the centralized software management center for the U. S. Department of Energy on October 1, 1991, replacing the National Energy Software Center. For further information and prices, call the ESTSC at (615) 576-2606, and request publication ESTSC-1 *Guide for Submitting and Ordering Software.* The mailing address is P. O. Box 1020, Oak Ridge, TN 37831-1020. Distribution of all KIVA software by Los Alamos is limited to our direct collaborators.

II. THE KIVA-3 PROGRAM

A. General Structure

The KIVA-3 computer program consists of a set of subroutines driven by a short main program, and is essentially identical to the KIVA-II structure. As before, there are a number of supporting subroutines that perform tasks for the primary subroutines. In KIVA-3, the chopper subroutine has been replaced by what we call snapper. Besides serving as a chopper that deactivates and reactivates cylinder planes in response to piston motion, the snapper incorporates logic that allows an automatic means for moving the piston past ports in the cylinder wall, when ports are present. A reversed snapper is also included for geometries having two opposed pistons in the same cylinder. Snapper is discussed in detail in Sec. II.H below.

Two features of KIVA-3 contribute to a speed improvement over KIVA-II. First, the departure from (i,j,k) logical space and the use of indirect addressing, which allows cells to be stored in random fashion, also allows storage to be sorted as described above. This results in the shortest possible vector lengths, whether in loops over all active cells or in loops over all active vertices. The shorter vector lengths counteract the penalty for using indirect addressing. Second, a new subroutine, SETUPBC, identifies all vertices or cell faces that lie on outside boundaries and thus require special treatment. It creates customized tables for use in vector loops by the various boundary condition setting subroutines. An enhancement that contributes to overall efficiency is the introduction of the cell-face boundary condition arrays. The goal of writing code in which the hydro routines never have to search when setting boundary values has been met. (Indeed, the SETUPBC and cell-face boundary condition concept would also benefit KIVA-II.) Although SETUPBC is the longest subroutine in KIVA-3, it is called only by SETUP and after snapper, so the time investment is negligible.

KIVA-3 is written specifically for use on the CRI Cray family of computers, in particular the more recent models with compilers that vectorize the many DO-loops involving indirect addressing, operating under either the Cray Time Sharing System (CTSS) or UNICOS. The export versions of KIVA-3, K3PREP, and K3POST are tailored for UNICOS, although they scarcely differ from their CTSS counterparts. The recommended compiler is CFT77, as it will produce the fastest object code. CFT77 requires several minutes to compile KIVA-3 on one processor of a Cray Y-MP. The user should create and maintain a library of relocatable binaries with the bld (build) utility to minimize time spent in the compiler.

B. The Computing Mesh

As in KIVA-II, the KIVA-3 formulation is based on (x,y,z) Cartesian coordinates. Rather than being confined to one logical block of cells in (i,j,k) space to encompass the entire region to be modeled, however, the geometry of a KIVA-3 mesh is composed of any arbitrary number of logical blocks that are patched together in a completely seamless fashion. This is made possible by the cell face boundary condition named FLUID between two fluid cells. Each block has six bounding faces, and although each block initially has its own (i,j,k) structure, any idea of a global (i,j,k) structure disappears when the blocks are patched together and storage is sorted. Figure 9 of the KIVA-II report [14] illustrated five mesh types, any one of which would be a possible block type (Fig. 2) in the K3PREP grid generator.

Patching allows complex geometries to be created block by block, while minimizing the number of deactivated zones. In fact, the only deactivated zones are the ghosts that surround the final mesh, as there are no ghosts between blocks where common faces are joined. The ghosts enable the application of boundary conditions at inflow/outflow cell faces, as they provide logically situated outside storage for fluxed quantities.

2-D and 3-D sector meshes use periodic front and derriere boundaries as in KIVA-II, and are identified in ITAPE by SECTOR = 1.0 and a value of THSECT that will divide evenly (within 10^{-3}) into 360^{o} (e.g., 72.0, 51.42857, 22.5) to satisfy the periodic condition. As before, a 2-D sector mesh has a single azimuthal plane of fluid cells of some finite volume, and again one specifies THSECT = 0.5. The 3-D full-circle mesh (SECTOR = 0.0, THSECT = 360.0) no longer requires periodic boundaries because of indirect addressing; all azimuthal cell faces have the boundary condition FLUID. Chapter III will illustrate generation of grids from these block types.

Several restrictions of KIVA-II carry over to KIVA-3. Cell faces must always coincide between neighboring cells; there are no partial cell faces or sliding interfaces. The mesh is structured to the extent that the computing cells are always logical hexahedrons with eight zones meeting at a (interior) vertex. This implies that the program does not allow, for example, a cylinder built from a rectangular central block surrounded by four curved blocks extending outward to define the circumference of the cylinder, because only six cells would meet at the corners of the central block (Fig. 3a). Although the concept of (i,j,k) space has vanished, the rule of left-right, front-derriere, bottom-top directions still applies. For example, one cannot construct a simple curved C-shaped block attached at two points to the right face of another block (Fig. 3b). The directional rule would be violated, as left would conflict with right at the second connection.

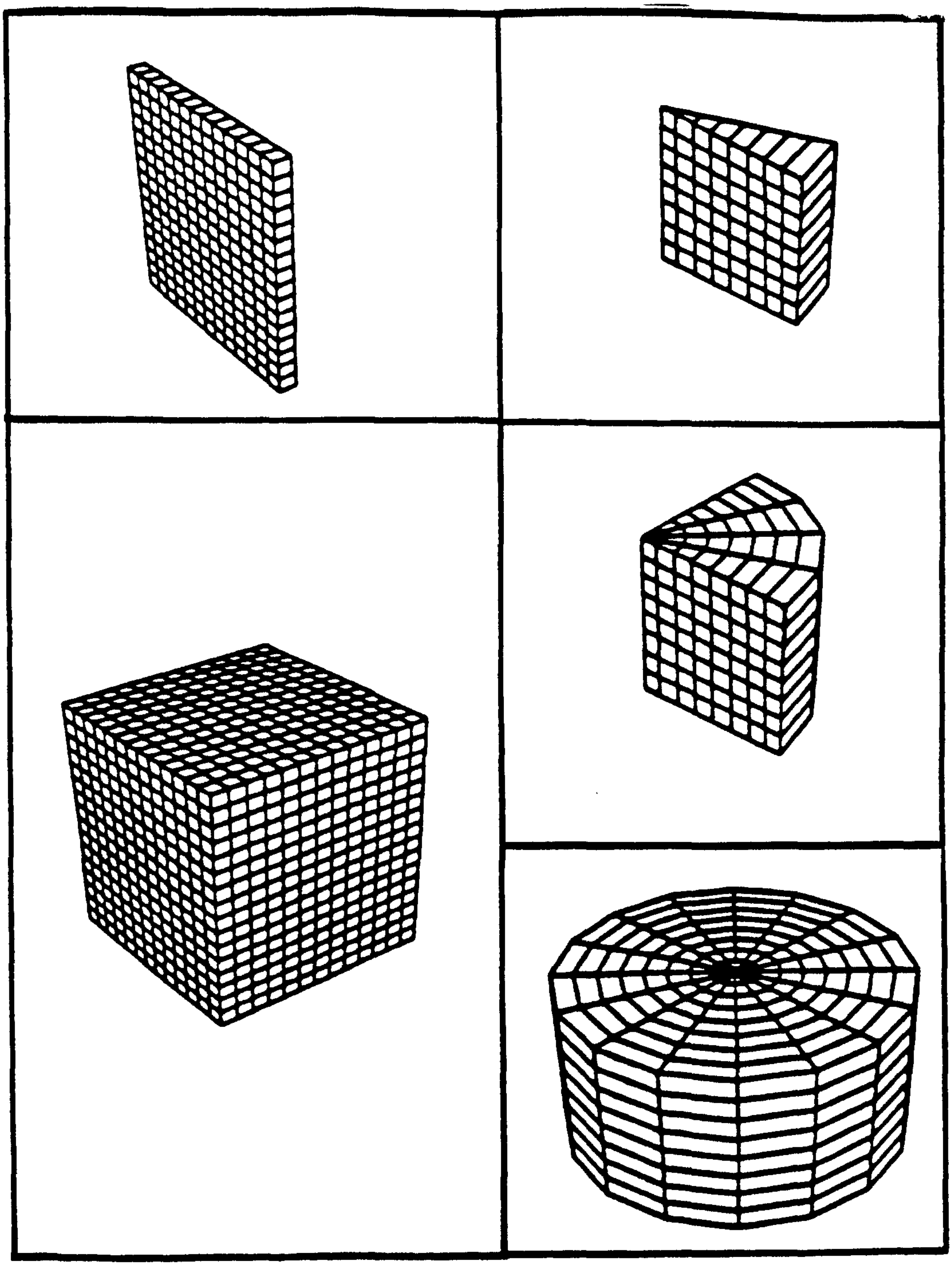
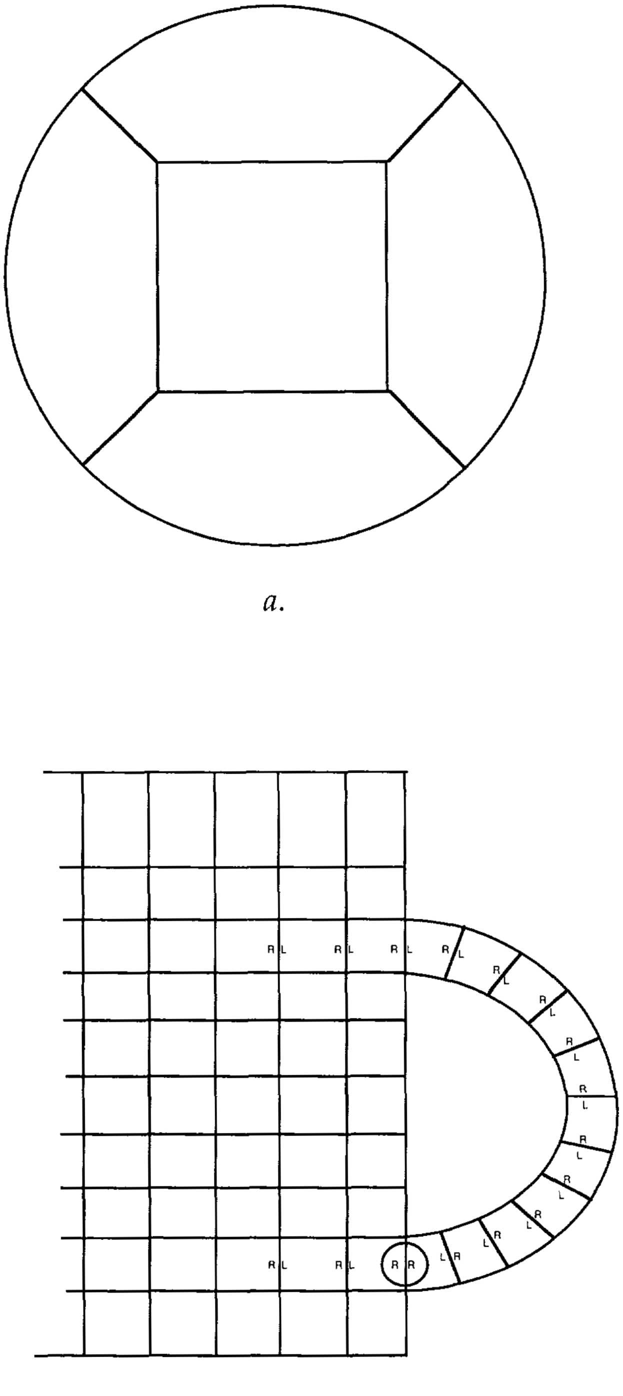


Fig. 2. K3PREP can build meshes or portions of meshes using these five generic block shapes.



a.

b.

Fig. 3. Two grids that are not allowed in KIVA-3 because the left-right, front-derriere, bottom-top rule is violated.

KIVA-II contained the option of a 2-D to 3-D converter, which has not been carried over to KIVA-3. The idea was to allow one to run a 2-D cylindrical mesh until the fuel spray event, while the flow was truly axisymmetric, and then map the solution to a 3-D sector or full-circle mesh. This allowed a significant reduction in computer time. KIVA-3 contains no mesh generation capability in the hydro code itself, and indeed expects the entire mesh with all its connectivity to have been established by the grid generator. Incorporation of a 2-D to 3-D converter would be awkward in this context and has not been addressed.

1. Blocks and Regions. The concept of "blocks" in the grid generator is distinct from the concept of fluid "regions" in the hydro code. This difference is *important and is illustrated by the following two cases.*

First, consider a simple cylinder with a piston bowl, as in our traditional baseline. In K3PREP, we can create this geometry by patching together several blocks, but KIVA-3 sees the resulting geometry as a single fluid region. It is not concerned with how many blocks were required to generate the geometry. Through the entire range of piston motion, it remains a single connected region in which every fluid cell always has a continuous fluid communication path with every other fluid cell.

Second, consider a simple cylinder with intake and exhaust ports around the cylinder wall, typical of many two-stroke engines. The grid generator might use one block to define the cylinder itself, and one or more blocks to define each port. Several dozen blocks might be patched together to create the entire geometry. The hydro code sees the geometry in terms of regions. For example, if several blocks had been patched together to create a complex intake port, the resulting port would always remain one continuous fluid region. When the piston is at bottom dead center, the flow in the cylinder can communicate with that in all the ports, and the entire computing domain behaves as a single fluid region, analogous to the first case above. When the piston is at top dead center, however, not only is the cylinder disconnected from all the ports, but the ports are all disconnected from one another, as they have no means to inter-communicate. At this point, we may have quite a number of fluid regions: the cylinder is one region and each complete port is yet another.

The importance of disconnected fluid regions becomes apparent if one considers the various iterative loops, such as for pressure, temperature, turbulence, etc. In each instance, convergence is scaled to the difference between the minimum and maximum values over the field. When there is one continuous region, as at bottom dead center, every cell has a communication path with all other cells. This is not true at top dead center, however, when the difference between the global maximum and minimum values may be literally orders of magnitude apart, although the difference between maximum and minimum values within any individual region may be slight. If the global difference were used, the solution would fail to converge. The implication is that the maximum and minimum values must be monitored separately in *each fluid region*. This allows the code to treat each region as a single separate computing domain when ports are disconnected from the cylinder. For this very purpose we have an integer array IDREG, supplied by the grid generator, to identify the native fluid region for every cell.

The discussion above also illustrates the usefulness of the cell face boundary condition arrays. When ports disconnect from the cylinder, the boundary condition at the common cell faces is switched automatically from FLUID to SOLID. The setting is reversed when ports reconnect.

2. Cylinder with Opposed Pistons. A number of internal combustion engine designs incorporate the use of two opposed pistons in the same cylinder. A further complication is that the geometry may not be symmetric across a plane midway between the opposing crankshafts, precluding any possibility of zoning only one half the full geometry. I have made such an application with KIVA-3, and to spare users from having to repeat my effort, have incorporated a two-piston geometry as an option in the code, and included an example below.

The lower piston is always the "master" piston. The starting crank angle ATDC, the current crank angle CRANK, piston velocity WPISTN, and piston position ZPISTN refer to the lower piston, just as they do in the usual single-piston geometries. The two pistons are not required to have the same crank angle history. The input variable DATDCT is used to specify the crank angle difference for the upper piston, and may be positive, negative, or zero. For example, if the upper piston reaches its TDC 15° crank angle before the lower piston reaches its TDC, then DATDCT = -15.0. Both pistons are moved in accordance with the standard piston logic in subroutine PISTON. Should this algorithm not accurately represent the piston motion, the user will have to patch the subroutine to provide it, typically by a tabular interpolation. The upper piston has a velocity WPISTNT and position ZPISTNT.

The existence of a two-piston geometry does not have to be explicitly stated by the user. KIVA-3 subroutine SETUP determines this case from the values of STROKE and SQUISH supplied in ITAPE, coupled with the fact that there are no head vertices (FV = FLHEAD). The quantity ZMID, calculated in subroutine SETUP, is the z at the middle of the cylinder in two-piston geometries, or a z just above the head in single-piston geometries. The value of ZMID is checked to ensure that no piston will pass it at its TDC, otherwise the code would be in error if it proceeded. Should the check fail, the problem will abort with a message that the values of STROKE and SQUISH should be reconsidered. Note that the value of SQUISH should be 1/2 the true minimum clearance between the two pistons.

C. Indexing Notation

Figure 4 illustrates the index nomenclature used in KIVA-3. As in KIVA-II, the index I4 in KIVA-3 refers either to cell I4 or its left-front-bottom corner vertex. The eight vertices of a cell are again referenced by the same I1 through I8 shorthand notation, and the neighbor cells on the logical right, derriere and top are identified as I1, I3, and I8 respectively. In KIVA-3, these three indices are maintained in arrays named I1TAB, I3TAB, and I8TAB. Three more index arrays are required to establish complete connectivity in all six directions, and these are the index arrays for neighbor cells on the logical left, front, and bottom. These are named IMTAB, JMTAB, and KMTAB respectively. [Despite the fact that the concept of (i,j,k) space becomes meaningless after the grid has been constructed, the KIVA-II user can visualize the directions implied by these names.] It is the responsibility of the grid generator to provide these six arrays. The arrays ITAB, JTAB, and KTAB that were present in KIVA-II no longer exist.

The DO index IFIRST in KIVA-3 is the first real (F=1.0) cell (vertex) in the vector, NCELLS is the last real cell, and NVERTS is the last real vertex. These may be thought of as corresponding to 1, IJKVEC, and IJKALL in KIVA-II, but because of sorting and indirect addressing, they may be physically located anywhere in the final mesh . For example, IFIRST might be out in the middle of a

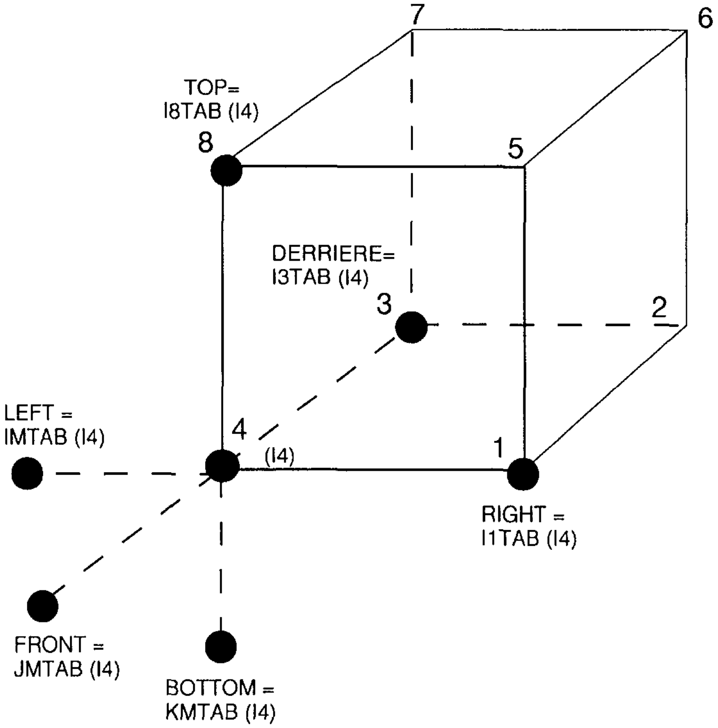


Fig. 4. Indexing notation in KIVA-3. Six neighbor indices establish complete connectivity.

large mesh, and NCELLS could conceivably be an immediately adjacent cell.

With reference to Fig. 1, it is evident that a DO loop over all active cells would read DO _ I4 = IFIRST, NCELLS, and a DO loop over all active vertices would read DO _ I4 = IFIRST, NVERTS. Indices for the eight corner vertices of cell I4 are obtained from the connectivity arrays as follows:

```
I1 = I1TAB(I4)
I2 = I3TAB(I1)
I3 = I3TAB(I4)
I5 = I8TAB(I1)
I6 = I8TAB(I2)
I7 = I8TAB(I3)
I8 = I8TAB(I4).
```

The rule that suggested a senior-to-junior array element replacement sequence of 6-7-5-8-2-3-1-4 in KIVA-II is impossible to observe in a scheme that uses arbitrary connectivity. However, it appears that violation of the rule is not as great a concern in posing vector dependency problems as it perhaps was ten years ago. Even back then, the observed differences in KIVA-II results were always in the noise level whatever storage sequence was used when updating the eight vertices of a cell. In tests at that time, the 6-7-5-8-2-3-1-4 sequence produced identical results to those obtained by breaking the DO loop into eight separate loops, one loop for each vertex, and was therefore considered the safest choice.

D. Storage of Cell Data

As in KIVA-II, the use of memory space in KIVA-3 is minimized by r ans of array equivalencing. The idea is to retain quantities during a calculational cycle only as long as they are needed, and then to reassign the available storage to other quantities. Again, cell storage consists of both equivalenced and dedicated arrays.

Figure 5 shows the allocation of the equivalenced arrays. The column labels from left to right correspond to the sequence in which subroutines are called during a cycle. Reading down a particular column, the appearance of a variable name signifies reference to it in the associated subroutine or in a supporting subroutine. Lines of asterisks indicate that the quantity to the left must be retained for later use.

Twelve other arrays (CLI, CLJ, CLK, CFI, CFJ, CFK, CBI, CBJ, CBK, FSUM14, FSUM34, and FSUM84) are included in parentheses in Fig. 5, although they are actually dedicated arrays at present. If the user is more pressed for memory space than computer time, these arrays could instead be equivalenced to arrays E4 through E15 as indicated. Because the pressure iteration requires virtually every available array, it will overwrite E4 through E15. Thus the user

KIVA-3 Storage Chart

SUB-	SETUP	VISC	TIMSTP	NEWCYC	INJECT	PMOVTV	BREAK	COLIDE	EVAP	LAWALL
ROUTINE	SORT			FULOUT	FRAN	FRAN	FRAN	FRAN		BC
\	APROJ			TAPEWR	PFIND	PFIND		REPACK		
\	SETUPBC					REPACK				
\	VOLUME									
\	STATE, BC									
ARRAY \	PISTON									
\										
E1	MV, RMV	•••••••••••••	•••••••••••••	RMV	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	RMV
E2	GAMMA	•••••••••••••	•••••••••••••	GAMMA	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	GAMMA	GAMMA
E3	EPS	EPS	•••••••••••••	EPS	•••••••••••••	EPS	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••
E4										
E5										
E6										
E7										
E8										
E9										
E10										
E11										
E12										
E13										
E14										
E15										
E16										
E17										
E18										
E19									VAPM	
E20									ENTHO	
E21										
E22	XL									
E23	YL									
E24	ZL									
E25	FXL	|•••••••••••••	FXL							
E26	FXF	|•••••••••••••	FXF							
E27	FXB	|•••••••••••••	FXB							
E28										
E29										
E30										
E31										
E32										
E33										
E34										
E35										
E36										
E37										
E38										
E39										
E40										
E41										
E42										
E43										
E44										
E45									TOTCM	
E46									DMTOT	•••••••••••••
E47									TOTH	
E48									DSIEP	•••••••••••••
E49									CPC	
E50				RON	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••
E51				SIEN	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••
E52										
E53										
E54				TKEN	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••
E55				EPSN	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••
E56				UN	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	UN
E57				VN	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	VN
E58				WN	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	WN

Fig. 5. The allocation of equivalenced arrays in KIVA-3 storage. Dotted lines indicate that the quantity to the left must be retained for later use. The 12 array names enclosed in parentheses are discussed in the text.

KIVA-3 Storage Chart

SUB-	NODCPL	CHEM	CHEMEQ	PMOM	PCOUPL	YSOLVE	(OPTIONAL)	EXDIF	PINIT	PGRAD (1.0)
ROUTINE	BCNODCPL		(OR	REPACK	BC	YIT	WRITE	BC		BCPGRAD
\	BC		CHMQGM)			RESY	E4	BCROT1		BC
\						BCDIFF	THROUGH	BCDIFF		
\						BCROT1	E15			
\						BCROXCEN	TO			
ARRAY \							SSD			
\										
E1	RMV, MV	•••••••••••••	•••••••••••••	•••••••••••••	MV, RMV	•••••••••••••	•••••••••••••	RMV	•••••••••••••	RMV
E2	•••••••••••••	•••••••••••••	GAMMA	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	GAMMA	GAMMA	•••••••••••••
E3	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	EPS	EPS	•••••••••••••	EPS	•••••••••••••	•••••••••••••
E4	AUGMV					(CLI)	(CLI)	(CLI)	•••••••••••••	•••••••••••••
E5						(CLJ)	(CLJ)	(CLJ)	•••••••••••••	•••••••••••••
E6						(CLK)	(CLK)	(CLK)	•••••••••••••	•••••••••••••
E7						(CFI)	(CFI)	(CFI)	•••••••••••••	•••••••••••••
E8						(CFJ)	(CFJ)	(CFJ)	•••••••••••••	•••••••••••••
E9						(CFK)	(CFK)	(CFK)	•••••••••••••	•••••••••••••
E10						(CBI)	(CBI)	(CBI)	•••••••••••••	•••••••••••••
E11						(CBJ)	(CBJ)	(CBJ)	•••••••••••••	•••••••••••••
E12						(CBK)	(CBK)	(CBK)	•••••••••••••	•••••••••••••
E13						(RFSUM14)	(RFSUM14)	(RFSUM14)	•••••••••••••	•••••••••••••
E14						(RFSUM34)	(RFSUM34)	(RFSUM34)	•••••••••••••	•••••••••••••
E15						(RFSUM84)	(RFSUM84)	(RFSUM84)	•••••••••••••	•••••••••••••
E16	DUDX	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DUDX		
E17	DUDY	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DUDY		
E18	DUDZ	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DUDZ		
E19	DVDX	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DVDX		
E20	DVDY	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DVDY		
E21	DVDZ	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DVDZ		
E22	DWDX	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DWDX		
E23	DWDY	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DWDY		
E24	DWDZ	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	DWDZ		
E25						PHID	•••••••••••••	PHID	•••••••••••••	•••••••••••••
E26						ENTHDF		DISPTIL	•••••••••••••	•••••••••••••
E27						SPD14,DYP14		TEM14		
E28						SPD34,DYP34		TEM34		
E29				RU	RU	SPD84,DYP84		TEM84		
E30				RV	RV	HISP		UTIL	•••••••••••••	•••••••••••••
E31				RW	RW	SPMTIL		VTIL	•••••••••••••	•••••••••••••
E32						ENTHTIL		WTIL	•••••••••••••	•••••••••••••
E33						DDY		RMVSU	•••••••••••••	RMVSU
E34						RES				
E35						RESOLD				
E36						DRES				
E37						RDRDY				
E38						DELTAY				
E39						DELYPH		TKE14		
E40						XCEN		TKE34	PN	PN
E41						YCEN		TKE84	PHIP	PHIP
E42	UB					ZCEN		EPS14		
E43	VB					YSPM		EPS34		
E44	WB					YSPD		EPS84		
E45								TT1MPH		
E46	•••••••••••••	•••••••••••••	•••••••••••••	DMTOT	DMTOT			TKE1MPH		
E47								EPS1MPH		
E48	•••••••••••••	•••••••••••••	•••••••••••••	DSIEP	DSIEP					
E49				DTKEP	DTKEP			CPC	•••••••••••••	•••••••••••••
E50	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	RON, HTCTIL	•••••••••••••	•••••••••••••
E51	•••••••••••••	•••••••••••••	SIEN	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	SIEN, TTIL	•••••••••••••	•••••••••••••
E52								TKETIL	•••••••••••••	•••••••••••••
E53								EPSTIL	•••••••••••••	•••••••••••••
E54	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••
E55	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••
E56	UN	•••••••••••••	•••••••••••••	UN	•••••••••••••	•••••••••••••	•••••••••••••	UN	•••••••••••••	•••••••••••••
E57	VN	•••••••••••••	•••••••••••••	VN	•••••••••••••	•••••••••••••	•••••••••••••	VN	•••••••••••••	•••••••••••••
E58	WN	•••••••••••••	•••••••••••••	WN	•••••••••••••	•••••••••••••	•••••••••••••	WN	•••••••••••••	•••••••••••••

Fig. 5. continued.

KIVA-3 Storage Chart

SUB-	VSOLVE	TSOLVE	PSOLVE	(OPTIONAL)	PGRAD (3.0)	PGRAD (1.0)	PHASEB	KESOLV	PACCEL	REZONE
ROUTINE	RESUVW	TINVRT	PGRAD (2.0)	READ	BCPGRAD	BCPGRAD		DRDKE, RESK		
\	BC	DRDT	BC, BCFC	E4	BC	BC		RESE, BCEPS		
\		REST	UFINIT,PEXDIF	THROUGH				BCRESEZ		
\		BCDIFF	DRDP, RESP	E15				BCDIFF		
\		BCROT1	BCRESP	to				BCROT1		
ARRAY \			BCPEXD	SSD						
\			BCPGRAD							
			BCROT1							
E1	RMV	RMV	RMV	••••••••••••••	RMV	RMV	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••
E2	••••••••••••••	GAMMA,RGAMI	RGAMMA	••••••••••••••	••••••••••••••	••••••••••••••	RGAMMA	••••••••••••••	••••••••••••••	••••••••••••••
E3	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	EPS	••••••••••••••	••••••••••••••
E4	••••••••••••••	(CLI)	PL	(CLI)	••••••••••••••	••••••••••••••	••••••••••••••	(CLI)		
E5	••••••••••••••	(CLJ)	PR	(CLJ)	••••••••••••••	••••••••••••••	••••••••••••••	(CLJ)		
E6	••••••••••••••	(CLK)	PF	(CLK)	••••••••••••••	••••••••••••••	••••••••••••••	(CLK)		
E7	••••••••••••••	(CFI)	PD	(CFI)	••••••••••••••	••••••••••••••	••••••••••••••	(CFI)		
E8	••••••••••••••	(CFJ)	PB	(CFJ)	••••••••••••••	••••••••••••••	••••••••••••••	(CFJ)		
E9	••••••••••••••	(CFK)	PT	(CFK)	••••••••••••••	••••••••••••••	••••••••••••••	(CFK)		
E10	••••••••••••••	(CBI)	PBALL	(CBI)	••••••••••••••	••••••••••••••	••••••••••••••	(CBI)		
E11	••••••••••••••	(CBJ)	PBALR	(CBJ)	••••••••••••••	••••••••••••••	••••••••••••••	(CBJ)		
E12	••••••••••••••	(CBK)	PBALF	(CBK)	••••••••••••••	••••••••••••••	••••••••••••••	(CBK)		
E13	••••••••••••••	(RFSUM14)	PBALD	(RFSUM14)	••••••••••••••	••••••••••••••	••••••••••••••	(RFSUM14)		XO
E14	••••••••••••••	(RFSUM34)	PBALB	(RFSUM34)	••••••••••••••	••••••••••••••	••••••••••••••	(RFSUM34)		YO
E15	••••••••••••••	(RFSUM84)	PBALT	(RFSUM84)	••••••••••••••	••••••••••••••	••••••••••••••	(RFSUM84)		ZO
E16	DUDX		UAL	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••
E17	DUDY	CV	UAF	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••
E18	DUDZ	R	UAB	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••
E19	DVDX	SIETIL	RPA	••••••••••••••	••••••••••••••	••••••••••••••	RPA			
E20	DVDY		PTEM							
E21	DVDZ		ML, RMLDT							
E22	DWDX	CVTERM	MF, RMFDT				XL	••••••••••••••	••••••••••••••	XL
E23	DWDY		MB, RMBDT				YL	••••••••••••••	••••••••••••••	YL
E24	DWDZ						ZL	••••••••••••••	••••••••••••••	ZL
E25	PHID	PHID	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	PHID		
E26	DISPTIL	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••					
E27	RESU	TEM14	UALA							
E28	RESV	TEM34	UAFA				VOLL	VOLL	••••••••••••••	••••••••••••••
E29	RESW	TEM84	UABA							
E30	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	UTIL					
E31	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	VTIL					
E32	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	WTIL					
E33	RMVSU	••••••••••••••	RMVSU	••••••••••••••	RMVSU	RMVSU		RDRDE		
E34	RESUO	RES	RES					RES		
E35	RESVO	RESOLD	RESOLD					RESOLD		
E36	RESWO	DRES	DRES					DRES		
E37	DRESU	RDRDT	RDRDP					RDRDK		
E38	DRESV	DTEMP	DP					DELTKE,DEPS		
E39	DRESW	TEMPHID	PPHIP					TKE14		
E40	••••••••••••••	PN	PN	••••••••••••••	PN	PN	PN	TKE34		
E41	••••••••••••••	••••••••••••••	PHIP	••••••••••••••	PHIP	PHIP		TKE84		
E42	UB	••••••••••••••	UB					EPS14		
E43	VB	••••••••••••••	VB					EPS34		
E44	WB	••••••••••••••	WB					EPS84		
E45	DISSIP	DISSIP	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	DISSIP	DISSIP		
E46	DUHAT	HTC	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	HTC	RTERMK		
E47	DVHAT, VOLB	VOLB	VOLB	••••••••••••••	••••••••••••••	••••••••••••••	VOLB	RTERME		
E48	DWHAT						RROVOLL	RROVOLL		
E49	••••••••••••••	CPC	••••••••••••••	••••••••••••••	••••••••••••••			TKEPHID		
E50	••••••••••••••	HTCTIL	••••••••••••••	••••••••••••••	••••••••••••••			EPSPHID		
E51	••••••••••••••	TTIL	••••••••••••••	••••••••••••••	••••••••••••••					
E52	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	TKETIL		
E53	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	EPSTIL		
E54	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	TKEN		
E55	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	••••••••••••••	EPSN		
E56	••••••••••••••	••••••••••••••	UN	••••••••••••••	••••••••••••••	••••••••••••••	UN			
E57	••••••••••••••	••••••••••••••	VN	••••••••••••••	••••••••••••••	••••••••••••••	VN			
E58	••••••••••••••	••••••••••••••	WN	••••••••••••••	••••••••••••••	••••••••••••••	WN			

Fig. 5. continued.

SUB-	VOLUME	APROJ	CCFLUX	MOMFLX	SNAP,SNAPT	STATE
ROUTINE			BCCCFL	BCMOMFL	SETUPBC	
\			BCEPS	BC	SORT, APROJ	
\			BCMOMXYZ	BCMOMVEL	BC, BCEPS	
\			BCROT1		GLOBAL	
\			BCROXCEN		PFIND	
ARRAY \			MFLUXES		VOLUME	
\						
E1	•••••••••••••	•••••••••••••	•••••••••••••	RMV	MV, RMV	•••••••••••••
E2	•••••••••••••	•••••••••••••	•••••••••••••	•••••••••••••	GAMMA	GAMMA
E3	•••••••••••••	•••••••••••••	EPS, SCL	•••••••••••••	EPS	•••••••••••••
E4			CLX, BNDL	•••••••••••••|		
E5			CLY, BNDR	•••••••••••••|		
E6			CLZ, BNDF	•••••••••••••|		
E7			CFX, BNDD	•••••••••••••|		
E8			CFY, BNDB	•••••••••••••|		
E9			CFZ, BNDT	•••••••••••••|		
E10			CBX, DMOML	•••••••••••••|		
E11			CBY, DMOMF	•••••••••••••|		
E12			CBZ, DMOMB	•••••••••••••|		
E13	•••••••••••••	•••••••••••••	XO	VCIM		
E14	•••••••••••••	•••••••••••••	YO	VCIP		
E15	•••••••••••••	•••••••••••••	ZO	VCJM		
E16	•••••••••••••	•••••••••••••	UAL	VCJP		
E17	•••••••••••••	•••••••••••••	UAF	VCKM		
E18	•••••••••••••	•••••••••••••	UAB	VCKP		
E19			DRDS	DUDS		
E20			DTDS	DVDS		
E21				DWDS		
E22	•••••••••••••	•••••••••••••	•••••••••••••	XL		
E23	•••••••••••••	•••••••••••••	•••••••••••••	YL		
E24			PERJD	PERJD		
E25			FXL	•••••••••••••	FXL	•••••••••••••
E26			FXF	•••••••••••••	FXF	•••••••••••••
E27			FXB	•••••••••••••	FXB	•••••••••••••
E28	•••••••••••••	•••••••••••••	VOLL	•••••••••••••|	VOLL	
E29			ROSIE	UMOM		
E30			ROTKE	VMOM		
E31			ROSCL	WMOM		
E32			ROSIEV	SMOM		
E33			ROTKEV	FXV		
E34			ROSCLV	FXVM		
E35			MVP	MVP		
E36			MP	UJP3		
E37			FXLM	FXLM, VJP3		
E38			FXFM	FXFM, WJP3		
E39			FXBM	FXBM, SJP3		
E40			XCEN	•••••••••••••|		
E41			YCEN	•••••••••••••|		
E42			ZCEN	•••••••••••••|		
E43			DSDS	DSDS		
E44			DVOL	•••••••••••••|		
E45			XJP3	FXI		
E46			YJP3	FXJ		
E47			ZJP3	FXK		
E48				S		
E49						
E50			CCIM	•••••••••••••|		
E51			CCIP	•••••••••••••|		
E52			CCJM	•••••••••••••|		
E53			CCJP	•••••••••••••|		
E54			CCKM	•••••••••••••|		
E55			CCKP	•••••••••••••|		
E56			DCCL	•••••••••••••|		
E57			DCCF	•••••••••••••|		
E58			DCCB	•••••••••••••|		

Fig. 5. continued.

will be forced either to recalculate CLI through FSUM84, or alternatively, to save them on an external storage device such as the SSD (solid state device) on the Cray. If this procedure is chosen, unformatted write and read statements should be used to maximize speed. The appropriate place to rewind and write the file is immediately after the CALL YSOLVE in the main driver KIVA. The file should be rewound and read back into memory immediately after the CALL PSOLVE statement. (It would appear preferable to have the read statement outside the big iteration, but if the big iteration is to be repeated, subroutine TSOLVE will require the saved quantities.)

A number of terms required for QSOU fluxing are calculated before the first subcycle in subroutine CCFLUX, taking advantage of the fact that equivalenced storage is available. This improves the efficiency of both CCFLUX and MOMFLX, as these terms are not recalculated every subcycle as they were in KIVA-II. In addition, the coding in both subroutines is simplified, because fewer tests are required. For CCFLUX, geometric terms CCIM, CCIP, CCJM, CCJP, CCKM, and CCKP are stored, along with distance terms DCCL, DCCF, and DCCB. For subroutine MOMFLX, boundary condition coefficients BNDL, BNDR, BNDF, BNDD, BNDB, and BNDT are stored, along with distance terms DMOML, DMOMF, and DMOMB.

The truly dedicated arrays in KIVA-3 (not included in Fig. 5) are X, Y, Z, U, V, W, RO, VOL, P, AMU, F, FV, TEMP, SIE, BCL, BCF, BCB, ALX, ALY, ALZ, AFX, AFY, AFZ, ABX, ABY, ABZ, TKE, PIT, PIT1, SUVW, the species densities SPD, and the integer arrays I1TAB, I3TAB, I8TAB, IMTAB, JMTAB, KMTAB, and IDREG.

One problem that KIVA-II users faced when using UNIX-based operating systems was the layout of the common blocks, which were written with the implicit assumption that they would be loaded exactly as they appeared in the listing. Because AA- and ZZ- names often spanned more than one common, problems resulted if the loader scattered the commons throughout memory. This particularly created havoc when arrays were zeroed in subroutine BC, causing the code itself to be overwritten. If the loader could be instructed to load the commons as a single module, the difficulty could be avoided, but if this was not an option, the user had to restructure the commons and make appropriate changes to subroutines BEGIN, TAPERD, and TAPEWR.

The KIVA-3 common blocks have been entirely rewritten, and should no longer present this problem, as the AA- and ZZ- names no longer span common blocks. In addition, the number of named commons has been reduced; the large number of commons in KIVA-II was a holdover from earlier days when the maximum length of a common block was 2^{17} words. Although integers are to a large extent grouped separately from real numbers in the common blocks, this is not rigorously true, but should cause no problem for most users if 60-64 bit words are specified for real numbers and integers alike.

E. Cell and Vertex Flags

The general definitions of the cell flags F and the vertex flags FV carry over from KIVA-II. Again, cells with F = 1.0 participate as fluid cells. Although complex KIVA-3 meshes can be built without the necessity of large blocks of deactivated cells, permanent ghost cells and cells that deactivate/reactivate in a moving mesh imply that we still need the F = 0.0 definition. In addition to the inherent ghost cells that KIVA has always had on the right, derriere, and top, ghost cells have been added on the left, front, and bottom, facilitating the application of boundary conditions at open boundaries.

The vertex flags FV carry over from KIVA-II, with one additional case added. As before, FV = 0.0 describes a vertex that is inactive and is not a vertex of a fluid (F = 1.0) cell. FV > 0.0 describes a fluid vertex. The mesh generator is expected to define flags according to the following six possible cases.

- FV = FLFACE = 2.0 for all vertices on a piston face.
- FV = FLBOWL = 3.0 for bowl vertices not on a piston face.
- FV = FLSQSH = 4.0 for the entire squish region. Vertices that lie on a plane separating a bowl and a squish region should be defined FLBOWL.
- FV = FLDOME = 5.0 for all vertices within a head volume, but not touching the head itself.
- FV = FLHEAD = 6.0 for vertices on a cylinder head surface.
- FV = FLFLUID = 1.0 describes a fluid vertex that does not fit into any of the above categories, which are specific to engine cylinders. FV = FLFLUID would be used in ports attached to cylinders and in all miscellaneous geometries. Note that the addition of the FLFLUID = 1.0 case has caused the other five cases to have a value 1.0 greater than they had in KIVA-II.

F. Cell-Face Boundary Conditions

Three arrays, BCL, BCF, and BCB, contain floating point integers whose values identify the boundary condition to be applied on all logical left, front, and bottom cell faces. The available possibilities are fluid, moving, solid, axis, periodic front, periodic derriere, specified velocity inflow, continuative outflow, pressure inflow, and pressure outflow. The numbers in parentheses below indicate the value to be assigned to the BCL, BCF, and BCB arrays for the named condition. Although a seemingly random choice, the number sequence was selected to expedite the many tests in subroutine SETUPBC.

- FLUID (4.0). At the common face shared by two active fluid cells, the cell-face boundary condition is FLUID, and as one would suspect, this is the most common value of BCL, BCF, and BCB.

All other cases apply to cell faces on bounding walls of the overall computing domain.

- MOVING (1.0). A cell face on a piston is the intended use, in which the cell face and its four vertices move according to the time-varying value of WPISTN. The option of a two-piston geometry (see Secs. II.B.2 and V.D), is made simpler by the fact that both the bottom and the top faces of the block that represents the cylinder may be given the cell face boundary condition MOVING. KIVA-3 can recognize this case, and if an upper piston is present its cell face and vertices move according to the time-varying value of WPISTNT.

- SOLID (2.0). A stationary wall is SOLID. No capability exists at present to distinguish among various types of possible solid walls, such as no-slip, free-slip, or law-of-the-wall. All fixed and moving walls will be treated according to the value of LWALL. The one exception is a half cylinder with a symmetry plane passing through y = 0.0, which is simple freeslip.

- AXIS (3.0). A degenerate left cell face with zero area that lies on an axis.

1. Periodic Boundaries.

- PERIODF (5.0) periodic front and PERIODD (6.0) periodic derriere faces apply to true sector meshes. (Unlike KIVA-II, a full-circle cylinder with an axis does *not* require periodicity to connect front and derriere faces, because connectivity is seamless through the I3TAB and JMTAB index arrays.)

Two tables are generated in subroutine SETUPBC (see Sec. II.G.1) that identify all vertices on front and derriere periodic faces, allowing quick access to them where required. Despite having ghost cells on all sides of the mesh in KIVA-3, regular storage is still inadequate for vertex quantities at the derriere plane in a sector mesh. Subroutine CCFLUX requires coordinates X and Y rotated through the sector angle THSECT, and subroutine MOMFLX requires velocity components U and V that are similarly rotated. Figure 6 illustrates the storage problem. Here, we take the liberty of using a j index nomenclature in a sector mesh that has four azimuthal zones, hence JBAR = 4. Because ghost vertices exist at J = 1 in KIVA-3 (the dotted line), JBAR+1 values can be rotated

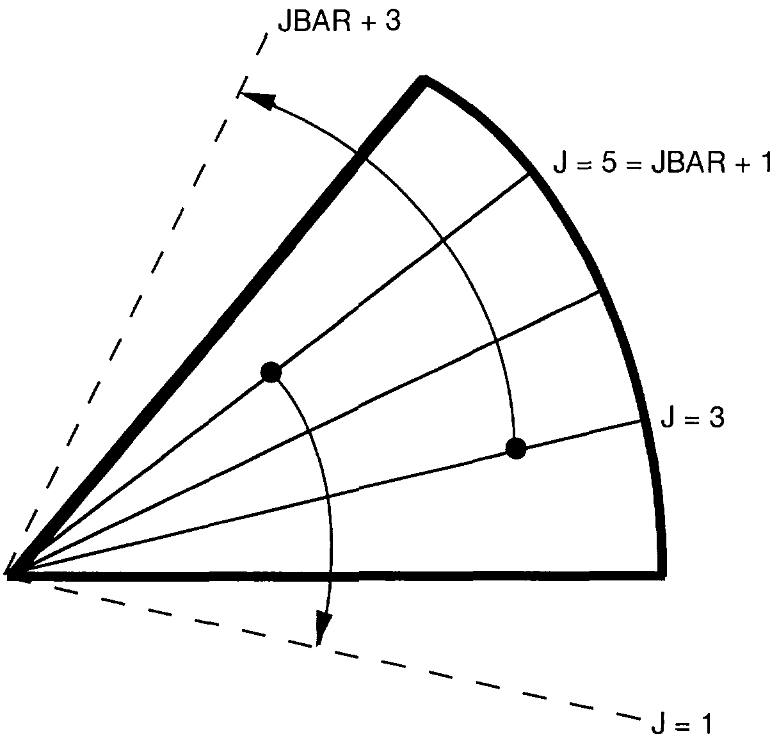


Fig. 6. Sector meshes require an extra plane at the derriere to store vertex quantities that have been rotated through +THSECT.

clockwise through -THSECT and stored there. However, J = 3 values have no available place to be stored when they are rotated counterclockwise through +THSECT, as no ghosts exist at JBAR + 3. Because of this, we need another plane of storage at the derriere. We can obtain the necessary space at no increase in storage by equivalencing to regular storage arrays that are no longer needed at the time of fluxing. Subroutine BCMOMPER performs the necessary rotations, and the JBAR + 3 storage uses arrays XJP3, YJP3, ZJP3, UJP3, VJP3, WJP3, and SJP3.

2. Inflow and Outflow Boundaries. The usefulness of the BCL, BCF, and BCB arrays in KIVA-3 is especially evident when defining inflow and outflow boundaries. Recall that in the export version of KIVA-II, inflow was available across the entire bottom boundary, outflow up the entire right and/or across the entire top boundaries. To make modifications to this, such as for a boundary open only along some portion of its length, the KIVA-II user had to patch a number of subroutines, using the existing coding as a guide.

In KIVA-3, however, any individual cell face or group of faces on an outside boundary, whatever the orientation, can be inflow or outflow. More specifically, the choices available are velocity inflow, continuative outflow, pressure inflow, and pressure outflow.

- INFLOW (7.0). To complete the specification of a velocity inflow boundary, supply a non-zero value for the normal velocity VELIN in the input file ITAPE, corresponding to the quantity WIN in KIVA-II. Subroutine SETUPBC will compute the actual u, v, and w velocity components to be applied based on the boundary orientation.

- OUTFLOW (8.0). At a continuative outflow boundary, the u, v, and w velocity components at the boundary vertex are set equal to those of the logical inside neighbor vertex.

- PRESIN (9.0) and PRESOUT (10.0). In most engine applications, pressure inflow and pressure outflow boundaries are more useful than a specified velocity condition, because time-varying pressure histories are often available from experimental data. Pressure boundaries are defined in conjunction with VELIN = 0.0 and a value of NUMPCC > 0 in the input file ITAPE, where NUMPCC is the number of entries for inflow (or crankcase) pressure data. NUMPEX is the number of entries for outflow (or exhaust) pressure data.

Tabular pressure histories may be supplied either in ITAPE or in an optional file TAPE27. Refer to the coding in KIVA-3 subroutine SETUP regions 115 - 120, and note that the expected data differ depending on whether it is ITAPE or TAPE27 that is used. It is, of course, impossible to provide a tabular format conforming to the needs of every user. The coding as supplied represents two formats that we have used, and can easily be changed. Note that if pressure data is supplied in units other than dynes/cm^2, it must be converted in subroutine SETUP. KIVA-3 subroutine PISTON interpolates on the crank angle between tabular entries to determine the appropriate pressures to be applied to each cycle.

Inflow and outflow boundaries require additional information supplied on ITAPE. This is described in the KIVA-II report[14] Sec. IV.K, but a brief synopsis is included here. The information required at an inflow boundary includes inflow densities SPDIN(M) for computing mass fluxes, where M is the index over species, and inflow reference densities SPDIN0(M), along with specific turbulent kinetic energy TKEAMB and turbulence length scale SCLAMB. The reference densities are at the reference pressure PAMB.

At an outflow boundary, the pressure PAMB is imposed either at the boundary (DISTAMB = 0.0), or some distance beyond the boundary (DISTAMB > 0.0). The purpose of DISTAMB is to enable the boundary to absorb acoustic waves, and reduce any tendency they may have to be reflected back in.

If the pressure gradient at an outflow boundary reverses, it will generate velocities directed into the computing mesh rather than outward, and it becomes necessary to describe the incoming fluid. This is accomplished by the reference species densities SPDAMB(M), referenced to pressure PAMB, and the inflow turbulence quantities TKEAMB and SCLAMB. This treatment is the same as that used in KIVA-II, but as an additional feature KIVA-3 continuative outflow and pressure boundaries have the option of reed valves to prevent flow reversal. These are controlled by the on/off settings of the ITAPE flags REEDIN and REEDOUT, for which a value of 1.0 turns the reed valve on, and 0.0 turns it off.

3. PGS and Pressure Boundaries. Pressure gradient scaling (PGS) is used in KIVA-3, as in KIVA-II, to scale up the magnitudes of the pressure fluctuations in far subsonic flows, in effect increasing the Mach number of the flow. Computational efficiency in solving the pressure equation is improved because PGS lowers the sound speed Courant number without changing other flow features of interest. When pressure boundaries are used, the implication is that the applied pressures must also be scaled up when PGS is used. In the studies we have performed, however, the pressure gradient between inflow and outflow boundaries is typically large enough so that PGS offers negligible benefit. Accordingly, we simply turn PGS off in applications having pressure boundaries by setting PGSSW = 0.0 in subroutine SETUP.

G. SETUPBC

In KIVA-3, boundary conditions have been removed from the hydro subroutines, and instead are grouped in special subroutines. For efficiency, it is clearly desirable to avoid testing inside the boundary condition subroutines, searching through the entire mesh every time the subroutine is called, to locate the vertices or cell faces whose values are to be reset. This goal is achieved through the use of a variety of tables, each of which is tailored to a specific task. These tables are created by subroutine SETUPBC, itself quite lengthy, but which is called only by SETUP, and thereafter only when the mesh is modified by the snapper. Because this occurs relatively infrequently, the time investment in SETUPBC is negligible overall. Subroutine SETUPBC is concerned with both solid and open boundaries.

1. Solid boundaries. If an axis is present, SETUPBC counts the number of zones azimuthally around (NASXISJ) and up (NAXISK) the axis, and identifies a starting I4 value at the lowest level (I4AXIS), which also serves as a flag as to whether an axis actually exists. If there is no axis, I4AXIS = 0.

Tables are loaded that identify all front and derriere periodic vertices. IPER is the number of front (or derriere) periodic vertices. The table IPERF contains the I4 indices of all front periodic vertices, and the table IPERD contains the corresponding I4 indices on the derriere periodic face.

The longest and most complex section of logic in SETUPBC is concerned with velocity setting at solid wall vertices, and is one of the prices paid for a staggered mesh with velocities defined at the vertices rather than at the cell centers. Because the normal velocity at each such vertex must be zeroed in subroutine BC, it is necessary to determine the orientation of the vertex with respect to neighboring solid wall vertices, which requires looking in all six logical directions from the vertex and selecting up to four neighbors that define the orientation. Out on the middle of a simple wall, this is easy enough to accomplish, and it is not too difficult on most curvilinear edges or on the top or bottom of an axis, but the logic can become quite tricky along oddly shaped edges, for example at the edge of a duct where it joins the cylinder, which becomes further complicated when it coincides with the edge of the piston. There are outside and inside corners, sharp, acute and obtuse edges and corners, and edges and corners at the lip of an open boundary. There are a large number of possible cases, and several orientations for each case.

The comments in SETUPBC should serve to guide the interested user through the logic used. The most common case occurs when four other boundary vertices are connected by cell edges to a given boundary vertex I4, as illustrated in Fig. 7. It is important to note that the figure is oriented such that the fluid lies *above* the plane of the drawing, that is, the observer is in the fluid. Planes are constructed surrounding vertex I4 and four selected neighboring vertices. The angles formed by the normals to these planes are used to determine the orientation of vertex I4. Given planes labeled a, b, c, and d, SETUPBC calculates

$$\cos\theta_{ab}, \sin\theta_{ab},$$
$$\cos\theta_{bc}, \sin\theta_{bc},$$
$$\cos\theta_{cd}, \sin\theta_{cd},$$
$$\text{and}\quad \cos\theta_{da}, \sin\theta_{da},$$

where θ_{ab} is the angle between normals to planes a and b, and the other angles are similarly defined. These angles are obtained from

$$n_a = \frac{(x_1 - x_c)\times(x_3 - x_c)}{|(x_1 - x_c)\times(x_3 - x_c)|},$$

$$n_b = \frac{(x_3 - x_c)\times(x_2 - x_c)}{|(x_3 - x_c)\times(x_2 - x_c)|},$$

$$n_c = \frac{(x_2 - x_c)\times(x_4 - x_c)}{|(x_2 - x_c)\times(x_4 - x_c)|},$$

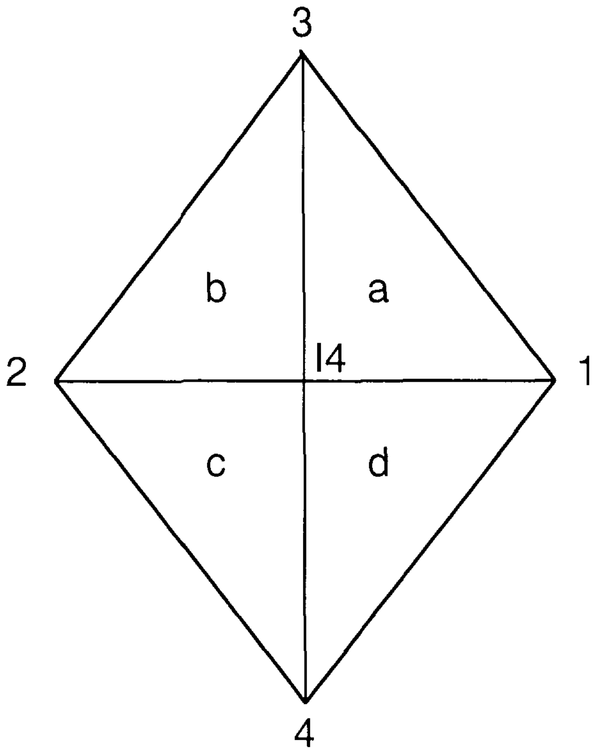


Fig. 7. Solid boundary vertex I4 and four selected neighboring vertices. Orientation is such that the fluid lies above the plane of the drawing.

$$n_d = \frac{(x_4 - x_c)\times(x_1 - x_c)}{|(x_4 - x_c)\times(x_1 - x_c)|},$$

from which

$$\cos\theta_{ab} = n_a \cdot n_b,$$

$$\sin\theta_{ab} = (n_a \times n_b)\cdot(x_3 - x_c),$$

and so forth. Based on the values of cos and sin, each pair of planes is classified as forming a *flat* surface, an *acute* angle, or an *obtuse* angle using the definitions

$\cos\theta_{ab} \geq 1/2$ for a flat surface,

$\cos\theta_{ab} < 1/2$ and $\sin\theta_{ab} > 0$ for an acute angle,

and $\cos\theta_{ab} < 1/2$ and $\sin\theta_{ab} < 0$ for an obtuse angle.

This classification is made for each pair of planes, from which we attempt to prescribe a "universal" boundary condition treatment for vertex I4. It is unlikely, however, that one can avoid considering special cases in any given geometry.

Considering symmetries and rotation, there are 21 cases that can be identified, as illustrated in Fig. 8, in which an acute angle is denoted by 'A', an obtuse angle by 'O', and a flat surface by 'F'. There are four possible velocity treatments for these 21 cases:

- Leave the velocity unchanged,
- Zero one component: $u \rightarrow u-(u \cdot n)n,$
- Zero two components: $u \rightarrow (u \cdot n)n,$
- Zero three components: $u \rightarrow 0.$

It is apparent which of these treatments to use in only three of the 21 cases shown in Fig. 8:

- $\begin{matrix} & \mathrm{O} & \\ \mathrm{F} & + & \mathrm{F} \\ & \mathrm{O} & \end{matrix}$ leave velocity unchanged,
- $\begin{matrix} & \mathrm{F} & \\ \mathrm{F} & + & \mathrm{F} \\ & \mathrm{F} & \end{matrix}$ zero one component, with n = average unit normal,
- $\begin{matrix} & \mathrm{A} & \\ \mathrm{F} & + & \mathrm{F} \\ & \mathrm{A} & \end{matrix}$ zero two components, with n = unit vector aligned with vertices A-----A.

The treatment of zeroing three components is applied in an inside corner, where the angles between the planes are all $< 120^{o}$. I have also provided placeholder coding for the flat-flat-acute-acute case. Fortunately, the set of treatments provided should cover over 99% of the wall vertices in any practical mesh. Vertices that do not belong to any of these treatments are left unchanged, but are identified on data file TAPE12 to warn the user.

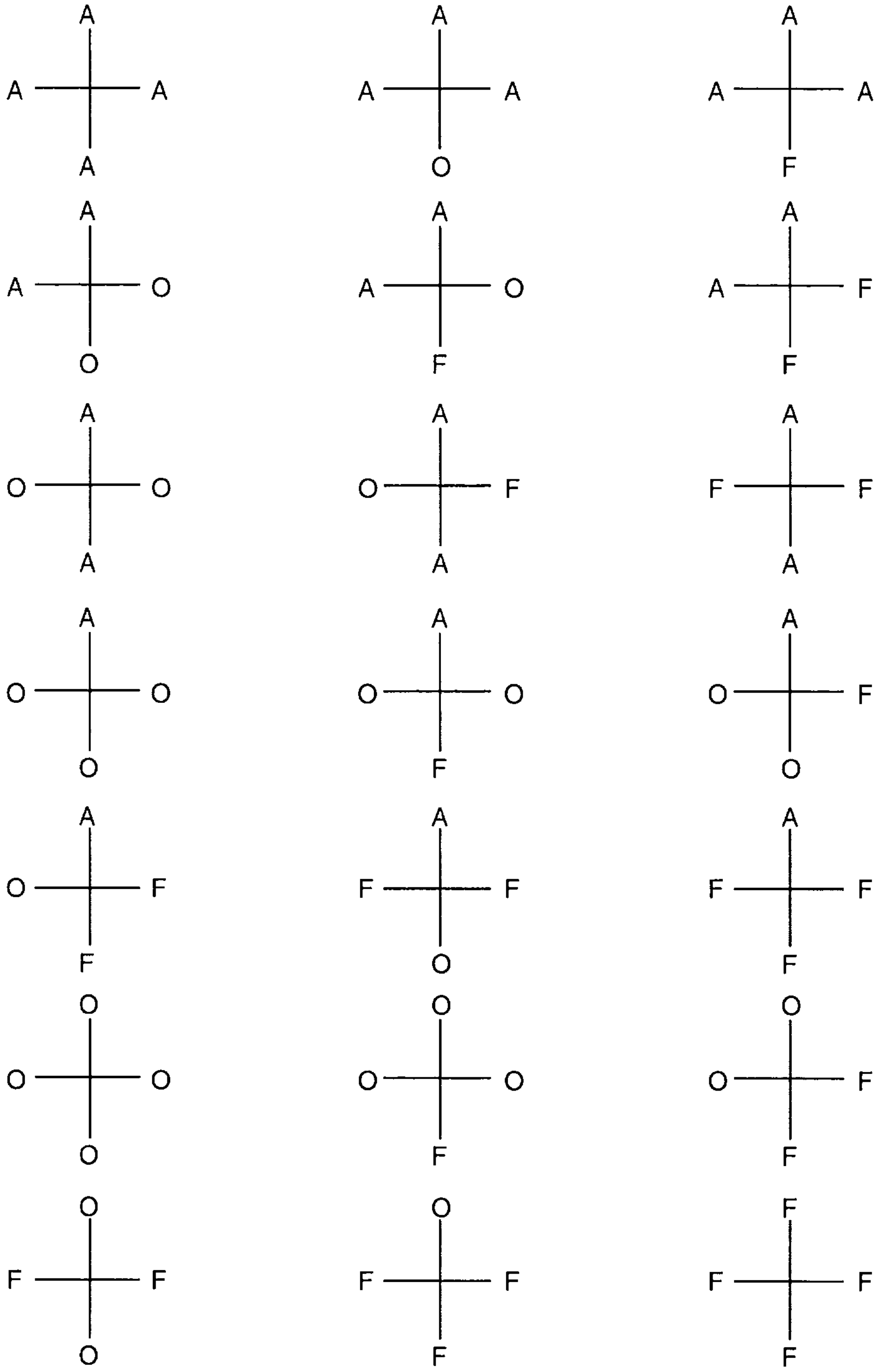


Fig. 8. Possible cases of planes meeting at a solid boundary vertex. Each pair of planes is classified as forming a flat (F) surface, an acute (A) angle, or an obtuse (O) angle.

Along the edges of a pressure inflow/outflow boundary, only three neighboring vertices exist. It is appropriate in this case to use vertex I4 itself as the fourth point. At the corners of a pressure inflow/outflow boundary, the fourth point is obtained by extrapolating the coordinates of the point inside.

Another type of boundary vertex occurs in a cylinder without an axis. Cells at the eight corners of the cube that was reformed into a cylinder are unique in that they have two faces on the cylinder wall, rather than one face. We call such cells "tent" cells, and they must be identified in SETUPBC and treated differently. Note that the vertex in the corner of a tent cell has only three neighboring vertices connected by cell edges. A fourth vertex is required for our logic as described above, and the vertex we choose is the one diagonally opposite the corner vertex on the cell face where the angle between the three planes is the greatest.

The total number of solid wall vertices is IBV, their I4 values are in the table I4BC, and the tables for the vector components n_x, n_y, and n_z for zeroing velocity components are ENX, ENY, and ENZ. A negative value of I4BC indicates that two components are to be zeroed. A positive value of I4BC with ENX = ENY = ENZ = 0.0 is used when three components are to be zeroed. Vertices that are to be left unchanged are flagged with ENX = 10^{50}. These vertices are maintained in the I4BC table only for graphics post-processor use in drawing the grid.

In addition to vertex velocity tables for subroutine BC, a number of other tables for solid walls are defined.

In subroutine BCFC, we need to zero the face velocities UAL, UAF, or UAB. NUAL counts the number of left cell faces to be reset, where the I4UAL table contains their I4 values, and the UALBC table contains a flag whose value is zero at rigid walls. Similarly, NUAF, I4UAF, and UAFBC relate to front faces to be reset, and NUAB, I4UAB, and UABBC relate to bottom faces to be reset.

In subroutine BCDIFF, the setting of specified wall temperatures TCYLWL, TPISTN, and THEAD is also automated. For the TEM14 component, NTEM14W counts entries, whose I4 values are in the table I4TEM14W. Similarly, NTEM34W and I4TEM34W are used for the TEM34 component, and NTEM84W and I4TEM84W for TEM84. To facilitate setting the temperature TPISTN on the piston face, NTEMF and I4TEMF are used; on the cylinder head, NTEMH and I4TEMH are used for setting THEAD.

In subroutines BCEPS, BCPGRAD, BCRESEZ, and LAWALL, we need to identify the solid wall faces of all F = 1.0 cells. The number of left faces is NLSOL, and their I4 values are in the table I4LSOL. It is necessary to know which side of the wall the fluid lies on, and we identify boundary orientation by the sign of I4LSOL; if the fluid cell lies to the right of the boundary, I4LSOL is positive, and if the fluid cell lies to the left of the boundary, I4LSOL is negative. Similarly, NFSOL and I4FSOL are used for solid front faces; if the fluid cell lies to the derriere of the boundary, I4FSOL is positive, and if it lies to the front of the boundary, I4FSOL is negative. Note that a symmetry plane across y = 0.0 is *not* a wall, and is not included in the I4FSOL table. Finally, NBSOL and I4BSOL are used for solid bottom faces; if the fluid cells lies above the boundary, I4BSOL is positive, and if it lies below the boundary, I4BSOL is negative.

2. Open Boundaries. Quite a number of tables are used in conjunction with inflow, outflow, and pressure boundaries. Both cell faces and vertices must be considered.

In subroutine BCFC, the flags UALBC, UAFBC, and UABBC that were described above take on values different from the 0.0 value used at a rigid wall. Here, a value of +1.0 identifies a specified inflow boundary, and -1.0 identifies a piston face. For pressure boundaries with reed valves, the I4UA- index is negative.

In subroutine BCCCFL (called by CCFLUX), NCLIN is the total number of left specified inflow cell faces, I4LIN is the table of all real cells adjacent to left inflow boundaries, and NELIN is the table of corresponding outside neighbor cells. Similar tables are created for front-derriere specified inflow boundaries, with "F" substituted for "L" in the four names, and for bottom-top specified inflow boundaries, with "B" substituted in the four names.

Pressure boundary conditions are required in a number of subroutines: BCCCFL (called by CCFLUX), BCMOMFL (called by MOMFLX), BCPEXD (called by PEXDIF), BCPGRAD (called by PGRAD), and BCRESP (called by RESP). In these subroutines, NCLPRES is the total number of left pressure cell faces, I4LPRES is the table of all real cells adjacent to left pressure boundaries, NELPRES is the table of corresponding outside neighbor cells, and I4LPPRES is the I4 table of left pressure cell faces on the boundary itself, and is therefore equal to either I4LPRES or NELPRES, depending on the left-right orientation of the boundary. (The only subroutine above that references NELPRES is BCCCFL, for setting conditions in the cells outside the boundary.) Note that these definitions are analogous to those in the above paragraph, but there is additional information in the I4- tables for pressure boundaries, in that the I4- value is flagged by being positive on pressure inflow boundaries and negative on pressure outflow boundaries. Similar tables with "F" rather than "L" are created for front-derriere pressure boundaries, and with "B" for bottom-top pressure boundaries.

Specified inflow velocities are set in subroutine BC, where NVLIN is the total number of vertices on left specified inflow boundaries, I4VLIN is the I4 table of all these vertices, and UULIN, VVLIN, and WWLIN are tables of the u, v, and w velocity components to be applied. Based on the specified velocity VELIN, these are calculated in subroutine SETUPBC so as to force the flow to follow the local mesh, and are directed from boundary vertex I4VLIN towards its inside logical neighbor vertex NEVLIN. Similar tables are created for front-derriere specified inflow vertices, with "F" substituted for "L" in the names, and for bottom-top specified inflow vertices, with "B" substituted in the names.

The velocities calculated at pressure boundary vertices are subject to being reset in two places: in subroutine BC, the normal velocity component must be zeroed if the flow tries to reverse when a reed valve is present. In subroutine BCMOMFL (called by MOMFLX), momenta at pressure boundaries must be reset to pick up the contribution for the fluxing face on the boundary. In these two subroutines, NVLPRES is the total number of vertices on left pressure boundaries, and I4VLPRES is the I4 table of all these vertices. I4VLPRES is flagged by being positive at a pressure inflow boundary, and negative at a pressure outflow boundary. Normals will be required for the flow reversal test in subroutine BC, so subroutine SETUPBC uses I4VLPRES and NEVLPRES, the I4 table of corresponding inside-neighbor vertices, to calculate XLPRES, YLPRES, ZLPRES, and XYZLPRES which are the x, y, and z components and denominator used in calculating a dot product for the test. These are used in conjunction with XNLPRES, YNLPRES, and ZNLPRES, which are the x, y, and z components of a vector directed in. Similar tables are created for front-derriere pressure vertices, with "F" substituted for "L" in the four names, and for bottom-top pressure vertices, with "B" substituted in the four names.

Continuative outflow boundaries require tables analogous to those used by pressure boundaries, with "-OUT" rather than "-PRES" in the names. I4 tables are always positive, as there is no need for a negative flag.

Subroutine MFLUXES is called by CCFLUX to keep running sums of the mass fluxes FLUXIN and FLUXOUT through specified inflow and pressure boundaries, for printing in subroutine GLOBAL. At a pressure boundary, MFLUXES can use NCLPRES and I4LPPRES, etc., which are described above and are already available. At a specified inflow boundary, MFLUXES can use NCLIN, etc., also described above, but in addition, it needs a table of I4 values on the specified velocity cell faces themselves. This is provided by SETUPBC as I4LFLX, along with its "F" and "B" counterparts. I4LFLX is equal to either I4LIN or NELIN, depending upon the left-right orientation of the boundary.

H. Snapper

Special piston motion logic has been devised that differs from that in KIVA-II to move the piston from its BDC position to its TDC position and back again. This procedure, called snapper, replaces and expands upon the chopper in KIVA-II. Subroutine SNAP behaves as a chopper in that it deactivates planes when the piston moves up, and activates planes when the piston moves down. In addition, though, snapper allows the connecting and disconnecting of port volumes with the in-cylinder volume. As in KIVA-II, the lowest plane of active cylinder vertices coincides with the piston crown and moves in Lagrangian fashion with the piston velocity. Periodically, however, a neighboring plane assumes the role of following the piston motion. If the piston is going up, the neighboring plane is the plane above the current plane; if the piston is going down, the neighboring plane is the plane below the current plane.

An example with upward piston motion past a port is shown schematically in Fig. 9. In the first drawing, the piston is at BDC. All the original z coordinates on the piston crown are saved in the table ZSVFACE, and their corresponding I4 values in the table I4FACE. These are the values at the plane labeled 2, where the saved z values are indicated by the small black squares. As the piston moves upward, the bottom plane of cylinder cells collapses, as shown in the second drawing. When plane 2 has collapsed to some specified fraction of its original height, typically a half, it is "snapped" back to its original position as saved in the ZSVFACE table, as shown in the third drawing. The z and I4 values of plane 3 replace the plane 2 values in ZSVFACE and I4FACE, and plane 3 assumes the role of the moving boundary. Plane 2, now below the piston crown, is deactivated by redefining its cells as ghosts. We perform a simple remap of the flow field onto the revised mesh, and volume average the cell quantities where volumes of cells in planes 2 and 3 are combined. Cell and vertex flags and cell-face boundary conditions are reset as necessary. As a port shuts off, its cell face BCL or BCF is switched from FLUID to SOLID, as indicated by the hatched line. Port walls are temporarily distorted at the cylinder edge as the piston moves past them. This changes the volumes of the associated port cells, and requires a density adjustment to conserve mass. Other than this, nothing special is required when the piston moves past a port. In the bottom drawing the piston, at plane 3, has moved up another half cell and is about to snap again. When the piston is moving down, the procedure is reversed and planes of cells are automatically activated as the piston moves past them.

The timing of snapper is controlled by the quantity SNAPFR in subroutine SNAP. When the piston crown does not coincide logically with a port edge, we generally snap when a plane has collapsed to half its height (SNAPFR = 0.5). At port edges, however, we snap at less than half a cell height (e.g, SNAPFR = 0.10-0.15) to minimize the temporary distortion of the port wall, such as seen in the second drawing of Fig. 9. Large fluid accelerations occur when a port first opens to the cylinder. Snapper sets a flag SNAPFLG = 1.0 when ports open, allowing the user the option to cut the timestep using the cut factor DTFSNAP in subroutine TIMSTP. The standard value of DTFSNAP is 0.2.

If there are no ports in the cylinder, or when the piston is above all the ports, I allow the mesh to "accordion" in subroutine REZONE as we did in KIVA-II. This must be done carefully, however, because there is a basic incompatibility

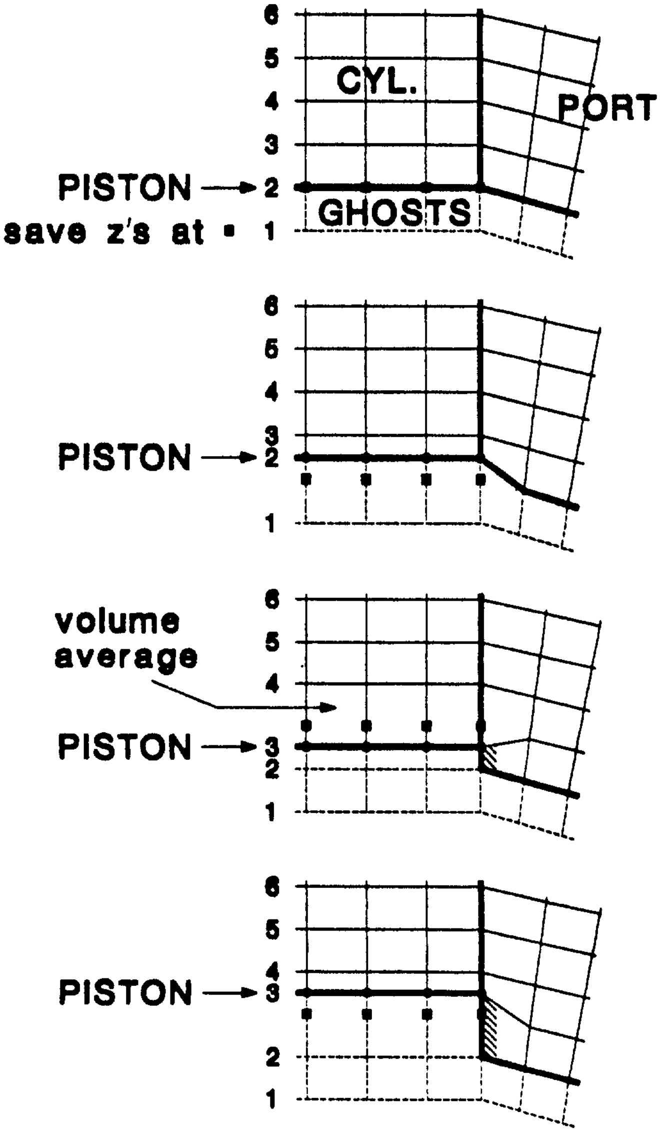


Fig. 9. Schematic showing the steps in the snapper, illustrating also the treatment at a cylinder-port interface.

between accordioning and snapping. The two procedures are mutually exclusive and neither is aware of what the other is doing to z coordinates. What can occur is that an accordioned grid line can *pass* a grid line saved in ZSVFACE, causing a plane of cells to invert in the next snap when the saved z's are restored. Because of this, I do not allow a vertex to be moved in subroutine REZONE if it would pass from one side its ZSVFACE to the other side. This restriction results in a narrow plane of cells in the squish region, but the condition is only temporary, and the zoning will regain its uniformity as time goes on.

A significant feature of the snapper is that connectivity never changes. Thus, the six index arrays I1TAB, ... , KMTAB never change. This avoids a lot of headaches during the run, but it implies that the complete mesh must be supplied by the grid generator. If the calculation is to start at BDC, one would want to start with the complete mesh anyway. However, if the calculation is to begin at TDC, as is common in engine applications, the connectivity for the full BDC mesh has to have been established by the grid generator, as KIVA-3 itself has no capability to create new cells and define their connectivity as the piston moves down. Accordingly, the variable ATDC does not appear in K3PREP, as a BDC mesh is always generated. If the calculation is to start at other than BDC, KIVA-3 subroutine SETUP moves the piston to the appropriate position by calling the snapper, repeatedly if necessary, to deactivate planes to achieve the appropriate starting mesh.

Note that in the snapper logic, there are never any partial cell faces to complicate the logic. Cell faces on the cylinder wall are simply either open (FLUID) or closed (SOLID). However, the approach is limited to square ports, implying that rounded port openings must be approximated by rectangles. In addition, snapper assumes that grid lines are completely horizontal around the periphery of the cylinder. Thus, all z coordinates around the edge of the piston must have the same value at any given instant.

After each snap, the storage is re-sorted to maintain minimum vector lengths, and subroutine SETUPBC is called to regenerate the tailored boundary condition tables for the new configuration.

For a cylinder with two opposed pistons, as described in Secs. II.B.2 and V.D, a reversed snapper for the upper piston is included in KIVA-3. This is supplied as subroutine SNAPT, which is called automatically if this option is present.

The snapper can be turned off entirely by setting the flag SNAPPER = 0.0 in the data statement in subroutine RINPUT, but this choice is acceptable only if there are no ports in the cylinder wall. As in KIVA-II, a minimum of two planes between piston and head is observed for proper application of the law-of-the-wall boundary condition.

I. Fuel Library

The fuel library in KIVA-3 block data FUELIB has been greatly expanded, and the tables of scalar quantities have been restructured to simplify future additions. At present, the library of thermophysical properties of various substances used as fuels or sprays contains the choices listed in Table I.

The data in the FUELIB tables was drawn from a variety of sources.[23-33] Most of the items listed above are pure substances that have been reasonably well characterized, with the exception of such quantities as enthalpies and liquid viscosities at the higher temperatures.

1. Liquid Fuel Mixtures. Unfortunately, most fuels of interest in the world outside a laboratory are liquid fuel mixtures and cannot be so easily characterized. The data for the KIVA-3 models for gasoline, kerosene, Jet A, and diesel #2 are each identified as some single hydrocarbon for use in fuel sprays with simple one- or two-step oxidation reaction.

In cases where the user knows the composition of the fuel, which is either simple or can be simplified sufficiently to render it tractable, a number of the principal paraffins, olefins, and aromatics are included in the library as a database for defining the fuel. This allows the use of multiple species to represent the fuel, providing that appropriate chemistry can also be supplied to make use of this information, or as a single species to be used in the same manner as the liquid fuels offered in the library.

Some guidance is available for creating an enthalpy table for a mixture of light hydrocarbons. According to Maxwell,[25] the enthalpies of the individual components of a mixture are additive in the liquid phase, that is, the molal heat content of the mixture equals the sum of the products of the molal heat contents of the components by their mole fractions. This assumption is substantially true at temperatures below the critical regions of all components. At temperatures near to or above the critical temperatures of any of the components, the liquid mixture is no longer an ideal solution of its components, and there is some deviation from the rule of additive heat contents. However, since these deviations are not too serious, and since no other simple method has been developed for determining the heat content of a liquid mixture, Maxwell suggests that the rule of additive enthalpies be used for all hydrocarbon mixtures irrespective of the critical temperatures and chemical composition of the components. A second assumption is that the enthalpies of individual components are additive in the vapor phase. This is strictly true only for vapor mixtures at infinite dilution (0 atm) but is a very close approximation for pressures up to 1 atm.

TABLE I. KIVA-3 Fuel Library

PARAFFINS (Alkanes):	
Methane	(CH_4)
Ethane	(C_2H_6)
Propane	(C_3H_8)
n-Butane	(C_4H_{10})
Isobutane	(2-Methylpropane) (C_4H_{10})
n-Pentane	(C_5H_{12})
n-Hexane	(C_6H_{14})
n-Heptane	(C_7H_{16})
n-Octane	(C_8H_{18})
Isooctane	(2,2,4-Trimethylpentane) (C_8H_{18})
n-Nonane	(C_9H_{20})
n-Decane	($C_{10}H_{22}$)
n-Dodecane	($C_{12}H_{26}$)
n-Tridecane	($C_{13}H_{28}$)
n-Tetradecane	($C_{14}H_{30}$)
n-Hexadecane	($C_{16}H_{34}$)
OLEFINS (Alkenes):	
Ethylene	(Ethene) (C_2H_4)
Propylene	(Propene) (C_3H_6)
1-Octene	(C_8H_{16})
AROMATICS (Alkyl benzenes):	
Benzene	(C_6H_6)
Methylbenzene	(Toluene) (C_7H_8)
Ethylbenzene	(C_8H_{10})
ALCOHOLS (Alkanols):	
Methanol	(Methyl Alcohol) (CH_3OH)
Ethanol	(Ethyl Alcohol) (C_2H_5OH)
ALKYNES:	
Acetylene	(Ethyne) (C_2H_2)
LIQUID FUEL MIXTURES:	
Gasoline	(C_8H_{17})
Kerosene	($C_{12}H_{23}$)
Jet A	($C_{12}H_{23}$, University of South Florida model)
Diesel #2	(DF2) ($C_{12}H_{26}$, Cummins model)
Diesel #2	(DI) ($C_{13}H_{23}$, University of Illinois model)
HYDROGEN COMPOUNDS:	
Hydrogen	(H_2)
Water	(H_2O)

Several observations follow concerning various liquid fuels and the models in the KIVA-3 fuel library.

a. Gasoline. An "Industry Average Gasoline" is blended to match physical property specifications and not necessarily a given chemical composition. Commercial gasolines vary between summer and winter and from region to region. Such an "average" gasoline will contain hundreds of hydrocarbons, including significant amounts of aromatics and olefins in addition to paraffins.[31] Most of them are saturated and contain from 4 to 12 carbon atoms per molecule, but they differ widely in structure.

In the past, we have typically used n-octane to represent gasoline, but it will be noted that the results are quite different using the new gasoline model (see the baseline calculation example in Sec. V.A). Principally, this is due to the much higher vapor pressure of gasoline as compared to n-octane, thus resulting in much greater evaporation rates. The vapor pressure table was created from the British Petroleum Company curve for "Motor Gasoline" as published by Rose and Cooper.[26] Published gasoline data has been used where available for the other tables in the fuel library, but for the time being I have taken the enthalpy to be that of n-octane. Taylor[33] suggests the chemical formula C_8H_{17}, which I have used in KIVA-3.

b. Kerosene. Kerosene is less volatile than gasoline and has a higher flash point to provide greater safety in handling. Although the chemical composition of kerosene depends upon its source, it typically consists of a mixture of about 10 hydrocarbons containing 10-16 carbon atoms per molecule; the constituents include n-dodecane, alkyl derivatives of benzene, napthalene and derivatives of napthalene. There are two grades of kerosene in the United States. Number 1 kerosene is a special low-sulfur grade fuel for critical kerosene burner applications. Number 2 kerosene is a regular-grade fuel suitable for use in flue-connected burner applications. JP1, the original jet fuel, is best described as kerosene #1 and is also similar to diesel #1.

I have labeled the general kerosene model as $C_{12}H_{23}$ and have used dodecane enthalpy as a placeholder, as the enthalpy is likely some average between that of decane and tetradecane. Otherwise, published kerosene data has been used wherever available.

c. Jet A. The most important requirements of aviation turbine fuels relate to freezing point, distillation range, and level of aromatics. Aromatics are objectionable because they increase coking deposits, and also increase the temperature of the combustor liner. Fluidity at low temperature is important to ensure proper atomization. Jet fuels for civil aviation are identified as Jet A and A1 (high-flash point, kerosene-type distillates), and Jet B (a relatively wide boiling range, volatile distillate). Jet fuels for military aviation are identified as

JP4 and JP5. JP4 has a low flash point and a wide boiling range, and JP5 has a high flash point and a narrow boiling range.[27]

The Jet A model in KIVA-3 was provided by Prof. Ben Ying of the University of South Florida, and has been used for some time in gas turbine studies in KIVA-II. It would appear that this model might also be appropriate to represent both diesel #1 and JP5. The chemical formula used is $C_{12}H_{23}$.

d. Diesel. Diesel #1 is a volatile distillate fuel oil for engines in service requiring frequent speed and load changes, and for service at extremely cold temperatures. Diesel #2 has lower volatility, and is the grade generally used in industrial and heavy mobile service. There is also a heavier #4 grade for low and medium speed engines.

The KIVA-3 fuel library contains two choices for diesel #2, labeled DF2 and DI. The DF2 model was assembled by T. L. McKinley of Cummins Engine Company. It uses diesel #2 vapor pressure, but the critical temperature, latent heat of vaporization, and liquid viscosity are set equal to those for n-hexadecane, with the remaining quantities set equal to those for n-dodecane. The chemical formula is $C_{12}H_{26}$.

The DI model, contributed by Constantine Varnavas of the University of Illinois at Urbana/Champlain,[30] offers an alternative to the DF2 model. The DI model was compiled from tables, graphs, and correlations from various sources, which Varnavas believes to be more representative of diesel #2, and uses the chemical formula $C_{13}H_{23}$.

J. Kinetic Chemistry

KIVA users often replace the kinetic chemistry subroutine CHEM with their own specialized model. This is not always the case, however, and Table II below provides suggested input values for using CHEM, assuming a simplified single-step oxidation model is used that converts fuel and oxygen to carbon dioxide and water.

The various columns in Table II contain ITAPE values for each basic hydrocarbon fuel appearing in the fuel library. CF is the forward pre-exponential factor, and the AM and BM stoichiometric coefficients for fuel, O_2, CO_2, and H_2O are labeled n_1, n_2, n_3, and n_4 respectively. The concentration exponents for the fuel and oxidizer are labeled AE_f and AE_o. (Negative values for AE_f imply that the reaction rate increases as the fuel concentration decreases.)

This data was abstracted to the extent possible from Table 4 of Westbrook and Dryer.[34] In all cases, an effective activation energy EF = 30 kcal/mole is

TABLE II. Suggested Input Data for Single-Step Oxidation Reactions

Fuel	CF	n_1	n_2	n_3	n_4	AE_f	AE_o
CH_4	8.3×10^5	1	2	1	2	-0.30	+1.30
C_2H_6	1.1×10^{12}	2	7	4	6	+0.10	+1.65
C_3H_8	8.6×10^{11}	1	5	3	4	+0.10	+1.65
C_4H_{10}	7.4×10^{11}	2	13	8	10	+0.15	+1.60
C_5H_{12}	6.4×10^{11}	1	8	5	6	+0.25	+1.50
C_6H_{14}	5.7×10^{11}	2	19	12	14	+0.25	+1.50
C_7H_{16}	5.1×10^{11}	1	11	7	8	+0.25	+1.50
C_8H_{18}	4.6×10^{11}	2	25	16	18	+0.25	+1.50
C_9H_{20}	4.2×10^{11}	1	14	9	10	+0.25	+1.50
$C_{10}H_{22}$	3.8×10^{11}	2	31	20	22	+0.25	+1.50
$C_{12}H_{26}$	3.0×10^{11}	2	37	24	26	+0.25	+1.50
$C_{13}H_{28}$	2.6×10^{11}	1	20	13	14	+0.25	+1.50
$C_{14}H_{30}$	2.2×10^{11}	2	43	28	30	+0.25	+1.50
$C_{16}H_{34}$	1.4×10^{11}	2	49	32	34	+0.25	+1.50
C_2H_4	2.0×10^{12}	1	3	2	2	+0.10	+1.65
C_3H_6	4.2×10^{11}	2	9	6	6	-0.10	+1.85
C_8H_{16}		1	12	8	8		
C_6H_6	2.0×10^{11}	2	15	12	6	-0.10	+1.85
C_7H_8	1.6×10^{11}	1	9	7	4	-0.10	+1.85
C_8H_{10}	1.2×10^{11}	2	21	16	10	-0.10	+1.85
CH_3OH	3.2×10^{12}	2	3	2	4	+0.25	+1.50
C_2H_5OH	1.5×10^{12}	1	3	2	3	+0.15	+1.60
C_2H_2	6.5×10^{12}	2	5	4	2	+0.50	+1.25
Gasoline	4.6×10^{11}	4	49	32	34	+0.25	+1.50
Kerosene	3.0×10^{11}	4	71	48	46	+0.25	+1.50
Jet A	3.0×10^{11}	4	71	48	46	+0.25	+1.50
DF2	3.0×10^{11}	2	37	24	26	+0.25	+1.50
DI	3.0×10^{11}	4	75	52	46	+0.25	+1.50

suggested, which is equivalent to about 1.5078 x 10^4 in KIVA input units when divided by the universal gas constant in kcal/mole degree Kelvin.

It must be emphasized that the value of CF is highly dependent upon the mesh type and resolution. Too high a value will result in excessive heat release and the code will fail, whereas too low a value will give insufficient heat release and combustion will not be sustained. In each application, an appropriate value must be determined empirically. (If the user installs a mixing-controlled chemistry model, this sensitivity can be mitigated or eliminated entirely, making the tabular values of CF more nearly valid.)

K. Data Integrity Checks

KIVA-3 subroutine SETUP performs a number of checks on the data it receives from the grid generator. Should a problem be detected, informative messages that describe the location and the complaint will be provided. With one exception noted below, failure to pass a check is fatal and will terminate the run.

- Invalid ghost cell index. An active cell (F(I4) = 1.0) with IMTAB(I4) = 0, JMTAB(I4) = 0, or KMTAB(I4) = 0.

- Invalid FLUID boundary. If (F(I4) + F(IMTAB(I4)) < 2.0) and BCL(I4) = FLUID, or if (F(I4) + F(JMTAB(I4)) < 2.0) and BCF(I4) = FLUID, or if (F(I4) + F(KMTAB(I4)) < 2.0) and BCB(I4) = FLUID.

- Invalid boundary between two fluid cells. If (F(I4) + F(IMTAB(I4)) = 2.0) and BCL(I4) ≠ FLUID, or if (F(I4) + F(JMTAB(I4)) = 2.0) and BCF(I4) ≠ FLUID, or if (F(I4) + F(KMTAB(I4)) = 2.0) and BCB(I4) ≠ FLUID.

- Invalid FV in fluid cell. If F(I4) = 1.0, the FV values of its eight corner vertices must all be > 0.0.

- Invalid BC- in fluid cell. If F(I4) = 1.0, the BC- values of its six cell faces must all be > 0.0.

- Concave cells. This is one error that is not treated as fatal. Although non-convex cells do not represent good grid generation practice, KIVA-3 will generally run with a few scattered non-convex cells. It has been necessary, however, to set the array of variable implicitness parameters PHIP = 1.0 in subroutine PINIT, which KIVA-3 will automatically do if there are any concave cells in the mesh. The particle-finding subroutine, PFIND, is predicted to fail should a spray particle ever enter a concave cell.

After subroutine SETUP has sorted storage into three categories by F and FV values, and has calculated the initial vertex masses MV, the following checks are made.

- Invalid FV or MV. If FV(I4) = 0.0 or MV(I4) = 0.0 between IFIRST and NVERTS.
- Invalid F. If F(I4) = 0.0 between IFIRST and NCELLS.

Several of the above data integrity checks are repeated in subroutines SNAP and SNAPT as an additional safeguard against errors.

L. Methodology Changes from KIVA-II

1. Improved Angular Momentum Conservation. The KIVA programs solve conservative difference approximations to the linear momentum equations, but because of truncation errors conservation of linear momentum does not imply conservation of angular momentum. Non-conservation of angular momentum is a common problem of compressible flow hydrodynamic codes and is one reason that special optional logic was installed in KIVA-II (and carried over to KIVA-3) to conserve the component of angular momentum about the z-axis when an axis is present. When this option is used, however, linear momentum conservation is surrendered, and other components of the angular momentum are still not conserved.

One of the main truncation errors giving rise to non-conservation of angular momentum in KIVA-II has been identified. In the transport equation for the cell-face velocities in subroutine UFINIT, there is a term that accounts for the fictitious forces, such as centrifugal forces, that arise because these cell-face velocities are velocity components in a curvilinear coordinate system. This source term is given in Eq. (82) of the KIVA-II report,[14] and its difference approximation is given in Eq. (86). The cell-face velocities are used to compute the cell volume changes in the Lagrangian phase by Eq. (102).

In rotating flows where the centrifugal force terms are large, the difficulty is that the cell volume change given by Eq. (102) can be considerably different from the cell volume change computed using the phase B vertex locations given by Eq. (115) in the cell volume formula Eq. (56). It can be shown that these two volume changes are much closer if we time-center the centrifugal force terms when computing the volume change in Eq. (102), thus reducing the truncation error and thereby obtaining better conservation of all components of angular momentum, without compromising linear momentum conservation. The time-centering is accomplished by multiplying the last term in Eq. (86) by a factor of 1/2. In KIVA, this means that the factor of 0.25 is replaced by 0.125 in the calculation of quantities DUAL, DUAF, and DUAB in subroutine UFINIT.

In addition to the above change, the centrifugal force terms (dA/dt) are now solved exactly. This eliminates the use of the previous finite difference approximation and provides a further increase in accuracy.

2. Pressure Gradient Terms at Boundaries. Experience with a variety of applications of KIVA-II and KIVA-3 has highlighted two boundary condition inconsistencies. The adjustment of the vertex velocities to account for pressure gradient terms in subroutine PGRAD has differed from the treatment accorded the face-centered velocities in subroutine RESP.

The first inconsistency concerns pressure (inflow or outflow) boundaries, at which we had neglected to adjust the vertex velocities to account for the pressure gradient. This omission appeared in the first release of KIVA-II (dated 050189), but was rectified in the second release (dated 053190).

The second inconsistency concerns vertices on solid walls, where the traditional treatment has been to set the normal pressure gradient (dp/dn) to zero. This was achieved in PGRAD by setting selected area projection terms to zero. The inconsistency manifested itself in a KIVA-3 application involving a hydrostatic equilibrium in a complex geometry, when spurious accelerations occurred at an inside corner precisely because this treatment differs from that in RESP. There, the pressure at the cell face is linearly extrapolated to obtain a boundary pressure for calculation of face velocities, and for consistency the same thing needs to be done for the vertex velocities in PGRAD.

In KIVA-3, both the treatment for pressure boundaries and the treatment for solid walls are performed by a new subroutine named BCPGRAD, which is called by PGRAD. It should be noted that the linear extrapolation of pressure, both in RESP and in BCPGRAD, is made in logical space and not physical space, and thus is subject to inaccuracy if cell sizes change significantly at the wall.

M. Troubleshooting

Abnormal termination of a calculation is caused by a floating point error, or an error that is about to occur but the condition has been detected by KIVA-3 and the calculation gracefully terminated with a message and a call to FULOUT.

The timestep is monitored every cycle, and if it drops below 10^{-4} times the time step of cycle 0, the run will be automatically terminated. If this occurs immediately in the first cycle, or by cycle 25, there is quite likely some incompatibility in initial or boundary conditions, and a comment to this effect is printed. Sometimes a smaller initial time step is all that is required to get a calculation off to a successful start.

All iteration loops in KIVA codes have counters, and the run will be terminated if the limit is reached, usually 500 iterations. This is modified, however, for the pressure iteration in KIVA-3. After 50-75 iterations in PSOLVE, control is returned to the driver, and the big iteration will be repeated up to 50 times. In our experience, this has added robustness and ties the pressure field to the solution better than having a limit of 500 pressure iterations and then terminating the run.

The test for pressure convergence in subroutine PSOLVE should also be considered in calculations that experience difficulty in convergence. We have yet to come up with a universal test appropriate for all applications. Currently, the standard code considers a cell converged if either $|$ SCALP * DP(I4) $|$ < EPSP * PTEST, in which EPSP has our usual value of 10^{-4} and PTEST is the difference between maximum and minimum pressures in the fluid region, or $|$ RES(I4) $|$ < 10^{-10} * VOL(I4). In some applications, however, these tests have evidently been too stringent, and iteration limits have been reached when the pressure field appears acceptable. Replacing the above pair of tests with $|$ RES(I4) $|$ < EPSP * VOL(I4) has worked successfully in this circumstance.

The snapper sets a flag to cut the timestep (see Sec. II.H) when ports first open, as the current time step may not be appropriate given the sudden changes in acceleration and rate of strain. Even with a small time step, sometimes the pressure field will become unstable, with pressures heading to + and - infinity. We have succeeded in avoiding this by running a fully implicit pressure calculation, which is achieved by setting the entire array of variable PHIP implicitness factors to 1.0. We do this automatically because it was found necessary in several meshes that have pressure boundaries. An easy way to force a confined flow to run implicitly is to set the flag CONCAVE = 1.0 in subroutine SETUP after the check on concave cells has been completed. This flag will be monitored in subroutine PINIT. Recall (Sec. II.K) that we have also found it necessary to run a fully implicit pressure iteration when concave cells are present.

Occasionally we have found it necessary to make the mass, momentum, heat, and turbulence calculations fully implicit, in addition to the pressure calculation. These cases have been determined by experimentation, in which we simply set the entire field of PHID implicitness terms to 1.0 in subroutine YSOLVE.

The reed valve option works reasonably well in long exhaust ports (REEDOUT = 1.0), but in very short ports, runs have failed with negative pressures and temperatures in exhaust port cells. Repeating the run with REEDOUT = 0.0 has overcome this difficulty. Real reed valves have some flow leakage, whereas the calculational valve is perfect and therefore may be considered less realistic. In general, we prefer to avoid the reed valve option, reasoning that any effect of a physical reed valve is already embodied in the experimentally-obtained pressure data applied at the boundary.

Calculations with combustion may fail at the time of ignition due to unrealistically high temperatures if the pre-exponential coefficient CF(1) is too large. Further comments appear in Sec. II.J above.

In a uniform flow, the timestep and the number of fluxing subcycles can climb in an unbounded fashion, as the timestep controls by acceleration and rate of strain are not affected. When the number of fluxing subcycles climbs above 10 or so, accuracy begins to be lost. Accordingly, it is suggested that the user effectively limit the timestep by limiting the number of fluxing subcycles allowed in uniform flows that exhibit this behavior.

III. MESH GENERATION

A. Input File From Grid Generator

KIVA-3 contains no mesh generation capability of its own, and instead reads a file TAPE17 to be provided by a separate grid generator. TAPE17 is read by KIVA-3 subroutine SETUP, and contains the following data:

- NAME (Format 10A8), a line of problem identification, up to 80 characters.

- NCELLS, NVERTS, NREGIONS (unformatted). After the grid generator has created all the blocks, patched them together, and packed storage to eliminate all vertices that have been deactivated as a result of patching, the storage vectors extend from 1 to NVERTS, where NVERTS is the final number of vertices. NCELLS is the index of the last real cell. NREGIONS is the total number of fluid regions that KIVA-3 will be concerned with. Section II.B.1 above discusses the concept of regions. Note that the grid generator is *not* asked to sort storage for the purpose of minimizing vector lengths, as KIVA-3 will sort the data read from TAPE17 after performing the basic data integrity checks on it.

- An unformatted block of data, to be read in a DO loop over I4 = 1, NVERTS, containing I4, X(I4), Y(I4), Z(I4), and FV(I4).

- A second block of data, again unformatted and over the same DO loop range, containing I1TAB(I4), I3TAB(I4), I4, I8TAB(I4),F(I4), BCL(I4), BCF(I4), BCB(I4), and IDREG(I4).

- MTABLE (unformatted), is an integer with a 0 or 1 value. MTABLES = 1 means that TAPE17 is supplying arrays of IMTAB, JMTAB, and KMTAB, and that KIVA-3 subroutine SETUP does not need to calculate these three arrays. Conversely, MTABLES = 0 means that KIVA-3 must generate these three arrays. This task is simple enough for KIVA-3 to accomplish, but requires a pair of nested DO loops that can be expensive if NVERTS is large. It is recommended that the grid generator supply these three arrays. If, however, MTABLES = 0, MTABLES is the last item appearing on file TAPE17. If MTABLES = 1, it is followed by

- A third block of data, again unformatted and over the same DO loop range as above, containing I4, IMTAB(I4), JMTAB(I4), and KMTAB(I4).

B. K3PREP

The KIVA-3 package includes a basic grid generator, K3PREP, that writes a TAPE17 conforming to the above specifications. K3PREP is not designed to generate the very complex geometries that KIVA-3 is capable of running, but it can define a variety of useful block shapes and patch them together in a straightforward manner, allowing moderately complex geometries to be constructed in a few seconds of computer time. Further, the user can modify subroutine SETUP in K3PREP to tailor it to specific needs.

K3PREP reads an input file named IPREP, which may be as short as 11 lines to define a simple single-block mesh, or may extend hundreds of lines for more complex meshes with curved boundaries. IPREP always begins with six lines that supply the following:

- NAME (Format 10A8), a line of problem identification, up to 80 characters.

- BORE, STROKE, SQUISH, THSECT (Format A8,F10.5). Values for these four quantities must be supplied, but they are applicable only to geometries containing an engine cylinder. BORE is the cylinder diameter, and STROKE is the piston travel from BDC to TDC. SQUISH is the clearance between the piston crown and the cylinder head at TDC, or half the minimum geometrical clearance between pistons in opposed-piston engines. All dimensions are in cm, as cgs units are used in KIVA programs. THSECT = 360.0 for a full-circle cylinder, = 180.0 for a half-circle cylinder with a y = 0 symmetry boundary, = 0.5 for a 2-D sector mesh, or some value that will divide evenly into 360.0 for a 3-D sector mesh.

- NBLOCKS (Format A8,I5) is the number of blocks to be generated, each of which has its own distinct (i,j,k) structure and six bounding faces. Blocks are generated one at a time, and stored contiguously in memory as they are created.

1. Defining Block Shapes. Following the above are sets of five or more lines necessary to define each *individual* block. Note in the descriptions below that all five generic block shapes illustrated in Fig. 2 can be generated. The basic five lines supply the following 36 items:

Line 1: NX, NY, NZ, NBO, NTYPE, NREGK3 (Format 6I4), where

- NX, NY, and NZ are the numbers of true fluid zones as in KIVA-II, exclusive of any possible ghosts, in the i, j, and k directions respectively.

- NBO, the number of points in an optional table of coordinates for outlining a curved block. This is discussed further below.

- NTYPE specifies the block type, where 1 = piston cup, 2 = squish, 3 = head dome, and 4 = all other cases.

- NREGK3 is the region identifier for the IDREG array supplied to KIVA-3. If an engine cylinder is present, the squish region and any piston bowl or head dome must always be numbered region 1, and any attached ports must be numbered regions 2, 3, etc.

Line 2: XC1, XC2, XC3, XC4, XC5, XC6, XC7, XC8 (Format 8F8.3),
Line 3: YC1, YC2, YC3, YC4, YC5, YC6, YC7, YC8 (Format 8F8.3),
Line 4: ZC1, ZC2, ZC3, ZC4, ZC5, ZC6, ZC7, ZC8 (Format 8F8.3), where

- XC1, ... , XC8, YC1, ... , YC8, and ZC1, ... , ZC8 are the (x,y,z) physical coordinates in cm of corners 1 through 8 of the block, following the same numbering orientation used for vertices of an individual cell. Based on the corner coordinates and NX, NY, and NZ, K3PREP locates the vertices of the block by trilinear interpolation, subject to subsequent modification by NBO data.

  If the block has a central axis, as for a cylinder (NTYPE < 4), input XC1 = XC2 = XC5 = XC6 = the cylinder radius, XC3 = XC4 = XC7 = XC8 = 0.0, and YC1 through YC8 = 0.0. K3PREP rotates x and y coordinates through the angle THSECT, and connects front and derriere faces when THSECT = 360.0. Note that the geometric center of the cylinder is placed at (x=0, y=0), corresponding to the standard values of X0 and Y0 in KIVA-3.

  A cylinder with no central axis is also an option, either full-circle (THSECT = 360.0) or half-circle (THSECT = 180.0). For a no-axis cylinder, input XC's and YC's as for a cube centered at (x = 0, y = 0). Thus, input XC1 = XC2 = XC5 = XC6 = + cylinder radius, XC3 = XC4 = XC7 = XC8 = - cylinder radius, YC2 = YC3 =YC6 = YC7 = + cylinder radius. YC1 = YC4 = YC5 = YC8 = - cylinder radius (THSECT = 360.0), or = 0.0 (THSECT = 180.0.)

Line 5: FACEL, FACER, FACEF, FACED, FACEB, FACET (Format 6F8.3), where

- FACEL, -R, -F, -D, -B, -T are the boundary conditions for the six bounding faces of the logical block, left, right, front, derriere, bottom, and top. Possible values are: 1.0 = moving, 2.0 = solid, 3.0 = axis, 4.0 = fluid, 5.0 = front periodic boundary (FACEF only), 6.0 = derriere periodic boundary (FACED only), 7.0 = specified inflow, 8.0 = continuative outflow, 9.0 = pressure inflow, and 10.0 = pressure outflow. These definitions are used by K3PREP to generate the BCL, BCF, and BCB arrays.

If NBO > 0, then NBO lines of tabular coordinate data follow to provide the detailed shape of a piston cup, piston crown, or head dome. Several possible choices are included in K3PREP.

a. **Cup or dome for a cylinder with an axis:** RBO, ZBO (Format 2F8.3). The tabular data consists of (r,z) coordinates that define all points on the bottom, right, and top edges. This is analogous to the silhouette data in KIVA-II. Again, the tabular data begins on the axis at the left-front-bottom corner of the block, but now runs counter-clockwise, and ends on the axis at the left-front-top corner of the block. The front azimuthal plane is first defined, and all points not on the bottom, right, and top edges are relaxed. Finally, the remaining azimuthal planes are created by rotation about the axis.

b. **Cup or dome for a cylinder with no axis:** XBO, YBO, ZBO (Format 3F8.3). The tabular data consists of (x,y,z) coordinates to define all points on the left, right, front, and derriere faces, one k level at a time, from bottom to top. The data starts at the right-front-bottom corner of the block and goes around the block counter-clockwise, ending at the last point before it would repeat the first point. This is repeated at each k level. All points not on the left, right, front, or derriere faces of the block are relaxed.

c. **Curved squish region with an axis:** RBO, ZBO (Format 2F8.3). The tabular data consists of (r,z) coordinates to reform the piston crown and the base of the head, giving each the same curvature. This is piston silhouette data much as in KIVA-II, with the table starting at the axis and ending at the cylinder wall, but here the silhouette is constrained to lie in one logical k plane, any cup or dome being handled as part of another logical block to be patched to this block. The front azimuthal plane is first defined, with a relaxation scheme to relocate interior points of the face. Finally, the remaining azimuthal planes are created by rotation about the axis.

2. **Patching Blocks Together.** If NBLOCKS = 1, the end of the IPREP file has now been reached. If more than one block has been defined (NBLOCKS > 1), however, K3PREP needs information to patch them together. Following all the block data are:

- NPATCH (Format A8, I5) is the number of lines of block patching commands that follow. Each command line contains the following six integers:

- NBLK1, NFACE, NBLK2, INDEX1, INDEX2, ICOOR (Format 6I4). The line may be interpreted as saying "Patch face NFACE of block NBLK1 onto the opposite face of block NBLK2, starting at logical vertices (INDEX1,INDEX2) of block NBLK2, and using the coordinates from block

ICOOR. Values of NFACE are: 1 = left, 2 = right, 3 = front, 4 = derriere, 5 = bottom, and 6 = top. ICOOR may refer either to NBLK1 or NBLK2.

Figure 10 illustrates the six patching possibilities. Note that our left-right-front-derriere-bottom-top rule implies that left faces can only be patched to right faces, right faces to left faces, front faces to derriere faces, derriere faces to front faces, bottom faces to top faces, and top faces to bottom faces. Vertex (1,1,1) of any block is always the real (FV > 0.0) vertex at the left-front-bottom corner. The possible existence of ghost vertices is to be ignored when writing patch commands.

Several features of K3PREP simplify the patching of a square port onto a cylinder wall. When defining the block for the cylinder, specify all the cylinder walls as SOLID; then when defining the port, specify the face that will be patched onto the cylinder wall as FLUID. K3PREP will automatically change BCL or BCF definitions at the joining cell faces from SOLID to FLUID as appropriate. In the patch command specify ICOOR = the cylinder block, to ensure that the cylinder coordinates prevail. This avoids having to specify the exact XC- and YC-coordinates of the intersection points on the cylinder wall when defining the square port.

Recall that it is not necessary to patch the front and derriere faces of a full-circle cylinder that has an axis, as K3PREP automatically joined these two faces when it created the block.

K3PREP flags all duplicate vertices with a negative FV value as it patches cell faces together. When patching is complete, it packs storage to eliminate these deactivated vertices. The resulting mesh has lost all memory of the number of blocks that went into its creation. Note that K3PREP does not generate a ghost plane at the i-, j-, or k- plane if the left, front, or bottom face respectively of the block is a fluid face. Bona-fide ghost planes are created automatically where needed by K3PREP, and have coordinates displaced outward in x,y,z space from the fluid boundary to ensure that they have a finite volume, and have F (or FV) = 0.0.

Patching in K3PREP takes place only over active cell faces. Ghost vertices are not patched together in the present code, because the logic would become considerably more complicated for all but the simplest meshes. As a result, the final IFIRST, NCELLS, and NVERTS have values greater than their theoretical minima. More important, however, is vector *length* between IFIRST and NCELLS and between IFIRST and NVERTS, and these are indeed at their absolute minima.

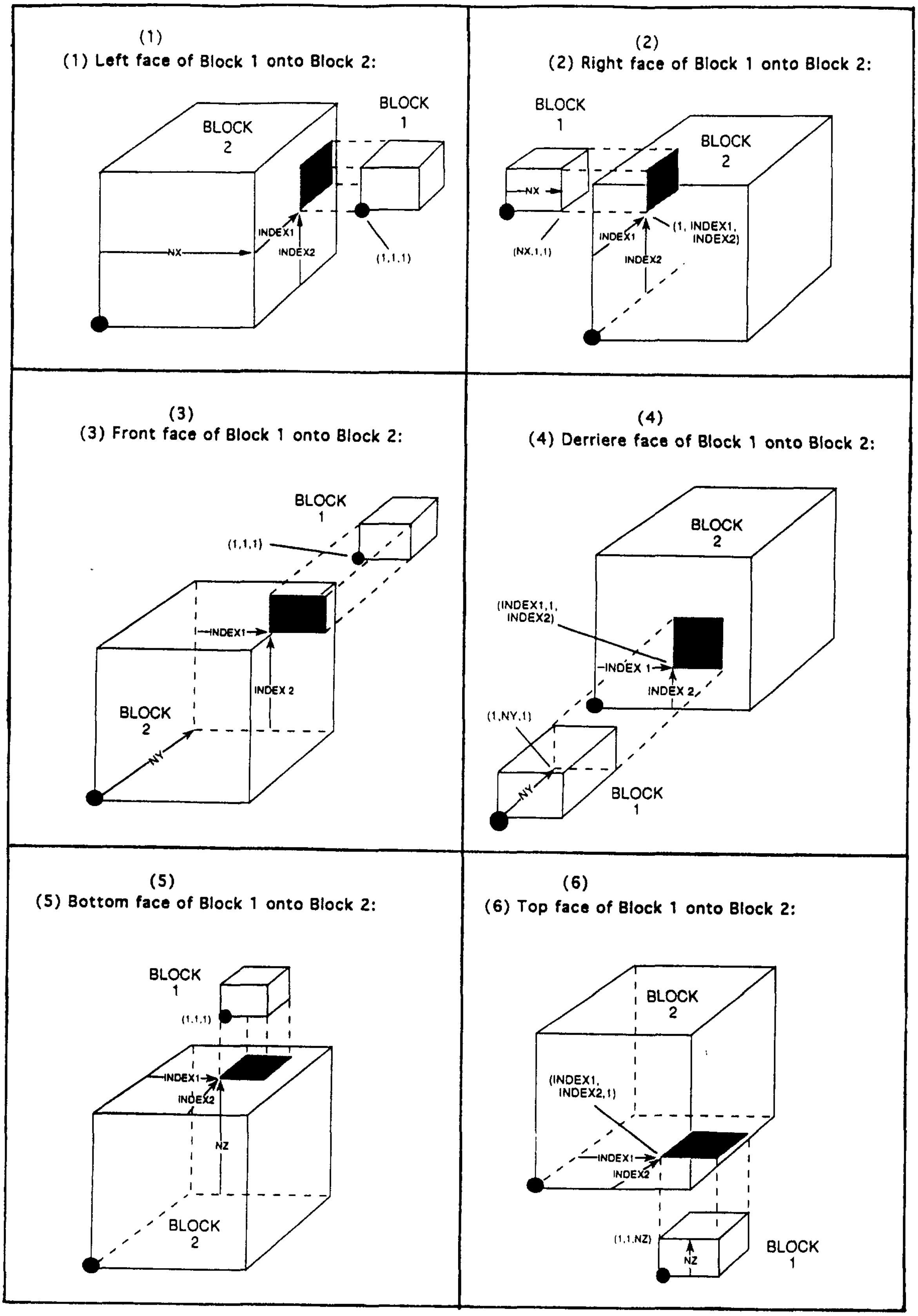


Fig. 10. The six possibilities for patching blocks together in K3PREP. Vertex (1,1,1) of any block is always the left-front-bottom corner.

IV. OUTPUT AND POST-PROCESSING

A. Output Files

Output from KIVA-3 is routed to three possible destinations, which are the user's terminal (TAPE59), a general alphanumeric information file (TAPE12), and a graphics post-processing file (TAPE9).

Monitor prints are produced at the user's terminal at least every 25th cycle. The variables KIVA-3 prints are listed in subroutines NEWCYC and TIMSTP. In addition, informational and error messages are always sent to the user's terminal. Except for the above, all alphanumeric output is written to file TAPE12, which can be scanned with a text editor and/or disposed to the operating system to be processed onto microfiche or paper listing. A four-line monitor print is written onto TAPE12 every cycle, in addition to copies of any messages sent to TAPE59. Whenever the graphics file is written, a short summary of system totals is written onto TAPE12 by subroutine GLOBAL. GLOBAL is also called immediately before and immediately after snapping, allowing the user to check for conservation. As in KIVA-II, detailed numerical cell data are optionally available, being produced only if the input flag LPR = 1.

Graphics generally prove to be the most useful form of output, and the effective presentation of three-dimensional results offers a career in itself. As in the original KIVA,[11,12] the graphics subroutines have been removed from the hydro code in KIVA-3, and information for computer-generated plots is written onto file TAPE9 for post-processing. Subroutine FULOUT is responsible for writing the necessary quantities, and dumps are made for cycles 0 and 1, and thereafter at cyclic intervals (NCFILM), crank angle intervals (CAFILM), or problem time intervals (TWFILM).

Because post-processing is the assumed standard in KIVA-3, rather than being an option as it was in KIVA-II, we have eliminated the post-processor flag IPOST and the crank angle controls CADUMP and DCADUMP that appeared in KIVA-II.

B. K3POST

The KIVA-3 package includes a basic post processor, K3POST, that can provide zone plots, velocity vector plots, and contour plots from the data dumped on TAPE9. The graphics subroutines in K3POST employ low level graphics calls, such as line drawing and point plotting, using local Los Alamos libraries, and accordingly can be modified to work on other graphics systems with minimal difficulty. Subroutines CONTUR and VELPLT were largely abstracted from KIVA-II, and hence the descriptions provided in the KIVA-II report[14] of their perspective-view logic and plot frame coordinate system remain applicable.

K3POST can be run interactively if the user wishes. If the flag NTRACTV = 1 in subroutine TAPERD, the user is prompted for a yes or no as each data set on TAPE9 is read. This allows selection for plotting only times or crank angles of interest. If NTRACTV = 0, all data on TAPE9 will be plotted without prompting.

Identifying labels are written at the bottom of each plot frame, as in KIVA-II. In K3POST, however, labeling can be omitted by setting LABELS = 0 in the common block COMD.

1. Perspective Plots with Hidden Lines Removed. K3POST can draw oblique perspective views of a three-dimensional KIVA-3 mesh with hidden lines removed, or cutaway planes through the mesh. The method was developed by Keith Meintjes of General Motors,[35] and is faster and easier to use than many competing algorithms. Most available hidden-line algorithms are limited because they assume that the data are connected in some particular way, or they contain specialized embedded plotting code. The GM algorithm has neither of these restrictions, and lends itself quite naturally to KIVA-3.

Subroutine ZONHIDE drives the subroutines that create the plots with hidden lines removed, which are identified by having names beginning with BB-. With the exception of BBCON3, these subroutines are general. In particular, BBHIDE is provided as a courtesy to the user of KIVA-3, and it would be inappropriate to incorporate it in a commercial graphics package.

BBCON3 creates a connectivity array for the KIVA-3 mesh. The cell face boundary condition arrays BCL, BCF, and BCB make the task simple. Because interior points are always hidden, these arrays quickly identify the solid cell faces (panels) of the mesh. Indices of the four vertices of each selected panel are stored in the ICON array. A second step in BBCON3 sets the MVIS array = 0 for all vertices that are not on a surface panel, and = 1 at all solid boundary vertices. The information to do this is readily available from the I4BC and the IPERF and IPERD tables, which were created by KIVA-3 subroutine SETUPBC (see Sec. II.G) for use in subroutine BC. Note that cutaway views can be drawn by making other choices in BBCON3 for setting the ICON and MVIS arrays. Unless cutaways or other custom plots are desired, the user need not be concerned with BBCON3.

Data statements in K3POST subroutine SETUP provide the necessary data for the plots. NVHIDE is the number of views of the mesh to be drawn. For each view XYROT, XZROT, and Z0 specify the rotation of the viewing axis (in a counter-clockwise direction), and the z plane of the perspective projection. An unexpected result, such as a plot that is a sunburst of straight lines, indicates that an invalid choice has been made. Occasionally, a plot will be correct except for a "hair" sticking out. In this case, a slight change in the view should remedy the problem. The example problems in Secs. V.C and V.D below include the plotting data for the perspective view plots shown.

2. Plots Made with Overlay Planes. Recall that velocity vector and contour plots in KIVA and KIVA-II were specified by requesting planar slices cutting through the mesh in i, j, or k logical space. Unfortunately, the disappearance of (i,j,k) space in KIVA-3 forces a KIVA-3 post-processor to deal with (x,y,z) space, which is somewhat more complex.

In K3POST, we define a simple two-dimensional overlay plane for each x, y, or z planar slice specified. Given some maximum number of cells available for the overlay plane, we choose an aspect ratio for it that most perfectly encloses the silhouette cut through the KIVA-3 mesh. We cycle through the overlay plane, and find where each vertex is located in the KIVA-3 mesh. The logic is analogous to subroutine PFIND in the hydro code, which starts out with some initial guess. Here, however, we have a set of initial I4 guesses to try in subroutine VFIND, where each guess is a cell in a different fluid region. A path is followed from each guess, which continues until either the cell is found, or a boundary is encountered.

Having a cell in every fluid region as an initial guess is generally adequate for locating the KIVA-3 cell, but VFIND can still fail in complex grids with curved ports. Evidence of this is seen in plots with incomplete regions in inside corners and in the vicinity of other odd shapes. In an attempt to pick up such points missed in the first pass, we have an optional but somewhat expensive second pass that tests all KIVA-3 cells lying in a band on both sides of the plane. The thicker the band, the more likely it will succeed in filling in all blank portions of the plots, but the code gets increasingly slower. Band thickness is controlled by the quantity SLFAC. The second pass is invoked by setting the flag BANDTST = 1.0 in subroutine SETUP. If this again fails to locate a KIVA-3 cell for the overlay plane vertex, we assume that the vertex must lie outside the KIVA-3 mesh, which it generally does. (If we were instead to blindly index through the entire KIVA-3 mesh from IFIRST to NCELLS, testing every cell until the correct cell was found, success would be guaranteed but the computational price would be intolerable. The use of educated guesses is clearly the way to go.)

When the KIVA-3 cell that contains a vertex has been identified, we next perform an inverse coordinate transformation, applying a Newton-Raphson iteration, to map from (x,y,z) physical space back to logical space, thus obtaining (ξ,η,ζ) interpolation coordinates that specify precisely where in the KIVA-3 cell the vertex lies.

Because velocities are defined at vertices in KIVA-3, they can be interpolated directly to the overlay plane using (ξ,η,ζ), but because the contour plots deal with cell-centered quantities, K3POST must calculate vertex-centered values before it can interpolate. This is performed by subroutine REMAP.

The finer the overlay plane, the more accurate the remapping, but it is evident that the procedure is computationally intensive and can become quite costly. We have been using 2500 cells per overlay plane, which on the average resolves the cut area as well if not better than the KIVA-3 mesh itself does.

Data statements in subroutine SETUP specify the number of views to be plotted (NVIEWS), and for each view a value of XSLV, YSLV, and ZSLV. Only one of the three can be non zero, and is some positive or negative x, y, or z coordinate in cm. Specifying all three equal to zero is not allowed. To plot across y = 0.0, for example, one might specify XSLV = ZSLV = 0.0 and YSLV = 10^{-3}.

Contour plots are drawn by subroutine CONTUR, and velocity vector plots are drawn by subroutine VELPLT. In addition, subroutine PARPLT plots the spray particles projected onto the overlay plane, and subroutine PVPLOT plots spray particle velocity vectors. Smooth and accurate contour plots are drawn when cell quantities are remapped onto the overlay plane, but often velocity vector plots are too finely resolved and appear too busy when vectors are plotted at every vertex of the overlay plane. In light of this, we have incorporated a stride control in subroutine VELPLT. ISTRIDE = 1 will draw all vectors of the remap plane, ISTRIDE = 2 will draw every other vector, and so forth. Vector length is controlled by the scaling factor SCALE. Subroutine OUTLINE accurately outlines each plot, using the KIVA-3 coordinates at solid boundaries in an interpolation scheme analogous to that used in a contour plotter.

3. 2-D Plots Across y = 0. In addition to the velocity vector and contour plotting subroutines PVPLOT, VELPLT, and CONTUR that use the overlay plane logic described above, K3POST also includes plotting subroutines ZONPLTY, PVPLOTY, VELPLTY, and CONTURY that can quickly draw plots across the y = 0.0 plane. These are appropriate for 2-D sector meshes, 180° meshes with a symmetry plane across y = 0.0, or 360° meshes that have a flat plane across y = 0.0. For 2-D sector meshes, a mirror image is also plotted if the flag MIRROR = 1. The 2-D plotting routines are turned on by setting PLOTY0 = 1.0 in subroutine SETUP. (For 2-D sector meshes, subroutine SETUP will automatically set PLOTY0 = 1.0 and both NVHIDE and NVIEWS = 0.) After drawing a plot of the grid at y = 0.0, ZONPLTY will draw a second plot of the grid outline with the spray particles projected onto it, using the x and z particle coordinates XP and ZP. Velocity vector lengths in PVPLOTY and VELPLTY are controlled by the scaling factor SCALEY0 in subroutine SETUP.

V. EXAMPLE PROBLEMS

In this section, we present four calculations that are intended to give the new KIVA-3 user some familiarity with the general features of the code and with generating meshes with K3PREP. For the users intending to switch to their own grid generator to replace K3PREP, the examples are also useful as a check as to whether that grid generator is writing equivalent data on TAPE17. None of these examples approaches the level of complexity that KIVA-3 is intended for, and the meshes used here are considerably coarser than in a practical application. Complete input files IPREP and ITAPE are listed in the Appendix.

Selected output plots from each example are included so that results may be compared with ours. To expedite comparison, subroutine SETUP of K3POST contains several sets of data statements to control views and plots. Comments indicate the appropriate dataset to be activated for each example. Examination of these data statements is helpful for gaining familiarity with the various plots that can be produced by K3POST.

Exact agreement with our calculated values should not be expected. Results are sensitive to word length and differences in various Fortran compilers, and even within any given compiler, to any change in the sequence of instructions it generates. Although we intend our pseudo-random number generator to be portable to all other computers, we have no independent verification of this. The non-Cray user should refer to the application comments in function FRAN.

Users whose machines have a word length shorter than 60-64 bits should put the cell arrays into double precision. Few complex CFD codes today can be run with 32- or 36-bit words, because roundoff errors are overwhelming and will reduce the results to nonsense.

The CPU times given below are for a Cray Y-MP8 operating in single processor mode with the UNICOS operating system, and using the Cray CFT77 Fortran compiler. Note that the CPU usage is influenced by the dynamic load on the resources in a time-sharing environment.

A. BASELINE: 2-D DISC ENGINE

Our example of a direct-injected stratified charge (DISC) engine is familiar to many users of KIVA-II. The chamfered bowl piston geometry illustrates the use of three blocks in K3PREP to generate a simple geometry with a single fluid region, while the calculation itself exercises the spray and chemistry submodels. Although the dimensions and timing of spray and ignition are typical, it is not our intent to represent any real engine.

Figure 11 is a mirror image plot of the 2-D cylindrical mesh across the y = 0.0 plane at -90° ATDC, when the calculation begins. Note that ten planes in the squish region were specified in IPREP. However, the number has been reduced to six planes by snapper calls in KIVA-3 subroutine SETUP, in adjusting the supplied BDC mesh to a configuration appropriate for the piston position at the specified starting crank angle. At this time, the cylinder contains only pure air with a Bessel function swirl profile.

During the compression stroke, liquid gasoline (see Sec. II.I.1.a) is sprayed into the cylinder from an injector with a single half sine wave pulse, located close to the axis. The injection begins at -52° ATDC, has a duration of 12.672°, and supplies 11.6 mg of fuel in the form of a hollow cone spray.

Figure 12 illustrates several quantities of interest at -30° ATDC, shortly before ignition. Although 400 computational spray particles (each representing some number of identical physical droplets) were injected, evaporation has reduced their number to 245, representing 3.84 mg of unevaporated fuel. Also evident in Fig. 12 is the effect of spray drag on the velocities, and a cool region in the temperature field due to spray evaporation, although the overall temperature in the cylinder has increased considerably due to compression. By this time, the snapper has reduced the squish region above the mesh to two planes of cells.

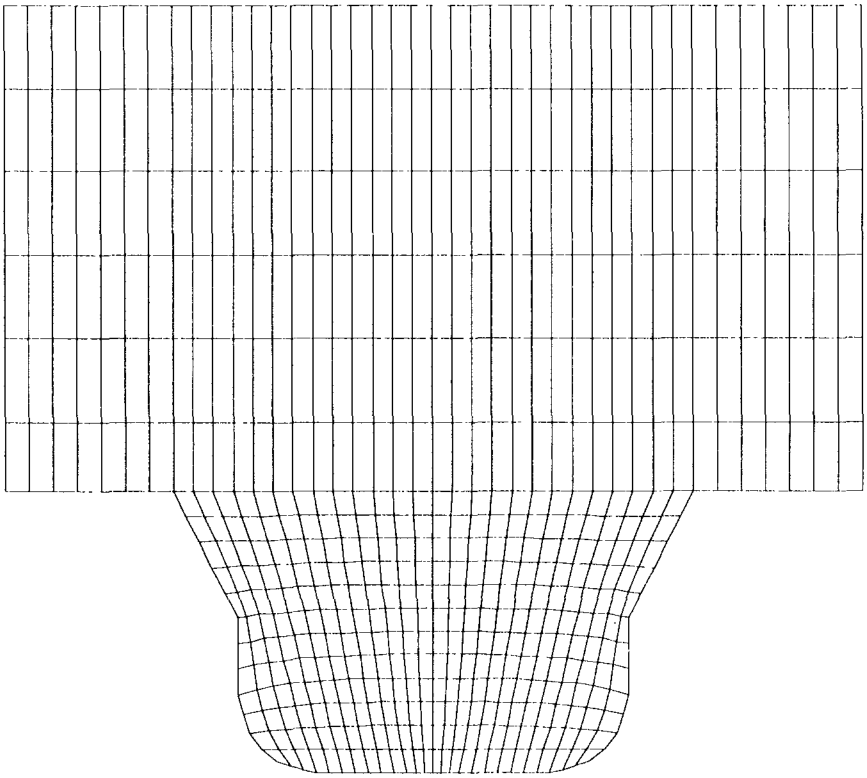


Fig. 11. The front (y = 0.0) plane of the computational mesh, and its mirror image, for the baseline 2-D DISC engine mesh at crank angle -90°ATDC, cycle 0.

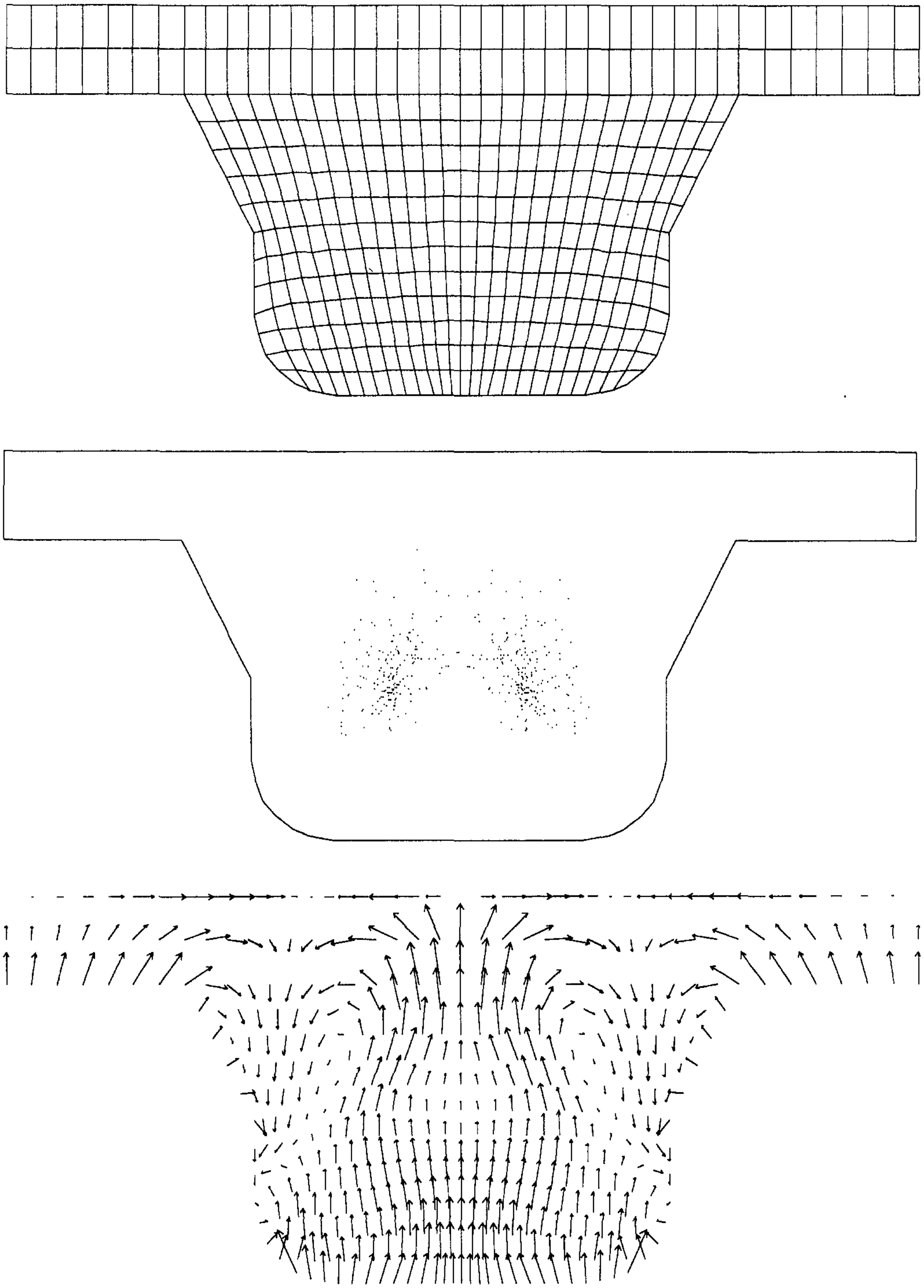


Fig. 12a. The baseline calculation at crank angle -29.33° ATDC, cycle 64. Top: the computing mesh. Center: the 245 spray particles. Bottom: velocity vectors; UMAX = 463 cm/sec, VMAX = 2993 cm/sec, WMAX = 716 cm/sec.

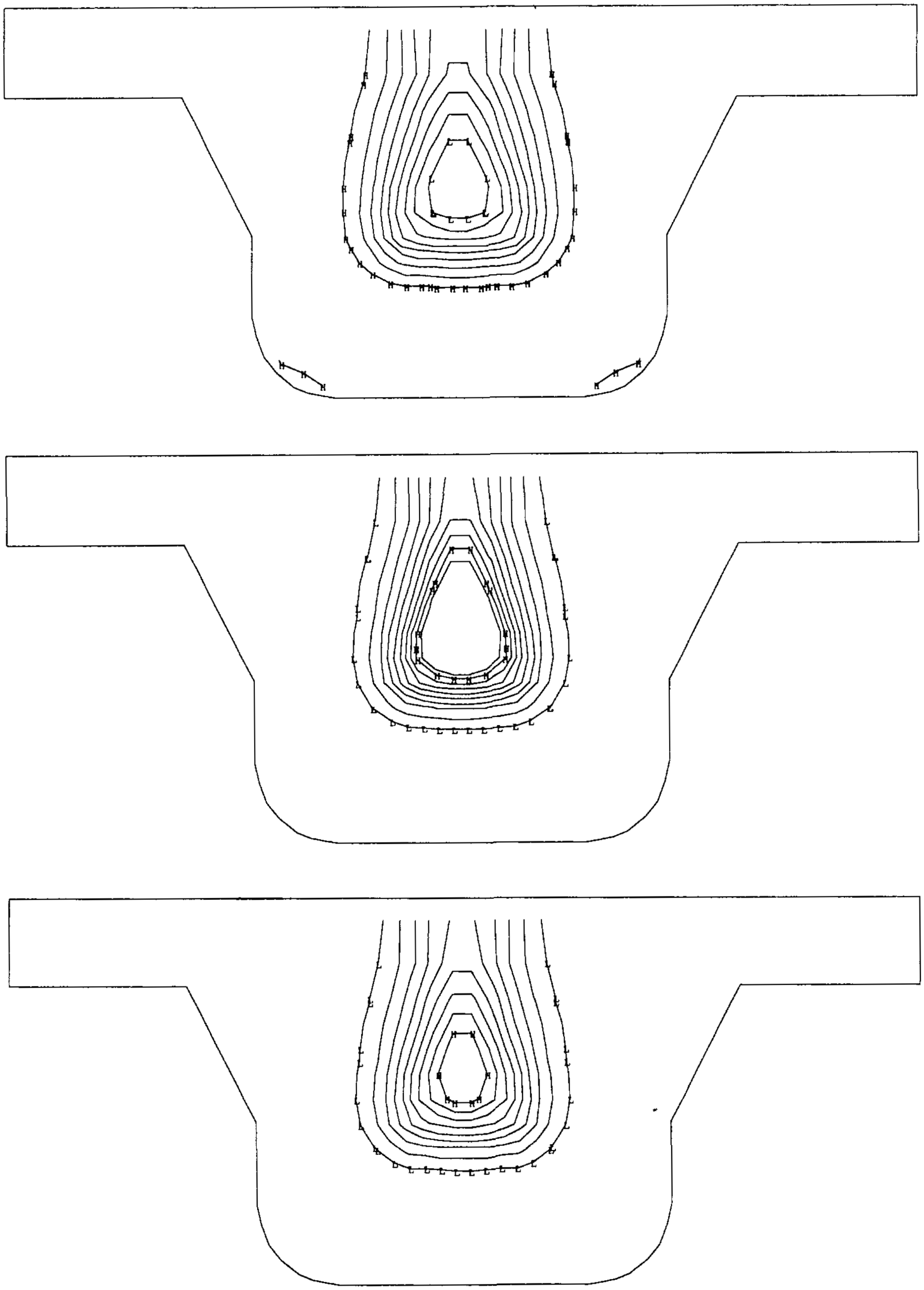


Fig. 12b. The baseline calculation at crank angle -29.33° ATDC, cycle 64. Top: isotherms; the minimum and maximum values are 434 K and 653 K. The L contour is 456 K, the H contour is 631 K, and the contour interval is 21.8 K. Center: equivalence ratio; the minimum and maximum values are 0.0 and 7.49. The L contour is 0.50, the H contour is 4.50, and the contour interval is 0.50. Bottom: fuel vapor mass fraction; the minimum and maximum values are 0.0 and 0.323. The L contour is 0.032, the H contour is 0.290, and the contour interval is 0.032.

Snapping occurs at crank angles -82.00°, -64.00°, -48.10°, and -33.33° ATDC. (After TDC, planes are reactivated at approximately the same piston positions.)

The fuel-air mixture is subsequently spark-ignited at -27° ATDC. The premixed fuel vapor begins to burn at about -24° ATDC, the burn rate becoming rapid after -20° ATDC. Selected views that illustrate the flow field at TDC are shown in Fig. 13. At this time, only 21 spray particles remain unevaporated, and most are on the wall of the piston bowl. The remaining particle mass is 0.102 mg, and the unburned fuel vapor mass is 0.397 mg.

K3PREP required 0.4 CPU seconds to generate the mesh, which contains 1363 vertices. The KIVA-3 run from -90° ATDC to TDC required 34.5 CPU seconds to run 221 cycles. (With a finish crank angle CAFIN = +90.0 selected rather than CAFIN = 0.0, the run required 56.3 CPU seconds to run 311 cycles.)

The IPREP file for the baseline can be easily modified (see Sec. III.A) to generate other sector meshes, such as THSECT = 72.0, or a 360° full circle mesh.

B. DOMED HEAD GEOMETRY

The purpose of this example is to illustrate an IPREP file for generating a squish region with a curved piston face and head surface with a dome, in a simple 2-D geometry.

Figure 14 shows the domed head mesh at cycle 0, and Fig. 15 shows the mesh and velocity vectors at -89° ATDC and again at +1° ATDC. The comparatively large velocities in the "tent" cells in the dome are similar to those seen in the corners of the piston bowl in the baseline grid. Because six out of eight vertices are on the wall in such cells, pressure accelerations for the center vertices on the wall are not accurate. Conversely, however, the poor coupling prevents these vertices from having a significantly adverse effect back on the overall flow field.

K3PREP required 0.3 CPU seconds to generate the mesh, which contains 783 vertices. The KIVA-3 run required 14.9 CPU seconds to run 361 cycles, from -180° ATDC to +181° ATDC. The ITAPE is the same as the baseline except: NAME, CAFIN = 180.0, ATDC = -180.0, T1INJ and T1IGN = 9.9E+9, and BORE, STROKE and SQUISH as on IPREP.

C. TEAPOT: TWO-STROKE ENGINE, COLD FLOW

This mesh, nicknamed for its appearance, was originally intended as a tool for developing K3PREP, to test pressure inflow and outflow boundaries, and for

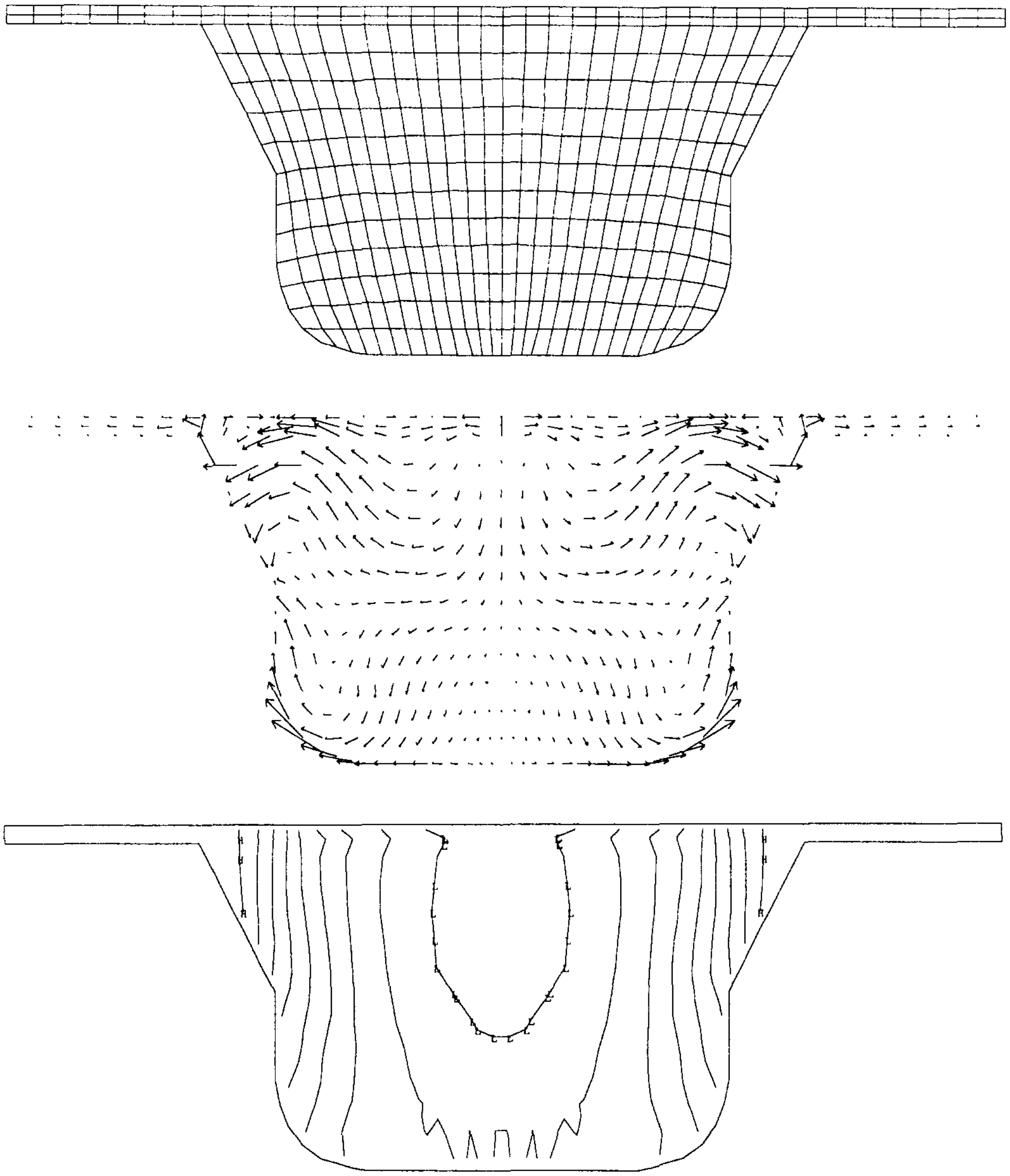


Fig. 13a. *The baseline calculation at 0.45° ATDC, cycle 221. Top: the computing mesh. Center: velocity vectors; UMAX = 591 cm/sec, VMAX = 2957 cm/sec, WMAX = 454 cm/sec. Bottom: pressure, in dynes/cm^2; the minimum and maximum values are 4.01 $\times 10^7$ and 4.12 $\times 10^7$. The L contour is 4.02 $\times 10^7$, the H contour is 4.11 $\times 10^7$, and the contour interval is 1.15 $\times$ 10^5.*

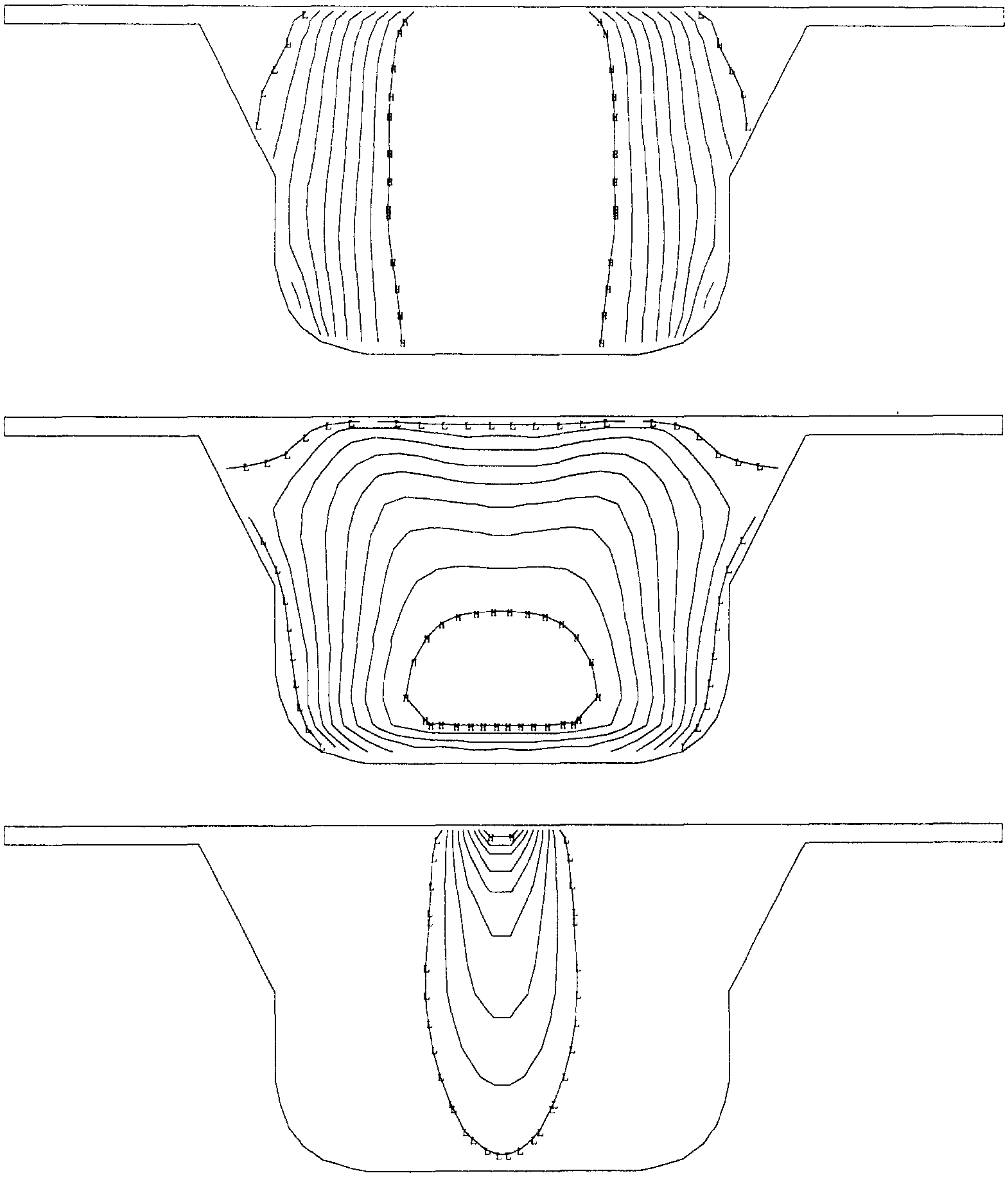


Fig. 13b. The baseline calculation at 0.45^{o} ATDC, cycle 221. Top: isotherms; the minimum and maximum values are 871 K and 2785 K. The L contour is 1062 K, the H contour is 2593 K, and the contour interval is 191 K. Center: turbulence kinetic energy, in (cm/sec)^2; the minimum and maximum values are 1.67×10^4 and 5.75×10^5. The L contour is 7.26×10^4, the H contour is 5.19×10^5, and the contour interval is 5.59×10^4. Bottom: fuel vapor mass fraction; the minimum and maximum values are 0.0 and 0.060. The L contour is 0.006, the H contour is 0.054, and the contour interval is 0.006.

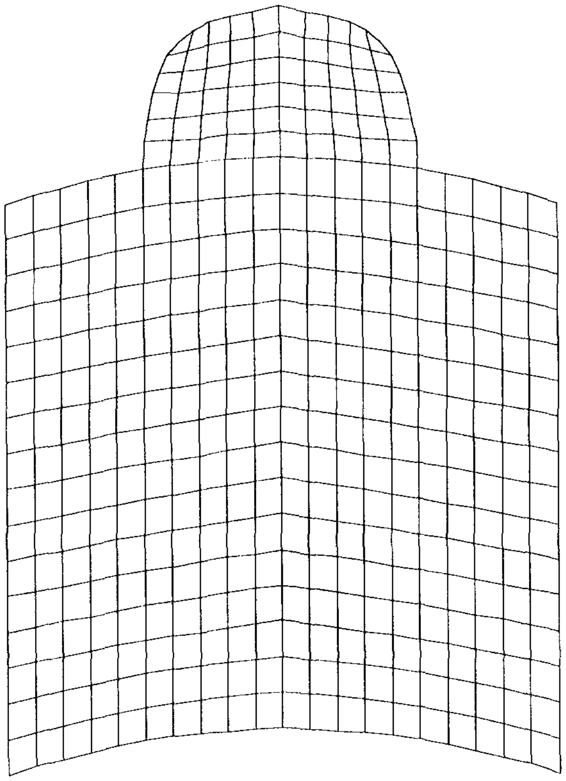


Fig. 14. The front (y = 0.0) plane of the computational mesh, and its mirror image, for the domed-head example at crank angle -180° ATDC, cycle 0.

debugging the snapper algorithm for moving the piston past the ports. The teapot was a prototype for our two-stroke engine modeling, and has been used in both 180° and 360° configurations.

Figure 16 shows the mesh for the 180° version. K3PREP uses seven blocks to create the mesh, one for the cylinder, four for the curved intake duct, and two for the straighter exhaust duct. Simple pressure histories are applied at the two open boundaries, using tabular data supplied on ITAPE following NUMPCC and NUMPEX.

Figures 17-23 are a sequence of mesh and velocity vector plots across the y = 0.0 plane at selected times from a calculation run from -180° ATDC to +180° ATDC. The intake port closes at -72.52° ATDC, and the exhaust at -47.24° ATDC. The exhaust reopens at +46.95° ATDC, and the intake at +72.89° ATDC.

K3PREP required 1.3 CPU seconds to generate the mesh, which contains 4552 vertices. The KIVA-3 run required 15.89 CPU minutes for 829 cycles to run one complete 360° revolution. The time step is limited at certain portions of the run by the high velocities in the system.

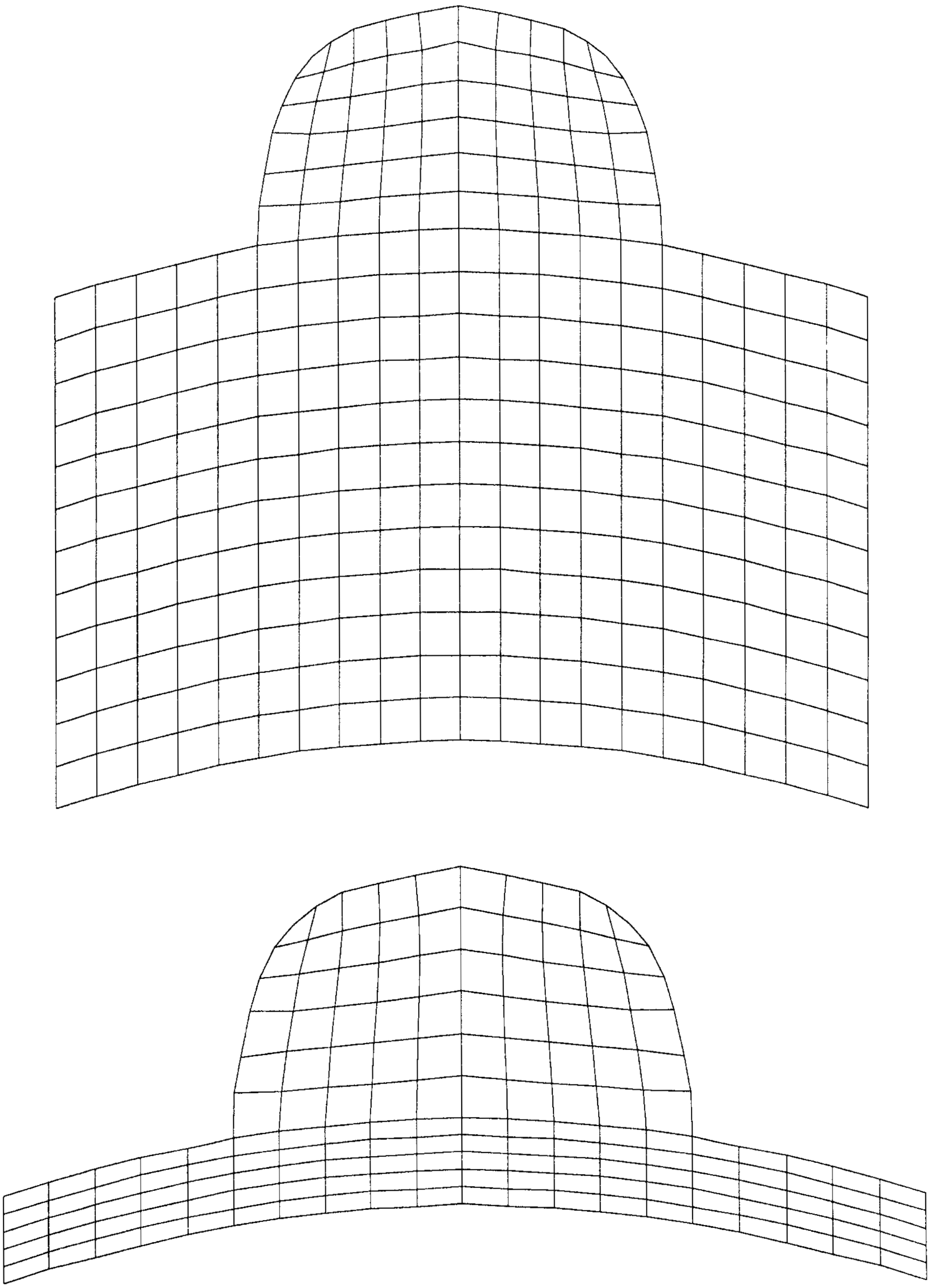


Fig. 15a. The computational mesh for the domed head example at crank angles -89.00° ATDC, cycle 91 (top), and 1.00° ATDC, cycle 181 (bottom).

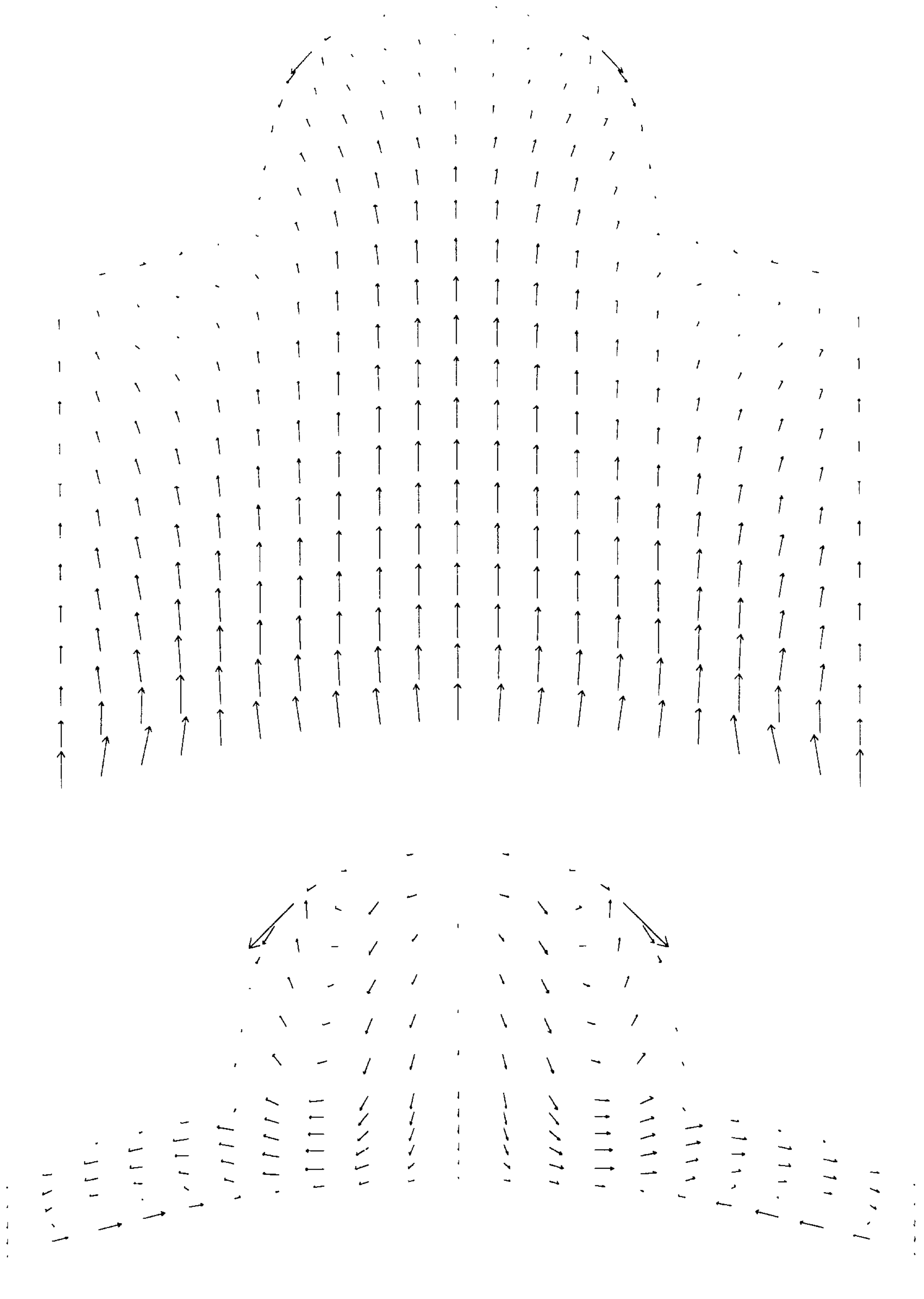


Fig. 15b. Velocity vectors across the y = 0.0 plane for the domed head example. Top: crank angle -89.00° ATDC, cycle 91; UMAX = 462 cm/sec, VMAX = 2061 cm/sec, WMAX = 852 cm/sec. Bottom: crank angle 1.00° ATDC, cycle 181; UMAX = 575 cm/sec, VMAX = 2357 cm/sec, WMAX = 575 cm/sec.

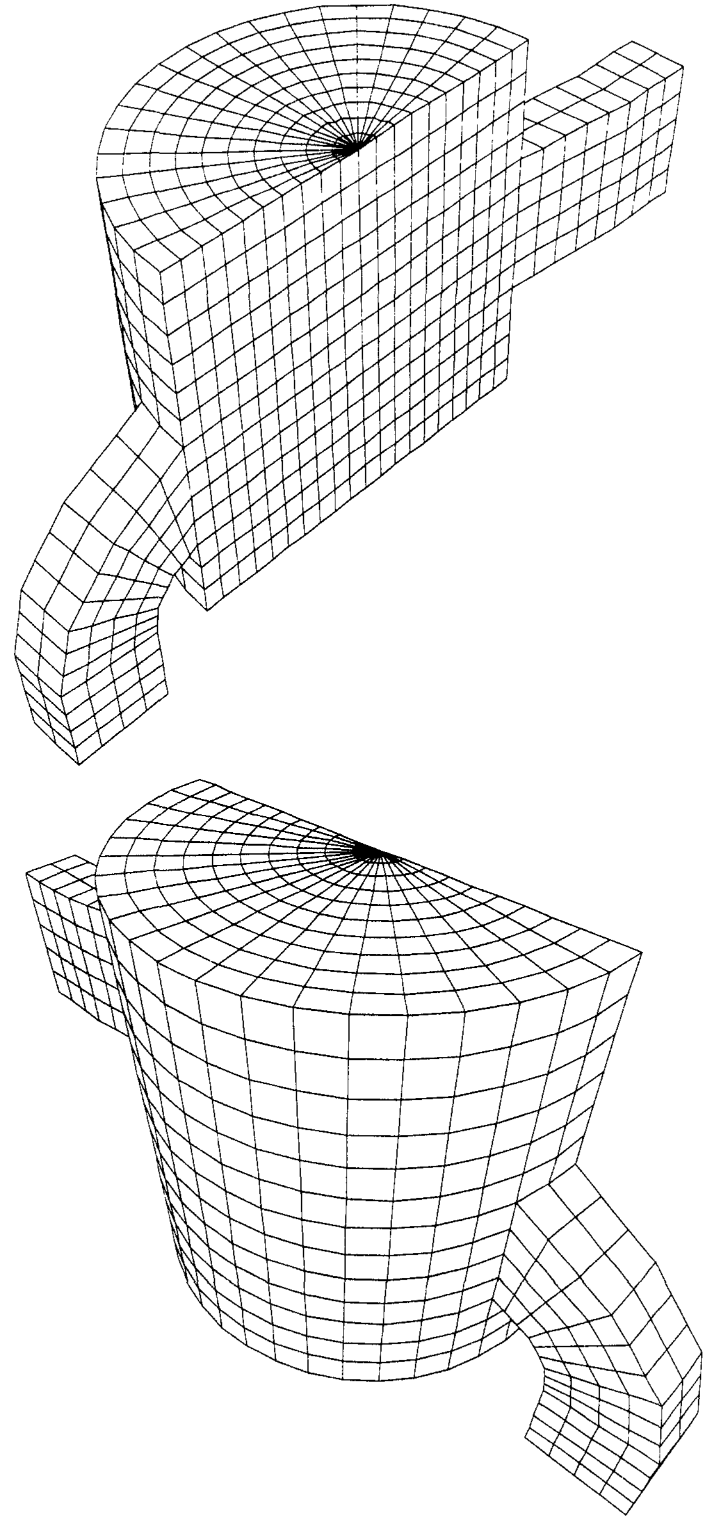


Fig. 16. The computational mesh for the 180° teapot example at crank angle -180° ATDC, cycle 0. The y = 0.0 plane is a symmetry boundary. In the upper plot, XYROT = XZROT = 45.0° and Z0 = 4.0. In the lower plot, XYROT has been rotated to -45.0°.

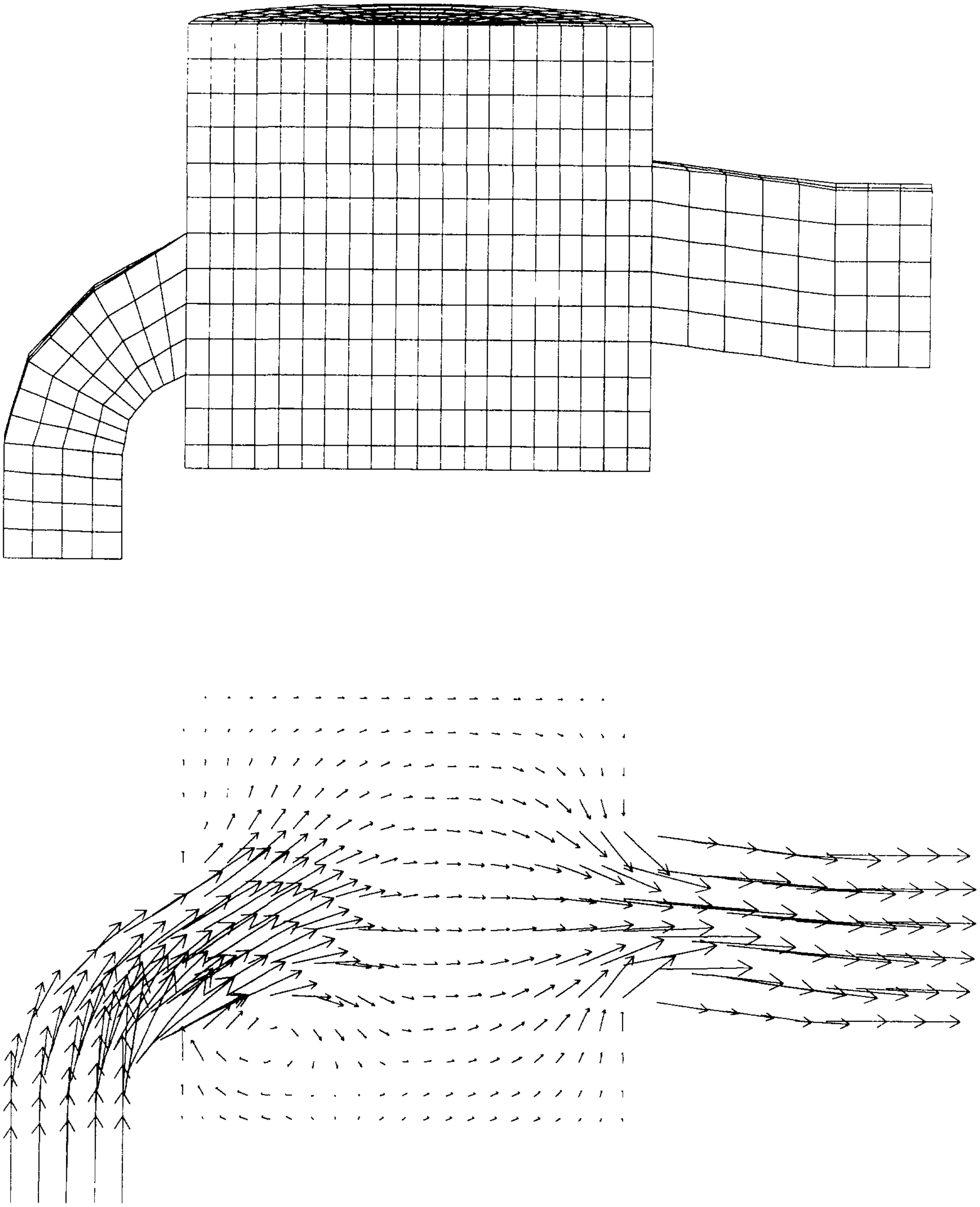


Fig. 17. The 180° teapot mesh (top) and velocity vectors across the symmetry plane (bottom) at crank angle -159.52° ATDC, cycle 41. In the perspective views of Figs. 17-23, XYROT = 90.0°, XZROT = 85.0°, and Z0 = 10^{10}. UMAX = 6325 cm/sec and WMAX = 6702 cm/sec.

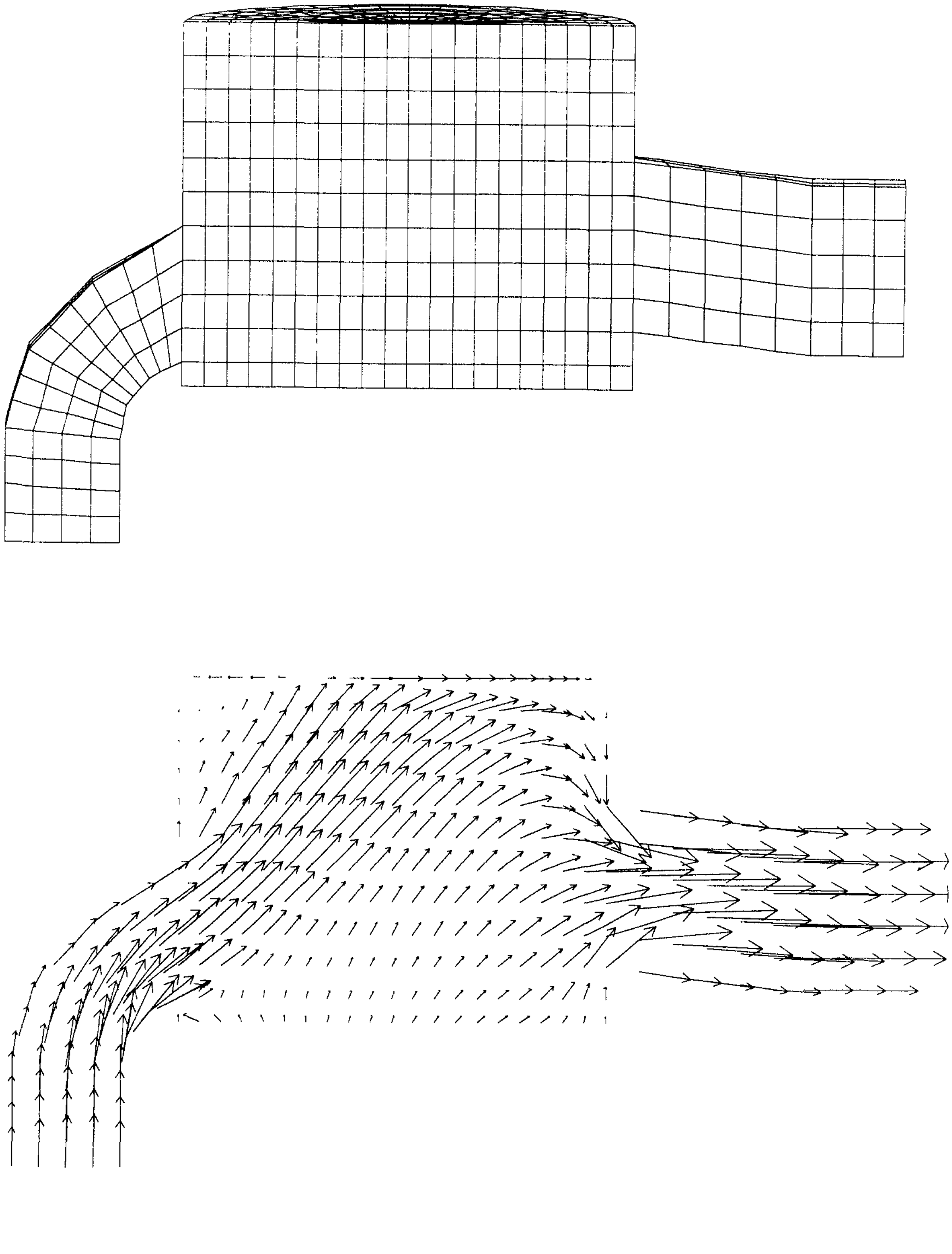


Fig. 18. The 180° teapot mesh (top) and velocity vectors across the symmetry plane (bottom) at crank angle -119.52° ATDC, cycle 121. UMAX = 8496 cm/sec and WMAX = 4993 cm/sec.

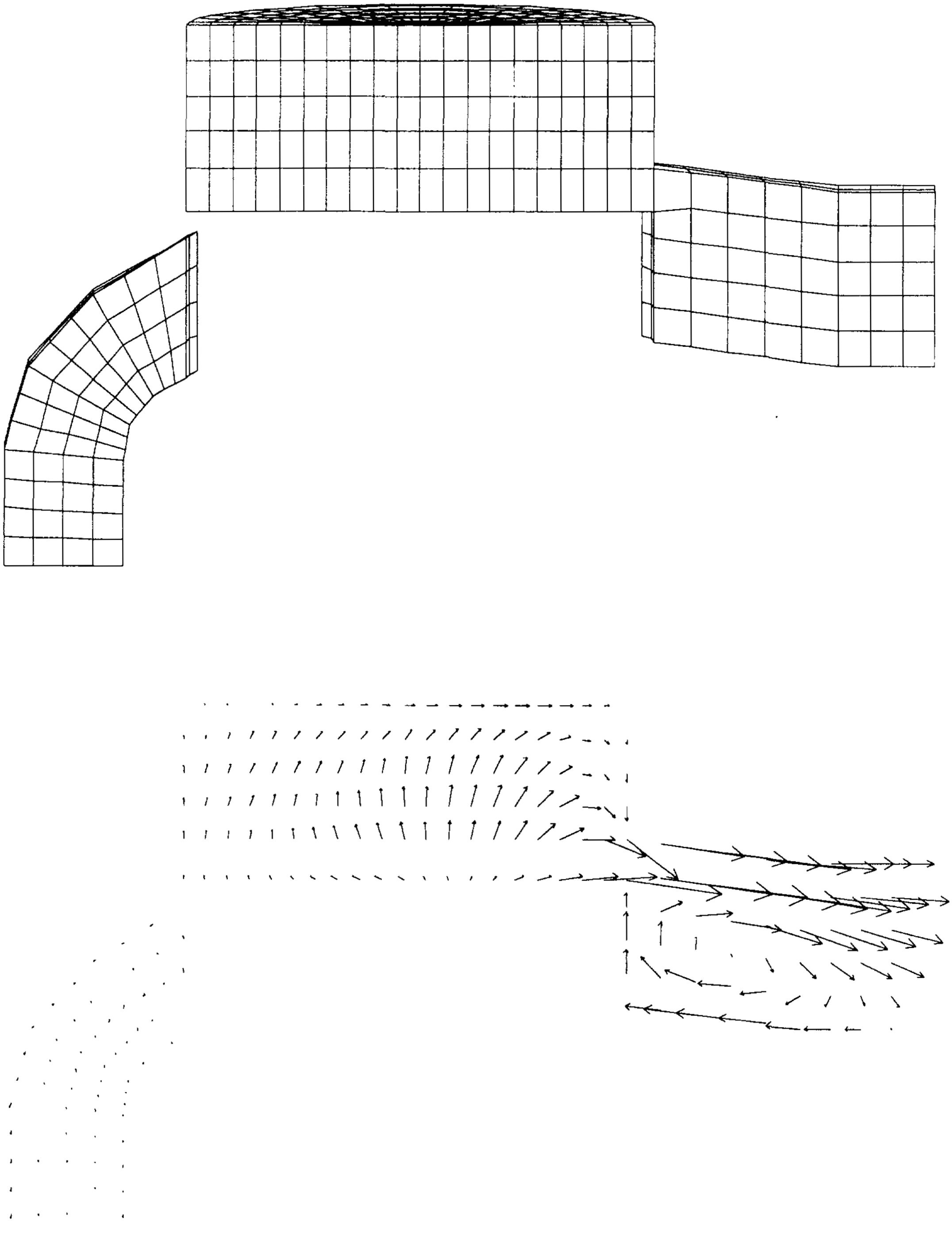


Fig. 19. The 180° teapot mesh (top) and velocity vectors across the symmetry plane (bottom) at crank angle -59.52° ATDC, cycle 241. UMAX = 12946 cm/sec and WMAX = 4553 cm/sec.

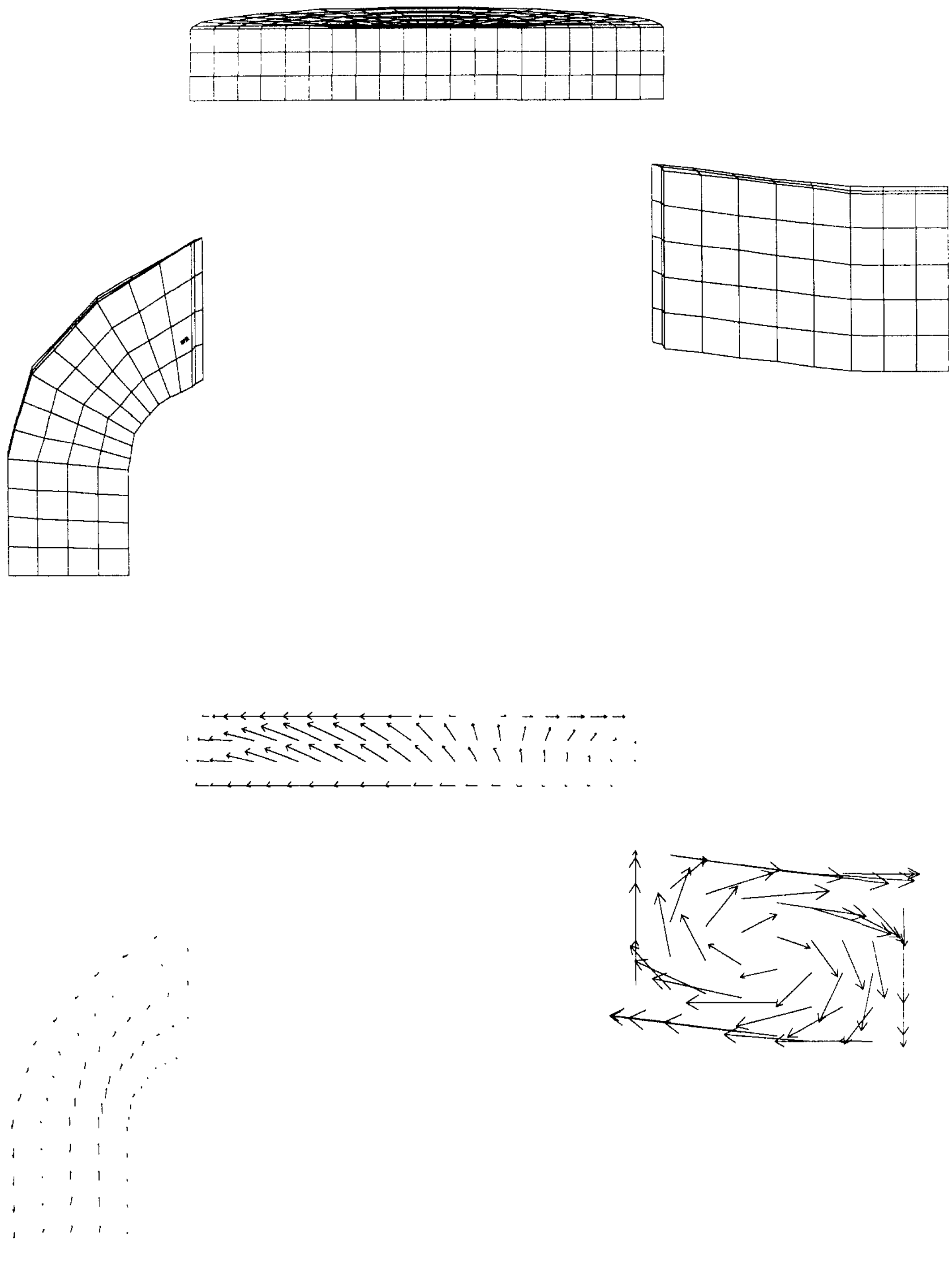


Fig. 20. The 180° teapot mesh (top) and velocity vectors across the symmetry plane (bottom) at crank angle 0.45° ATDC, cycle 368. UMAX = 2584 cm/sec and WMAX = 1599 cm/sec.

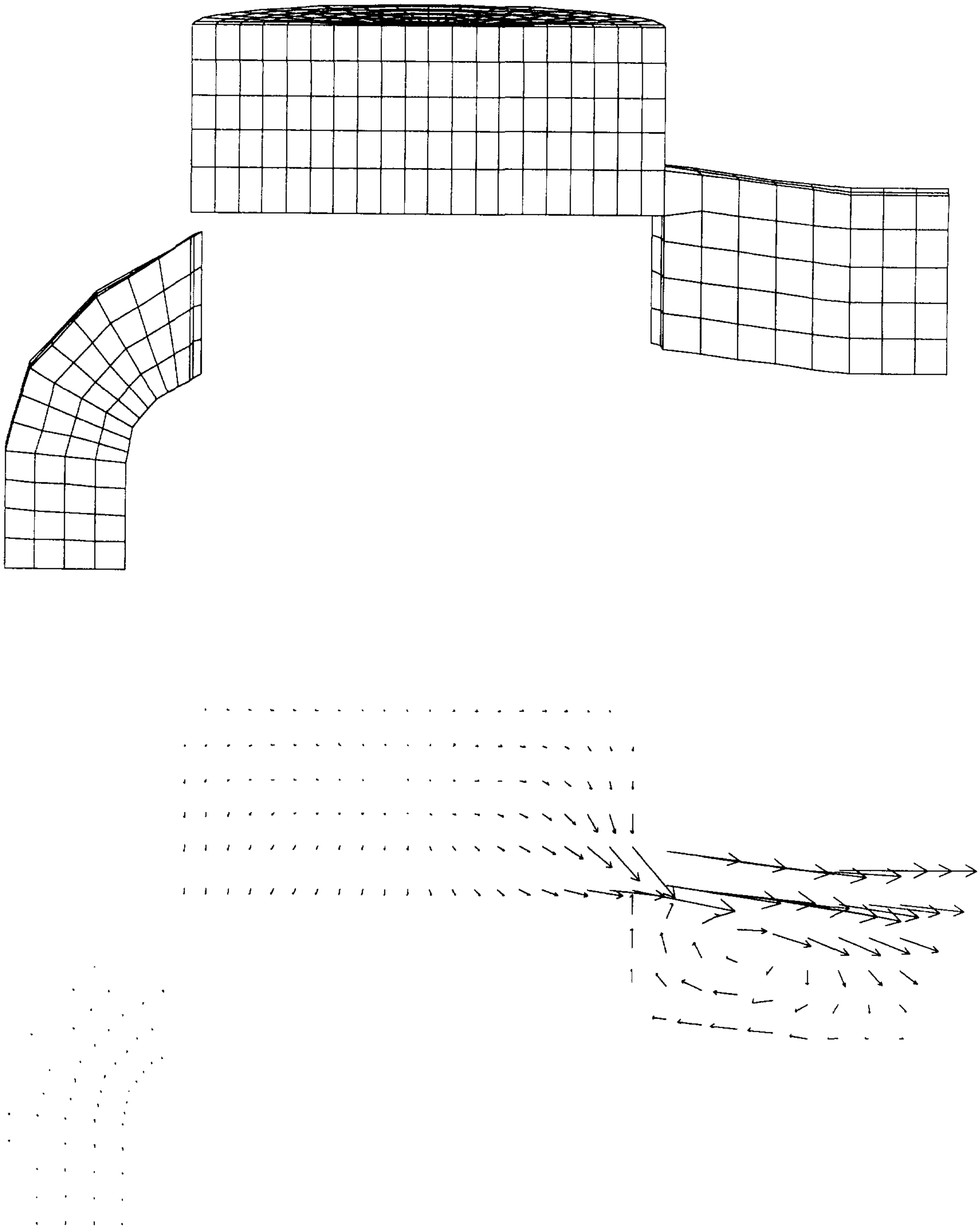


Fig. 21. The 180° teapot mesh (top) and velocity vectors across the symmetry plane (bottom) at crank angle 60.15° ATDC, cycle 499. UMAX = 12589 cm/sec and WMAX = 5763 cm/sec.

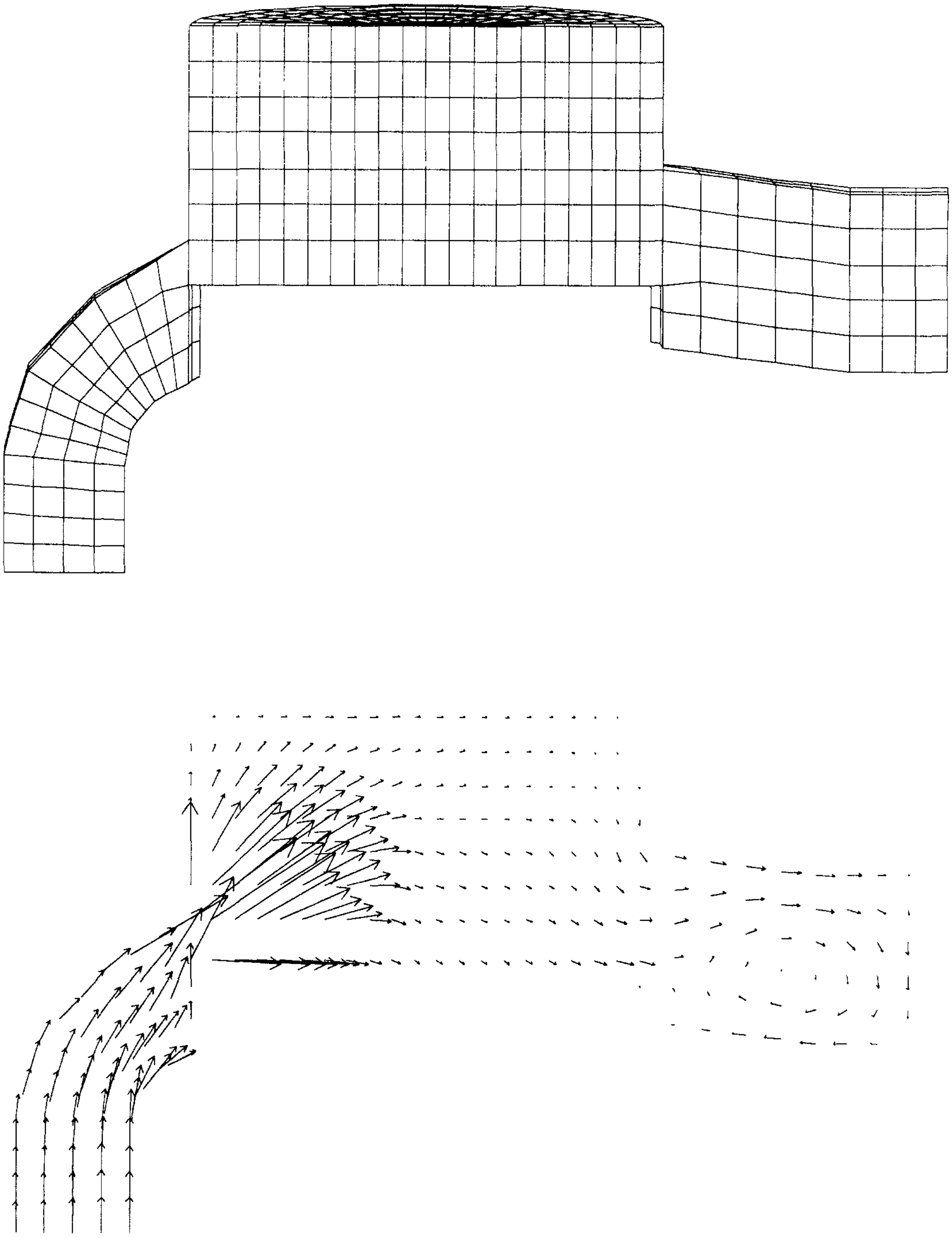


Fig. 22. The 180° teapot mesh (top) and velocity vectors across the symmetry plane (bottom) at crank angle 80.17° ATDC, cycle 554. UMAX = 21233 cm/sec and WMAX = 16184 cm/sec.

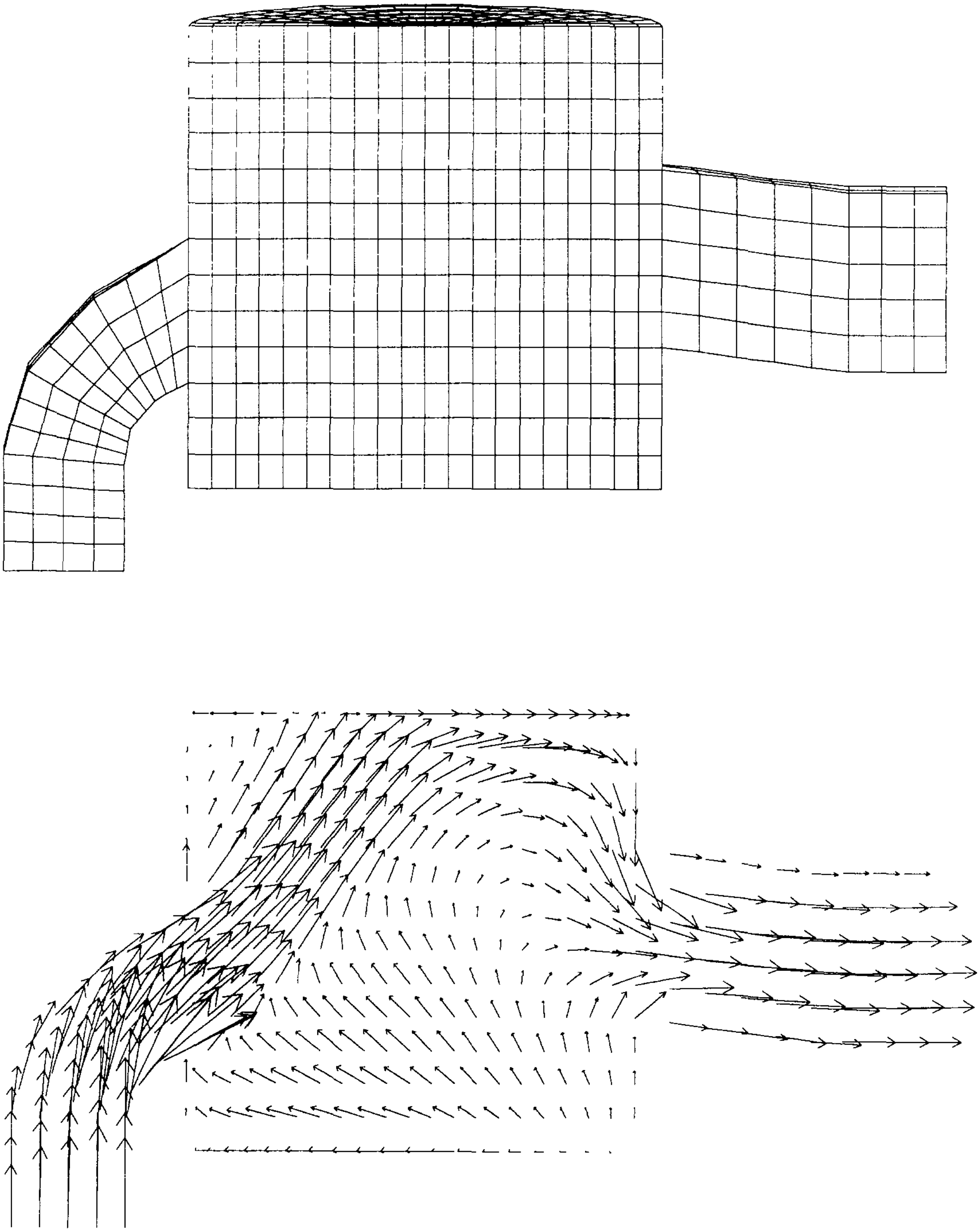


Fig. 23. The 180° teapot mesh (top) and velocity vectors across the symmetry plane (bottom) at crank angle 180.42° ATDC, cycle 829. UMAX =10206 cm/sec and WMAX = 12732 cm/sec.

D. OPPOSED-PISTON GEOMETRY WITH NO AXIS

The final example (Figs. 24-29) illustrates a geometry with two moving pistons. The lower (intake) piston moves past 12 inlet ports, each angled 5^o to induce a clockwise swirl in the cylinder, and the upper (exhaust) piston moves past 8 outlet ports. K3PREP creates this geometry from 21 blocks. The cylinder, which has no central axis, is built from a block of 12 x 12 x 19 cells, has its center at x = y = 0, and a radius of 5.0 cm and a wall thickness of 1 cm. Because the standard treatment in K3PREP is to distribute vertices evenly around the circumference of the cylinder, each of the 48 cell faces on the cylinder wall subtends 7.5^o.

The ports are patched onto the right, derriere, left, and front faces of the cylinder block. The 12 inlets are all identical; each subtends 15^o, with a 15^o separation from the next. The 5^o swirl angle is obtained by displacing the outer vertices $+5^o$ from the inner vertices. For example, the first inlet (block 2), which is patched onto the logical right face of the cylinder, has inner vertices at 322.5^o and 337.5^o, and outer vertices at 327.5^o and 342.5^o.

The 8 outlets are also identical to one another; each subtends 22.5^o, with a 22.5^o separation from the next. Each outlet is 2^o narrower on a side at its pressure boundary end. For example, the first outlet (block 14), which is patched onto the logical right face of the cylinder, has inner vertices at 330^o and 352.5^o, and outer vertices at 332^o and 350.5^o.

The cold flow calculation begins at BDC, when the pistons are at their maximum separation of 19 cm. The pistons are in phase with one another, i.e., DATDCT = 0.0. The inlets close at -115.58^o ATDC, then the exhausts close at -99.22^o ATDC. The exhausts reopen at 99.69^o ATDC, and the inlets reopen at 115.65^o ATDC. A hypothetical pressure history is applied at the inflow and outflow boundaries, and as in the previous example, the tabular pressure data is supplied as part of ITAPE.

The velocity fields plotted in Figs. 25-29 are all remapped onto 2-D overlay planes (see Sec. IV.A.2), and hence the vectors originate at overlay plane vertices rather than at actual KIVA-3 vertices. Interpretation of these plots must take into consideration where the x, y, or z slice is made. In this grid it is impossible to define a single x or y plane that passes through the axis of the cylinder and also cuts through both inlet and outlet ports. In Fig. 25, for example, the y = 0.0 (10^{-3}) cm slice passes through inlets but not outlets, which gives the velocities the appearance of heading towards zero at the top of the mesh, when in reality they are turning normal to the view plane. The y = 2.40 cm slice, on the other hand, cuts through both inlets and outlets and clearly shows the flow entering and leaving the mesh. But because the plot is far off-axis, the projection of the swirl

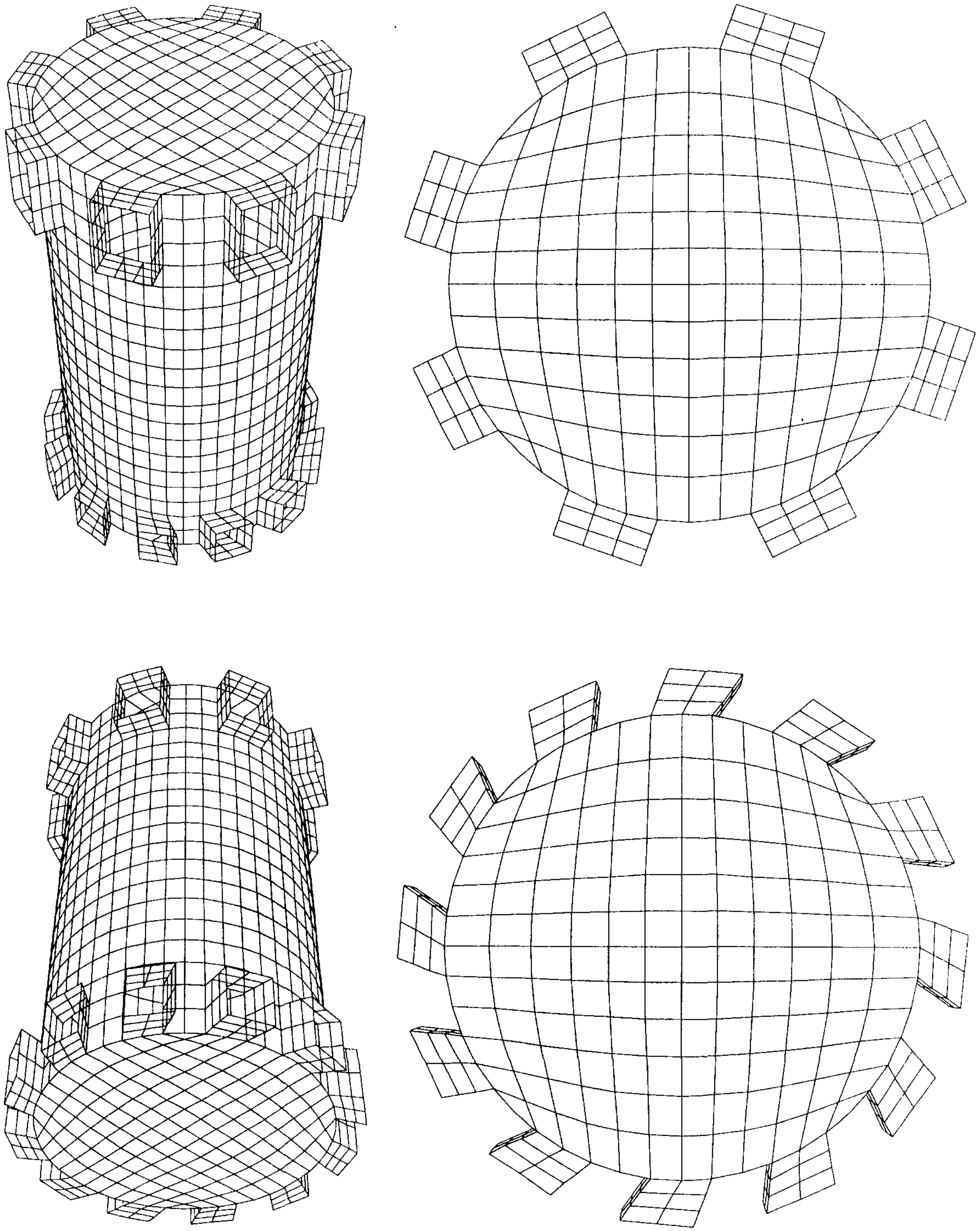


Fig. 24. The computing mesh for the opposed-piston geometry, with 8 exhaust ports and 12 intake ports. Upper left: XYROT = XZROT = 45.0° and Z0 = 4.0. Upper right (plan view): XYROT = 90.0°, XZROT = 0.0°, and Z0 = 5.0. Lower left: XYROT = 45.0°, XZROT = 135.0°, and Z0 = 4.0. Lower right (from below looking straight up): XYROT = 90.0°, XZROT = 180.0°, and Z0 = 5.0.

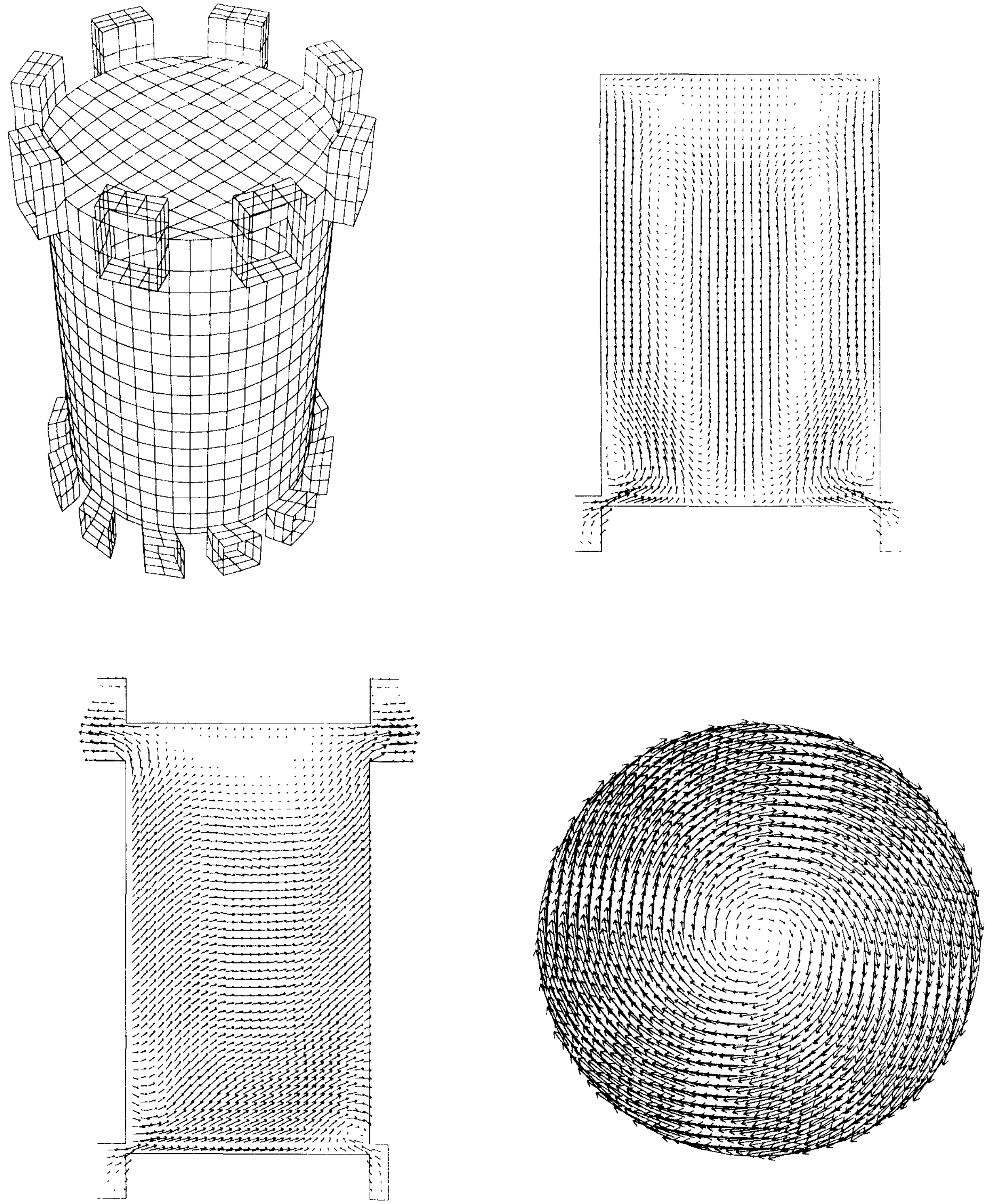


Fig. 25. The opposed-piston mesh and remapped velocity fields at crank angle -119.55° ATDC, cycle 96. Upper right: $y = 10^{-3}$ cm; UMAX = 3645 cm/sec, VMAX = 4421 cm/sec, WMAX = 2699 cm/sec. Lower left: y = 2.40 cm; UMAX = 5815 cm/sec, VMAX = 4844 cm/sec, WMAX = 2846 cm/sec. Lower right: z = 9.50 cm; UMAX = 4058 cm/sec, VMAX = 4059 cm/sec, WMAX = 2354 cm/sec.

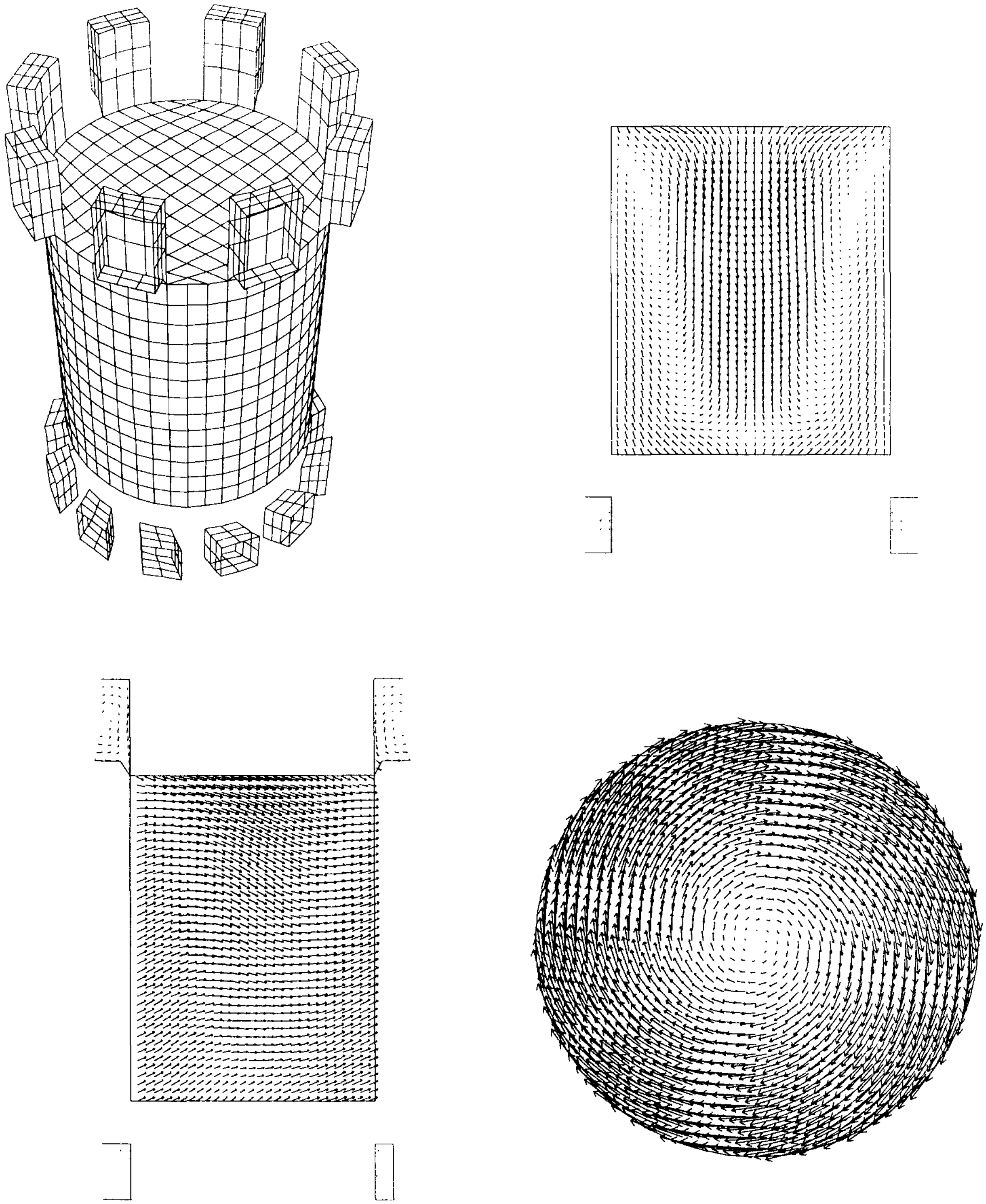


Fig. 26. The opposed-piston mesh and remapped velocity fields at crank angle -89.53° ATDC, cycle 139. Upper right: $y = 10^{-3}$ cm; UMAX = 1106 cm/sec, VMAX = 3596 cm/sec, WMAX = 2340 cm/sec. Lower left: $y = 2.40$ cm; UMAX = 3583 cm/sec, VMAX = 3110 cm/sec, WMAX = 1107 cm/sec. Lower right: $z = 9.50$ cm; UMAX = VMAX = 3435 cm/sec, WMAX = 2224 cm/sec.

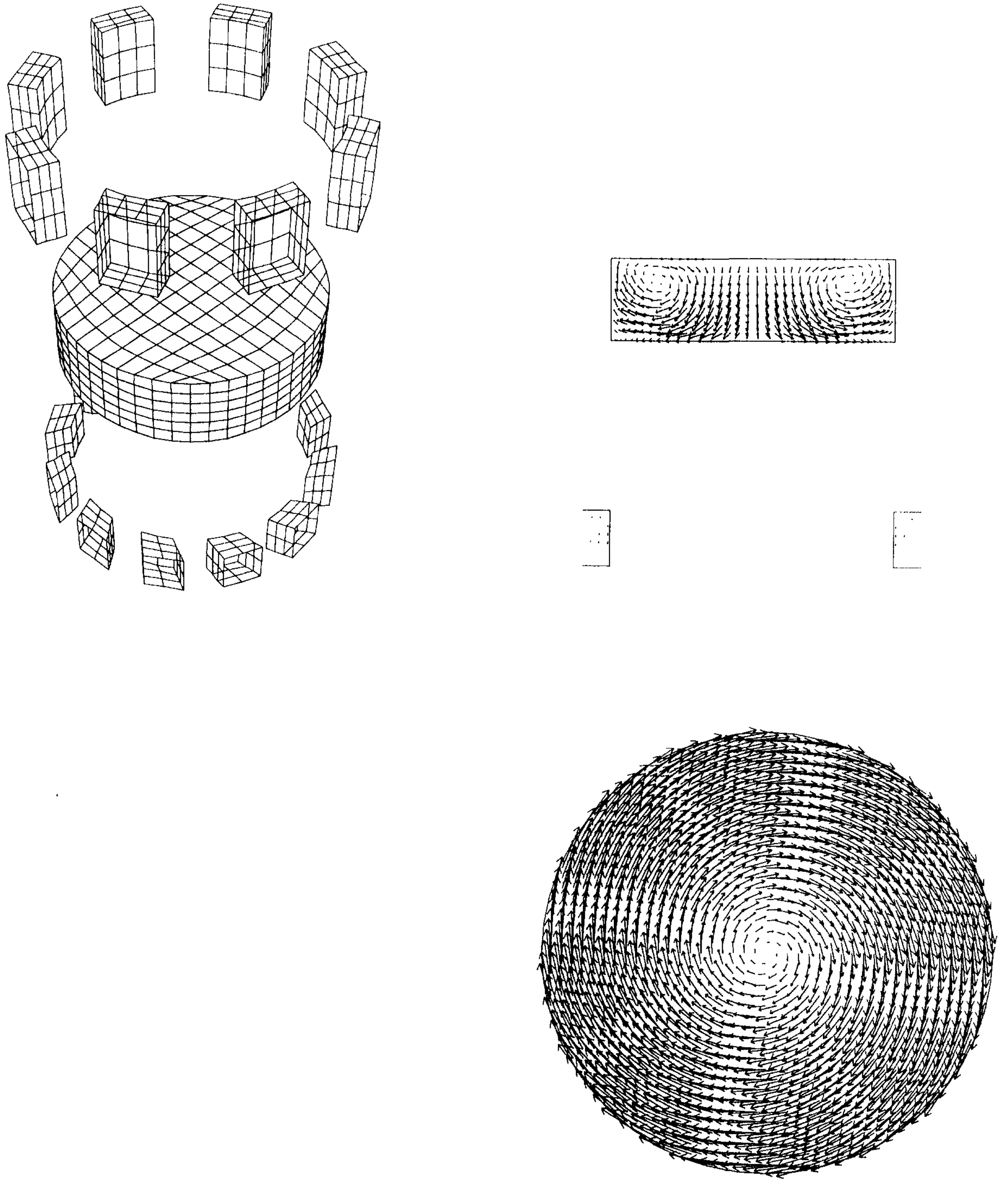


Fig. 27. The opposed-piston mesh and remapped velocity fields at crank angle 0.47° ATDC, cycle 229. Upper right: $y = 10^{-3}$ cm; UMAX = 221 cm/sec, VMAX = 2795 cm/sec, WMAX = 183 cm/sec. Lower right: z = 9.50 cm; UMAX = VMAX = 2798 cm/sec, WMAX = 185 cm/sec.

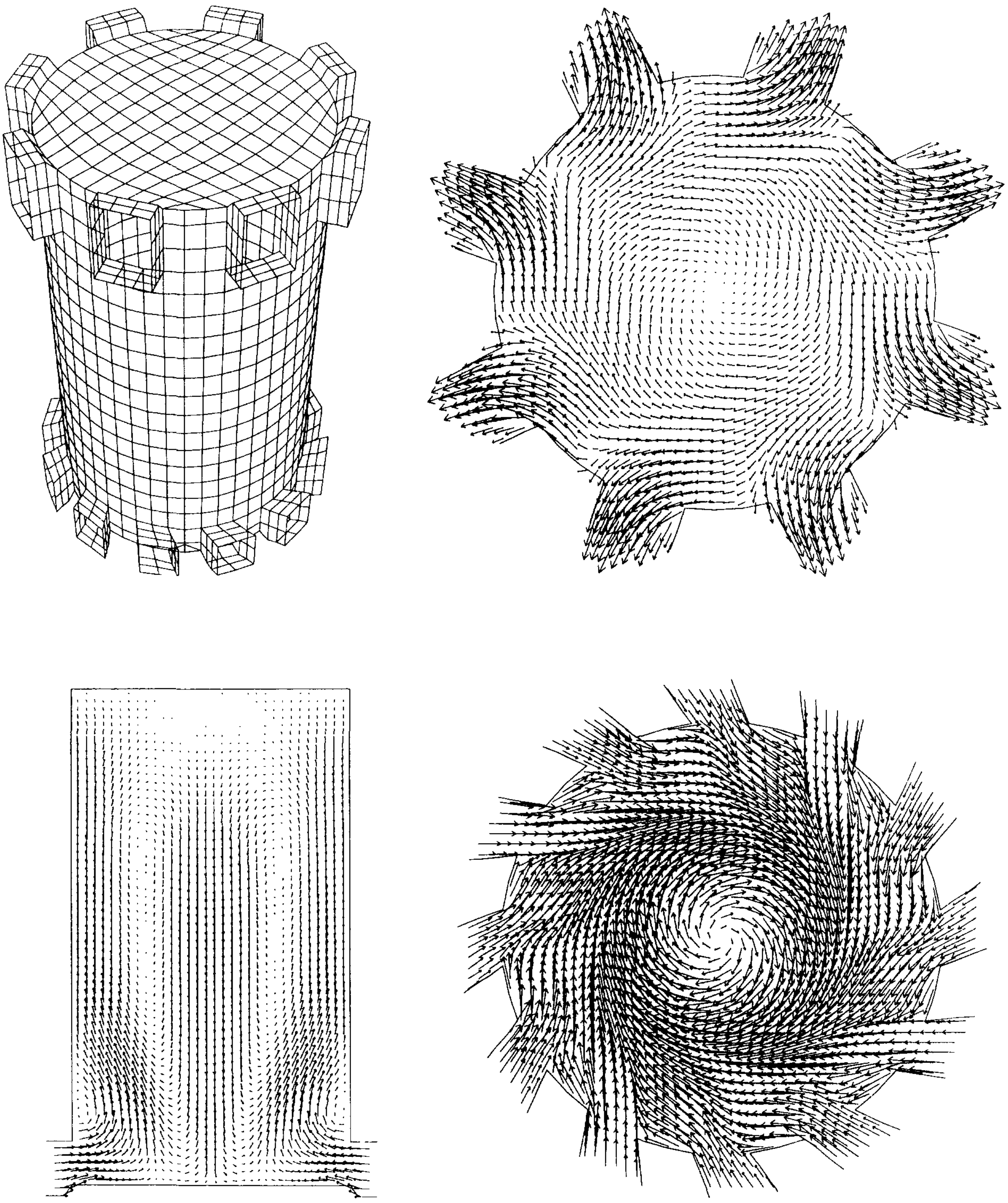


Fig. 28. The opposed-piston mesh and remapped velocity fields at crank angle 150.01° ATDC, cycle 491. Upper right: z = 17.50 cm; UMAX = VMAX = 3476 cm/sec, WMAX = 1754 cm/sec. Lower left: y = 10^{-3} cm; UMAX = 7686 cm/sec, VMAX = 11182 cm/sec, WMAX = 6079 cm/sec. Lower right: z = 1.00 cm; UMAX = 12017 cm/sec, VMAX = 12018 cm/sec, WMAX = 2694 cm/sec.

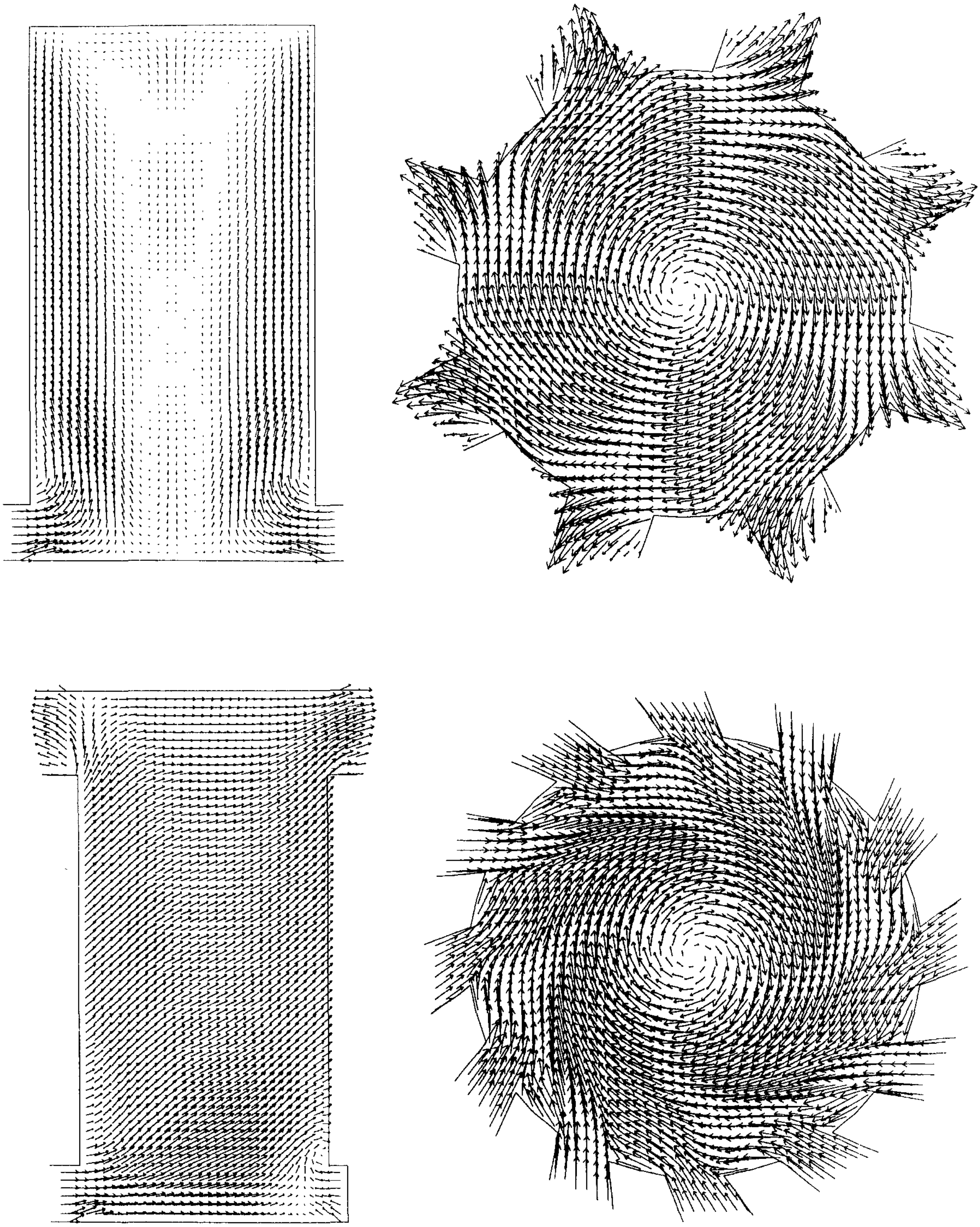


Fig. 29. Remapped velocity fields for the opposed-piston mesh at crank angle 180.10° ATDC, cycle 550. Upper left: $y = 10^{-3}$ cm; UMAX = 4273 cm/sec, VMAX = 7003 cm/sec, WMAX = 3915 cm/sec. Upper right: z = 17.50 cm; UMAX = VMAX = 6189 cm/sec, WMAX = 3085 cm/sec. Lower left: y = 2.40 cm; UMAX = 6799 cm/sec, VMAX = 6542 cm/sec, WMAX = 4422 cm/sec. Lower right: z = 1.00 cm; UMAX = VMAX = 7776 cm/sec, WMAX = 1513 cm/sec.

component of velocity onto the view plane causes the vectors to point to the upper right through much of the cylinder.

K3PREP required 1.9 CPU seconds to generate the mesh, which contains 5644 vertices. Relaxation of the block that formed the cylinder required 346 iterations with a convergence criterion of 10^{-12}. The KIVA-3 run through the full 360^o required 8.00 CPU minutes to run 550 cycles, averaging 0.87 CPU second per cycle. To achieve a repeatable solution, at least two engine revolutions would have to be calculated to wash out the effect of starting with a null velocity field.

A variation on this cold flow calculation could be used to determine such things as the scavenging efficiency, trapping efficiency, and delivery ratio . This calculation would begin at TDC rather than BDC, and would require several patches to KIVA-3. One possible approach would use "old" air and "new" air. Species 6-9 (new air) are identical to species 2-5 (old air), and thus are also defined as O_2, N_2, CO_2, and H_2O. Therefore, the molecular weights MW6-MW9 are the same as MW2-MW5 on ITAPE, and the enthalpy tables HK6-HK9 must be made identical to tables HK2-HK5 in subroutine RINPUT.

Initially in the cylinder (region 1) and the exhaust ports (region 3), and supplied as ambient (SPDAM2-5) is old air. Initially in the inlet ports (region 2) and supplied for the inflow (SPDIN6-9) is new air. Values for SPDAM6-SPDAM9 and SPDIN2-SPDIN5 should all be zero. Subroutine SETUP requires patches to initialize densities (both RO and SPD) and temperatures (TEMP) appropriately for each of the three regions. Because the cylinder is at TDC, the species densities and temperature in region 1 must be elevated to correspond to the estimated TDC cylinder pressure.

KIVA-3 calculates several quantities helpful for calculating efficiencies and the delivery ratio. Written on TAPE12 are FLUXIN (the cumulative mass of delivered air) and TOTM1 (the mass of the trapped cylinder charge). Because subroutine GLOBAL lists species masses by fluid region, the mass of delivered air retained in the above example would be the sum of species masses 6, 7, 8, and 9 in the cylinder. Also written are the cumulative mass flux leaving the system, the cumulative wall heat loss, and the current swirl ratio.

ACKNOWLEDGMENTS

I wish to acknowledge first and foremost my colleagues Peter J. O'Rourke, T. Daniel Butler, and Michael C. Cline for their useful advice during the KIVA-3 development. Peter supplied the algorithm for treating solid boundary vertices as described in Sec. II.G.1, along with a basic outline for the snapper.

Second, I wish to acknowledge my colleagues outside Los Alamos who have taken the pre-released KIVA-3, without benefit of documentation beyond what existed in the code listings, and successfully applied it to their own research. The code bugs they have encountered and the improvements they have suggested have contributed to a better program. Among these users have been Keith Meintjes of General Motors Powertrain Division, along with Ramachandra Diwakar of General Motors Research Laboratories, Tom McKinley and Carol Hoover of Cummins Engine Company, and Randy Hessel at the University of Wisconsin/Madison. Keith also donated the basic subroutines for drawing perspective plots with hidden line removal, and one of his suggestions became the inspiration for the snapper.

Finally, I thank Margaret Findley for creating many of the drawings and assisting in the final production of this report.

REFERENCES

1. F. Grasso, M. J. Wey, F. V. Bracco, and J. Abraham, "Three Dimensional Computations of Flows in a Stratified-Charge Rotary Engine," SAE Paper 870409 (1987).

2. P. J. O'Rourke and A. A. Amsden, "Three-Dimensional Numerical Simulations of the UPS-292 Stratified Charge Engine," SAE Paper 870597 (1987).

3. D. Gilaber and P. Pinchon, "Measurements and Multidimensional Modeling of Gas-Wall Heat Transfer in a S. I. Engine," SAE Paper 880516 (1988).

4. J. D. Naber and R. D. Reitz, "Modeling Engine Spray/Wall Impingement," SAE Paper 880107 (1988).

5. T. W. Kuo and R. D. Reitz, "Computation of Premixed-Charge Combustion in Pancake and Pent-Roof Engines," SAE Paper 890670 (1989).

6. Y. Takenaka, Y. Aoyagi, Y. Tsuji, and I. Joko, "3D Numerical Simulation of Fuel Injection and Combustion Phenomena in DI Diesel Engines," SAE Paper 890668 (1989).

7. R. D. Reitz and T. W. Kuo, "Modeling of HC Emissions Due to Crevice Flows in Premixed-Charge Engines," SAE Paper 892085 (1989).

8. K. Naitoh, H. Fujii, T. Urushihara, Y. Takagi, and K. Kuwahara, "Numerical Simulation of the Detailed Flow in Engine Ports and Cylinders," SAE Paper 900256 (1990).

9. W. K. Cheng and J. A. Diringer, "Numerical Modeling of SI Engine Combustion with a Flame Sheet Model," SAE Paper 910268 (1991).

10. Diwakar *et al*, "Engine and High Turbulence Piston Therefor," U.S. Patent No. 4,955,338 (1990).

11. A. A. Amsden, J. D. Ramshaw, P. J. O'Rourke, and J. K. Dukowicz, "KIVA: A Computer Program for Two- and Three-Dimensional Fluid Flows with Chemical Reactions and Fuel Sprays," Los Alamos National Laboratory report LA-10245-MS (February 1985).

12. A. A. Amsden, J. D. Ramshaw, L. D. Cloutman, and P. J. O'Rourke, "Improvements and Extensions to the KIVA Computer Program," Los Alamos National Laboratory report LA-10534-MS (October 1985).

13. A. A. Amsden, T. D. Butler, and P. J. O'Rourke, "The KIVA-II Computer Program for Transient Multidimensional Chemically Reactive Flows with Sprays," SAE Paper 872072 (1987).

14. A. A. Amsden, P. J. O'Rourke, and T. D. Butler, "KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays," Los Alamos National Laboratory report LA-11560-MS (May 1989).

15. R. Taghavi, "CRI/TurboKiva Delivers the Power of Insight," *Cray Channels* **13**, No. 4, 26–27 (1992).

16. R. Taghavi and A. Dupont, "Multidimensional Flow Modeling," *Cray Channels* **10**, No. 4, 18–21 (1989).

17. M. Zellat, Th. Rolland, and F. Poplow, "Three-Dimensional Modeling of Combustion and Soot Formation in an Indirect Injection Diesel Engine," SAE Paper 900254 (1990).

18. P. Epstein, "A Computer Simulation of the Scavenging Flow in a Two-Stroke Engine," MSME Thesis, University of Wisconsin/Madison (1990).

19. P. H. Epstein, R. D. Reitz, and D. E. Foster, "Computations of a Two-Stroke Engine Cylinder and Port Scavenging Flows," SAE Paper 910672 (1991).

20. T. W. Kuo, "Multidimensional Port-and-Cylinder Gas Flow, Fuel Spray, and Combustion Calculations for a Port-Fuel Injection Engine," SAE Paper 920515 (1992).

21. A. A. Amsden, P. J. O'Rourke, T. D. Butler, K. Meintjes, and T. D. Fansler, "Comparisons of Computed and Measured Three-Dimensional Velocity Fields in a Motored Two-Stroke Engine," SAE Paper 920418 (1992).

22. R. Diwakar, T. D. Fansler, D. T. French, J. B. Ghandhi, C. J. Dasch, and D. M. Heffelfinger, "Liquid and Vapor Fuel Distribution from an Air-Assist Injector -- An Experimental and Computational Study," SAE Paper 920422 (1992).

23. N. B. Vargaftik, *Tables on the Thermophysical Properties of Liquids and Gases* (John Wiley & Sons, New York, 1975).

24. *TRC Thermodynamic Tables* (Thermodynamics Research Center, Texas A & M University, College Station, Texas, 1987).

25. J. B. Maxwell, *Data Book on Hydrocarbons* (D. Van Nostrand Company, Inc., Princeton, New Jersey, 1958).

26. *Technical Data on Fuel* (J. W. Rose and J. R. Cooper, John Wiley & Sons, New York, 1977).

27. R. J. Reed, *Mechanical Engineers' Handbook* (John Wiley & Sons, New York, 1986).

28. *Marks' Standard Handbook for Mechanical Engineers* (T. Baumeister *et al,* McGraw-Hill, New York, 1978).

29. *The Science of Petroleum , Vol. II* (A. E. Dunstan *et al,* Oxford University Press, London, 1938).

30. C. A. Varnavas, "Studies of Diesel Engine Combustion Using a Multidimensional Computer Simulation," MSME Thesis, University of Illinois at Urbana-Champaign (1990).

31. W. R. Leppard, General Motors Research Laboratories, Fuels and Lubricants Department, personal communication (1992).

32. C. R. Ferguson, *Internal Combustion Engines* (John Wiley & Sons, New York, 1986).

33. C. F. Taylor, *The Internal Combustion Engine in Theory and Practice, Vol. 1: Thermodynamics, Fluid Flow, Performance* (MIT Press, Cambridge, Massachusetts, 1980).

34. C. K. Westbrook and F. L. Dryer, "Chemical Kinetic Modeling of Hydrocarbon Combustion," *Prog. Energy Combust. Sci. 1984* **10**, 1–57 (1984).

35. K. Meintjes, "Hidden-Line Removal for Viewing Three-Dimensional Surfaces and Objects," General Motors Research Laboratories, Research Publication GMR-5343 (1988).

APPENDIX

INPUT FILES FOR EXAMPLE PROBLEMS

BASELINE: 2-D DISC ENGINE, File IPREP for K3PREP:

```
T3AAA K111292 KIVA-3 3-BLOCK BASELINE, 1/2 DEG, ROUNDED CUP 111892
 BORE        9.843
 STROKE      9.55
 SQUISH    0.181934
 THSECT      0.5
NBLOCKS      3
  13    1   12   39    1    1
 3.0       3.0       0.0       0.0       3.0       3.0       0.0       0.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
 0.0       0.0       0.0       0.0       3.3       3.3       3.3       3.3
 3.0       1.0       5.0       6.0       1.0       4.0
 0.0       0.0
 0.1       0.0
 0.2       0.0
 0.3       0.0
 0.45      0.0
 0.60      0.0
 0.75      0.0
 0.90      0.0
 1.05      0.0
 1.20      0.0
 1.35      0.0
 1.50      0.025
 1.65      0.07
 1.80      0.125
 1.98      0.275
 2.12      0.45
 2.20      0.675
 2.25      0.9
 2.25      1.2
 2.25      1.5
 2.25      1.8
 2.40      2.1
 2.55      2.4
 2.70      2.7
 2.85      3.0
 3.00      3.3
 2.769     3.3
 2.538     3.3
 2.308     3.3
 2.077     3.3
 1.846     3.3
 1.615     3.3
 1.385     3.3
 1.154     3.3
 0.923     3.3
 0.692     3.3
 0.462     3.3
 0.231     3.3
 0.0       3.3
 13     1   10    0    2    1
 3.0       3.0       0.0       0.0       3.0       3.0       0.0       0.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
 3.3       3.3       3.3       3.3       3.3       3.3       3.3       3.3
 3.0       4.0       5.0       6.0       4.0       2.0
  7     1   10    0    2    1
 4.9215    4.9215    3.0       3.0       4.9215    4.9215    3.0       3.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
 3.3       3.3       3.3       3.3       3.3       3.3       3.3       3.3
 4.0       2.0       5.0       6.0       1.0       2.0
 NPATCH      2
  2     5    1    1    1    2
  3     1    2    1    1    3
```

BASELINE: 2-D DISC ENGINE, File ITAPE for KIVA-3:

```
T3AAA K111292 KIVA-3 1/2-DEGREE BASELINE, GASOLINE, 111892
 IREST      0
 LWALL      1
 LPR        0
 IREZ       2
 NCFILM 9999
 NCTAP8 9999
 NCLAST 9999
 CAFILM   30.0
 CAFIN     0.0
 ANGMOM     1.0
 PGSSW      1.0
 SAMPL     0.0
 DTI    1.04167E-4
 DTMXCA     1.0
 DTMAX    9.99E+9
 TLIMD      1.0
 TWFILM   9.99E+9
 TWFIN    9.99E+9
 FCHSP     0.25
 BORE      9.843
 STROKE    9.55
 SQUISH  0.181934
 RPM        1.6E+3
 ATDC    -90.0
 DATDCT    0.0
 CONROD   16.269
 SWIRL     3.0
 SWIPRO    3.11
 THSECT    0.5
 SECTOR     1.0
 EPSY       1.0E-3
 EPSV       1.0E-3
 EPSP       1.0E-4
 EPST       1.0E-3
 EPSK       1.0E-3
 EPSE       1.0E-3
 GX        0.0
 GY        0.0
 GZ        0.0
 TCYLWL  400.0
 THEAD   400.0
 TPISTN  400.0
 TVALVE  400.0
 TEMPI   400.0
 PARDON    0.0
 AO        0.0
 BO         1.0
 ANC4      0.05
 ADIA      0.0
 ANUO      0.0
 VISRAT-.66666667
 TCUT    800.0
 TCUTE  1200.0
 EPSCHM    0.02
 OMGCHM     1.0
 TKEI      0.10
 TKESW      1.0
 SGSL      0.0
UNISCAL    0.0
 AIRMU1  1.457E-5
 AIRMU2  110.0
 AIRLA1  252.0
 AIRLA2  200.0
 PRL       0.74
 RPR        1.11
 RPRQ       1.0
 RPRE    0.769231
 RSC        1.11
 XIGNIT     1.0E+4
```

```
  T1IGN      -1.0
  TDIGN      -1.0
  CA1IGN   -27.0
  CADIGN     9.6
  XIGNL1     0.25
  XIGNR1     0.75
  YIGNF1     0.0
  YIGND1     0.238
  ZIGNB1    11.75
  ZIGNT1    12.50
  XIGNL2     0.0
  XIGNR2     0.0
  YIGNF2     0.0
  YIGND2     0.0
  ZIGNB2     0.0
  ZIGNT2     0.0
  KWIKEQ     0
  NUMNOZ      1
  NUMVEL      1
 INJDIST      1
  KOLIDE      1
  T1INJ      -1.0
  TDINJ      -1.0
  CA1INJ   -52.0
  CADINJ    12.672
  TSPMAS     0.0116
  PULSE       1.0
  TNPARC   400.0
  TPI      350.0
  TURB        1.0
 BREAKUP      1.0
  EVAPP       1.0
  DRNOZ      0.0
  DZNOZ     -0.01
  DTHNOZ     0.0
  TILTXY     0.0
  TILTXZ     0.0
  CONE      62.5
  DCONE     12.5
  ANOZ-       1.0
  SMR       5.00E-4
  AMPO       0.0
       4000.0
  NSP       12
GASOLINE    RHO1        0.0
       O2   RHO2    3.1096E-4  MW2      32.000   HTF2      0.0
       N2   RHO3    1.0809E-3  MW3      28.016   HTF3      0.0
      CO2   RHO4    1.4007E-5  MW4      44.011   HTF4    -93.965
      H2O   RHO5    7.0030E-6  MW5      18.016   HTF5    -57.103
        H   RHO6       0.0     MW6       1.008   HTF6     51.631
       H2   RHO7       0.0     MW7       2.016   HTF7      0.0
        O   RHO8       0.0     MW8      16.000   HTF8     58.989
        N   RHO9       0.0     MW9      14.008   HTF9    112.520
       OH   RHO10      0.0     MW10     17.008   HTF10     9.289
       CO   RHO11      0.0     MW11     28.011   HTF11   -27.200
       NO   RHO12      0.0     MW12     30.008   HTF12    21.456
  NRK        4
 CF1    8.0000E10  EF1   1.5780E+4  ZF1       0.0
 CB1       0.0     EB1      0.0     ZB1       0.0
 AM1      4    49      0     0     0     0     0     0     0     0     0     0
 BM1      0     0      0    32    34     0     0     0     0     0     0     0
 AE1     0.250    1.500    0.000    0.000    0.000    0.000    0.000    0.000
         0.000    0.000    0.000    0.000
 BE1     0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
         0.000    0.000    0.000    0.000
 CF2    1.5587E14  EF2   6.7627E+4  ZF2       0.0
 CB2    7.5000E12  EB2      0.0     ZB2       0.0
 AM2      0     1      2     0     0     0     0     0     0     0     0     0
 BM2      0     0      0     0     0     0     0     0     2     0     0     2
 AE2     0.000    0.500    1.000    0.000    0.000    0.000    0.000    0.000
         0.000    0.000    0.000    0.000
 BE2     0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
         1.000    0.000    0.000    1.000
 CF3    2.6484E10  EF3   5.9418E+4  ZF3       1.0
 CB3    1.6000E+9  EB3   1.9678E+4  ZB3       1.0
```

```
146  AM3     0      2      1      0      0      0      0      0      0      0      0      0
147  BM3     0      0      0      0      0      0      0      2      0      0      0      2
148  AE3    0.000   1.000  0.500   0.000   0.000   0.000   0.000   0.000
149         0.000   0.000  0.000   0.000
150  BE3    0.000   0.000  0.000   0.000   0.000   0.000   0.000   1.000
151         0.000   0.000  0.000   1.000
152  CF4   2.1230E14 EF4   5.7020E+4 ZF4      0.0
153  CB4     0.0     EB4     0.0     ZB4      0.0
154  AM4     0      0      1      0      0      0      0      0      0      2      0      0
155  BM4     0      0      0      0      0      2      0      0      0      0      0      2
156  AE4    0.000   0.000  0.500   0.000   0.000   0.000   0.000   0.000
157         0.000   1.000  0.000   0.000
158  BE4    0.000   0.000  0.000   0.000   0.000   1.000   0.000   0.000
159         0.000   0.000  0.000   1.000
160   NRE        6
161  AS1   0.990207  BS1   -51.7916  CS1   0.993074  DS1  -0.343428  ES1   0.0111668
162  AN1     0      0      0      0      0      0      1      0      0      0      0      0
163  BN1     0      0      0      0      0      2      0      0      0      0      0      0
164  AS2   0.431310  BS2   -59.6554  CS2   3.503350  DS2  -0.340016  ES2   0.0158715
165  AN2     0      1      0      0      0      0      0      0      0      0      0      0
166  BN2     0      0      0      0      0      0      0      2      0      0      0      0
167  AS3   0.794709  BS3  -113.2080  CS3   3.168370  DS3  -0.443814  ES3   0.0269699
168  AN3     0      0      1      0      0      0      0      0      0      0      0      0
169  BN3     0      0      0      0      0      0      0      0      2      0      0      0
170  AS4  -0.652939  BS4    -9.8232  CS4   3.930330  DS4   0.163490  ES4  -0.0142865
171  AN4     0      1      0      0      0      0      1      0      0      0      0      0
172  BN4     0      0      0      0      0      0      0      0      0      2      0      0
173  AS5   1.158882  BS5   -76.8472  CS5   8.532155  DS5  -0.868320  ES5   0.0463471
174  AN5     0      1      0      0      2      0      0      0      0      0      0      0
175  BN5     0      0      0      0      0      0      0      0      0      4      0      0
176  AS6   0.980875  BS6    68.4453  CS6  -10.5938   DS6   0.574260  ES6  -0.0414570
177  AN6     0      1      0      0      0      0      0      0      0      0      2      0
178  BN6     0      0      0      2      0      0      0      0      0      0      0      0
```

DOMED HEAD GEOMETRY, File IPREP for K3PREP:

```
T3AAA KIVA-3 DOMED HEAD EXAMPLE CALCULATION, 111892
 BORE       10.0
 STROKE      9.55
 SQUISH      0.955
 THSECT      0.5
NBLOCKS      2
   10    1   16   11    2    1
 5.0       5.0       0.0       0.0       5.0       5.0       0.0       0.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
 3.0       2.0       5.0       6.0       1.0       2.0
 0.0       0.86
 0.5       0.84
 1.0       0.81
 1.5       0.77
 2.0       0.71
 2.5       0.64
 3.0       0.54
 3.5       0.42
 4.0       0.29
 4.5       0.16
 5.0       0.0
    5    1    6   17    3    1
 2.5       2.5       0.0       0.0       2.5       2.5       0.0       0.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
 0.86      0.86      0.86      0.86      3.64      3.64      3.64      3.64
 3.0       2.0       5.0       6.0       4.0       2.0
 0.0       0.86
 0.50      0.84
 1.00      0.81
 1.50      0.77
 2.00      0.71
 2.50      0.64
 2.48      1.14
 2.41      1.54
 2.31      2.04
 2.20      2.39
 2.04      2.73
 1.82      2.99
 1.58      3.19
 1.30      3.34
 0.90      3.44
 0.50      3.54
 0.00      3.64
 NPATCH      1
    2    5    1    1    1    1
```

TEAPOT: TWO-STROKE ENGINE, File IPREP for K3PREP:

```
T3AAA K111292 180-DEGREE TEAPOT MESH SAMPLE CALCULATION, 111892
 BORE      9.843
 STROKE    8.25
 SQUISH    1.50
 THSECT  180.0
NBLOCKS    7
  10  20  13   0   2   1
 4.9215  4.9215  0.0     0.0     4.9215  4.9215  0.0     0.0
 0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 0.0     0.0     0.0     0.0     9.7500  9.7500  9.7500  9.7500
 3.0     2.0     2.0     2.0     1.0     2.0
   3   2   4   0   4   2
-5.6250 -5.6250 -4.9215 -4.6806 -6.8750 -6.8750 -4.9215 -4.6806
 1.5209  0.0000  0.0000  1.5209  1.5209  0.0000  0.0000  1.5209
 1.8750  1.8750  2.2500  2.2500  4.0000  4.0000  5.2500  5.2500
 4.0     4.0     2.0     2.0     2.0     2.0
   3   2   4   0   4   2
-6.1250 -6.1250 -5.6250 -5.6250 -8.2500 -8.2500 -6.8750 -6.8750
 1.5209  0.0000  0.0000  1.5209  1.5209  0.0000  0.0000  1.5209
 1.2500  1.2500  1.8750  1.8750  2.5000  2.5000  4.0000  4.0000
 4.0     4.0     2.0     2.0     2.0     2.0
   3   2   4   0   4   2
-6.2500 -6.2500 -6.1250 -6.1250 -8.7500 -8.7500 -8.2500 -8.2500
 1.5209  0.0000  0.0000  1.5209  1.5209  0.0000  0.0000  1.5209
 0.5000  0.5000  1.2500  1.2500  0.7500  0.7500  2.5000  2.5000
 4.0     4.0     2.0     2.0     2.0     2.0
   4   2   4   0   4   2
-6.2500 -6.2500 -6.2500 -6.2500 -8.7500 -8.7500 -8.7500 -8.7500
 1.5209  0.0000  0.0000  1.5209  1.5209  0.0000  0.0000  1.5209
-1.7500 -1.7500  0.5000  0.5000 -1.7500 -1.7500  0.7500  0.7500
 4.0     9.0     2.0     2.0     2.0     2.0
   5   2   5   0   4   3
 8.7500  8.7500  4.6806  4.9215  8.7500  8.7500  4.6806  4.9215
 0.0000  1.5209  1.5209  0.0000  0.0000  1.5209  1.5209  0.0000
 2.5000  2.5000  3.0000  3.0000  6.2500  6.2500  6.7500  6.7500
 4.0     4.0     2.0     2.0     2.0     2.0
   3   2   5   0   4   3
10.7500 10.7500  8.7500  8.7500 10.7500 10.7500  8.7500  8.7500
 0.0000  1.5209  1.5209  0.0000  0.0000  1.5209  1.5209  0.0000
 2.5000  2.5000  2.5000  2.5000  6.2500  6.2500  6.2500  6.2500
 4.0    10.0     2.0     2.0     2.0     2.0
 NPATCH    6
   6   1   1   1   5   1
   7   1   6   1   1   7
   2   1   1  19   4   1
   3   1   2   1   1   3
   4   1   3   1   1   4
   5   1   4   1   1   5
```

TEAPOT: TWO-STROKE ENGINE, File ITAPE for KIVA-3:

```
T3AAA K111292 KIVA-3, MOVING PISTON IN 180-DEG. TEAPOT MESH 111892
 IREST     0
 LWALL     1
 LPR       0
 IREZ      2
 NCFILM 9999
 NCTAP8  100
 NCLAST 9999
 CAFILM   20.0
 CAFIN  +180.0
 ANGMOM    0.0
 PGSSW     0.0
 SAMPL     0.0
 DTI       5.0E-5
 DTMXCA    0.5
 DTMAX    9.99E+9
 TLIMD     1.0
 TWFILM   9.99E+9
 TWFIN    9.99E+9
 FCHSP     0.25
 BORE      9.843
 STROKE    8.25
 SQUISH    1.5
 RPM       1.6E+3
 ATDC   -180.0
 DATDCT    0.0
 CONROD   16.269
 SWIRL     0.0
 SWIPRO    3.11
 THSECT  180.0
 SECTOR    0.0
 EPSY      1.0E-3
 EPSV      1.0E-3
 EPSP      1.0E-4
 EPST      1.0E-3
 EPSK      1.0E-3
 EPSE      1.0E-3
 GX        0.0
 GY        0.0
 GZ        0.0
 TCYLWL  300.0
 THEAD   300.0
 TPISTN  300.0
 TVALVE  300.0
 TEMPI   300.0
 PARDON    0.0
 AO        1.0
 BO        0.0
 ANC4      0.05
 ADIA      0.0
 ANUO      0.0
 VISRAT-.66666667
 TCUT    800.0
 TCUTE  1200.0
 EPSCHM    0.02
 OMGCHM    1.0
 TKEI      0.10
 TKESW     1.0
 SGSL      0.0
UNISCAL    0.0
 AIRMU1  1.457E-5
 AIRMU2  110.0
 AIRLA1  252.0
 AIRLA2  200.0
 PRL       0.74
 RPR       1.11
 RPRQ      1.0
 RPRE    0.769231
 RSC       1.11
 XIGNIT    1.5E+3
 T1IGN    9.99E+9
 72   TDIGN     9.99E+9
 73   CA1IGN  -27.0
 74   CADIGN    9.6
 75   XIGNL1    0.1
 76   XIGNR1    0.45
 77   YIGNF1    0.0
 78   YIGND1    0.0
 79   ZIGNB1   11.75
 80   ZIGNT1   12.50
 81   XIGNL2    0.0
 82   XIGNR2    0.0
 83   YIGNF2    0.0
 84   YIGND2    0.0
 85   ZIGNB2    0.0
 86   ZIGNT2    0.0
 87   KWIKEQ    0
 88   NUMNOZ    1
 89   NUMVEL    1
 90  INJDIST    1
 91   KOLIDE    1
 92   T1INJ    9.99E+9
 93   TDINJ    9.99E+9
 94   CA1INJ  -52.0
 95   CADINJ   12.672
 96   TSPMAS    0.0116
 97   PULSE     1.0
 98   TNPARC  400.0
 99   TPI     350.0
100   TURB      1.0
101  BREAKUP    1.0
102   EVAPP     1.0
103   DRNOZ     0.0
104   DZNOZ    -0.01
105   DTHNOZ    0.0
106   TILTXY    0.0
107   TILTXZ    0.0
108   CONE     62.5
109   DCONE    12.5
110   ANOZ_     1.0
111   SMR      5.00E-4
112   AMPO      0.0
113       4000.0
114   NSP      12
115    C8H18   RHO1      0.0
116       O2   RHO2  2.9100E-4 MW2      32.000  HTF2      0.0
117       N2   RHO3  1.0115E-3 MW3      28.016  HTF3      0.0
118      CO2   RHO4  1.3108E-5 MW4      44.011  HTF4    -93.965
119      H20   RHO5  6.5535E-6 MW5      18.016  HTF5    -57.103
120        H   RHO6      0.0   MW6       1.008  HTF6     51.631
121       H2   RHO7      0.0   MW7       2.016  HTF7      0.0
122        O   RHO8      0.0   MW8      16.000  HTF8     58.989
123        N   RHO9      0.0   MW9      14.008  HTF9    112.520
124       OH   RHO10     0.0   MW10     17.008  HTF10     9.289
125       CO   RHO11     0.0   MW11     28.011  HTF11   -27.200
126       NO   RHO12     0.0   MW12     30.008  HTF12    21.456
127   NRK          4
128  CF1    8.0000E10 EF1    1.5780E+4 ZF1       0.0
129  CB1       0.0    EB1       0.0    ZB1       0.0
130  AM1      2    25     0     0     0     0     0     0     0     0     0     0
131  BM1      0     0     0    16    18     0     0     0     0     0     0     0
132  AE1     0.250   1.500   0.000   0.000   0.000   0.000   0.000   0.000
133          0.000   0.000   0.000   0.000
134  BE1     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
135          0.000   0.000   0.000   0.000
136  CF2    1.5587E14 EF2    6.7627E+4 ZF2       0.0
137  CB2    7.5000E12 EB2       0.0    ZB2       0.0
138  AM2      0     1     2     0     0     0     0     0     0     0     0     0
139  BM2      0     0     0     0     0     0     0     0     2     0     0     2
140  AE2     0.000   0.500   1.000   0.000   0.000   0.000   0.000   0.000
141          0.000   0.000   0.000   0.000
142  BE2     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
143          1.000   0.000   0.000   1.000
144  CF3    2.6484E10 EF3    5.9418E+4 ZF3       1.0
145  CB3    1.6000E+9 EB3    1.9678E+4 ZB3       1.0
```

```
146  AM3       0     2     1     0     0     0     0     0     0     0     0     0
147  BM3       0     0     0     0     0     0     0     2     0     0     0     2
148  AE3     0.000    1.000   0.500   0.000    0.000   0.000   0.000   0.000
149          0.000    0.000   0.000   0.000
150  BE3     0.000    0.000   0.000   0.000    0.000   0.000   0.000   1.000
151          0.000    0.000   0.000   1.000
152  CF4   2.1230E14  EF4   5.7020E+4 ZF4       0.0
153  CB4      0.0      EB4     0.0     ZB4       0.0
154  AM4       0     0     1     0     0     0     0     0     0     2     0     0
155  BM4       0     0     0     0     0     2     0     0     0     0     0     2
156  AE4     0.000    0.000   0.500   0.000    0.000   0.000   0.000   0.000
157          0.000    1.000   0.000   0.000
158  BE4     0.000    0.000   0.000   0.000    0.000   1.000   0.000   0.000
159          0.000    0.000   0.000   1.000
160   NRE        6
161  AS1   0.990207   BS1   -51.7916  CS1   0.993074  DS1  -0.343428  ES1    0.0111668
162  AN1       0     0     0     0     0     0     1     0     0     0     0     0
163  BN1       0     0     0     0     0     2     0     0     0     0     0     0
164  AS2   0.431310   BS2   -59.6554  CS2   3.503350  DS2  -0.340016  ES2    0.0158715
165  AN2       0     1     0     0     0     0     0     0     0     0     0     0
166  BN2       0     0     0     0     0     0     0     2     0     0     0     0
167  AS3   0.794709   BS3  -113.2080  CS3   3.168370  DS3  -0.443814  ES3    0.0269699
168  AN3       0     0     1     0     0     0     0     0     0     0     0     0
169  BN3       0     0     0     0     0     0     0     0     2     0     0     0
170  AS4  -0.652939   BS4    -9.8232  CS4   3.930330  DS4   0.163490  ES4   -0.0142865
171  AN4       0     1     0     0     0     0     1     0     0     0     0     0
172  BN4       0     0     0     0     0     0     0     0     0     2     0     0
173  AS5   1.158882   BS5   -76.8472  CS5   8.532155  DS5  -0.868320  ES5    0.0463471
174  AN5       0     1     0     0     2     0     0     0     0     0     0     0
175  BN5       0     0     0     0     0     0     0     0     0     4     0     0
176  AS6   0.980875   BS6    68.4453  CS6  -10.5938   DS6   0.574260  ES6   -0.0414570
177  AN6       0     1     0     0     0     0     0     0     0     0     2     0
178  BN6       0     0     0     2     0     0     0     0     0     0     0     0
179  DISTAMB     2.0
180   PAMB     1.1439E+6
181   TKEAMB    3600.0
182   SCLAMB      1.0
183  SPDAM1   0.0000E+0
184  SPDAM2   2.9100E-4
185  SPDAM3   1.0115E-3
186  SPDAM4   1.3108E-5
187  SPDAM5   6.5535E-6
188  SPDAM6       0.0
189  SPDAM7       0.0
190  SPDAM8       0.0
191  SPDAM9       0.0
192  SPDAM10      0.0
193  SPDAM11      0.0
194  SPDAM12      0.0
195   VELIN       0.0
196   REEDIN      0.0
197  REEDOUT      1.0
198   NUMPCC      4
199       0.0     1.08E+6
200      70.0     1.30E+6
201     110.0     1.24E+6
202     360.0     1.08E+6
203   NUMPEX      8
204       0.0     1.10E+6
205      50.0     1.00E+6
206      70.0     7.60E+5
207      90.0     7.60E+5
208     110.0     1.14E+6
209     290.0     1.10E+6
210     310.0     1.20E+6
211     360.0     1.10E+6
212  SPDIN1   0.0000E+0
213  SPDIN2   2.9100E-4
214  SPDIN3   1.0115E-3
215  SPDIN4   1.3108E-5
216  SPDIN5   6.5535E-6
217  SPDIN6       0.0
218  SPDIN7       0.0
219  SPDIN8       0.0
220  SPDIN9       0.0
221  SPDIN10      0.0
222  SPDIN11      0.0
223  SPDIN12      0.0
```

2-PISTON GEOMETRY WITH NO AXIS, File IPREP for K3PREP:

```
T3AAA KIVA-3 CYLINDER WITH 2 PISTONS, REPORT EXAMPLE 111892
 BORE      10.00
 STROKE     8.0
 SQUISH     1.5
 THSECT   360.0
NBLOCKS    21
 12  12  19    0   2   1
 5.000   5.000  -5.000  -5.000   5.000   5.000  -5.000  -5.000
-5.000   5.000   5.000  -5.000  -5.000   5.000   5.000  -5.000
 0.0     0.0     0.0     0.0    19.0    19.0    19.0    19.0
 2.0     2.0     2.0     2.0     1.0     1.0
  3   2   2    0   4   2
 5.060   5.722   4.619   3.967   5.060   5.722   4.619   3.967
-3.224  -1.804  -1.913  -3.044  -3.224  -1.804  -1.913  -3.044
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 4.0     9.0     2.0     2.0     2.0     2.0
  3   2   2    0   4   2
 5.994   5.858   4.957   4.957   5.994   5.858   4.957   4.957
-0.262   1.299   0.653  -0.653  -0.262   1.299   0.653  -0.653
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 4.0     9.0     2.0     2.0     2.0     2.0
  3   2   2    0   4   2
 5.322   4.424   3.967   4.619   5.322   4.424   3.967   4.619
 2.770   4.054   3.044   1.913   2.770   4.054   3.044   1.913
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 4.0     9.0     2.0     2.0     2.0     2.0
  2   3   2    0   4   2
 3.044   3.224   1.804   1.913   3.044   3.224   1.804   1.913
 3.967   5.060   5.722   4.619   3.967   5.060   5.722   4.619
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 2.0     2.0     4.0     9.0     2.0     2.0
  2   3   2    0   4   2
 0.653   0.262  -1.299  -0.653   0.653   0.262  -1.299  -0.653
 4.957   5.994   5.858   4.957   4.957   5.994   5.858   4.957
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 2.0     2.0     4.0     9.0     2.0     2.0
  2   3   2    0   4   2
-1.913  -2.770  -4.054  -3.044  -1.913  -2.770  -4.054  -3.044
 4.619   5.322   4.424   3.967   4.619   5.322   4.424   3.967
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 2.0     2.0     4.0     9.0     2.0     2.0
  3   2   2    0   4   2
-4.619  -3.967  -5.060  -5.722  -4.619  -3.967  -5.060  -5.722
 1.913   3.044   3.224   1.804   1.913   3.044   3.224   1.804
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 9.0     4.0     2.0     2.0     2.0     2.0
  3   2   2    0   4   2
-4.957  -4.957  -5.994  -5.858  -4.957  -4.957  -5.994  -5.858
-0.653   0.653   0.262  -1.299  -0.653   0.653   0.262  -1.299
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 9.0     4.0     2.0     2.0     2.0     2.0
  3   2   2    0   4   2
-3.967  -4.619  -5.322  -4.424  -3.967  -4.619  -5.322  -4.424
-3.044  -1.913  -2.770  -4.054  -3.044  -1.913  -2.770  -4.054
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 9.0     4.0     2.0     2.0     2.0     2.0
  2   3   2    0   4   2
-1.804  -1.913  -3.044  -3.224  -1.804  -1.913  -3.044  -3.224
-5.722  -4.619  -3.967  -5.060  -5.722  -4.619  -3.967  -5.060
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 2.0     2.0     9.0     4.0     2.0     2.0
  2   3   2    0   4   2
 1.299   0.653  -0.653  -0.262   1.299   0.653  -0.653  -0.262
-5.858  -4.958  -4.958  -5.994  -5.858  -4.958  -4.958  -5.994
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 2.0     2.0     9.0     4.0     2.0     2.0
  2   3   2    0   4   2
 4.054   3.044   1.913   2.770   4.054   3.044   1.913   2.770
-4.424  -3.967  -4.619  -5.322  -4.424  -3.967  -4.619  -5.322
 0.0     0.0     0.0     0.0     2.0     2.0     2.0     2.0
 2.0     2.0     9.0     4.0     2.0     2.0
```

```
  3    3     3    0     4    3
 5.298     5.918     4.957     4.330     5.298     5.918     4.957     4.330
-2.817    -0.990    -0.653    -2.500    -2.817    -0.990    -0.653    -2.500
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
 4.0      10.0       2.0       2.0       2.0       2.0
  3    3     3    0     4    3
 5.738     4.884     3.967     4.830     5.738     4.884     3.967     4.830
 1.754     3.484     3.044     1.294     1.754     3.484     3.044     1.294
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
 4.0      10.0       2.0       2.0       2.0       2.0
  3    3     3    0     4    3
 2.500     2.817     0.990     0.653     2.500     2.817     0.990     0.653
 4.330     5.298     5.918     4.957     4.330     5.298     5.918     4.957
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
 2.0       2.0       4.0      10.0       2.0       2.0
  3    3     3    0     4    3
-1.294    -1.754    -3.484    -3.044    -1.294    -1.754    -3.484    -3.044
 4.830     5.738     4.885     3.967     4.830     5.738     4.885     3.967
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
 2.0       2.0       4.0      10.0       2.0       2.0
  3    3     3    0     4    3
-4.957    -4.330    -5.298    -5.918    -4.957    -4.330    -5.298    -5.918
 0.653     2.500     2.817     0.990     0.653     2.500     2.817     0.990
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
10.0       4.0       2.0       2.0       2.0       2.0
  3    3     3    0     4    3
-3.967    -4.830    -5.738    -4.885    -3.967    -4.830    -5.738    -4.885
-3.044    -1.294    -1.754    -3.484    -3.044    -1.294    -1.754    -3.484
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
10.0       4.0       2.0       2.0       2.0       2.0
  3    3     3    0     4    3
-0.990    -0.653    -2.500    -2.817    -0.990    -0.653    -2.500    -2.817
-5.918    -4.957    -4.330    -5.298    -5.918    -4.957    -4.330    -5.298
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
 2.0       2.0      10.0       4.0       2.0       2.0
  3    3     3    0     4    3
 3.484    3.044      1.294     1.754     3.484     3.044     1.294     1.754
-4.885    -3.967    -4.830    -5.738    -4.885    -3.967    -4.830    -5.738
16.000    16.000    16.000    16.000    19.000    19.000    19.000    19.000
 2.0       2.0      10.0       4.0       2.0       2.0
 NPATCH     20
  2    1     1    2     1    1
  3    1     1    6     1    1
  4    1     1   10     1    1
  5    3     1   10     1    1
  6    3     1    6     1    1
  7    3     1    2     1    1
  8    2     1   10     1    1
  9    2     1    6     1    1
 10    2     1    2     1    1
 11    4     1    2     1    1
 12    4     1    6     1    1
 13    4     1   10     1    1
 14    1     1    3    17    1
 15    1     1    9    17    1
 16    3     1    8    17    1
 17    3     1    2    17    1
 18    2     1    8    17    1
 19    2     1    2    17    1
 20    4     1    3    17    1
 21    4     1    9    17    1
```

2-PISTON GEOMETRY WITH NO AXIS, File ITAPE for KIVA-3:

```
T3AAA K111292 CYLINDER WITH 2 PISTONS. REPORT EXAMPLE 111892
 IREST     0
 LWALL     1
 LPR       0
 IREZ      2
 NCFILM 9999
 NCTAP8  100
 NCLAST 9999
 CAFILM   30.0
 CAFIN   180.0
 ANGMOM    0.0
 PGSSW     0.0
 SAMPL     0.0
 DTI    1.00000E-5
 DTMXCA    1.0
 DTMAX    9.99E+9
 TLIMD     1.0
 TWFILM   9.99E+9
 TWFIN    9.99E+9
 FCHSP     0.25
 BORE     10.00
 STROKE    8.0
 SQUISH    1.5
 RPM       1.6E+3
 ATDC   -180.0
 DATDCT    0.0
 CONROD   16.269
 SWIRL     0.0
 SWIPRO    3.11
 THSECT  360.0
 SECTOR    0.0
 EPSY      1.0E-3
 EPSV      1.0E-3
 EPSP      1.0E-4
 EPST      1.0E-3
 EPSK      1.0E-3
 EPSE      1.0E-3
 GX        0.0
 GY        0.0
 GZ        0.0
 TCYLWL  325.0
 THEAD   325.0
 TPISTN  325.0
 TVALVE  325.0
 TEMPI   325.0
 PARDON    0.0
 AO        0.0
 BO        1.0
 ANC4      0.05
 ADIA      0.0
 ANUO      0.0
 VISRAT-.66666667
 TCUT     9.99E+9
 TCUTE    9.99E+9
 EPSCHM    0.02
 OMGCHM    1.0
 TKEI      0.10
 TKESW     1.0
 SGSL      0.0
UNISCAL    0.0
 AIRMU1  1.457E-5
 AIRMU2  110.0
 AIRLA1  252.0
 AIRLA2  200.0
 PRL       0.74
 RPR       1.11
 RPRO      1.0
 RPRE    0.769231
 RSC       1.11
 XIGNIT    1.0E+4
 T1IGN    9.99E+9
 72    TDIGN     9.99E+9
 73    CA1IGN   -27.0
 74    CADIGN    9.6
 75    XIGNL1     0.0
 76    XIGNR1     0.623
 77    YIGNF1     0.0
 78    YIGND1     0.238
 79    ZIGNB1    11.75
 80    ZIGNT1    12.50
 81    XIGNL2     0.0
 82    XIGNR2     0.0
 83    YIGNF2     0.0
 84    YIGND2     0.0
 85    ZIGNB2     0.0
 86    ZIGNT2     0.0
 87    KWIKEQ     0
 88    NUMNOZ     1
 89    NUMVEL     1
 90   INJDIST     1
 91    KOLIDE     1
 92    T1INJ     9.99E+9
 93    TDINJ     9.99E+9
 94    CA1INJ   -52.0
 95    CADINJ    12.672
 96    TSPMAS     0.0116
 97    PULSE      1.0
 98    TNPARC  2000.0
 99    TPI      350.0
100    TURB       1.0
101   BREAKUP     1.0
102    EVAPP      1.0
103    DRNOZ      0.0
104    DZNOZ     -0.01
105    DTHNOZ     0.0
106    TILTXY     0.0
107    TILTXZ     0.0
108    CONE      62.5
109    DCONE     12.5
110    ANOZ       1.0
111    SMR       5.00E-4
112    AMPO       0.0
113          4000.0
114    NSP        5
115      C8H18   RHO1       0.0
116         O2   RHO2  2.6862E-4 MW2       32.000  HTF2       0.0
117         N2   RHO3  9.3371E-4 MW3       28.016  HTF3       0.0
118        CO2   RHO4  1.2100E-5 MW4       44.011  HTF4     -93.965
119        H2O   RHO5  6.0495E-6 MW5       18.016  HTF5     -57.103
120    NRK        0
121    NRE        0
122   DISTAMB     0.0
123    PAMB    1.1439E+6
124    TKEAMB   3600.0
125    SCLAMB     1.0
126   SPDAM1   0.0000E+0
127   SPDAM2   2.6862E-4
128   SPDAM3   9.3371E-4
129   SPDAM4   1.2100E-5
130   SPDAM5   6.0495E-6
131    VELIN      0.0
132    REEDIN     0.0
133   REEDOUT     0.0
134    NUMPCC     4
135       0.0    1.08E+6
136     115.0    1.30E+6
137     150.0    1.21E+6
138     360.0    1.08E+6
139    NUMPEX     8
140       0.0    1.10E+6
141     100.0    1.04E+6
142     115.0    7.60E+5
143     130.0    7.60E+5                      148    SPDIN1   0.0000E+0
144     150.0    1.14E+6                      149    SPDIN2   2.6862E-4
145     245.0    1.08E+6                      150    SPDIN3   9.3371E-4
146     260.0    1.06E+6                      151    SPDIN4   1.2100E-5
147     360.0    1.10E+6                      152    SPDIN5   6.0495E-6
```

☆ U.S. GOVERNMENT PRINTING OFFICE 1992—774-208/63058