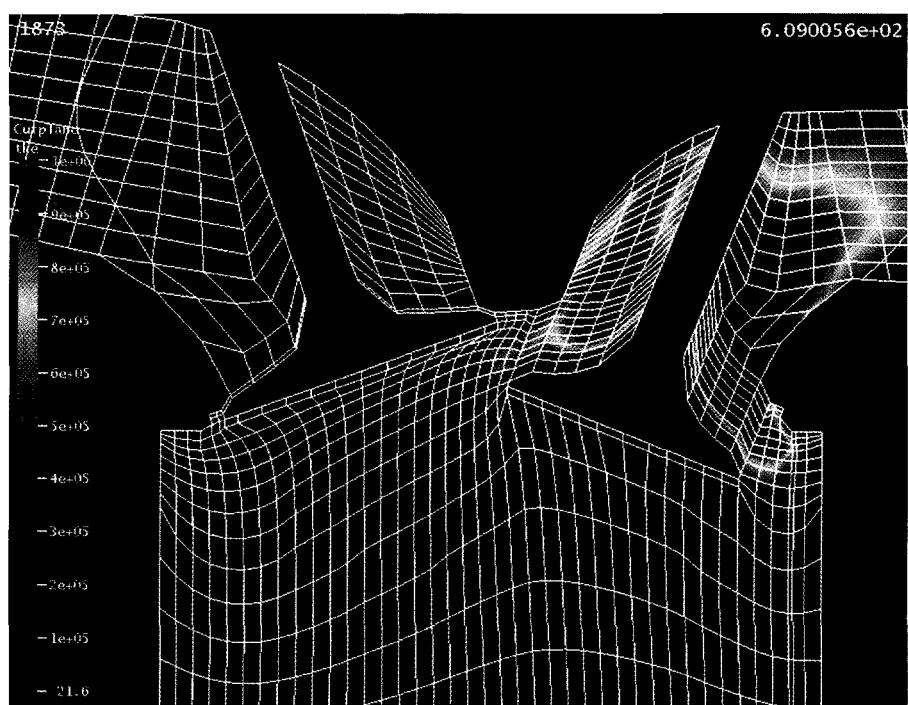
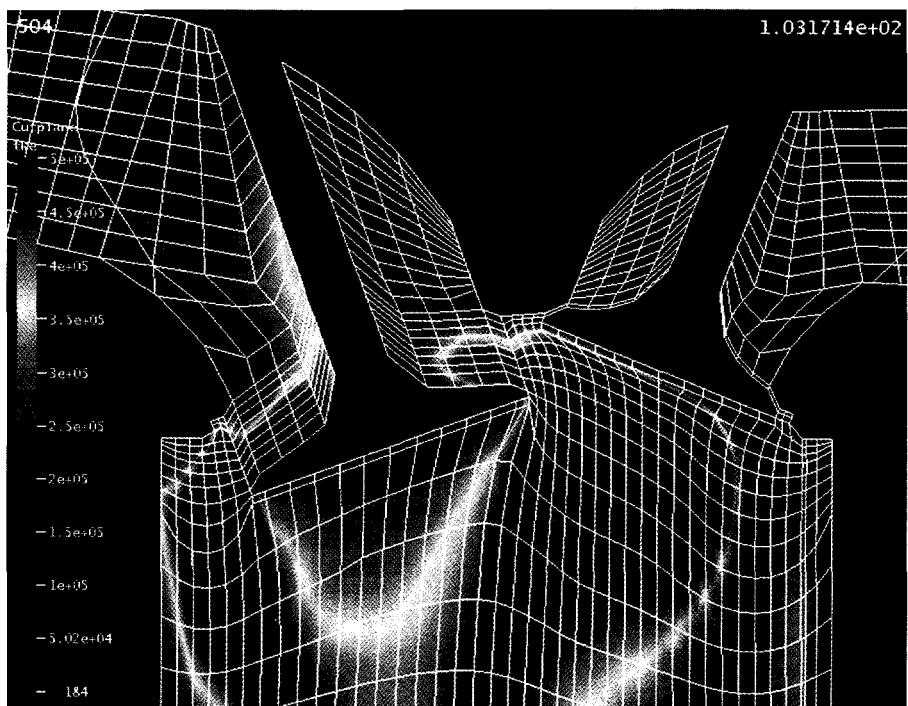


*KIVA-3V: A Block-Structured KIVA Program  
for Engines with Vertical or Canted Valves*

REIVED  
JUL 28 1997  
OSTI



**Los Alamos**  
NATIONAL LABORATORY

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

*Prepared by Margaret Findley, Group T-3*

*This work was supported by the US Department of Energy, Energy Efficiency and Renewable Energy, Office of Transportation Technologies and Office of Utility Technologies.*

**Cover photos:** Pictured here is a KIVA-3V calculation of a modern, high-performance 4-valve gasoline engine with a pentroof combustion chamber. This is a cold-flow run at 1600 rpm, with a constant 98-kPa manifold pressure. The cross-sectional views are through one intake valve (left side of each picture) and one exhaust valve (right side of each picture). Because the intake and exhaust valves in this engine share the same physical space at different times in the engine cycle, the structured computing mesh must be continually adjusted to conform to the current valve configuration. The white lines show the mesh at the two extremes: when the intake valves are fully open (103° crank angle) and, later, when the exhaust valves are fully open (609° crank angle). The curved lines cutting through the ports are fictitious. Plotted in color is the turbulent kinetic energy. At 103°, the highest turbulence levels induced in the cylinder by the incoming flow around the valve occur both off the near-side top surface and the far edge of the valve. At 609°, the highest turbulence values are confined to the exhaust port.

The initial computing mesh was created by the K3PREP parametric grid generator. The color plots were drawn with the General Mesh Viewer (GMV) graphics postprocessor, a public-domain program written at Los Alamos National Laboratory.

*An Affirmative Action/Equal Opportunity Employer*

*This report was prepared as an account of work sponsored by an agency of the United States Government. Neither The Regents of the University of California, the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by The Regents of the University of California, the United States Government, or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of The Regents of the University of California, the United States Government, or any agency thereof. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.*

*KIVA-3V: A Block-Structured KIVA Program  
for Engines with Vertical or Canted Valves*

*Anthony A. Amsden*

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

**Los Alamos**  
NATIONAL LABORATORY

Los Alamos, New Mexico 87545

## **DISCLAIMER**

**Portions of this document may be illegible  
in electronic image products. Images are  
produced from the best available original  
document.**

## CONTENTS

ABSTRACT .....	1
I. INTRODUCTION AND BACKGROUND .....	2
II. THE KIVA-3V PROGRAM .....	5
A. Storage of Cell Data.....	5
B. Input and Output Files .....	10
III. THE KIVA-3V VALVE MODEL.....	11
A. Valve Data in File ITAPE17.....	11
B. Valve Data in File ITAPE5.....	13
C. Valve Lift Data: File ITAPE18.....	13
D. Valve Movement.....	13
E. Valve Snapping.....	14
F. Valve Closing and Opening.....	14
G. Valve Grids and Continuous Rezoning .....	15
H. Grid Generation with Valves.....	27
IV. EXAMPLES WITH VERTICAL AND CANTED VALVES.....	28
A. Vertical Valves .....	28
B. OHV Engine with Wedge Combustion Chamber and 2 Valves....	32
C. DOHC Pentroof Engine (Lift Paths Overlap).....	37
D. DOHC Asymmetric Pentroof Engine .....	41
V. MESH GENERATION .....	45
A. File ITAPE17.....	45
B. K3PREP: Extensions to the IPREP File .....	46
1. Block Identification and Ghost Blocks.....	46
2. Moving Boundary Identification.....	47
3. Block Copying.....	47
4. Flag Cylinder Vertices to Tilt.....	47
5. Flag Central Pentroof Vertices.....	48
6. Flag Vertices Around Wedge Top.....	48
7. Block Translation.....	48
8. Localized Geometry Refinement.....	48
9. Block Reshaping .....	49
10. Pentroof (x, y) Coordinates .....	49
11. Shaping of the Valve Ports .....	50
12. Shaping Runners.....	50
13. Joining Pairs of Runners.....	50
14. Block Patching.....	50
15. Relaxing the Interior of the Piston Bowl .....	51
16. Valve Profiling.....	51
a. Valve Tops.....	51

b. Valve Faces.....	51
17. Specifying z's up the Cylinder Wall.....	52
18. Valve Canting.....	52
19. Dish in Piston Face .....	53
20. Scallops in Piston Face .....	53
21. Offset Piston Bowl or Dome.....	53
 VI. OTHER NEW FEATURES IN KIVA-3V.....	55
A. Particle-Based Model for Wall Film Dynamics.....	55
1. The Splash Model.....	55
2. Valve Seat Clearing .....	56
B. Initial Conditions Specified by Region.....	57
C. Improved Subroutine SORT .....	59
D. Improved Law-of-the-Wall Heat Transfer Model .....	59
E. Mixing-Controlled Turbulent Combustion Model.....	60
F. Optional RNG k- $\epsilon$ Turbulence Model .....	61
G. Fuel Library .....	61
1. New Fuel Options .....	61
a. Natural Gas .....	62
b. Dimethyl Ether (DME).....	62
2. Tables of Surface Tension and Thermal Conductivity.....	62
H. No-Hydro Option.....	63
I. Low Mach-Number Flows .....	63
J. High-Speed Flows .....	64
K. Soot Model.....	65
 VII. OUTPUT AND POST-PROCESSING.....	68
A. CGS (Common Graphics System) and Postscript Files.....	68
B. K3POST: Optional IPOST File .....	69
C. Monitor Prints for IC Engines.....	70
D. GMV (General Mesh Viewer) .....	71
 VIII. OTHER CHANGES TO FILE ITAPE5.....	74
 ACKNOWLEDGMENTS.....	75
 REFERENCES .....	76

# **KIVA-3V: A BLOCK-STRUCTURED KIVA PROGRAM FOR ENGINES WITH VERTICAL OR CANTED VALVES**

by

**Anthony A. Amsden**

## **ABSTRACT**

This report describes an extended version of KIVA-3, known as KIVA-3V, that can model any number of vertical or canted valves in the cylinder head of an internal combustion (IC) engine. The valves are treated as solid objects that move through the mesh using the familiar "snapper" technique used for piston motion in KIVA-3. Because the valve motion is modeled exactly, and the valve shapes are as exact as the grid resolution will allow, the accuracy of the valve model is commensurate with that of the rest of the program. Other new features in KIVA-3V include a particle-based liquid wall film model, a new sorting subroutine that is linear in the number of nodes and preserves the original storage sequence, a mixing-controlled turbulent combustion model, and an optional RNG  $k-\epsilon$  turbulence model. All features and capabilities of the original KIVA-3 have been retained. The grid generator, K3PREP, has been expanded to support the generation of grids with valves, along with the shaping of valve ports and runners. Graphics output options have also been expanded. The report discusses the new features, and includes four examples of grids with vertical and canted valves that are representative of IC engines in use today.

## I. INTRODUCTION AND BACKGROUND

Computational fluid dynamics (CFD) has become an established tool for the design and understanding of practical combustion systems.

Multidimensional models have proven their value in reducing the need for physical experimentation, the benefit of which has been a reduction in product development time and cost.

Although there are a number of commercial CFD packages available on the market, the most widely used for engine research has been the KIVA family of three-dimensional programs. This popularity is based on the accuracy of KIVA, and the fact that the complete source code is available to a worldwide user community at a modest cost. In addition to their use in industry and government laboratories, KIVA programs are widely used in university engineering departments, which are graduating a new generation of engineers familiar with KIVA and CFD modeling in general. As background, the story of KIVA as a successful case history of technology transfer has been discussed in the literature.<sup>1</sup>

The original KIVA program was publicly released in 1985,<sup>2-4</sup> and was replaced by the improved KIVA-II in 1989.<sup>5,6</sup> These earlier versions lent themselves well to confined in-cylinder flows<sup>7</sup> and a variety of open combustion systems, but were quite inefficient when applied to complex geometries that included such features as long transfer ports or diesel prechambers. This situation resulted from the fact that the entire domain of interest had to be encompassed within a single tensor-product mesh with fixed index offsets in all three directions, which could result in a large number of deactivated cells.

KIVA-3<sup>8</sup> removed this handicap by the use of a block-structured mesh that entirely eliminated the need to create regions of unused cells. In addition, the use of indirect addressing for neighbor connectivity allowed data storage arrays to be sorted, which minimized the length of vector loops and eliminated testing on cell and vertex flags. Further, tailored boundary condition data was carried in tables that allowed KIVA-3 to sweep in shorter vectors over only those vertices or cells involved. When KIVA-3 was developed, the immediate application was to IC engines with ports in the cylinder walls. This included both crankcase scavenged 2-stroke engines with transfer and boost ports,<sup>9</sup> and opposed-piston diesel engines.<sup>10</sup>

With its release in 1993 in both Cray and workstation versions, KIVA-3 has attracted more KIVA users. An analysis of the technical papers on multidimensional modeling presented at the 1997 SAE (Society of Automotive Engineers) International Congress reveals that a large fraction involved the use of KIVA programs. A representative sample illustrates that they are in worldwide use.<sup>11-18</sup> KIVA programs have also been extensively used for gas turbine<sup>19-23</sup> and ramrocket<sup>24,25</sup> applications.

The KIVA-3 workstation version became quite popular with the availability of increased computing power on high-end platforms, such as those offered by IBM, H-P, SGI, and Sun. Although not as efficient on Cray platforms as the Cray-specific version (about 10% slower), the workstation version ports easily to the Cray. To eliminate the maintenance of multiple versions of KIVA codes, in the future only a workstation version will be updated; KIVA-II and the Cray version of KIVA-3 will no longer be supported by Los Alamos. A commercial version of KIVA-3, known as CRI/TurboKiva™, is distributed by Cray Research/SGI for use on Cray platforms.<sup>26</sup>

KIVA-3V represents a significant increment in capability through the addition of an effective model for intake and exhaust valves, while retaining all previous features of KIVA-3. The valve model is an extension of that created by R. P. Hessel,<sup>27-29</sup> in which valves are treated as solid objects that move through the mesh, using the familiar "snapper" technique already used for piston motion in KIVA-3. A pre-release (beta) version of KIVA-3V has been applied to studies recently reported.<sup>13,14</sup>

KIVA-3V retains the distinct three-part structure of KIVA-3, in which the grid generator and graphics are separated from the hydro program. The KIVA-3V package includes both a pre-processor (K3PREP), which has been expanded to support the generation of grids with valves, 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-3V uses the same solution algorithms and solves the same set of equations as KIVA-3 and KIVA-II, with the exception of the improvements and new features that are described in this report. This report should be considered a companion report to the comprehensive KIVA-II report<sup>6</sup> and the KIVA-3 report.<sup>8</sup> It is assumed that the reader has these earlier reports available, as only new or updated information is presented here.

Check our Website for more information on KIVA programs, and how to subscribe to kiva-talk and kiva-news on the Internet. You will find links to other installations around the world that are involved in KIVA research, plus information on commercial pre- and post-processors that are compatible with KIVA. The home page for our Fluid Dynamics Group is

<http://gnarly.lanl.gov/home.html> .

Distribution of the KIVA-3V package (source codes for KIVA-3V, K3PREP, and K3POST, along with all input data files for eight 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 (423) 576-2606, and request publication ESTSC-1 *Guide for Submitting and Ordering Software*. The e-mail address for ESTSC is [estsc@adonis.osti.gov](mailto:estsc@adonis.osti.gov). In addition, they have a Website at <http://www.doe.gov/html/osti/estsc/estsc.html>. The mailing address is P. O. Box 1020, Oak Ridge, TN 37831-1020.

## II. THE KIVA-3V PROGRAM

The general structure of the KIVA-3V program remains unchanged from that of KIVA-3. The discussions<sup>8</sup> concerning the computing mesh, the concepts of blocks and fluid regions, and the indexing notation continue to apply.

### A. Storage of Cell Data

As in KIVA-II and KIVA-3, the use of memory space in KIVA-3V is minimized by means of array equivalencing, which provides efficient memory management. 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.

Despite a significant increase in the number of named arrays to support the additional features in KIVA-3V, particularly for the liquid wall film model, the number of new storage arrays has grown only slightly. The dedicated arrays have increased by **three**, two of which support the improved sorting algorithm, and the third to identify moving cell faces. (If the optional soot model is turned on, another **three** additional arrays will be activated.) The number of equivalenced arrays has also gone up by **three**, from 58 to 61 to support the high-speed flow option and the RNG k- $\epsilon$  turbulence option.

Table 1 shows the allocation of the equivalenced arrays in KIVA-3V, and is an updated version of that in the KIVA-3 report. Again, the column labels from left to right correspond to the sequence in which subroutines are called during the KIVA-3V 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.

The common blocks in KIVA-3V, like those in the workstation version of KIVA-3 from which it was built, are arranged to completely separate real numbers from integers. Because workstations typically express integers in 32 bits (4 bytes), while real numbers are required to be 64 bits (8 bytes), a rigorous separation of the two data types is necessary to ensure that real numbers will never have an address at a half-word boundary.

KIVA-3V contains two built-in functions, LOCI and LOCR, that determine the lengths of all the integer and real arrays in the common blocks, using the word lengths appropriate for the platform in use, which are calculated in subroutine BEGIN.

Table 1. The Allocation of Equivalenced Arrays in KIVA-3V Storage

SUB-ROUTINE	SETUP	VISC	TIMSTP	NEWCYC	INJECT	PMOVTV	BREAK	COLIDE	EVAP	LAWALL
1	ADJPISTN			FULOUT	FRAN	FRAN	FRAN	FRAN		BC
1	ADJVALVE			MONITOR	PFIND	PFIND		REPACK		
1	APROJ_RC			PLTGMV		SPLASH				
1	CONVEX			TAPEWR		REPACK				
1	PISTON									
ARRAY 1	SETUPBC									
1	SORT_STATE									
1	VOLUME									
E1	MV, RMV	.....	.....	RMV	.....	.....	.....	.....	RMV	
E2	GAMMA	GAMMA	.....	GAMMA	.....	.....	.....	GAMMA	GAMMA	
E3	EPS	EPS	.....	EPS	.....	EPS	.....	.....	.....	
E4	WORK			WORK					TAUL	
E5	WORK1								TAUF	
E6									TAUB	
E7									TAULX	
E8									TAULY	
E9									TAULZ	
E10									TAUFX	
E11									TAUFY	
E12									TAUFZ	
E13									TAUBX	
E14									TAUBY	
E15									TAUBZ	
E16					PLM	.....	.....	.....	.....	
E17					PFM	.....	.....	.....	.....	
E18					PBM	.....	.....	.....	.....	
E19									VAPM	
E20									ENTHO	
E21					PLX	.....	.....	.....	.....	
E22	XL				PLY	.....	.....	.....	.....	
E23	YL				PLZ	.....	.....	.....	.....	
E24	ZL				PFX	.....	.....	.....	.....	
E25	FXL	.....	FXL		PFY	.....	.....	.....	.....	
E26	FXF	.....	FXF		PFZ	.....	.....	.....	.....	
E27	FXB	.....	FXB		PBX	.....	.....	.....	.....	
E28					PBY	.....	.....	.....	.....	
E29	VSOLID				PBZ	.....	.....	.....	.....	
E30				HL	HL				HL	
E31				HF	HF				HF	
E32				HB	HB				HB	
E33										
E34										
E35										
E36										
E37										
E38										
E39										
E40										
E41										
E42									HTL	
E43									HTF	
E44									HTB	
E45								TOTCM	HYL	
E46								DMTOT	.....	
E47								TOH	HYF	
E48								DSIEP	.....	
E49								CPC	HYB	
E50		RON			.....	.....	.....	.....	.....	
E51		SIEN			.....	.....	.....	.....	.....	
E52										
E53										
E54		TKEN			.....	.....	.....	.....	.....	
E55		EPSN			.....	.....	.....	.....	.....	
E56		UN			.....	.....	.....	.....	UN	
E57		VN			.....	.....	.....	.....	VN	
E58		WN			.....	.....	.....	.....	WN	
E59										
E60										
E61										

Dotted lines indicate that the quantity to the left must be retained for later use. The 12 arrays whose names are enclosed in parentheses are normally defined as dedicated arrays, as discussed in the KIVA-3 report.

Table 1. (continued)

SUB-ROUTINE	WALLFILM	NODCPL	CHEM	CHEMEO	PMOM	PCOPL	YSOLVE	(OPTIONAL)	EXDIF	PINIT
	REPACK	BCNODCPL		(OR CHMQGM)	REPACK	BC	YIT	WRITE	BC	
	CLEAR	BC					RESY	E4	BCROT1	
							BCDIFF	THROUGH	BCDIFF	
							BCROT1	E15		
							BCROXCEN	TO		
ARRAY								SSD		
E1	.....	RMV, MV	.....	.....	MV, RMV	.....	.....	RMV	.....	
E2	GAMMA	.....	.....	GAMMA	.....	.....	.....	GAMMA	GAMMA	
E3	.....	.....	.....	.....	EPS	EPS	.....	EPS	.....	
E4	TAUL	AUGMV					(CLI)	(CLI)	(CLI)	
E5	TAUF						(CLJ)	(CLJ)	(CLJ)	
E6	TAUB						(CLK)	(CLK)	(CLK)	
E7	TAULX						(CFI)	(CFI)	(CFI)	
E8	TAULY						(CFJ)	(CFJ)	(CFJ)	
E9	TAULZ						(CFK)	(CFK)	(CFK)	
E10	TAUFX						(CBI)	(CBI)	(CBI)	
E11	TAUFY						(CBJ)	(CBJ)	(CBJ)	
E12	TAUFZ						(CBK)	(CBK)	(CBK)	
E13	TAUBX						(RFSUM14)	(RFSUM14)	(RFSUM14)	
E14	TAUBY						(RFSUM34)	(RFSUM34)	(RFSUM34)	
E15	TAUBZ						(RFSUM84)	(RFSUM84)	(RFSUM84)	
E16	PLM	DUDX	.....	.....	.....	.....	.....	DUDX	.....	
E17	PFM	DUDY	.....	.....	.....	.....	.....	DUDY	.....	
E18	PBM	DUDZ	.....	.....	.....	.....	.....	DUDZ	.....	
E19		DVDX	.....	.....	.....	.....	.....	DVDX	.....	
E20		DVDY	.....	.....	.....	.....	.....	DVDY	.....	
E21	PLX	DVDZ	.....	.....	.....	.....	.....	DVDZ	.....	
E22	PLY	Dwdx	.....	.....	.....	.....	.....	Dwdx	.....	
E23	PLZ	Dwdy	.....	.....	.....	.....	.....	Dwdy	.....	
E24	PFX	Dwdz	.....	.....	.....	.....	.....	Dwdz	.....	
E25	PFY						PHID	.....	PHID	.....
E26	PFZ						ENTHDF	.....	DISPTIL	.....
E27	PBX						SPD14,DYP14	.....	TEM14	
E28	PBY						SPD34,DYP34	.....	TEM34	
E29	PBZ				RU	RU	SPD84,DYP84	.....	TEM84	
E30	HL				RV	RV	HISP	.....	UTIL	.....
E31	HF				RW	RW	SPMTIL	.....	VTL	.....
E32	HB						ENTHTIL	.....	WTIL	.....
E33	UFILML,TMNUML						DDY	.....	RMVSU	.....
E34	VFILML,TMDENL						RES	.....		
E35	WFILML						RESOLD	.....		
E36	UFILMF,TMNUMF						DRES	.....		
E37	VFILMF,TMDENF						RDRDY	.....		
E38	WFILMF						DELTAY	.....		
E39	UFILMB,TMNUMB						DELYPH	.....	TKE14	
E40	VFILMB,TMDENB						XCEN	.....	TKE34	PN
E41	WFILMB						YCEN	.....	TKE84	PHIP
E42	HTL	UB					ZCEN	.....	EPS14	
E43	HTF	VB					YSPM	.....	EPS34	
E44	HTB	WB					YSPD	.....	EPS84	
E45	HYL							TT1MPH		
E46	.....	.....	DMTOT	DMTOT	.....	.....	.....	TKE1MPH		
E47	HYF							EPS1MPH		
E48	.....	.....	DSIEP	DSIEP	.....	.....	.....			
E49	HYB		DTKEP	DTKEP	.....	.....	.....	CPC	.....	
E50	.....	.....	.....	.....	.....	.....	.....	RON, HTCTL	.....	
E51	.....	.....	SIEN	.....	.....	.....	.....	SIEN, TTIL		
E52	.....	.....						TKETIL	.....	
E53	.....	.....						EPSTIL	.....	
E54	.....	.....								
E55	.....	.....								
E56	.....	UN	.....	UN	.....	.....	.....	UN	.....	
E57	.....	VN	.....	VN	.....	.....	.....	VN	.....	
E58	.....	WN	.....	WN	.....	.....	.....	WN	.....	
E59										
E60										
E61										

Table 1. (continued)

SUB-ROUTINE	PGRAD (1.0)	VSOLVE	TSOLVE	PSOLVE	(OPTIONAL)	PGRAD (3.0)	PGRAD (1.0)	PHASEB	KESOLV	PACCEL
\ BC	BCPGRAD	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	RMV	*****	*****	*****
E2 *****			GAMMARGAM	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)	
E14 *****			(RFSUM34)	PBALB	(RFSUM34)	*****	*****		(RFSUM34)	
E15 *****			(RFSUM84)	PBALT	(RFSUM84)	*****	*****		(RFSUM84)	
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	*****
E23 *****		DWDY		MB, RMBDT					YL	*****
E24 *****		DWDZ							ZL	*****
E25 *****		PHID	PHID		*****	*****	*****		PHID	
E26 *****		DISPTIL			*****	*****	*****			
E27 RESU		RESU	TEM14	UALA						
E28 RESV		RESV	TEM34	UAFA					VOLL	VOLL
E29 RESW		RESW	TEM84	UABA						*****
E30 *****					UTIL					
E31 *****					VTL					
E32 *****					WTIL					
E33 RMVSU		RMVSU	*****	RMVSU	*****	RMVSU	RMVSU		RDRDE	
E34 RESUO		RESUO	RES	RES					RES	
E35 RESVO		RESVO	RESOLD	RESOLD					RESOLD	
E36 RESWO		RESWO	DRES	DRES					DRES	
E37 DRESU		DRESU	RDRDT	RDRDP					RDRDK	
E38 DRESV		DRESV	DTEMP	DP					DELTKEDEPS	
E39 DRESW		DRESW	TEMPHID	PPHIP					TKE14	
E40 PN		PN	PN	PN	*****	PN	PN	PN	TKE34	
E41 PHIP		PHIP	*****	PHIP	*****	PHIP	PHIP		TKE84	
E42 UB		UB	*****	UB					EPS14	
E43 VB		VB	*****	VB					EPS34	
E44 WB		WB	*****	WB					EPS84	
E45 DISSIP		DISSIP	*****	*****	*****	*****	*****	DISSIP	DISSIP	
E46 DUHAT		DUHAT	HTC	*****	*****	*****	*****	HTC	RTERMK	
E47 DVHAT, VOLB		DVHAT, VOLB	VOLB	VOLB	*****	*****	*****	VOLB	RTERME	
E48 DWHAT		DWHAT							RROVOLL	RROVOLL
E49 *****			CPC	*****	*****	*****	*****		TKEPHID	
E50 HTCTIL		HTCTIL	*****	*****	*****	*****	*****		EPSPHID	
E51 TTIL		TTIL	*****	*****	*****	*****	*****			
E52 *****					*****	*****	*****		TKETIL	
E53 *****					*****	*****	*****		EPSTIL	
E54 *****					*****	*****	*****		TKEN	
E55 *****					*****	*****	*****		EPSN	
E56 UN		UN	*****	*****	*****	*****	*****	UN		
E57 VN		VN	*****	*****	*****	*****	*****	VN		
E58 WN		WN	*****	*****	*****	*****	*****	WN		
E59 DIVERG		DIVERG	*****	*****	*****	*****	*****		DIVERG	
E60										
E61										

Table 1. (continued)

SUB-ROUTINE	REZONE	VOLUME	APROJ	CCFLUX	MOMFLX	(SNAPPERS)	STATE
	REZPENT			BCCFL	BCMOMFL	SETUPBC	
	REZWEDGE			BCEPS	BC	SORT, APROJ	
	REZCOMB			BCMOMXYZ	BCMOMVEL	BC, BCEPS	
				BCROT1		GLOBAL	
				BCROXCEN		PFIND	
ARRAY				MFLUXES		VOLUME	
						FILMSNAP	
						(REZONERS)	
E1	.....	.....	.....	.....	RMV	MV, RMV	.....
E2	.....	.....	.....	.....	.....	GAMMA	GAMMA
E3	.....	.....	.....	EPS, SCL	.....	EPS	.....
E4				CLX, BNDL	.....	WORK	
E5				CLY, BNDR	.....	WORK1	
E6				CLZ, BNDF	.....		
E7				CFX, BNDD	.....		
E8				CFY, BNDB	.....		
E9				CFZ, BNDT	.....		
E10				CBX, DMOML	.....		
E11				CBY, DMOMF	.....		
E12				CBZ, DMOMB	.....		
E13	XO	.....	.....	XO	VCIM		
E14	YO	.....	.....	YO	VCIP		
E15	ZO	.....	.....	ZO	VCJM		
E16	.....	.....	.....	UAL	VCJP		
E17	.....	.....	.....	UAF	VCKM		
E18	.....	.....	.....	UAB	VCKP		
E19				DRDS	DUDS		
E20				DTDS	DVDS		
E21					DWDS		
E22	XL	.....	.....	.....	XL		
E23	YL	.....	.....	.....	YL		
E24	ZL			PERJD	PERJD		
E25				FXL	.....	FXL	.....
E26				FXF	.....	FXF	.....
E27				FXB	.....	FXB	.....
E28	.....	.....	.....	VOLL	.....	VOLL	
E29				ROSIE	UMOM		
E30				ROTKE	VMOM		
E31				ROSLC	WMOM		
E32				ROSLV	SMOM		
E33				ROTKEV	FXV		
E34				ROSLV	FXVM		
E35				MVP	MVP		
E36					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				MP	.....		
E50				CCIM	.....		
E51				CCIP	.....		
E52				CCJM	.....		
E53				CCJP	.....		
E54				CCKM	.....		
E55				CCKP	.....		
E56				DCCL	.....		
E57				DCCF	.....		
E58				DCCB	.....		
E59				ROSLV	.....		
E60				ROTKEV	.....		
E61				ROSLV	.....		

## B. Input and Output Files

Because the number of files is often greater in KIVA-3V than it was in KIVA-3, the file naming procedure has been modified to eliminate confusion between input and output files. Input files now have an 'T' as the first character in their names, and output files have an 'O' as the first character. Thus, for example, K3PREP output file OTAPE17 must be renamed ITAPE17 for use in KIVA-3V.

The ITAPE17 for KIVA-3V differs from that supplied to KIVA-3 only by the addition of one new vertex flag array, IDFACE, which relates to valves. If the user is not running a valve problem, and has only an old TAPE17 that doesn't include IDFACE, it can still be used. A comment near the beginning of subroutine SETUP in KIVA-3V provides three lines of coding that can be added to easily set appropriate IDFACE values at all vertices. When the valve model is used, KIVA-3V requires input file ITAPE18 to supply the lift data.

For postprocessing with K3POST, the KIVA-3V output file OTAPE9 must be renamed ITAPE9 for use in K3POST. The user now has the option of supplying the plot data in an input file named IPOST. Other optional output from KIVA-3V includes four files of monitor data as a function of crank angle for IC engines, and dumps for the GMV (General Mesh Viewer) graphics postprocessor.

### III. THE KIVA-3V VALVE MODEL

KIVA-3V can model any number of valves in the cylinder head. Each valve can have its own size and profile, accurate to the fineness of the grid, and its own lift history. The valves may be vertical, with the valve axis parallel to the cylinder axis, or canted at some angle with respect to the cylinder axis. Canting is presently allowed in (x,z) space only, implying that the camshaft(s) are parallel with the y-axis. In logical space, valves can move only in a bottom-top direction. There is currently no provision for horizontal, splayed, or annular valves.

The valve model is an extension of that originally developed by R. P. Hessel,<sup>27-29</sup> in which a valve is treated as a solid object that moves through the mesh using the familiar "snapper" technique already used for piston motion in KIVA-3. Because the valves are modeled exactly, the accuracy of the model is commensurate with that of the rest of KIVA-3. Hessel's approach was to extend subroutine SNAPB to snap the valve upper surfaces in addition to piston faces, and to reassign subroutine SNAPT to snap the lower valve faces. His model considered only vertical valves, as his immediate application was to a heavy-duty diesel engine.

KIVA-3V, however, has two separate new snapper subroutines, SNAPVFCE and SNAPVTOP, tailored specifically for valves, and retains SNAPB and SNAPT in their original roles as piston snappers. SNAPVFCE considers the lower surfaces that face the cylinder, and SNAPVTOP considers the upper surfaces that face the valve ports. To accommodate canted valves, the new subroutines include the x-direction component of motion in addition to the z-direction component. The end result is that the purpose of each of the four snapper subroutines is quite distinct. All previous KIVA-3 capabilities have been retained.

#### A. Valve Data in File ITAPE17

As in the original KIVA-3, the grid supplied to KIVA-3V as input file ITAPE17 from the grid generator is required to have the piston at its BDC position, and the region array IDREG must be supplied. There are typically three physical regions in a valved IC engine (Fig. 1). As before, IDREG = 1 is assigned to all cells in the cylinder. IDREG = 2 may refer to the intake port(s) and IDREG = 3 to the exhaust port(s), or vice versa. The only possible ambiguity is in the path of valve travel. Here, the cells directly above the valve, i.e., out to the valve diameter, have the IDREG of the port (2 or 3). As the valve opens during the KIVA-3 calculation, the region number 2 or 3 automatically follows downward in the wake of the valve in SNAPVTOP as cells above the valve are activated. As the valve closes, SNAPVFCE assigns the cells below the valve face to region 1, as they become part of the cylinder. Thus, the same physical cells will have different values of IDREG during the run, depending on whether the valve is

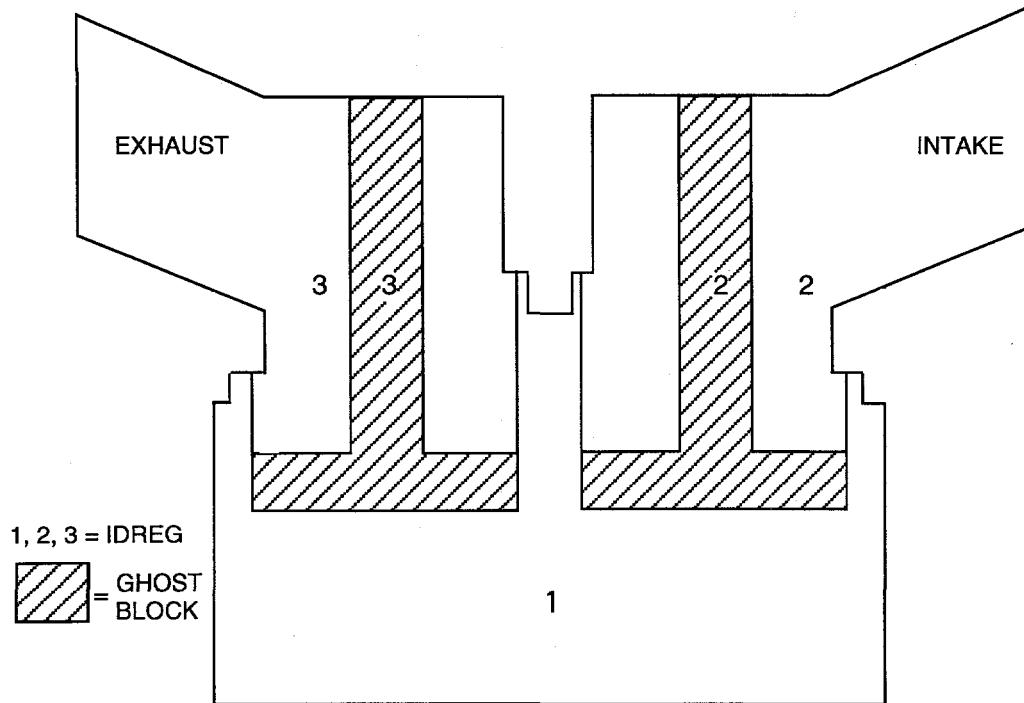


Fig. 1. The three physical regions in a KIVA-3V grid for a typical valved IC engine.

currently above or below them. Cells in a valve recess or pocket always have IDREG = 1.

A new vertex flag array, IDFACE, must also be supplied by the grid generator as part of the ITAPE17 data file. *The addition of IDFACE is the only change to the ITAPE17 file.* In the original KIVA-3, there were only two possible moving surfaces: the piston and the optional upper piston. The program was able to use z-coordinates and vertex and flag information to identify moving surfaces, without requiring additional input data. The implementation of valves and their associated multiple moving surfaces requires additional information to identify *which* moving surface a vertex or cell face is identified with. By the new definition, a lower piston is always moving surface 0, and all vertices on the piston face have a flag IDFACE = 0. Vertices on the upper piston face in an opposed-piston geometry have vertex flags IDFACE = 1.

Although both the lower and upper surface of a valve move with the same velocity, each surface is identified separately because each moving surface is treated as a separate entity by the valve snappers. Because an upper piston is not an option in a valved geometry, the value IDFACE = 1 is available. The vertices of bottom surfaces of valves (i.e., the valve face) always have *odd* values for IDFACE (1, 3, ...), and the top valve surfaces and stems always have *even* values for IDFACE (2, 4, ...). By this definition, any moving surface, be it piston or valve, that has fluid above it has an even IDFACE. Conversely, if fluid lies below

the face, the IDFACE is odd. All the remaining vertices in the grid, which are not associated with any moving surface, are assigned IDFACE = -1.

## B. Valve Data in File ITAPE5

- NVALVES is the number of valves,  
and for *each* valve:
  - VLIFTMIN, the minimum lift in cm, below which the valve is closed;
  - SKIRTTH, the thickness of the vertical edge of the valve, in cm ;
  - TMOVE, the valve temperature in Kelvins. (Because a temperature for each valve can now be specified, the previous TVALVE line in earlier ITAPE5 files has been deleted);
  - VTILTXZ, the valve cant angle from cylinder axis, in degrees: +, -, or 0.0;
  - NLIFT, the number of lift entries for the valve on file ITAPE18.

## C. Valve Lift Data: File ITAPE18

Input data file ITAPE18 (free format) is a table of crank angles (integers) and corresponding valve lifts in cm (real numbers). The table must provide the lift information for one complete engine cycle:  $0^\circ$  to  $720^\circ$  for a 4-stroke engine, or  $0^\circ$  to  $360^\circ$  for a 2-stroke engine. KIVA-3V determines the length of an engine cycle from the new input quantity REVREP (revolutions between repetition), which should be specified as 2.0 for a 4-stroke engine, or 1.0 for a 2-stroke engine. All crank angles for which the lift is zero may be excluded from ITAPE18, which minimizes the length of the file. In addition, the crank angle increment from one line to the next is not required to be uniform throughout the file. The history for each valve appears in succession, with NLIFT lines per valve.

## D. Valve Movement

Each cycle, subroutine VALVE interpolates the lifts and velocities of the valves at the current crank angle from the data on ITAPE18. If the current crank angle lies outside the table range, the subroutine will use the appropriate equivalent crank angle inside the table range, permitting multiple engine cycles to be calculated. In physical space, the vertices that lie on a valve surface move as a unit each cycle in subroutine REZONE, using the current valve velocity components (UMOVE, WMOVE). Vertices on the valve stem are assigned the valve velocity, which allows the proper wall stress to be calculated and spray particles on the stem to be moved. To preserve the original zoning in the port, these vertices are never actually moved in subroutine REZONE.

## E. Valve Snapping

In grids that have vertical valves, one can use uniform fine zoning across the region of valve travel. This case is treated in a manner entirely analogous to piston snapping, in which a valve surface is snapped when the grid line representing the surface has moved half the distance between its saved initial position and the next grid line in the direction of travel.

In grids with canted valves, it is assumed that there is no nice uniform zoning in the valve lift regions that the snappers can reliably test. Instead, KIVA-3V subroutine SETUP creates tables of fractions of the maximum valve lift, which are used to determine when to snap. This approach requires that the grid line immediately below the valve face must be kept fairly close to the valve face to avoid sudden large changes to the grid when snapping occurs, which could result in cell inversions.

The top and bottom surfaces of valves are snapped from grid plane to grid plane by SNAPVTOP and SNAPVFCE. During the run, the valve will alternate between being one and two cells thick, although its dimensions are always preserved. The snappers are phased so that when the body is one cell thick, they will not make a snap that would cause the valve bottom to become the top or vice versa, which would result in a zero-thickness cell. When the valve is closing, SNAPVTOP will make the valve two cells thick before SNAPVFCE comes up from behind and puts it back to one cell. Conversely, when the valve is opening, SNAPVFCE will make the valve two cells thick before SNAPVTOP comes down from above and puts it back to one cell.

When the valve body is two cells thick, the value of IDFACE is ambiguous at the intermediate vertices around the valve skirt. Because these vertices are not part of either a lower or an upper surface, they do not require specific treatment in the valve snappers, other than to keep them properly placed. The arbitrary convention is to assign the same IDFACE to these intermediate vertices as to the (odd-numbered) valve face below. The valve bodies created by the grid generator should have no more than a two-cell thickness at the valve skirt.

## F. Valve Closing and Opening

As a valve closes during the KIVA-3V run, the grid finally snaps to a minimum one-cell clearance between the valve top and its seat. As the valve closes further, this final cell begins to collapse. If the valve were allowed to seat entirely, a zero-thickness cell would result. To prevent this situation, valve closure automatically terminates when a specified minimum clearance or lift (VLIFTMIN), typically 0.20–0.50 mm, is reached. At this point, the cell face boundary conditions BCL and BCF of the appropriate left, right, front, or derriere

cell faces around the valve periphery are changed from FLUID to SOLID, which closes off the flow through the valve. In subroutine BC, the velocities are simply set to zero on these cell faces.

As the valve opens, the above procedure is reversed: when the valve lift exceeds VLIFTMIN, the cell face boundary conditions are changed from SOLID to FLUID, and flow through the valve commences.

The smaller the value of VLIFTMIN, the closer KIVA-3V will come to matching the physical closing and opening crank angles. The error is most pronounced in lift histories that tail out gradually at very low lift values, but because the mass flux is small at low lift, the error is not as egregious as it would appear. Much smaller values of VLIFTMIN, e.g. 0.10 mm, are observed to have little effect on the overall results, despite the improved accuracy in valve timing, but the time step can be severely reduced when exhaust valves first open and the pressure gradient is large.

Because the law-of-the-wall model in KIVA programs is designed to modify velocities one vertex out from a solid wall, the model is turned off in the valve seat area during opening or closing when the lift is less than two cell widths.

It should be noted that KIVA-3V continues to track the *true* valve lift after the lift falls below VLIFTMIN. The true lift is used in the wall film model to force any remaining liquid fuel out of the valve seat area when intake valve(s) close in port fuel injection applications.

## G. Valve Grids and Continuous Rezoning

To date, KIVA-3V calculations have been either cold flow or full combustion calculations of four different types of IC engine geometries with moving valves:

- 1) Pancake combustion chamber with vertical valves (Fig. 2a-b),
- 2) OHV engine with wedge combustion chamber (Fig. 3a-c),
- 3) DOHC engine with quasi-symmetric pentroof combustion chamber and 4 valves whose paths of travel overlap (Fig. 4a-c), and
- 4) DOHC engine with asymmetric pentroof "cloverleaf" combustion chamber and 4 valves (Fig. 5a-b).

Vertical valve geometries should present no unusual problems in mesh generation, valve snapping, or rezoning. The grids are comparatively robust and

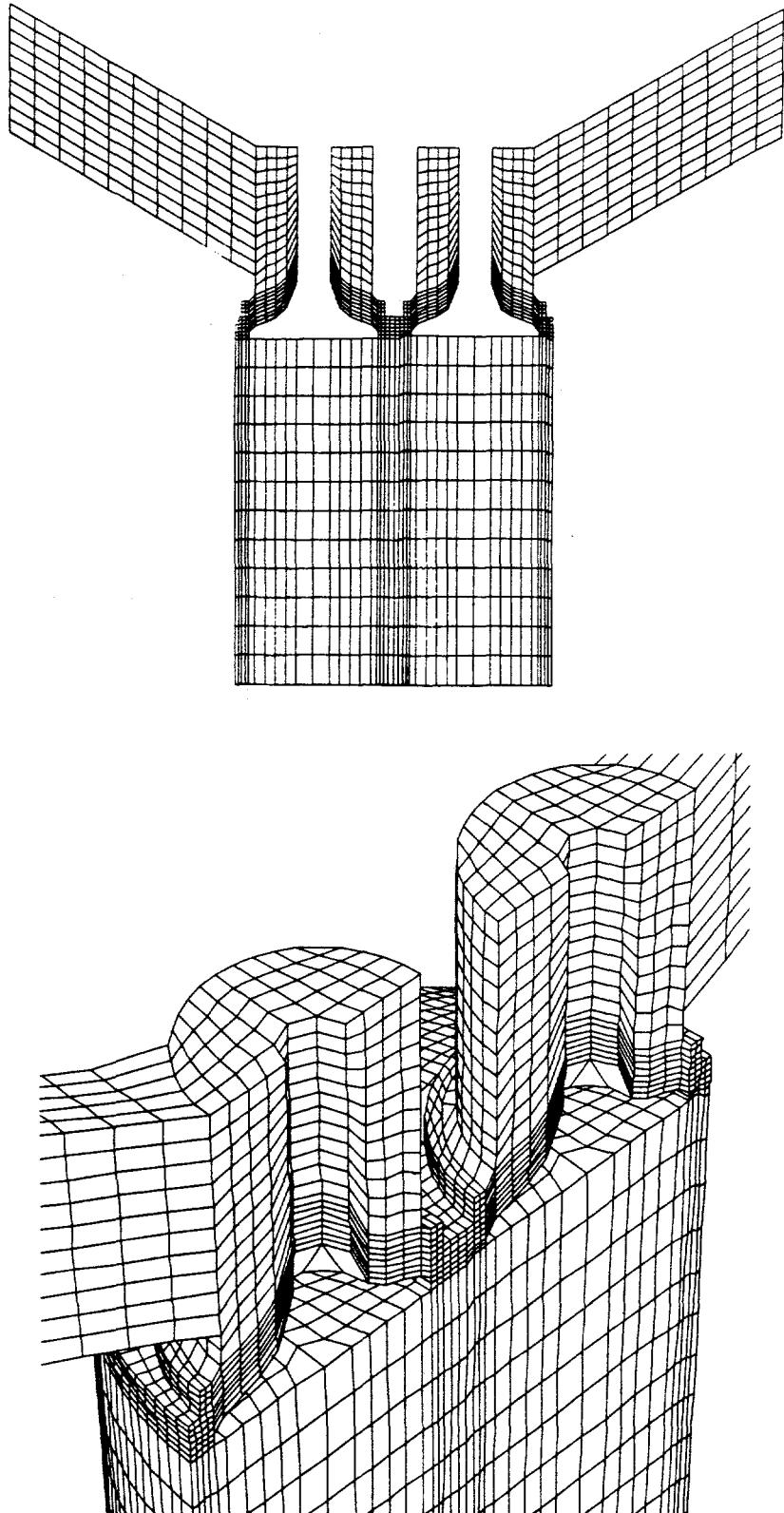


Fig. 2a. Pancake combustion chamber with two vertical valves. Because the engine is symmetric about the  $y = 0.0$  line, only one half of the full geometry is zoned. Top: the initial grid as created by K3PREP, with the valves in their BDC positions. Bottom: a perspective view that shows details of the valve lift region.

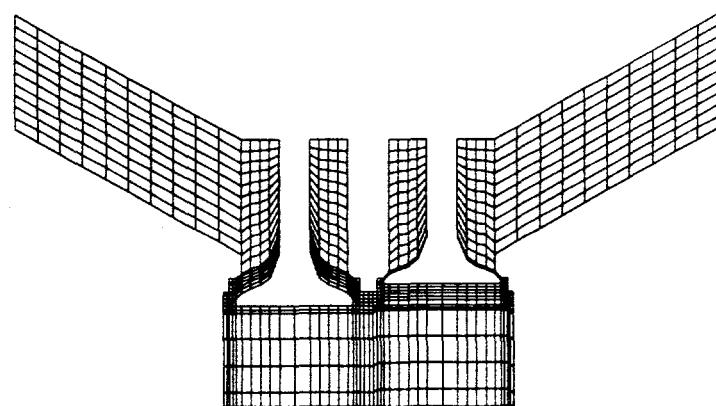
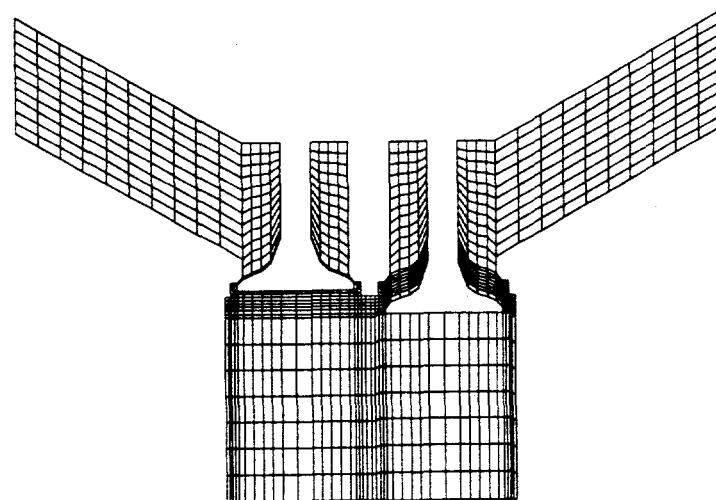
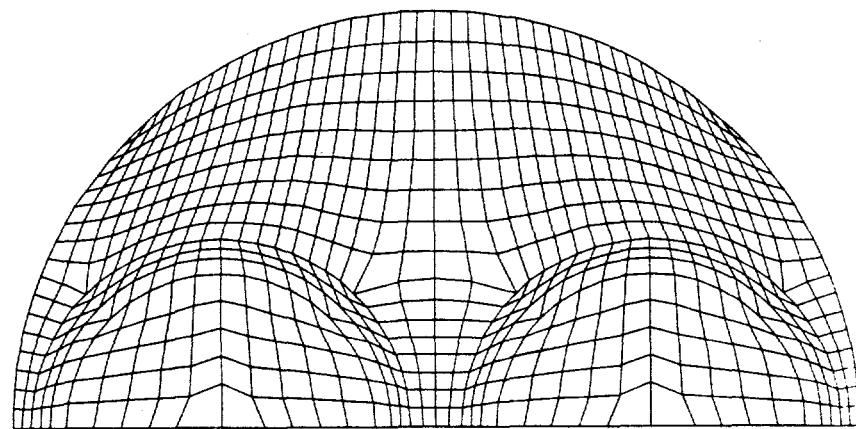
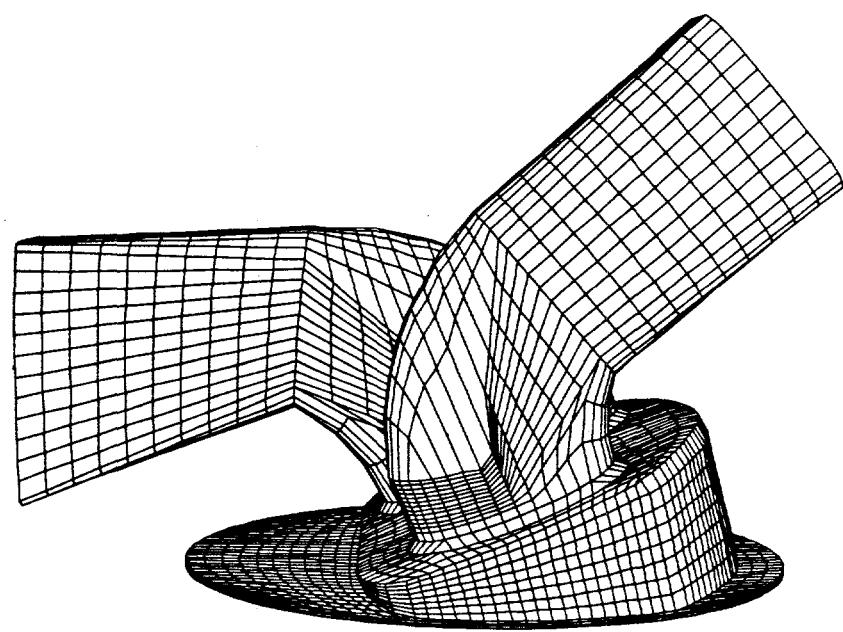
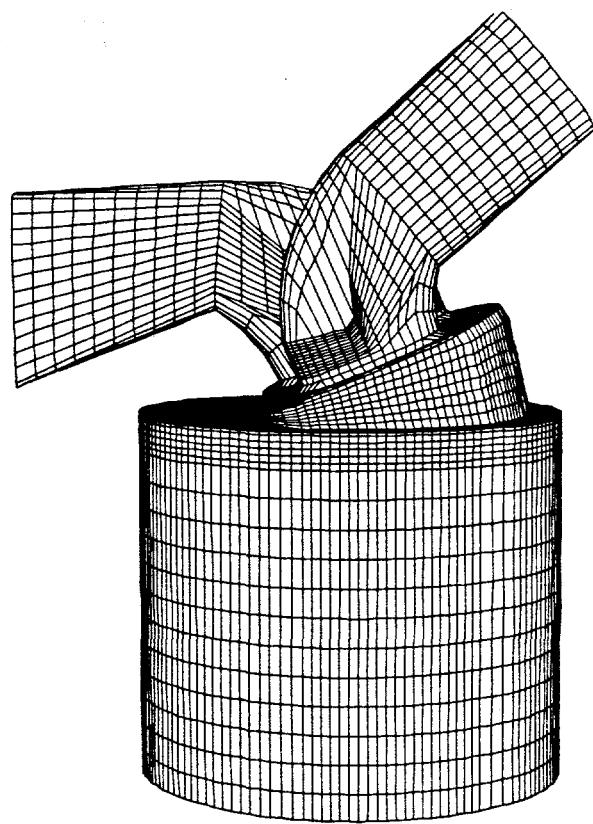
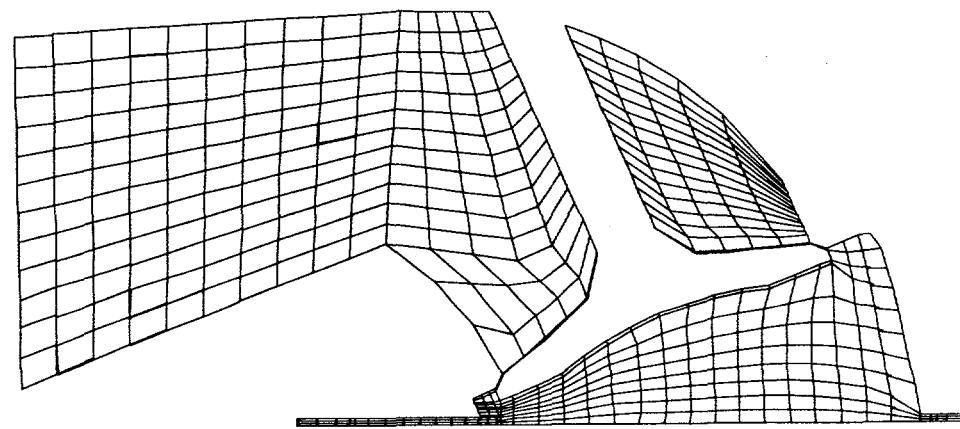
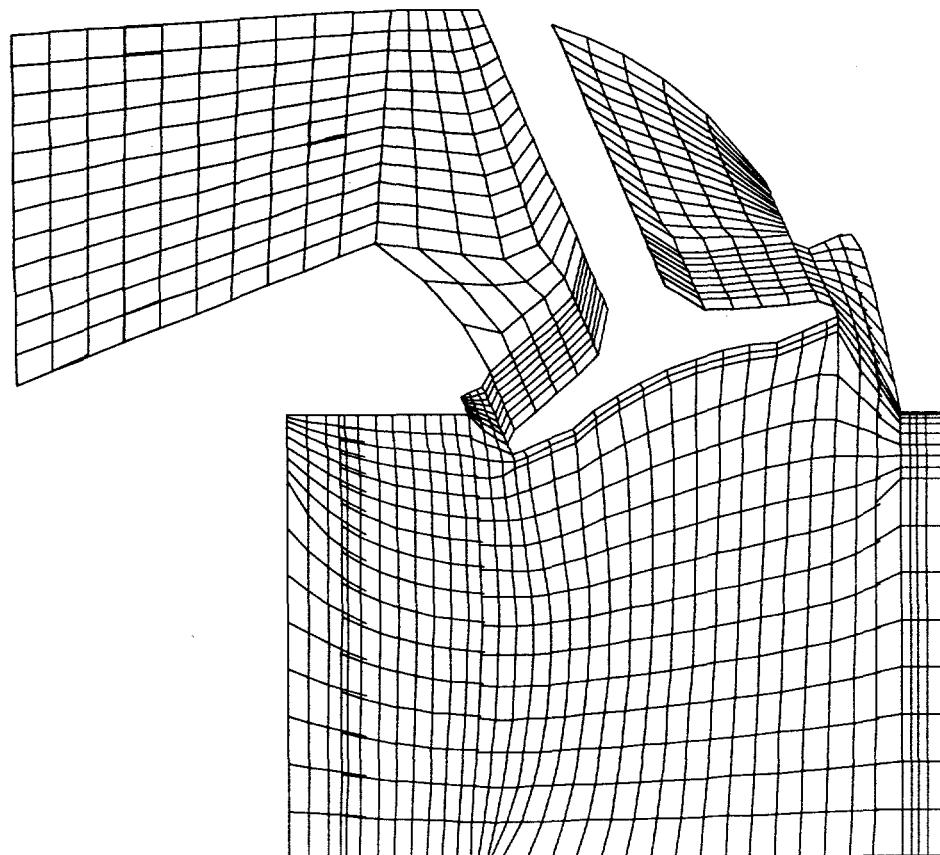


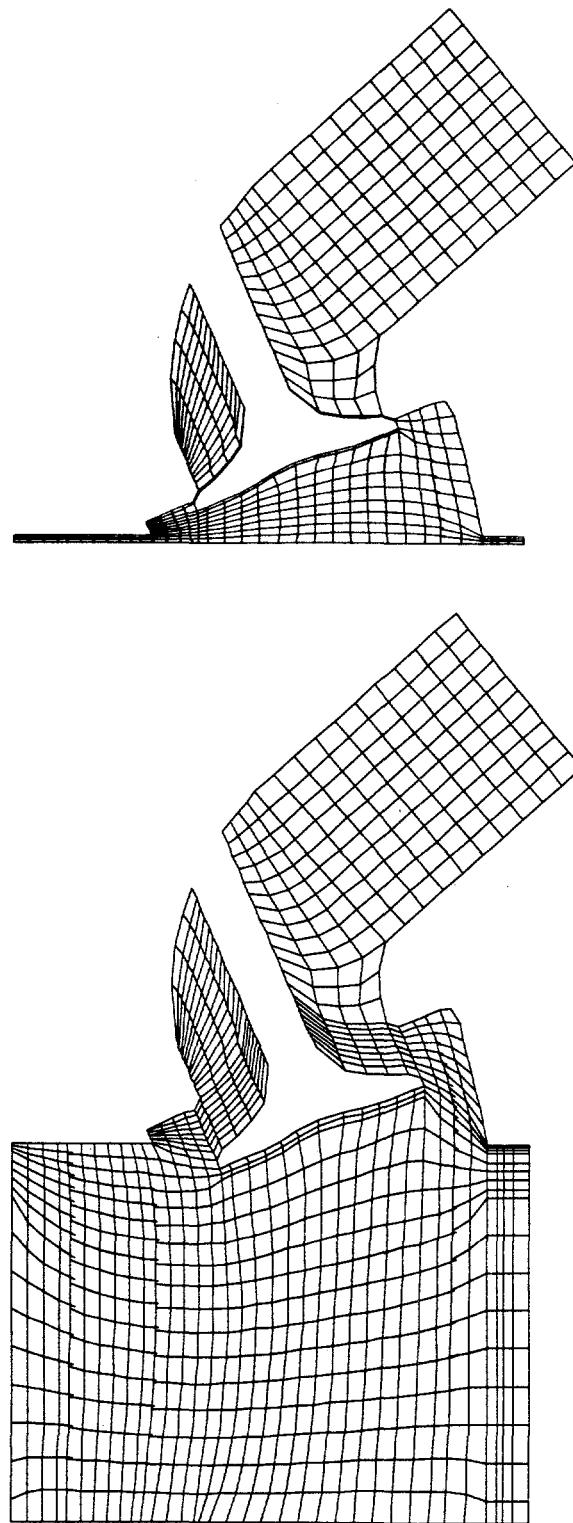
Fig. 2b. Top: plan view of the piston face in the vertical valve grid. Middle: the grid at 90° ATDC, when the intake valve (right) is at its BDC and the exhaust valve (left) is closed. Bottom: the grid at 660° ATDC, when the intake valve is closed and the exhaust valve is mostly open.



*Fig. 3a. OHV engine with wedge combustion chamber and 2 valves canted at  $-23^\circ$ . Top: the grid with the piston at BDC. Bottom: the grid with the piston at TDC.*



*Fig. 3b. Cross sections of the OHV wedge combustion chamber grid through the intake valve ( $y = 2.286$ ). Top: at 104° ATDC, when the valve is at its BDC. Bottom: at 360° ATDC. Note that at least one logical plane is maintained close to the valve face, to minimize the risk of cell inversions when the valve is snapped.*



*Fig. 3c. Cross sections of the OHV wedge combustion chamber grid through the exhaust valve ( $y = -2.540$ ). Top: at  $360^\circ$  ATDC. Bottom: at  $609^\circ$ , when the valve is at its BDC.*

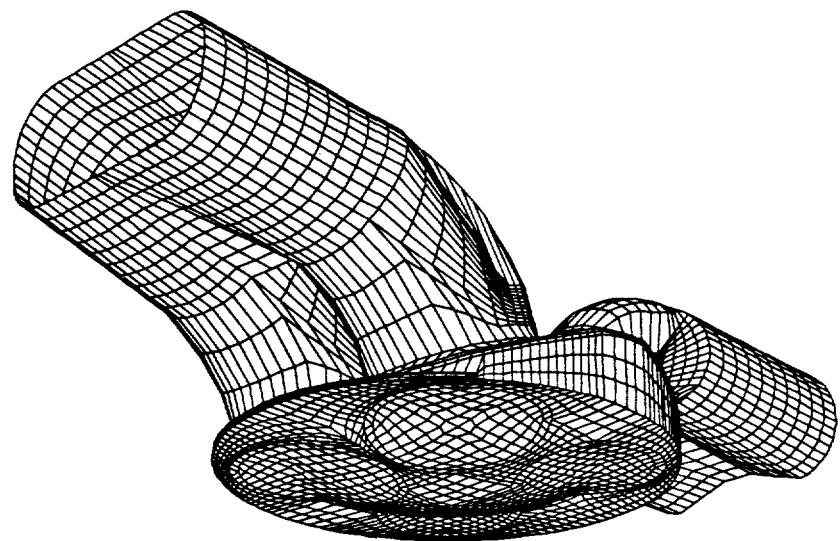
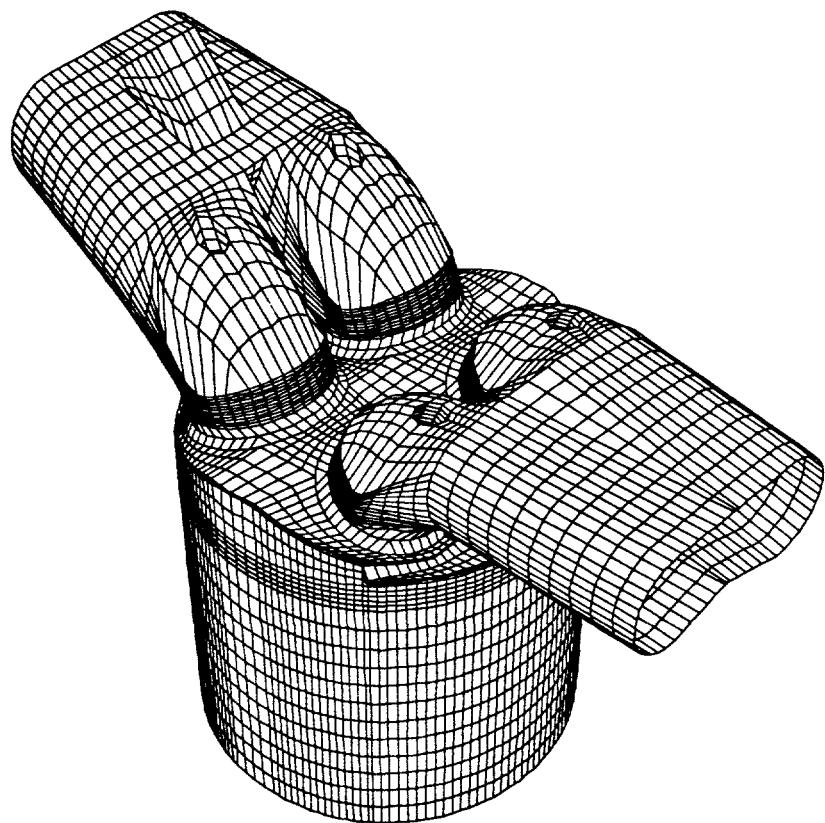
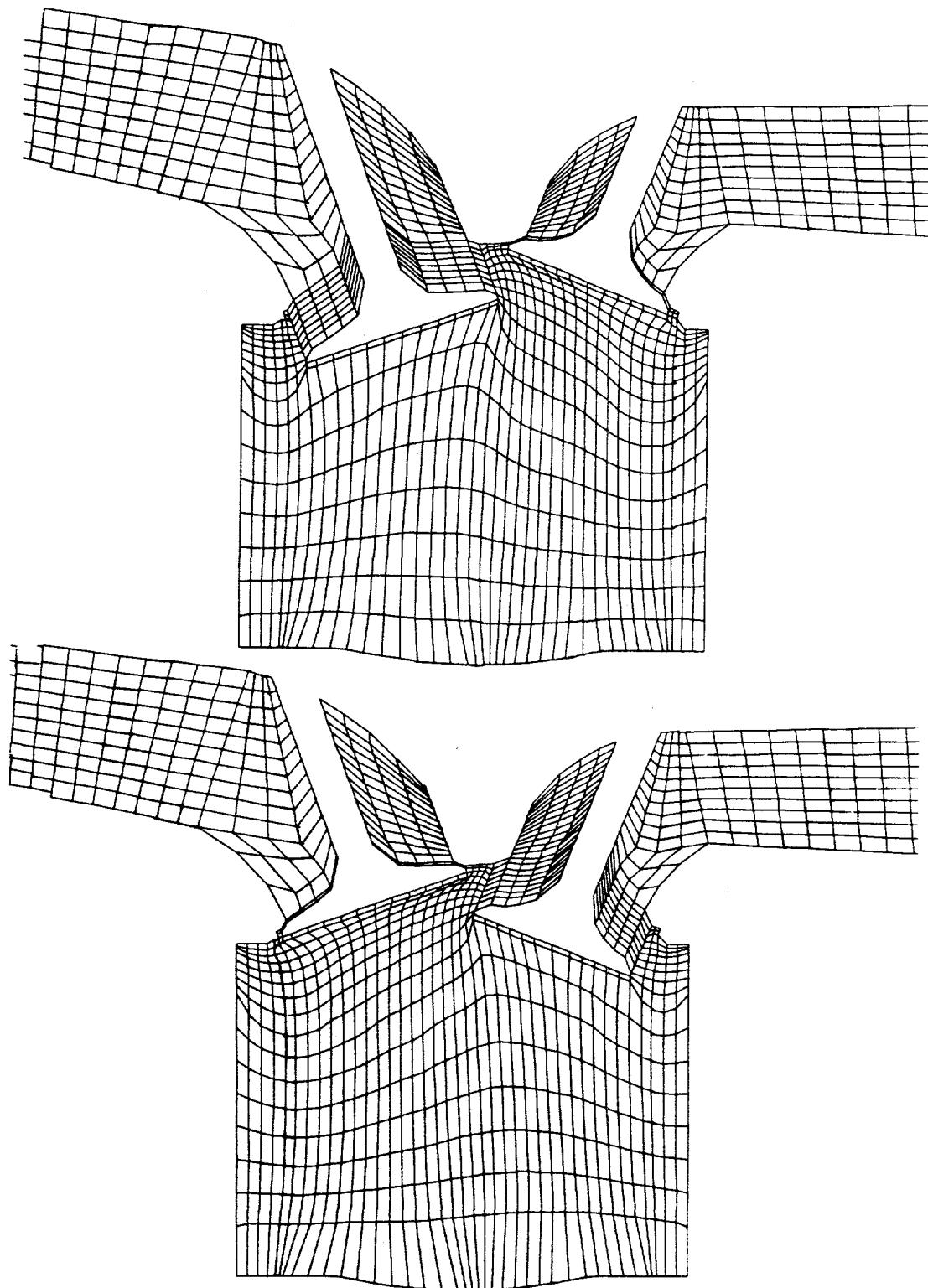
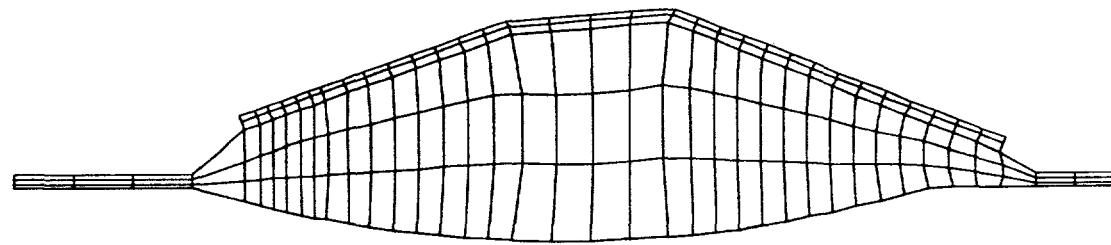
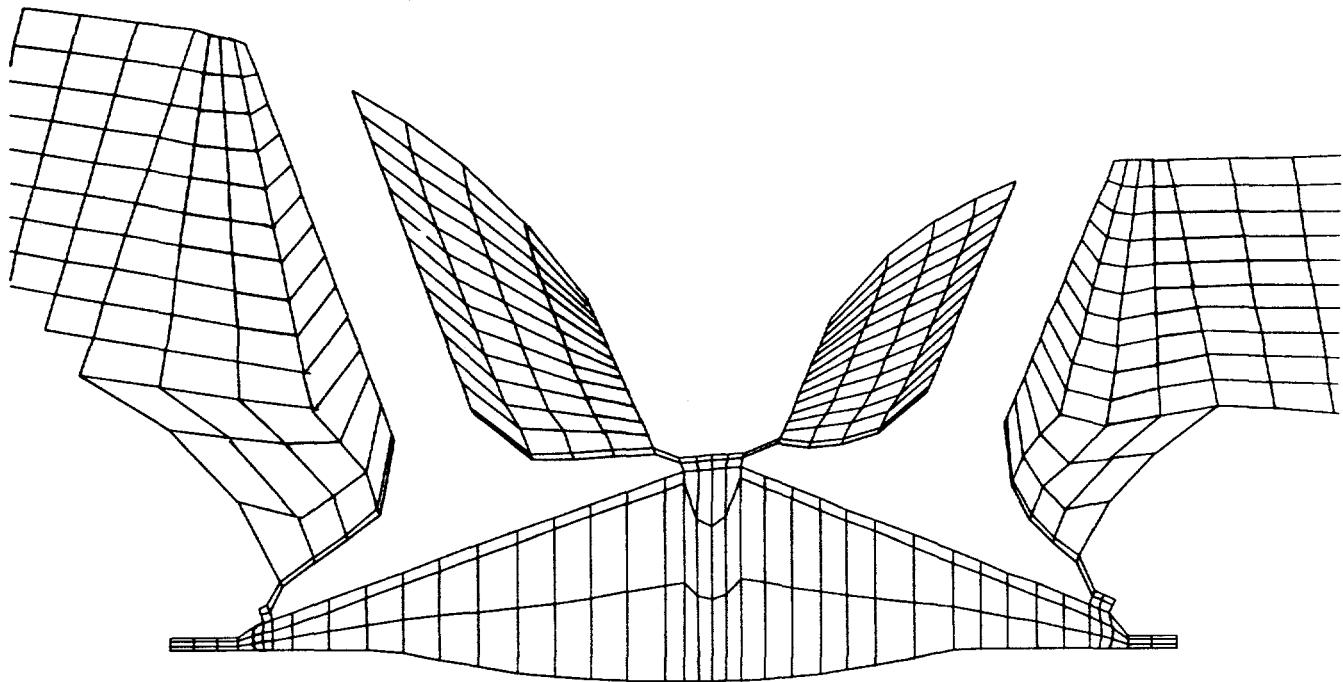


Fig. 4a. DOHC engine with quasi-symmetric pentroof combustion chamber and 4 valves. The intake valves are canted  $-20^\circ$  and the exhaust valves are canted  $+22^\circ$ . Top: the grid with the piston at BDC, showing the fuel injection port. Bottom: the grid with the piston at TDC, showing the shallow dish in the piston crown.



*Fig. 4b. Cross sections of the quasi-symmetric pentroof combustion chamber grid, through one intake valve (left) and one exhaust valve (right), at  $y = +$  or  $-1.95$ . Top: at  $104^\circ$  ATDC, when the intakes are at their BDC. Bottom: at  $610^\circ$ , when the exhausts are at their BDC.*



*Fig. 4c. Cross sections of the quasi-symmetric pentroof combustion chamber grid at 360° ATDC. Top: through one intake valve (left) and one exhaust valve (right), at  $y = +$  or  $-1.95$ . Bottom: through the  $y$ -axis of the cylinder, at  $y = 0.0$ .*

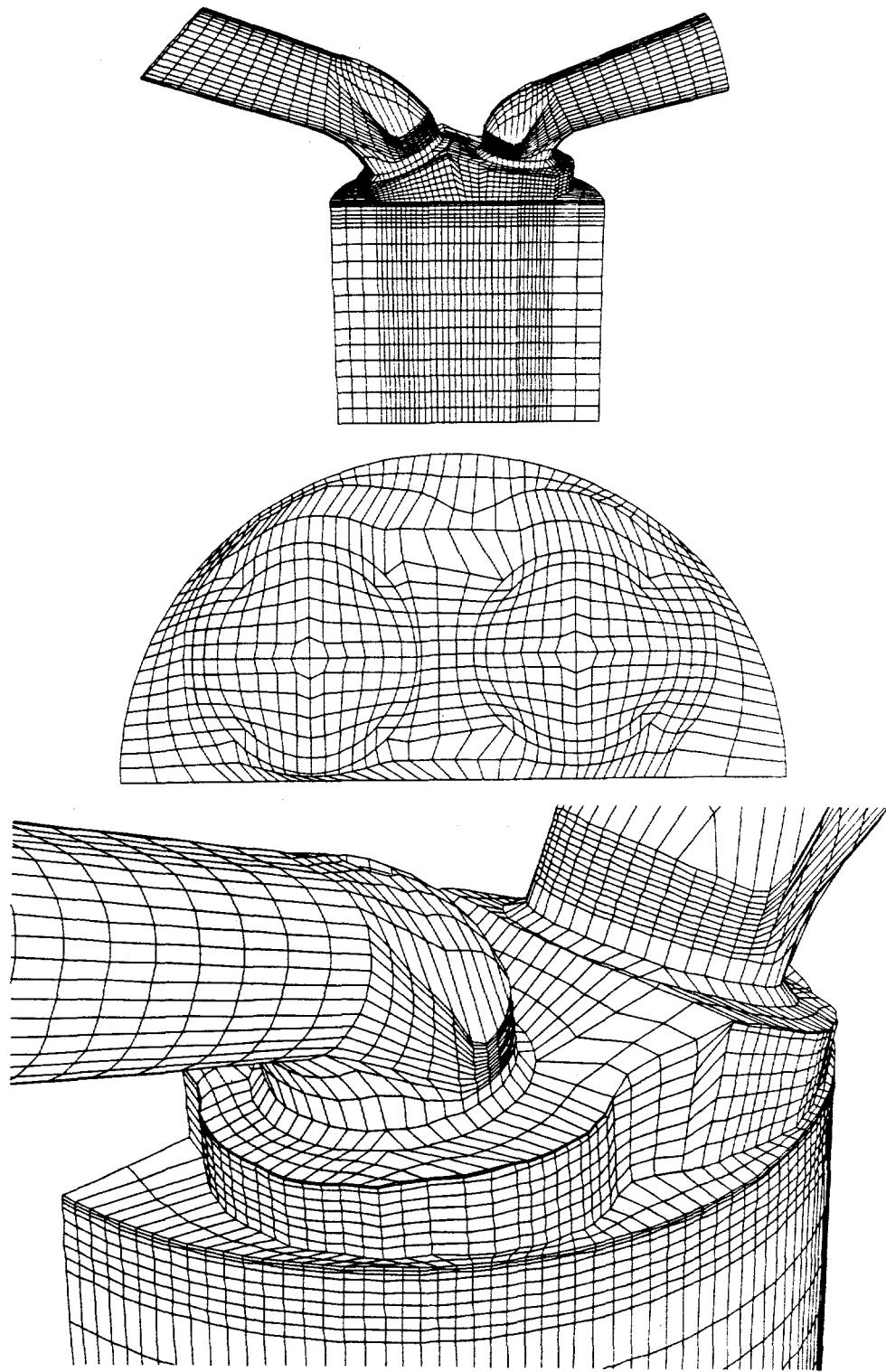


Fig. 5a. DOHC engine with asymmetric pentroof cloverleaf combustion chamber and 4 valves. The intake valves are canted  $-25^\circ$  and the exhaust valves are canted  $+7^\circ$ . Because the geometry is symmetric about the  $y = 0.0$  line, only one half of the full geometry is zoned. Top: a perspective view of the initial grid as created by K3PREP. Middle: a plan view of the piston face. Bottom: a perspective view showing the cloverleaf shape.

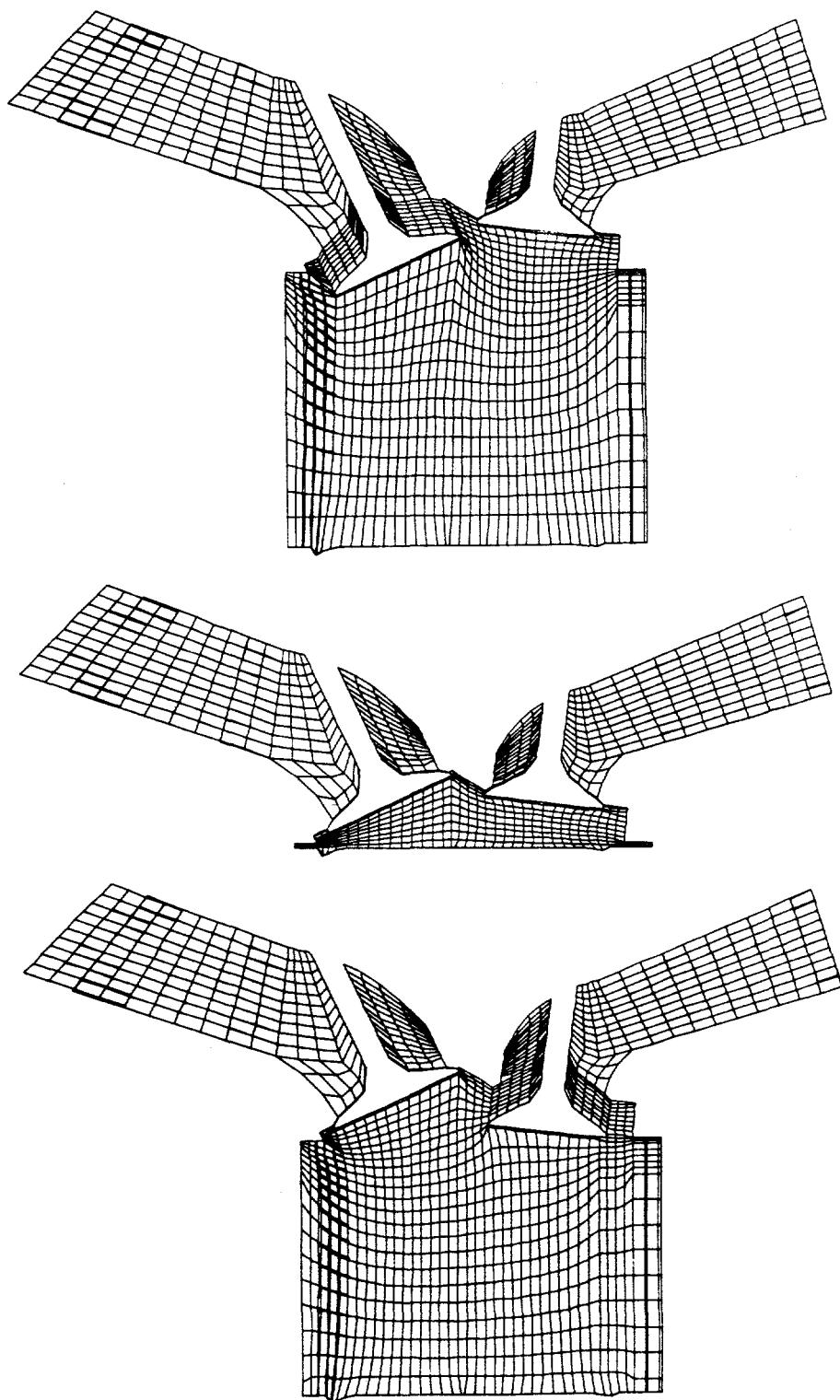


Fig. 5b. Cross sections of the asymmetric pentroof mesh through the valves, at  $y = 1.75$ .  
 Top: at  $117^\circ$  ATDC, when the intake valves are at their BDC. Middle: at  $360^\circ$ . Bottom: at  $609^\circ$ , when the exhaust valves are at their BDC.

may be expected not to create any inverted cells during the KIVA-3V run. The initial mesh typically will have the valves at their lowest (BDC) positions. At the beginning of the KIVA-3V run, the valves are snapped upward as necessary in subroutine ADJVALVE to match the starting crank angle ATDC.

Unfortunately, in the canted-valve cases 2) to 4), there is usually no single grid that will be satisfactory throughout the full engine cycle. *This is not simply a matter of generating the initial grid: The grid must dynamically change during the run in response to the changing valve positions.* Bear in mind also that we are always dealing with a block-structured mesh of hexahedral cells, which imposes restrictions on the rezoning of the mesh.

Note also that we cannot generate the grid for case 3) with the valves in their lowest positions, because the intake and exhaust valves would simultaneously occupy the same space. Furthermore, it is impossible to generate a mesh with nice uniform planes of cells through the region of valve lift. Accordingly, these grids should be generated with the valves in their *closed* (TDC) positions, and provide a sufficient number of planes near the top of the cylinder to allow for valve snapping. In addition, one must consider combustion chamber resolution at the time of combustion, when the piston is at its TDC. As shown in the sample meshes, one should maintain at least a bare minimum of three planes of cells in a pentroof combustion chamber between the valves and the piston, and at least nine planes to resolve a wedge combustion chamber.

The approach of a dynamically changing grid is made possible by using a continuous rezoner that will attempt to maintain an acceptable grid throughout the full engine cycle. Until a fully general rezoner is developed, subroutines REZWEDGE, REZPENT, and REZCOMB in KIVA-3V should only be considered placeholders, as they are tailored to the specific geometries of cases 2), 3), and 4), and contain some hard-wired dimensions in various tests. However, these rezoners provide a useful starting point for users of KIVA-3V, because they illustrate a number of rezoning problems that arise during the engine cycle, and possible ways to deal with them. The problem areas include

- the central region shared by overlapping intake and exhaust valves, which is helped by biasing the grid relaxation upward when (but only when) valves are open,
- the narrow region between the face of an opening valve and the piston crown near TDC, in which the grid lines need to be kept nearly vertical,
- the grid just below where the wedge meets the bottom of the head, in which the vertical spacing needs to be carefully distributed to avoid cell inversions, and

- the grid lines that go outward in (i,j) logical space from the skirts of moving valves.

## H. Grid Generation with Valves

The grids in Figs. 2 to 5, which were created by K3PREP without modification, illustrate a number of geometric capabilities that are desirable in any generator for KIVA-3V grids with valves. These include:

- Tilting portions of the grid to the left or right.
- Shaping of ports and runners.
- Joining paired runners.
- Profiling valve tops and faces.

Along with new K3PREP subroutines and associated input data that support the above, other new subroutines and input data have been added to K3PREP. These include:

- Copying of identical blocks, such as paired valves.
- Translating blocks originally centered at the axis to their actual locations.
- Reshaping of interior cylinder grid points to conform to circular shapes.
- Specifying z coordinates up the cylinder wall.
- Specifying (x,y) points on the periphery of the combustion chamber.
- Creating a shallow dish or valve scallops in the piston crown.

The usage of K3PREP with these new features is now discussed in the context of cases 1) to 4).

## IV. EXAMPLES WITH VERTICAL AND CANTED VALVES

### A. Vertical Valves

Figure 2 showed several views of the computing grid for a two-valve engine with vertical valves, square runners, and a flat piston crown. This is about as simple a practical valve geometry as possible, but it is one that proved quite satisfactory in a real programmatic application. Because a plane of symmetry can be passed through the centers of the valves and ports at  $y = 0.0$ , only one half of the full geometry is zoned.

The valve lift region is more finely zoned, starting at the level of the valve face when the valve is at its maximum lift, and the curved profile of the top of each valve is repeated uniformly in the port up through the region of valve lift, which is 8 cells in this example.

No special rezoning is required in this mesh, and the movement of moving surfaces is handled by subroutine REZONE in the traditional manner. The interior mesh does not move, in order to preserve the original zoning in the valve lift region.

The circular shapes of the valve stems, skirts, and recesses are carried downward through the cylinder all the way to the piston face. Although this is not an absolute requirement, as will become evident in the subsequent examples, it is nonetheless desirable to maintain vertical grid lines to the greatest extent possible through the squish region. Such zoning will allow the greatest accuracy in the remapping of cell quantities in the piston snapper, and reduce the risk of cell inversions during the run.

Figure 6 is a drawing of the logical mesh that corresponds to the grid shown in Fig. 2. A schematic drawing is useful for writing and understanding the IPREP file. A total of 41 blocks are required in this implementation, of which *only the first 9* are actually saved at the end of grid generation. Blocks 10–41 are used only for repositioning vertices inside the cylinder and the vertical ports into circles to conform to the circular shapes of the valve stems, skirts, and recesses, and are then discarded. Hessel came up with this idea, and he called these temporary blocks "reshaping blocks." Figure 7 shows the detailed dimensions in the valve seats, which differ between the intake and exhaust sides. Figure 8 illustrates the K3PREP block types required in a mesh with valves. Note that the NVALPRT blocks extend straight down, and stop at the level of the head. A plan view drawing (Fig. 9) of the logical mesh with the blocks numbered is helpful when writing the lines of reshaping and patching commands.

**Cylinder:** From left to right, then moving up, the fluid blocks in the cylinder encompass four distinct volumetric levels: the level between the piston

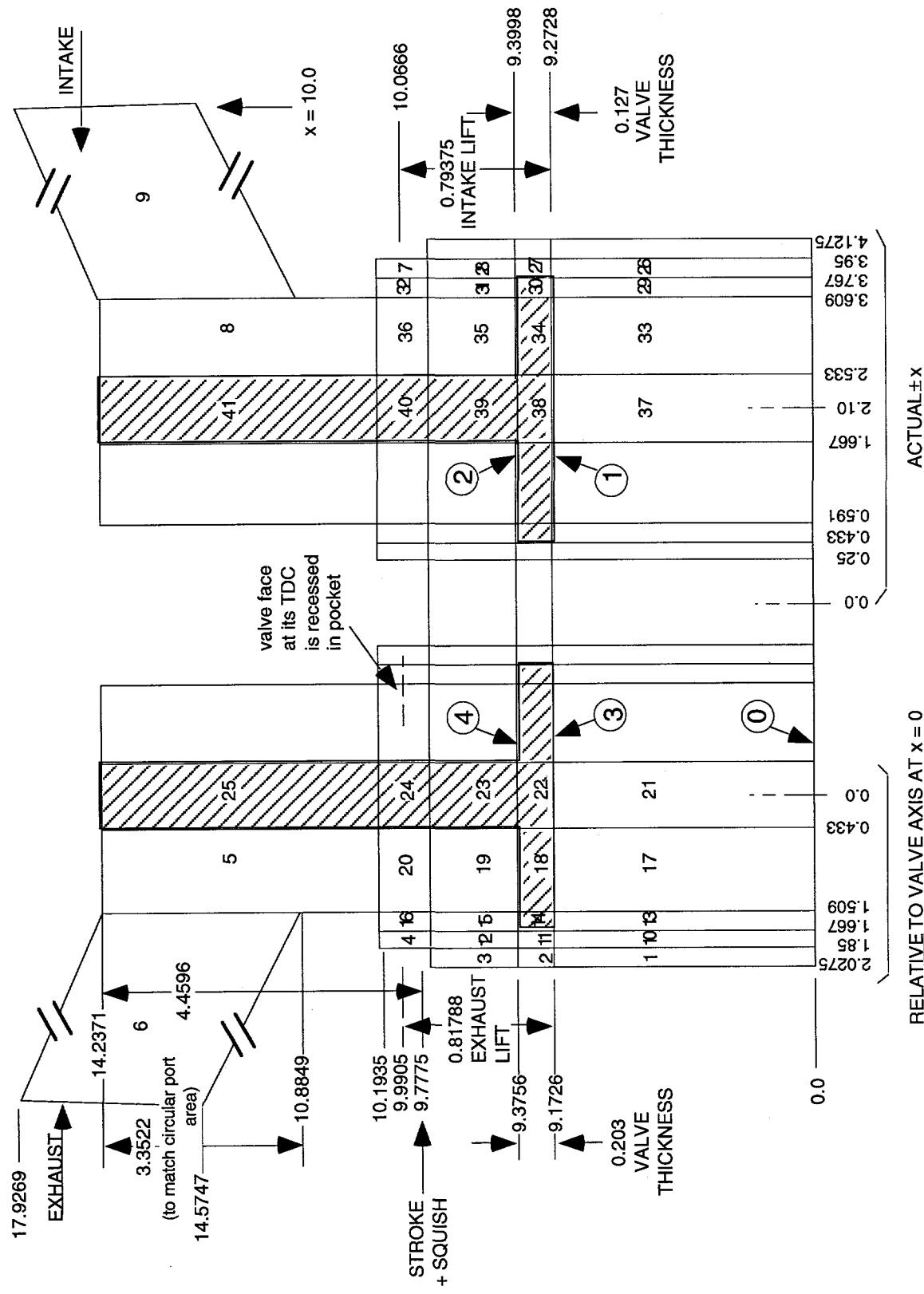
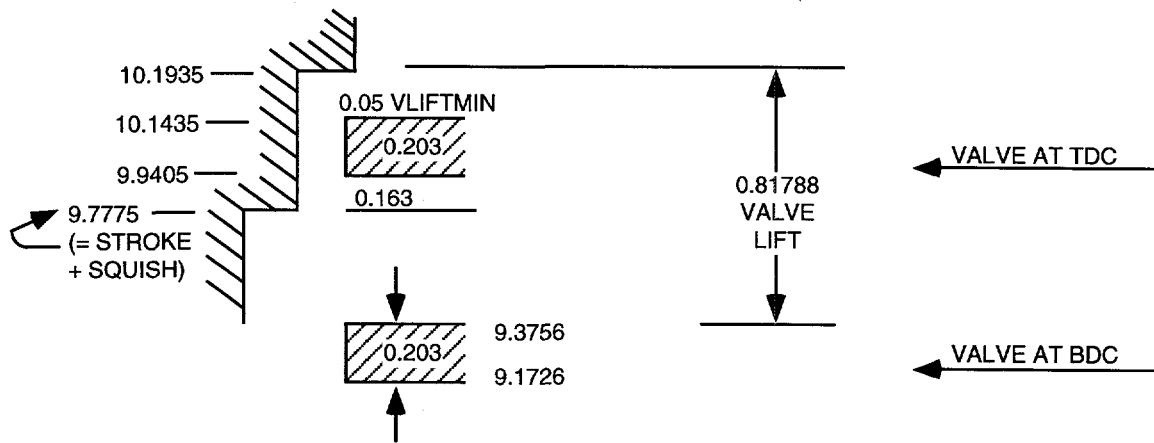


Fig. 6. The logical mesh for the vertical-valve example, showing the numbering of the blocks for K3PREP, all necessary dimensions, and the numbering of the moving surfaces.

## EXHAUST VALVE:



(Note that Z's at BDC of valves are chosen to give the correct maximum valve opening. Accordingly, the minimum valve lift (VLIFTMIN = 0.05) must be included in the valve lift)

## INTAKE VALVE:

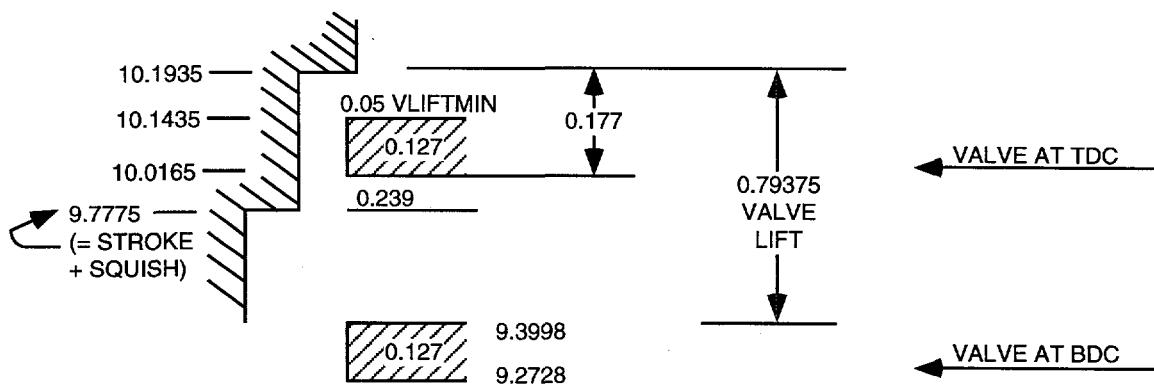


Fig. 7. Dimensions of the exhaust and intake valves and in the valve seat area, for the vertical-valve example.

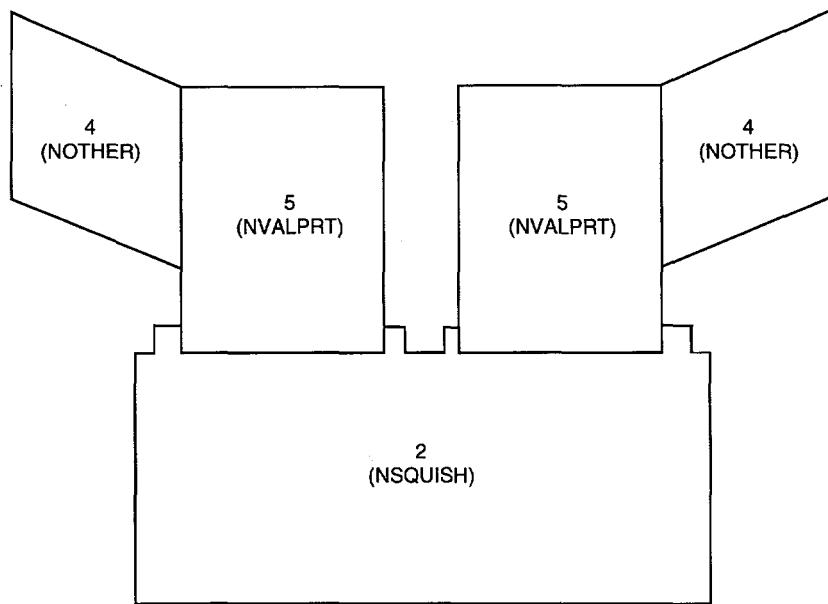


Fig. 8. The K3PREP block types in a mesh with valves. Note that the NVALPRT blocks extend straight down, and stop at the level of the head.

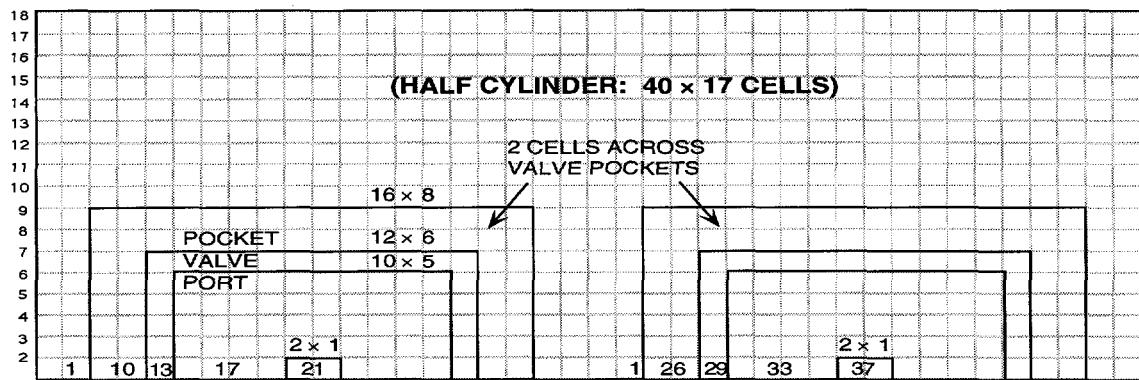


Fig. 9. Plan view of the logical mesh for the vertical valve example, with vertex and block numbering, useful for writing the reshape and patch commands for K3PREP.

and the valves (blocks 1, 10, 13, 17, 21, 26, 29, 33, 37), the level of the valve body (2, 11, 27), the level of valve travel below the valve pockets (3, 12, 15, 19, 28, 31, 35), and the level of the valve pockets (4, 7, 16, 20, 32, 36).

**Valves:** The valves themselves are treated as internal obstacles in the mesh. These are "ghost" blocks, and the cells that comprise them have  $F = 0.0$ . Only the bodies of the valves (blocks 14, 18, 22, 30, 34, and 38) are moved and snapped. Vertices on the stems (blocks 23–25 and 39–41) are assigned the valve velocity for use by the liquid wall film model in KIVA-3V, but are never moved during the calculation.

If the valves are recessed into the head when they close, as in this example, the pockets (blocks 4 and 7) must be zoned with a minimum two-cell radial clearance between the valve skirt and the head, to avoid the calculation of fictitiously large velocities in the pockets when the valves are closed.

**Ports:** The ports and runners in this example are about as simple as possible, and are formed from blocks 5, 6, 8, and 9, without any attempt to shape them. The (x,y) coordinates at port-runner interfaces are chosen to give the same area open to flow as that of the circular port.

## B. OHV Engine with Wedge Combustion Chamber and 2 Valves

Figure 3 showed several views of a K3PREP mesh patterned after a typical wedge geometry, and was based in part on the plan view of the combustion chamber and valves shown in Fig. 10. It is evident that no symmetry plane can be passed through this engine, thus requiring the user to create a grid for the full geometry. Because grid lines around the cylinder wall are required to be horizontal through the range of the piston stroke (as in the original KIVA-3), the combustion chamber is built from blocks that sit on top of the cylinder. In the drawing of the logical mesh (Fig. 11), note that block 2 is six planes high, to provide at least modest resolution on the spark-plug side of the chamber. Unfortunately, these six planes are very compressed on the left where they meet the squish region. To match the real engine, they should be squeezed down to nearly zero height, a more accurate representation that has not yet been pursued. Even so, care must be taken when the valves are opening (especially the intake) to prevent cell inversions. Note also that the valves are generated in their *closed* positions, with the heights of blocks 4, 12, 16, 20, 25, 29, and 33 set equal to  $VLIFTMIN = 0.025$  cm.

Figure 12 illustrates the z coordinates that are specified up the cylinder wall. Here, twelve uniform planes are used up to the average valve lift level ( $z = 8.20005$ ). Above this level, the zoning is finer. Note in particular that the coordinates are carefully chosen so that three cylinder planes will remain at TDC.

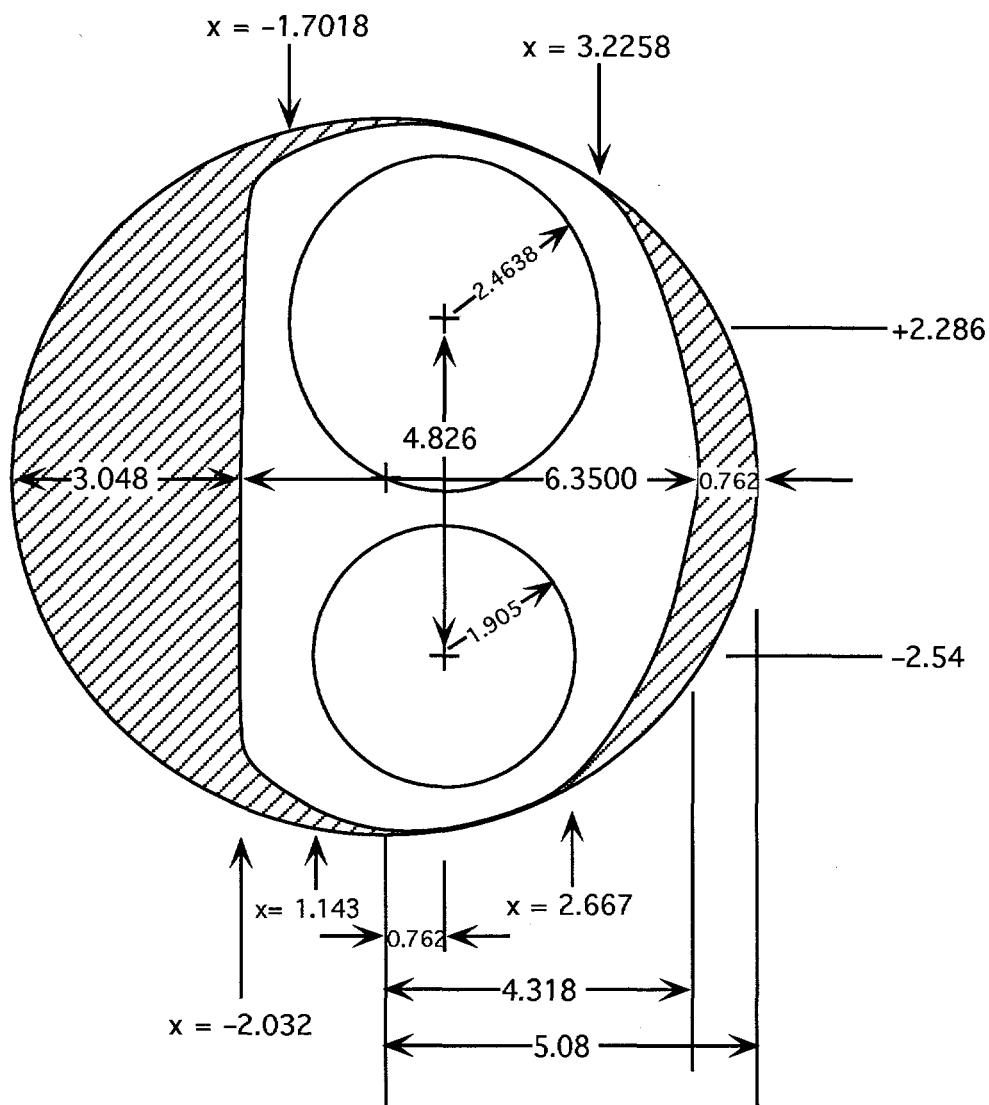


Fig. 10. Combustion chamber and valve dimensions for the OHV wedge geometry example.

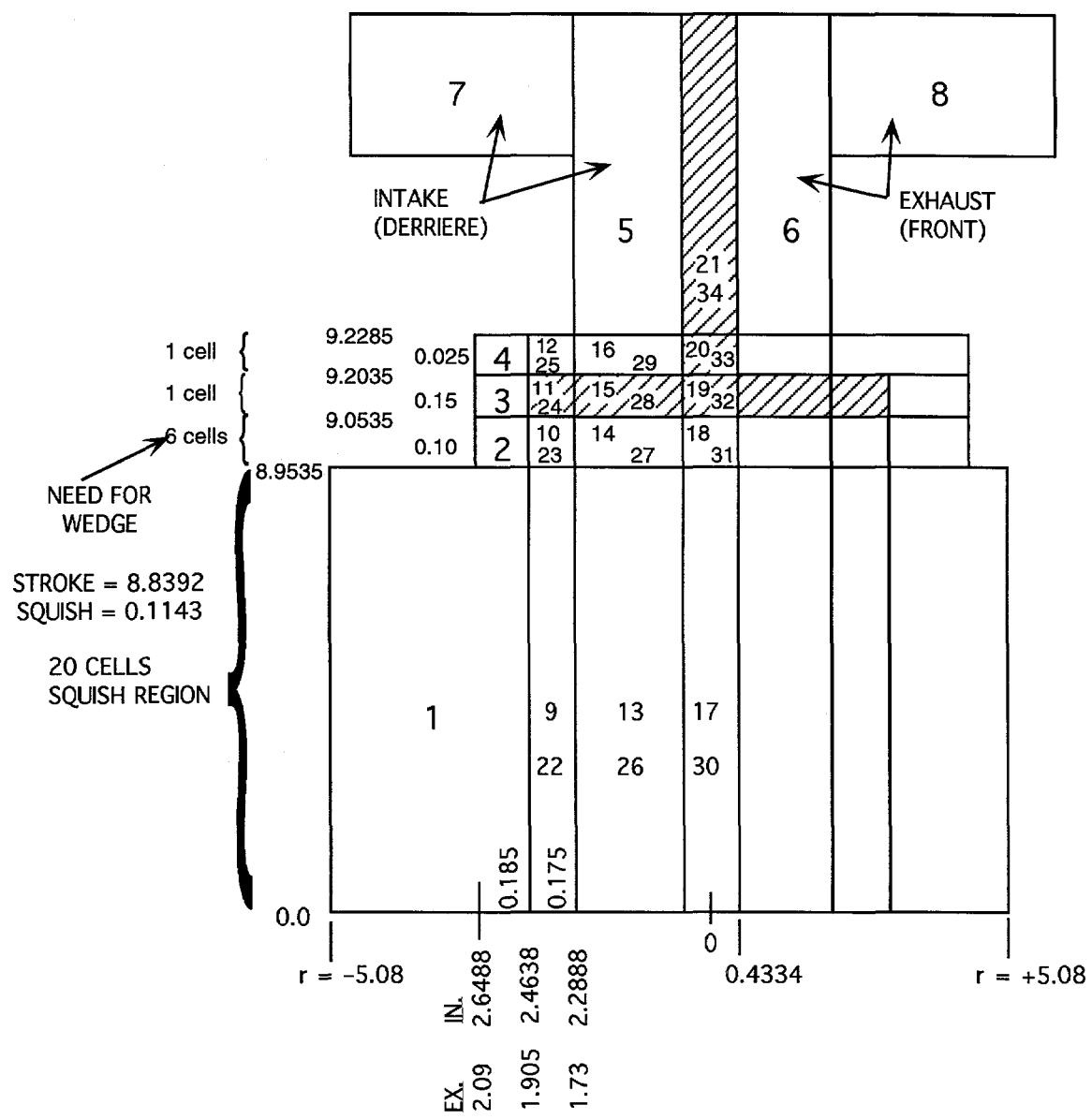


Fig. 11. The K3PREP logical mesh for the OHV wedge geometry example.

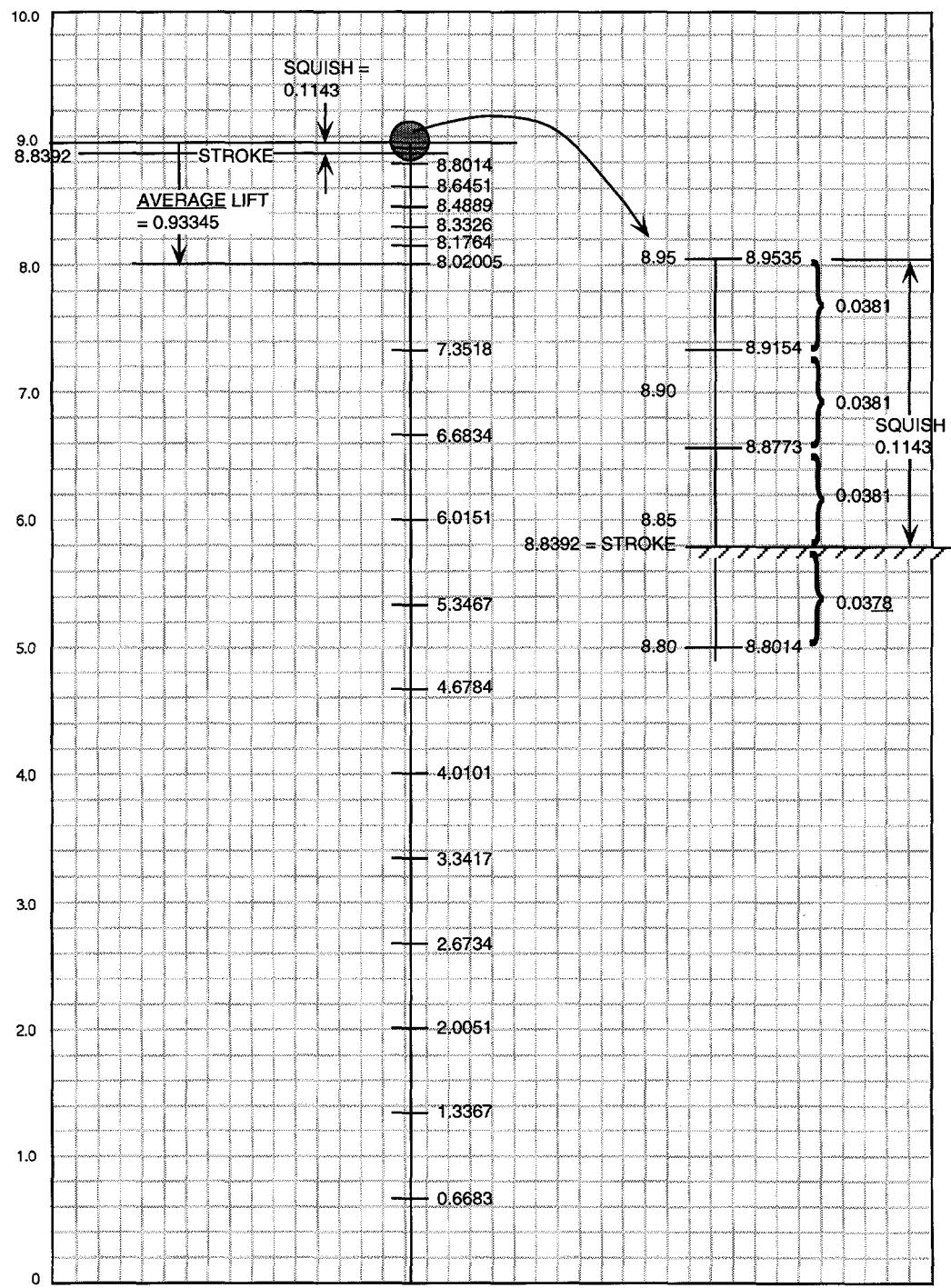


Fig. 12. The z coordinates on the cylinder wall for the OHV wedge geometry example are carefully chosen so that a minimum of three cylinder planes will remain when the piston is at TDC.

Figure 13 is a plan view of the logical mesh, as used for writing the reshaping and patch commands.

The new NLOCXY option is used to enforce the desired combustion chamber shape in (x,y) space for points not lying on the cylinder wall. An expanded NBO option (described below), used for block 1, allows the specification of points where the combustion chamber meets the cylinder wall.

K3PREP subroutine TILTWEDG is a template, and is intended for use as a starting point for the generation of 2-valve OHV wedge geometries.

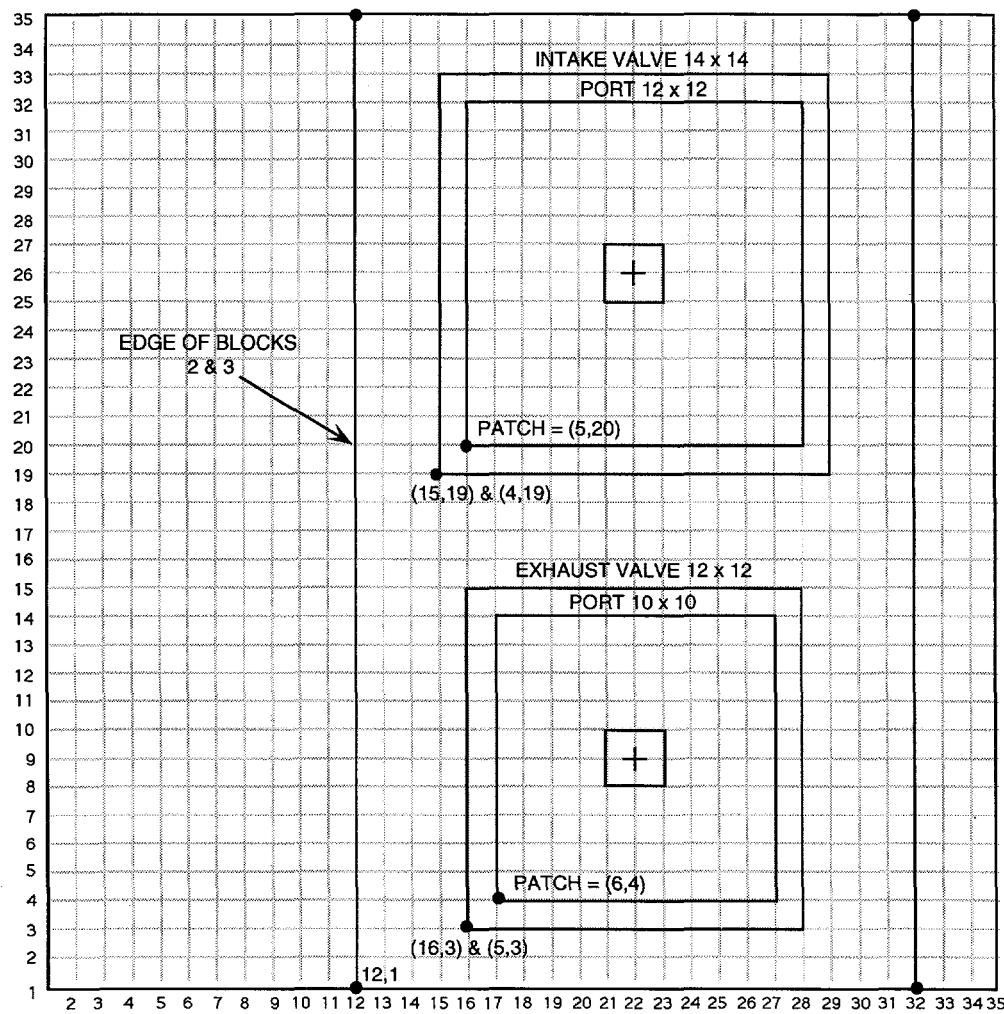


Fig. 13. Plan view of the K3PREP logical mesh for the OHV wedge geometry example.

### C. DOHC Pentroof Engine (Lift Paths Overlap)

The 4-valve pentroof geometry of Fig. 4 presents some different grid problems. As one might infer from the plan view of the geometry in Fig. 14, the mesh is symmetric about the  $y = 0$  line. In our intended application, however, the intake valves are slightly out of phase with one another by several crank angle degrees, and the exhaust valves are similarly out of phase. (The purpose of this design is to damp vibrations and oscillations in the valve train, but the intent is nevertheless to achieve symmetric gas flow within the cylinder.) As a test, this mesh for the full geometry was created first, to verify that the flow is indeed symmetric for all practical purposes even with the slightly out-of-phase valve lifts. Compared to the weeks that were required to create the initial mesh, it only took an afternoon to modify IPREP to generate instead a half-circle symmetry mesh for our subsequent studies, which allowed a 40% reduction in storage and computer time.

Note also in Fig. 14 that the distance between intake and exhaust valves is such that in reality the circles must overlap one another before the valves are canted (Fig. 15), and indeed this is the case in the IPREP data for this mesh.

Figure 16 is a logical mesh for this pentroof geometry. Note that only blocks 1–16 are saved and patched together, while blocks 17–56 are used for reshaping, and are then discarded. Blocks 17–36 are almost identical to blocks 37–56, differing only in their  $(x,y)$  location, region number IDREG, and moving face identifiers IDFFACE. The block copying option in K3PREP (described below) allows defining only blocks 1–36, then simply to copy 17 to 37, 18 to 38, ..., 36 to 56.

Figure 17 is a plan view of the logical mesh, indicating block numbers and vertex indexes, for writing the block copying, reshaping, and patching commands.

The pentroof combustion chamber and valve seating regions are defined by blocks 2 and 3, but at TDC the top three planes of block 1 also form part of the combustion chamber. Just as in the wedge example, the  $z$  coordinates at the top of the cylinder are chosen to ensure the existence of three cylinder planes.

The intake valves are canted  $20^\circ$  to the left, and the exhaust valves are canted  $22^\circ$  to the right. The logic for tilting and continuous rezoning for the left and right sides of the mesh is nearly the same because these two angles are nearly the same.

K3PREP subroutine TILTPENT is a template, and is intended for use as a starting point for tilting and relaxing 4-valve pentroof geometries, either half-circle or full-circle. The trickiest piece of this puzzle, the continuous rezoning of

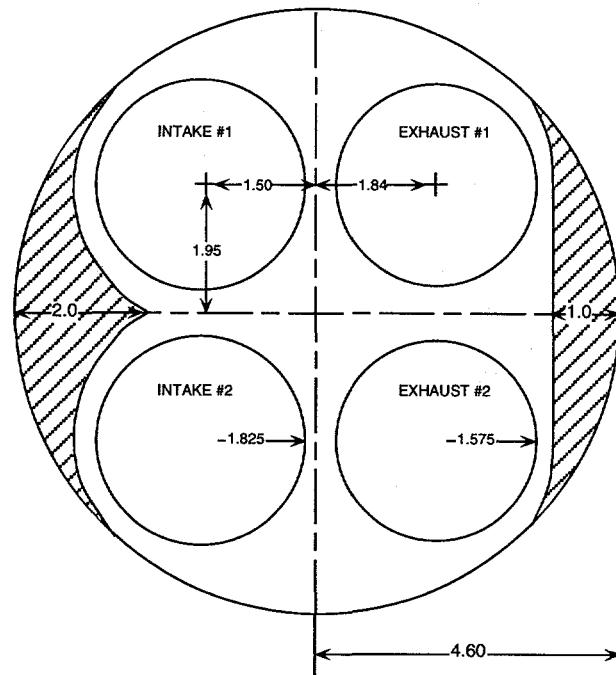


Fig. 14. Plan view of combustion chamber in the quasi-symmetric DOHC 4-valve pentroof example.

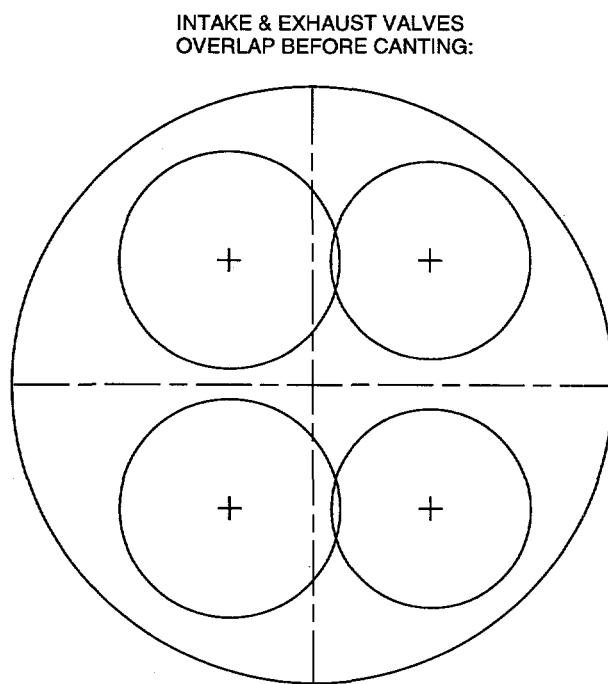


Fig. 15. The intake and exhaust valves overlap in physical space before they are canted, which is evident from the dimensions given in Fig. 14.

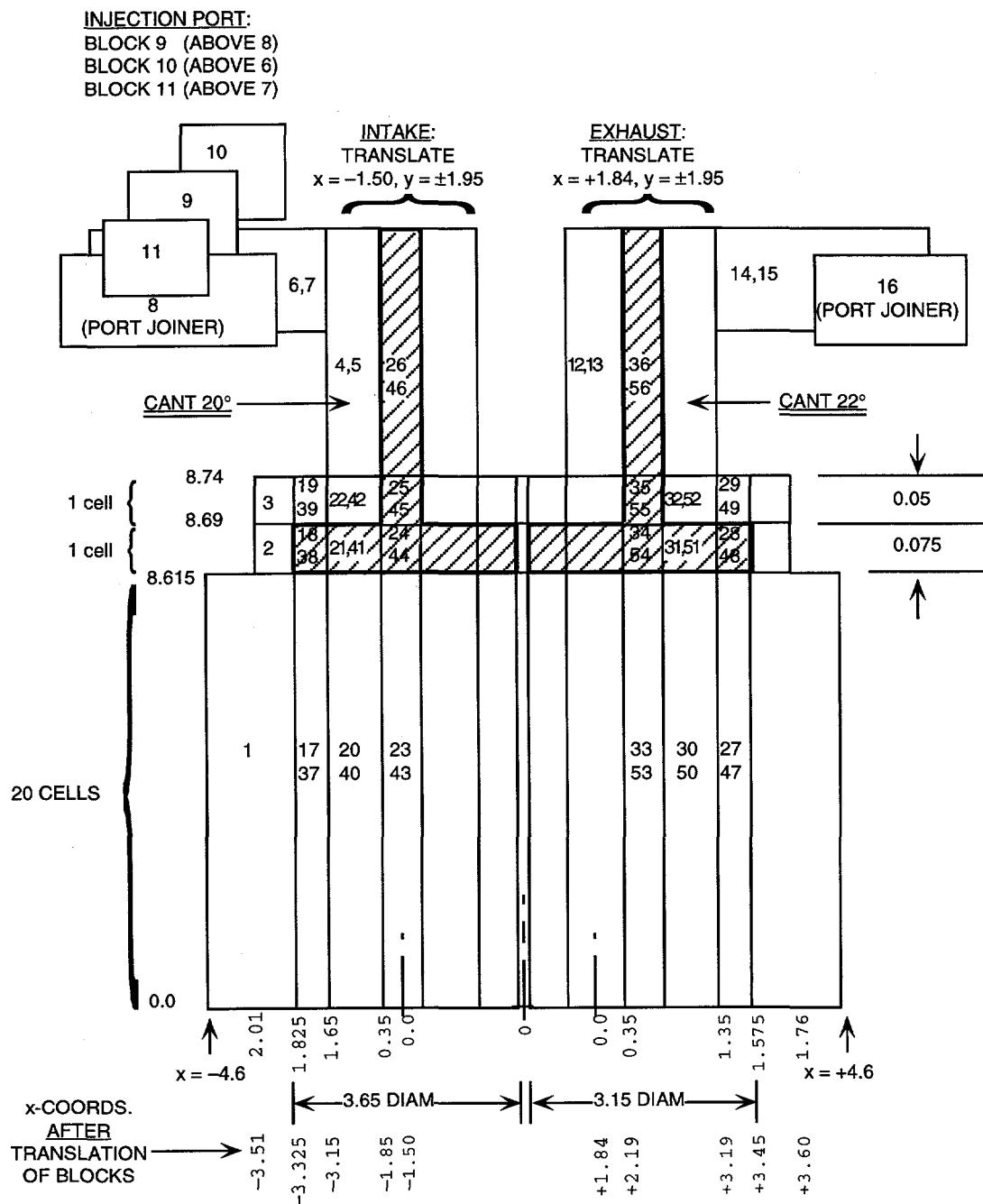


Fig. 16. The K3PREP logical mesh for the quasi-symmetric DOHC 4-valve pentroof example.

**4-VALVE GEOMETRY**  
 (BLOCK 1 IS  $39 \times 34$  CELLS,  
 BLOCKS 2 & 3 ARE  $32 \times 34$  CELLS)

RESHAPING IS RELATIVE TO BLOCK 1 IF  $z < 8.615$ ;  
 OR TO BLOCK 2 OR 3 IF  $z > 8.615$ .

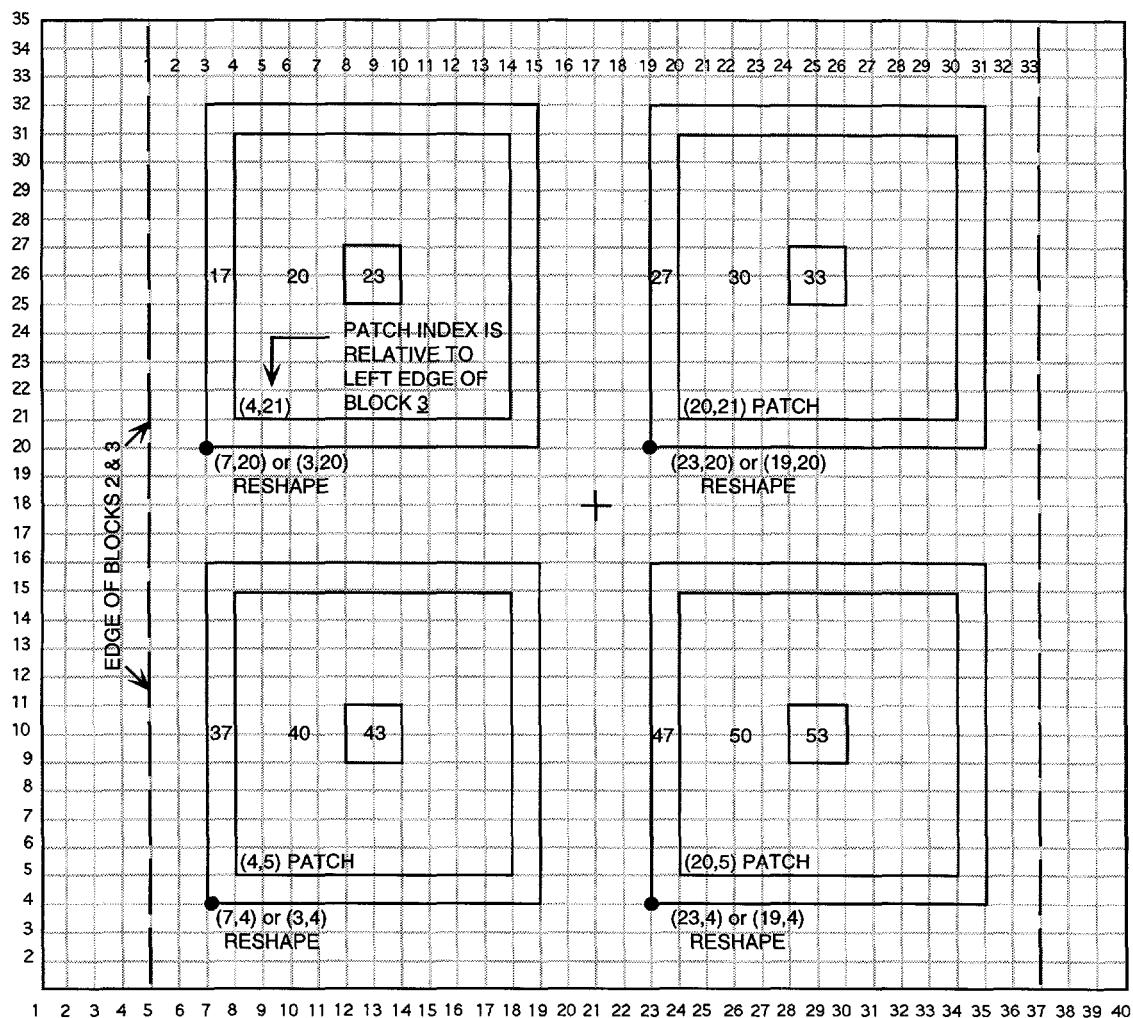


Fig. 17. Plan view of the K3PREP logical mesh for the quasi-symmetric DOHC 4-valve pentroof example.

a geometry in which the intake and exhaust valves alternately share the same physical space, is handled by subroutine REZPENT in KIVA-3V.

#### D. DOHC Asymmetric Pentroof Engine

The 4-valve geometry of Fig. 5 combines some features of the previous two examples. This is a rough representation of a different 4-valve engine, intended only to capture the principal features of the geometry. In Fig. 18, the intake valves are canted 25°, but the exhaust valves only 7°. As a result, the combustion chamber is much higher on the exhaust side, and therefore strongly asymmetric in comparison with the case in Fig. 5. Furthermore, the combustion chamber has a more complex outline (Fig. 19), although it has a front-derriere symmetry that allows only one-half the full geometry to be zoned. This grid uses the pentroof as its starting point, with six planes in block 2 (Fig. 20), as in the wedge geometry, to provide acceptable vertical resolution on the exhaust side. Figure 21 is a plan view of the logical mesh, in which it should be noted that blocks 2-4 not only do not extend to the left and right edges of block 1, but now do not extend to the derriere edge of block 1. This choice of block sizes helped in specifying the "cloverleaf" outline of the combustion chamber.

K3PREP subroutine TILTCOMB is a template for DOHC geometries such as this, that combine features of both the traditional wedge and the symmetric pentroof designs.

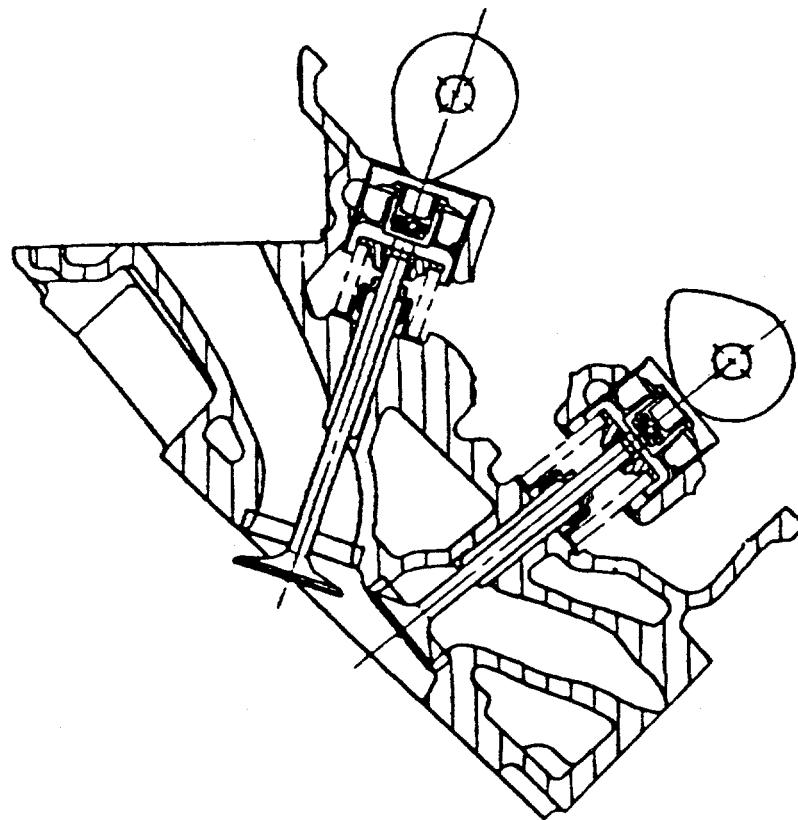


Fig. 18. DOHC engine in which the intake valves are canted  $-25^\circ$  and the exhaust valves are canted  $+7^\circ$ . This results in a significantly asymmetric pentroof combustion chamber.

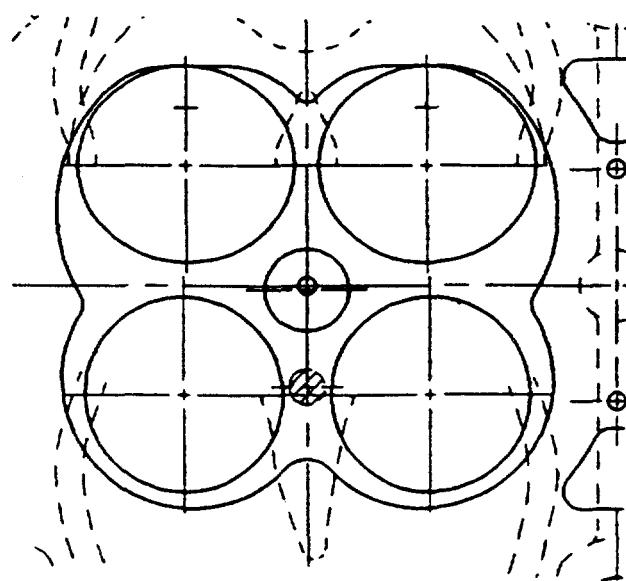


Fig. 19. Outline of the combustion chamber for the example of Fig. 18.

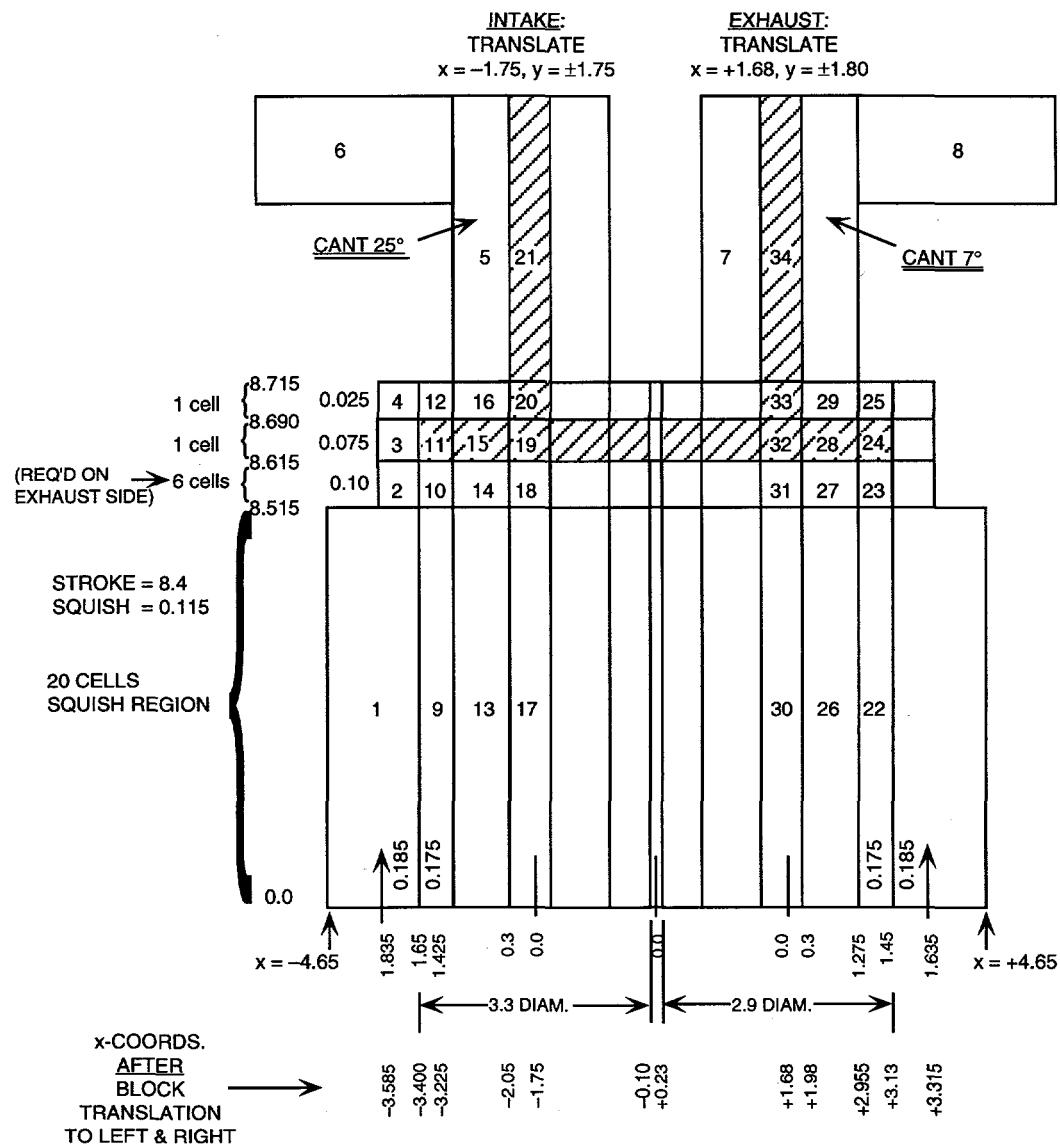
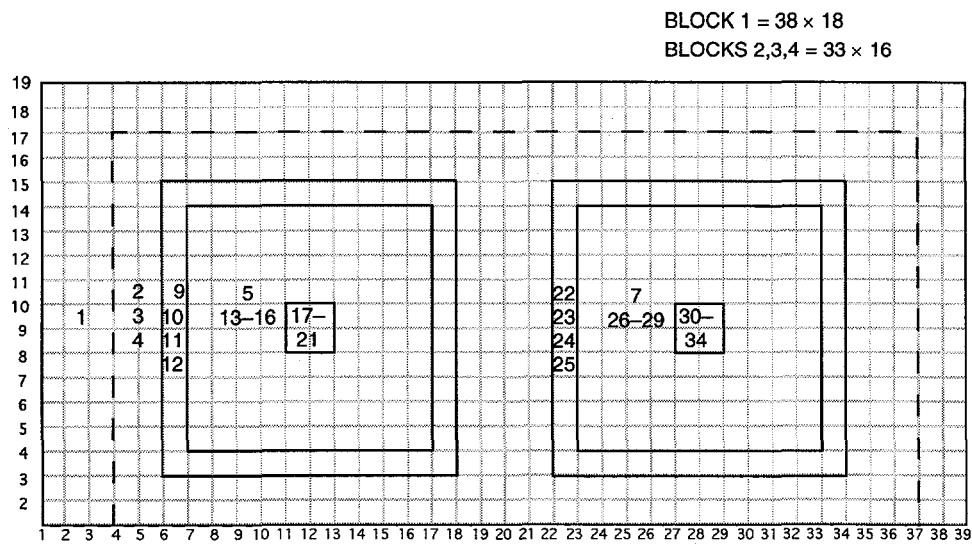


Fig. 20. The K3PREP logical mesh for the asymmetric DOHC 4-valve pentroof example.



*Fig. 21. Plan view of the K3PREP logical mesh for the asymmetric DOHC 4-valve pentroof example.*

## V. MESH GENERATION

Users of KIVA have traditionally relied on the supplied parametric mesh generator to build grids. However, in the past few years the increasing use of KIVA in the analysis of complex engine geometries has led to an increasing use of commercial grid generators. Today, a relatively complex combustion chamber, port, and valve assembly may require a month of meshing, while the actual KIVA analysis of such a test case rarely exceeds a day's worth of computer time. This imbalance between pre-processing and analysis time will become more critical as geometries become increasingly complex.

To meet this need, vendors are creating automatic grid generators in which the user can specify the input geometry as a discretized polygonal surface, as produced by virtually all CAD packages. For current information on what is available, check out our Website and Internet links, described in Sec. I above.

### A. File ITAPE17

KIVA-3V contains no mesh generation capability of its own, and instead relies on a file (ITAPE17) to be provided by a separate grid generator. ITAPE17 is read by KIVA-3V 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-3V will be concerned with. Note that the grid generator is *not* asked to sort storage for the purpose of minimizing vector lengths, as KIVA-3V will sort the data read from ITAPE17 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), FV(I4), and IDFAC(I4). (Note that the addition of the IDFAC array is the only change to the file from the original KIVA-3.)
- 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).

- MTABLES (unformatted) is an integer with a 0 or 1 value. MTABLES = 1 means that ITAPE17 is supplying arrays of IMTAB, JMTAB, and KMTAB, and that KIVA-3V subroutine SETUP does not need to calculate these three arrays. Conversely, MTABLES = 0 means that KIVA-3V must generate these three arrays. This task is simple enough for KIVA-3V 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 always supply these three arrays. If, however, MTABLES = 0, MTABLES is the last item appearing on file ITAPE17. 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: Extensions to the IPREP File

The KIVA-3V package includes a basic grid generator, K3PREP, that writes a file ITAPE17 conforming to the above specifications. K3PREP is not intended to generate the really sophisticated geometries that KIVA-3V 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 reasonable amount of computer time. Further, the user can modify the various subroutines in K3PREP to tailor it to specific needs.

The IPREP file has been greatly expanded, and includes a number of useful new options for the generation of grids with valves. Refer to the IPREP files for the various examples.

**1. Block Identification and Ghost Blocks.** The first line of data for each block (format 8I4) has two new quantities, one at the beginning of the line, and the other at the end. The line now reads: N, NX, NY, NZ, NBO, NTYPE, NREGK3, IGHOST. N is simply the block number, an identifier that counts up sequentially from 1. IGHOST identifies whether the block is a normal fluid block or a ghost block (part of the valve). In Fig. 6, for example, the 12 blocks that comprise the valves (14, 18, 22–25, and 30, 34, 38–41) are ghost blocks and are identified by IGHOST = 1. All other blocks have IGHOST = 0. The other quantities retain their previous meaning, except that the definition NTYPE = 5 has been added to identify valve ports and the portion of valve stems in the ports (blocks 5, 8, 20, 24, 25, 36, 40, and 41).

The NBO option has been expanded: If the block is in the *squish* region, NBO > 0 will specify the number of points around the periphery of a cylinder (with no axis) whose points are to be fixed in (x,y) space. This is useful for the precise placement of ports in the cylinder wall, or for positioning a combustion chamber at the top of the cylinder. The use of this option is illustrated in cases 2)

and 4) above. As with the previous usages of NBO, command line(s) follow the basic set of lines (now 6) for the block, where:

- IBO, JBO, XBO, YBO (format 2I4,2F8.3) are the (i,j) of the vertex and the desired (x,y).

**2. Moving Boundary Identification.** Because K3PREP can generate multiple moving boundaries, it is necessary to supply the IDFACE for each block face:

- FACEIDL, FACEIDR, FACEIDF, FACEIDD, FACEIDB, FACEIDT (format 6F8.3) is an additional (6th) line of data for each block. This is a string of identifiers that correspond to the FACEL, FACER, FACEF, FACED, FACEB, and FACET values in line 5. In accordance with the definition of IDFACE, if a FACE-value is 1.0 (moving), then the corresponding FACEID- has a value of 0.0 or greater to identify *which* moving surface the block face is on. If FACE- has any value other than moving, its corresponding FACEID- value is -1.0.

**3. Block Copying.** In some geometries, several blocks will have the same dimensions, for example, the two intake valves or the two exhaust valves in a 4-valve engine. In the 4-valve pentroof example (case 3), blocks 37-56 are almost identical to blocks 1-6. They differ only in some region identifiers (2 or 3), moving boundary identifiers (1 and 2 become 3 and 4; 5 and 6 become 7 and 8), and in their (x,y) location in the mesh. The block identifier on the first line of block data is used to take advantage of this similarity whenever it occurs, and it allows the user to easily duplicate identical blocks:

- NCOPY (format A8, I5) is the number of lines of block copy commands that follow. If NCOPY > 0, each command line contains the following information:
  - NORIG, NDUP, NREGDUP, FACEIDADD (format 3I4, F8.3). These are the block numbers of the original block, the duplicate block, the IDREG of the duplicate block, and the increment or decrement for the FACEID- of moving faces. As a result of block copying, the 120 lines of data that would otherwise be required to specify blocks 37-56 have been reduced to 20 lines.

**4. Flag Cylinder Vertices to Tilt.** One of the last steps in the grid generation will be the actual tilting of valves and their associated ports and runners. When the time comes, these will be identified by their IDREG flags (2 or 3). However, it is generally necessary to include in the tilting some portion of the combustion chamber that is part of IDREG = 1. This is easiest to specify at this point, while the block number identification is still available:

- TILTFLAG (format A10,I5) is the number of sub-volumes to be flagged. For each sub-volume, a command line contains the following information:

- N, ITILTFAG, XLEFT, XRIGHT, ZBOT (format 2I4,3F8.3), where N identifies the block, ITILTFAG (2 or 3) specifies the KIVA-3 region number that this volume is to be associated with in tilting. XLEFT and XRIGHT specify the x-range of the sub-volume to be flagged. Only vertices lying above ZBOT can be tilted.

**5. Flag Central Pentroof Vertices.** After tilting regions to the left and right in a pentroof geometry, the ridgeline between the intake and exhaust sides is likely to be uneven. An experienced user will probably modify K3PREP to define the surface accurately, especially around the spark plug location. However, the worst of the problem is in the z direction, which can be smoothed after tilting by adjusting the z coordinates of flagged vertices, where:

- NPENTFLAG (format A10,I5) is the number of pentroof layers or blocks to be smoothed. For each layer, a command line follows, where:
- N, ILEFT, IRIGHT (format 3I4), where N is the block number whose vertices that lie between ILEFT and IRIGHT are to be flagged for smoothing after tilting. The i-index counts from 1 at the left edge of the block.

**6. Flag Vertices Around Wedge Top.** This option flags vertices around the top edge of the combustion chamber, so that the edge will be slightly rounded after it is tilted:

- NWEDGEFLAG (format A10,I5) is the number of lines that follow, where:
- N (format I4) is the block number at the top of the combustion chamber.

**7. Block Translation.** Complex grids with valves contain a large number of circular blocks. Note in the IPREP files for cases 1) to 4) that all of these blocks are specified as being centered at (x = 0, y = 0). This is done to minimize the chances for arithmetic error, and as an aid in fine-tuning the positions of the valves. At this point, all such blocks are translated to their true locations:

- TRANSLATE (format A10, I5) is the number of lines of block translation commands that follow. Each command line contains the following information:
- NBLK1, TRANSX, TRANSY, TRANSZ (format I4, 1X, 3F10.5) identify the block number, and the + or - distance in cm that the block is to be translated from (0,0,0) in x and/or y and/or z space.

**8. Localized Geometry Refinement.** Any number of logically vertical grid lines can have their x and y values set to specific values from top to bottom. Subroutine LOCXY is a tool for geometry refinement, particularly around combustion chambers and ports, and it can reduce or possibly eliminate the need

to insert special patches in K3PREP. The following commands are used for this purpose:

- NLOCXY (format A10, I5): NLOCXY is the number of lines of tailoring commands that follow. If NLOCXY > 0, each command line that follows contains the following information:
- NBLK1, INDEXI, INDEXJ, XLOC, YLOC (format 3I4, 2F8.3) says: in each k plane of block NBLK1, for the vertex whose i and j are INDEXI and INDEXJ, set the x and y to XLOC and YLOC. These vertices are flagged to prevent their being moved in the reshaping process that follows:

**9. Block Reshaping.** Block reshaping is a way to make concentric circular grid regions nest inside one another. Subroutine RESHAPEB preserves the circular shape in (x,y) space of ports, valve stems, valve skirts, and valve pockets all the way down to the piston crown (subject to some relaxation in the tilting process), in order to obtain the most orthogonal possible zoning. The reshaping around a valve starts with the valve stem as the innermost preserved shape, and proceeds radially outward next to the port, then to the valve skirt, and finally to the valve pocket. The outer radius of each circle is preserved, but the x and y of points inside this radius are relaxed to conform to the circle lying inside. Figures 9, 13, 14, and 17 are plan views of the cylinder in logical space for the four examples, as used for writing the reshaping commands.

- RESHAPE (format A10, I5) is the number of block reshaping commands.
- NBLK1, NBLK2, INDEX1, INDEX2, INTRP, IRELAX (format 6I4) says: reshape grid points of NBLK2 (the outer block) to conform to the peripheral shape of NBLK1 (the inner block), starting at (INDEX1, INDEX2) of NBLK2. INTRP = 1 means that any non-assigned points between the two surfaces are to be interpolated. The value of INTRP is best determined on a case-by-case basis. IRELAX = 1 will relax INTRP = 1 vertices after their interpolation.

It is important to note that all reshaping blocks are discarded after their job of reshaping is done, and they must appear *last* in the block data.

**10. Pentroof (x,y) Coordinates.** Given the basic cylinder, which is typically block 1, and which has had its (x,y) coordinates adjusted in subroutines LOCXY and/or RESHAPEB, subroutine PENTXY makes the initial (x,y) coordinates of the block(s) above it, which will later be tilted to form the pentroof, match the (x,y) coordinates at the top of the cylinder block.

- NPENTXY (format A10,I5) is the number of pentroof blocks.
- NORIG, NCOPY, IOFFSET, JOFFSET (format 4I4) make the (x,y) coordinates of block NCOPY match those of block NORIG below. The values of

IOFFSET and JOFFSET, if  $> 0$ , specify the offset of the copied block with respect to the template block. This offset allows for a pentroof that does not reach to the cylinder edge.

**11. Shaping of the Valve Ports.** (Note that the shaping of ports and runners takes place *after* reshaping, but *before* block patching or tilting.)

- NVALVPORT (format A10,I5) is the number of valve ports. For each, the following two lines are read by subroutine VALVPORT:

- N (format I4) is the block number of the valve port;
- XROT0, ZROT0, RADCURV, RADF, RADD, ANGRUNR, XYROTATE (format 7F8.3), which says: Rotate outer radius of valve port N about (XROT0, ZROT0), using a radius of curvature RADCURV. The region of the left or right face of the port where the runner will be patched is a logical rectangle, with its front and derriere edges curved with RADF and RADD. This face is then angled ANGRUNR degrees from vertical in (x,z) space, and rotated XYROTATE degrees in (x,y) space, viewed from above. A positive angle is measured clockwise from the  $y = 0.0$  line, and a negative angle is measured counter-clockwise.

**12. Shaping Runners.** Subroutine RUNNER is used to round the edges of intake and exhaust runners.

- NRUNNER (format A10,I5) is the number of runners to be shaped. For each runner, two lines are read by subroutine RUNNER, where:

- N, NE (format 2I4), and
- RADF, RADD (format 2F8.3) say: Shape runner (block N) to fit the shape of the valve port (block NE), which has already been prepared for this in subroutine VALVPORT. The open (pressure boundary) end of the runner is curved with radius RADF on its logical front edge, and with radius RADD on its logical derriere edge (either of which may be zero to leave the face square in cross-section, for example, when paired runners will be later joined.)

**13. Joining Pairs of Runners.** Subroutine SIAMESE is used to smoothly join pairs of intake or exhaust runners in 4-valve engines.

- NSIAMESE (format A10,I5) is the number of port-pairs to be siamesed. For each pair, the following line is read:
- N, NF, ND (format 3I4) says to join the central block (N) with runner block (NF) on the front, and runner block (ND) on the derriere.

**14. Block Patching.** This remains unchanged from the original K3PREP. It is important to remember that the *only* blocks to be patched are the low-numbered blocks that remain after reshaping.

## 15. Relaxing the Interior of the Piston Bowl.

- IRELAXB (format A10,I5), if = 1, will call subroutine RELAXB to relax interior points in a piston bowl.

**16. Valve Profiling.** Subroutine PROVTOP is called to shape the curved top of each valve to a user-specified silhouette. Similarly, if the face of the valve is not flat, subroutine PROVFCE is called to shape the face vertices. These two subroutines allow the true shape of the valve to be replicated to the fineness of the grid. To avoid complicating the process, valves are profiled in their original vertical orientation, before any tilting.

### a. Valve Tops

- NPROVTOP (format A10, I5) is the number of valve top surfaces to be profiled. The following information is supplied for each valve:
  - IDSURF, NZSEAT, NZEDGE, NZPORT, NPTSPRO (format 5I4), and
  - XSHIFT, YSHIFT, RADPORT (format 3F8.3) say: Profile valve top surface IDSURF (2, 4, 6, 8, etc.); there are NZSEAT identical planes up to the valve seat, based on whether the valve was generated in either its BDC position or its TDC position. Above the seat level, there are NZPORT planes, whose z's are to be smoothly interpolated between the profile at the seat and the z at the top of the port. If NZEDGE < NZPORT, it means that a runner will be patched to the side of the port, so at the edge profile only up to NZEDGE. Because the following data lines are relative to (x=0, y=0), (XSHIFT, YSHIFT) specify the true position of the valve axis, which resulted from an earlier TRANSLATE command.
  - RADPORT is the radius of the port, which is typically the inner radius of the valve seat. Following these 2 lines are NPTSPRO lines of (r,z) values, in cm:
  - RVALVE, ZVALVE (format 2F8.3). The first r in this list is the radius of the valve stem, and the final r is the radius of the valve skirt. The z values are relative to z = 0.0 at the valve skirt. NPTSPRO does *not* have to correspond to the actual number of grid points between stem and radius; one can supply, however, as many (r,z) measurements as one wishes. The program will interpolate the (RVALVE, ZVALVE) dataset to produce as accurate a replica of this profile as the local grid resolution will allow.

### b. Valve Faces

- NPROVFCE (format (A10,I5) is the number of valve faces to be profiled (typically with a shallow dish). The following information is supplied for each face:
  - IDSURF, NPTSPRO (format 2I4), and

- XSHIFT, YSHIFT (format 2F8.3) say: Profile valve face IDSURF, where IDSURF = 1, 3, 5, 7, etc. The valve has its center at (XSHIFT, YSHIFT) cm. Following these 2 lines are NPTSPRO lines of (r,z) values, in cm:

- RVALVE, ZVALVE (format 2F8.3). The first r in this list is at the valve axis, and therefore must = 0.0. The z values are relative to the z at the valve edge; thus the final z must = 0.0, and the last r is at the radius at the edge of the dish in the valve face. Outside this radius, the valve face remains flat.

**17. Specifying z's up the Cylinder Wall.** It is often desirable to assign z coordinates on the cylinder wall. This is particularly true in grids with valves, and KIVA-3V subroutine REZONE does not accord grid that contain valves. As we have seen in the examples, it is desirable to control the z-coordinates, especially near the top of the cylinder, to preserve the original cell spacing in vertical-valve geometries, and to ensure some minimum number of planes above the piston at TDC in canted-valve geometries (Fig. 12).

- NZCYLWALL (format A10,I5), if greater than zero, is the number of lines of z-coordinates that follow, which are read by subroutine ZCYLWALL:
- ZWALL (format F8.3), where typically the first value is 0.0, and the final value = (STROKE + SQUISH).

**18. Valve Canting.** Grid volumes may be tilted to cant valves and the adjoining ports and runners. The tilt is in (x,z) space only, as the camshaft(s) is (are) assumed to be parallel to the y axis. In logical space, valve travel remains in the bottom-to-top direction, just as it was before tilting. There is currently no provision for horizontal or annular valves.

- TILT (format A10, I5). For vertical valve geometries, or geometries with no valves, one simply specifies TILT = 0. If TILT is greater than 0, it specifies the number of tilt command lines. The very next line, however, is INDTILT:
- INDTILT (format I4) is an integer (that matches the value of IREZ in KIVA-3V) that specifies which of the three provided tilting templates is to be used, where: 4 = TILTPENT, 5 = TILTWEDG, and 6 = TILTCOMB. The INDTILT line is followed by (TILT) lines of commands, each of which contains:
- IDREGION, XCROT, ZCROT, TILTXZ (format I4,3F8.3), says: Tilt the region whose cells have IDREG = IDREGION (typically 2 or 3), about a center of rotation (XCROT, ZCROT) by an angle TILTXZ degrees from vertical. When viewed from the logical front, a positive value of TILTXZ tilts the valve stem to the right, and a negative value tilts it to the left. Cant angles range from zero to 20 or more degrees. Recall that TILTFLAG, read earlier, has allowed the inclusion of cylinder vertices (IDREG = 1) in the tilting process.

**19. Dish in Piston Face.** This option is used to put a shallow dish in the center of the piston face, which is assumed to have been flat until now. The dish is formed by depressing the z-coordinates of vertices on the face to values less than the original 0.0. (In contrast, a true piston bowl is much deeper, and requires specification as one or more fluid blocks that are patched below the squish region.) If

- NDISH (format A10,I5) is = 1, subroutine DISH is called, which reads the following command line:
- DMAX, RBOWLMAX, VOLWANT (format 3F8.3), where DMAX is the maximum depth of the dish, RBOWLMAX is its radius, and VOLWANT is its desired volume. Vertices that lie inside this radius are simply pushed straight down, with no attempt to make the outline of the dish exact. This feature can be used to tailor the compression ratio of the engine precisely when a dish is present. An example of its use is the 4-valve pentroof engine, case 3) described previously.

**20. Scallops in Piston Face.** This option is used to put small curved scallops in the piston face. These are used to provide TDC clearance from valves that are not closed. As in subroutine DISH, the z values on the piston face are simply pushed straight down, with no attempt to make the outline of the scallop exact. If

- NSCALLOP (FORMAT A10,I5) is greater than zero, subroutine SCALLOP is called, which in turn reads (NSCALLOP) command lines:
- XS, YS, ZS, RS, TILTS (format 5F8.3), in which (XS, YS, ZS) are the coordinates at the center of the scallop, which is visualized as a tilted circle that intersects the piston face. RS is the radius of the scallop, and TILTS is its tilt angle, which is typically the same as the tilt angle of the valve above the scallop.

**21. Offset Piston Bowl or Dome.** Some piston designs have an offset bowl in the piston face, not concentric with the axis of the cylinder, and/or a nonconcentric head dome. K3PREP now contains an option to offset bowls and domes in a full circle mesh that has a central axis, as in Figs. 16-19 in the KIVA-II report. The following two lines are read:

- XOFFSET (format A10,F8.3), and
- YOFFSET (format A10,F8.3) are offsets, in cm, and either of them may be positive, negative, or zero. If either is nonzero, subroutine OFFSET is called.

Information regarding the desired output from K3PREP now appears at the end of the IPREP file. In the original program, this existed in data statements.

More flexibility is now available, as the associated quantities have been moved to the IPREP file:

- WRITE17, PLOTMESH, NXPLOTS, NYPLOTS, NZPLOTS, NVHIDE.

Note the addition of zone plots in the x and z directions. In all three of these subroutines, the desired slice(s) can now be specified. If the selected slice does not cut through a perfect plane, the plot may have gaps, mothholes, or overlaps. However, imperfect plots are nevertheless often valuable for locating grid problems.

## VI. OTHER NEW FEATURES IN KIVA-3V

### A. Particle-Based Model for Wall Film Dynamics

A particle numerical method<sup>30</sup> has been incorporated in KIVA-3V to model the dynamics and evaporation of liquid wall films, and is primarily intended for port-injection spark ignition engines. **Caveat:** *It is important to note that the entire wall film model is a work-in-progress, and will undoubtedly undergo revisions and additions in the future. Careful comparison with experimental data remains to be done.*

A formulation based on particles has the advantage of accurate calculation of convective transport of the film, and is compatible with the existing spray model. The disadvantage is the need to track particles on curved surfaces, which in a 3-D program requires a considerable amount of coding.

Another programming consideration is that the model requires a large number of cell-storage arrays, starting in subroutine PMOVTV, then in subroutine LAWALL and subroutine WALLFILM. Fortunately, **no** new storage is required, because during this early portion of the cycle, there are almost no other demands on equivalenced storage (refer to Table 1 in Sec. II.A).

The dynamics of the wall film involve the interactions of the impinging spray, the wall, and the gas flow near the wall. The spray influences the film through mass, tangential momentum, and energy addition. The wall affects the film through the no-slip boundary condition and heat transfer. The gas alters film dynamics through tangential stresses and heat and mass transfer in the boundary layers above the film. New wall functions have been developed to predict transport in the boundary layers above the vaporizing film. The model in its present state assumes that the film is sufficiently thin that the film flow is laminar, and that liquid inertial forces are negligible. Because liquid Prandtl numbers are typically about 10, unsteady heating of the film may play a role, and is accounted for in the model. The thin film approximation breaks down near sharp corners, where an inertial separation criterion is used.

Because the referenced paper describes the model in detail, the information will not be repeated here. However, two new features have recently been added: (1) a model for the splash caused by an impinging spray, and (2) a model for the expulsion of liquid from the intake-valve seating surfaces as the valve closes.

**1. The Splash Model.** When subroutine PMOVTV determines that an airborne spray droplet has just collided with a solid surface, it calls subroutine SPLASH, which will either reflect the droplet, or transform it into a wall film

particle. When the approach is finalized, we plan to describe it in full detail in a SAE paper. For the present, however, only a brief sketch of the approach is presented.

The first decision, of course, is whether to splash the impinging droplet or to incorporate it in the wall film. This determination is made by calculating an impact parameter for the particle, and comparing it with some defined critical value that was based on experimental observation. The impact parameter is a function of the Weber number and Reynolds number of the impinging droplet, along with the current wall-film thickness at the impact location. If the critical value is exceeded, along with a random number function of the difference between the impact parameter and the critical value, the droplet is said to have splashed.

If neither of these tests is met, the droplet becomes a wall particle. The conversion of a spray particle to a film particle has been previously described.<sup>30</sup> In this case, subroutine SPLASH accounts for the contribution of the droplet to the mass, momentum, and energy of the wall film, and increments the film height on the surface.

If the droplet rebounds and remains airborne, it is assigned a new radius that is based on a cumulative distribution function whose values in turn are again based upon experimental observation of splashing drops.

A great area of uncertainty, however, lies in the specification of velocity components for a splashed particle. The current distribution of splash product drop velocities, which includes random tangential and normal components, is based on very limited experimental data.

**2. Valve Seat Clearing.** A significant amount of the exhaust emissions in a port fuel-injected engine are formed during cold starting. Before the intake valve is heated by combustion, fuel spray impinging on the valve evaporates quite slowly. As a consequence, liquid is still present on the valve and in the valve seat region for some number of engine revolutions. Because the valve shuts with considerable authority, any liquid remaining on the seating surfaces is strongly forced either into the engine cylinder or back into the intake port.

KIVA-3V subroutine CLEAR is designed to model this process. The approach is based on the analogy of liquid being forced out from between two colliding parallel plates. The contact length of the plates is analogous to the length of the valve seat, and different pressures at either end of the plates are analogous to different pressures on the intake port and cylinder sides in an engine.

There may be wall film particles on both the top edge of the valve and on the valve seat above it. These are used collectively to determine the film height between opposing cell faces as the valve closes. As soon as the current true lift of the valve (VLIFTTRU) becomes less than the current film height, all wall film particles between the two faces are flagged to indicate that they are to be handled in a different manner. They no longer communicate with the gas, and are assigned the average wall temperature.

As the valve continues to close, a test is made each cycle on each valve-seat cell face to determine if any of these flagged particles are to be ejected. The test formula is based on the area of the cell face and the current values of valve clearance and film height, and produces an integer that represents the number of particles to be ejected. If the value of this integer is greater than zero, the film height is reduced to the current clearance, and ejection velocities are determined for both the port and cylinder sides. These are functions of the film height, seat width, pressure gradient, and the liquid viscosity. Finally, the decision whether to eject a spray particle into the cylinder or into the intake port is in turn a random function involving these two velocities. Newly launched spray particles are properly conservative of mass.

The cycle in which VLIFTTRU finally goes to zero represents the last opportunity to clear any remaining liquid off the valve seat surfaces. Subroutine CLEAR checks for this situation and will force the final clearing if necessary.

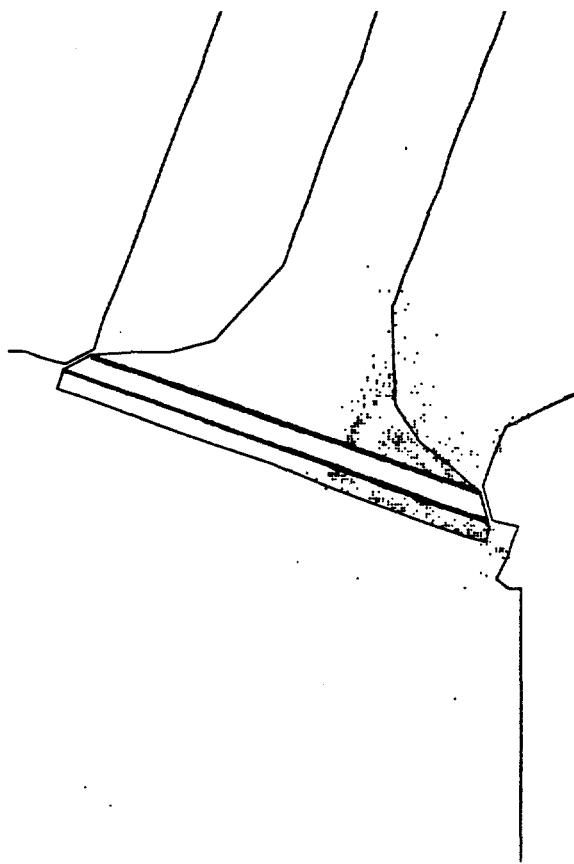
Figure 22 illustrates a valve seat area that has been cleared of film particles. Note that ejected particles have been reentrained into both the cylinder and the intake port.

Along with the splash model, the valve seat clearing model will be more fully described in the planned SAE Paper.

## B. Initial Conditions Specified by Region

A useful new feature allows the user to define the initial conditions keyed to the region number IDREG. A new section of input data has been added to file ITAPE5 that contains the following:

- STOIFUEL, STOIO2. These are the stoichiometric coefficients for the fuel and oxygen in the fuel oxidation reaction, and are required if the fuel is pre-mixed.
- NREG, the number of fluid regions in the mesh. Then, for *each* region, the user supplies:
- PRESI, TEMPI, TKEI, SCLI, ER, and MFRAC--. These are the initial values of cell pressure, temperature, turbulence kinetic energy, turbulence length scale, equivalence ratio, and species mass fractions for all (NSP) species.



*Fig. 22. The liquid film, represented by wall film particles, is cleared from the valve seat area when the valve closes.*

In engine applications, the researcher generally knows the initial pressures, temperatures, and mixture compositions in the cylinder and ports. Given this information, along with the equivalence ratio for fuel-air mixtures in premixed applications, the new subroutine SETUPRO in KIVA-3V calculates the initial species densities. As a check, the input data are reverse-calculated using these densities. SETUPRO then writes to file OTAPE12 all information that the code will be using, region by region, so that the user can verify it. Reported are the equivalence ratio, average molecular weight of the non-fuel species, total density, total moles per unit volume (the sum of the density/molecular weight ratios), pressure, temperature, species densities and mass fractions.

If  $SCLI > 0.0$ , its value is used as a uniform length scale (in cm) by subroutine SETUP when it calculates the initial EPS array. If  $SCLI = 0.0$ , the initial EPS will instead be proportional to the distance to the nearest solid wall.

KIVA-3V will ensure that the initial length scale is less than or equal to SGSL when the subgrid scale turbulence model is used.

The new initialization procedure also permits the removal of the specification of inlet and ambient species densities (SPDIN-- and SPDAMB--) from ITAPE5. Instead, the user simply specifies the region number (2 or 3) to associate with inflow (NREGIN0) and open (NREGAMB) boundaries. In addition, the following lines are also obsolete: TEMPI, TKEI, UNISCAL.

### C. Improved Subroutine SORT

When the original KIVA-3 was written, an effort was made to minimize computer storage. Accordingly, the original subroutine SORT was written in a manner that simply interchanged information between two cells, using scalar temporaries but no storage arrays. The scheme also involved a pair of nested do loops over all vertices. At the time, our largest calculations involved perhaps 30,000 grid points. It wasn't too long, however, before KIVA-3 users had grids of 100,000+ grid points, and a scheme that used nested loops became totally unacceptable, as the CPU usage went from negligible to horrible.

Subroutine SORT was completely rewritten in December 1993 without nested loops, making it linear in the number of vertices. The improved subroutine uses a negligible amount of time, at most a few seconds for large meshes.

Two new storage arrays were introduced in the rewrite, I4ORIG and I4SORT. For every new sort, all data is first put back where it originally was, using the I4ORIG array. This prevents the storage order from gradually changing and deteriorating during the calculation. Because nearby neighbors are more likely to also remain stored in close proximity, the rest of the program is observed to run slightly faster. The I4SORT array contains the new index for quantities as they are sorted.

Finally, all the equivalenced arrays that are left unsorted are zeroed out. This eliminates the chance of garbage numbers reappearing in some different location, such as at an open boundary, and destroying the calculation.

### D. Improved Law-of-the-Wall Heat Transfer Model

The wall heat loss calculation in KIVA-3V has been revised to conform closely to the expression originally proposed by Launder and Spalding.<sup>31</sup>

$$Q_w = \frac{\rho v_\ell c_p}{Pr_\ell} \frac{F}{y} (T - T_w), \quad (1)$$

where

$$F = \begin{cases} \frac{R' Pr_\ell * RPR}{1/\kappa \ln R' + B + 11.05(Pr_\ell * RPR - 1)} & R' > 11.05 \\ 1.0 & R' < 11.05, \end{cases} \quad (2)$$

and

$$R' = \frac{c_\mu^{1/4} K^{1/2} y}{v_\ell}. \quad (3)$$

In these equations,  $v_\ell$  is the laminar kinematic viscosity of air,  $Pr_\ell$  is the laminar Prandtl number,  $\kappa$  is Karmann's constant,  $RPR$  is the reciprocal of the turbulent Prandtl number, and  $T_w$  is the wall temperature. The gas temperature  $T$ , turbulent kinetic energy  $K$ , and distance from the wall  $y$  are all evaluated at the midpoint of the face that lies opposite the wall face of the computational cell next to a wall. Equations (1)–(3) can be obtained from the equations of the KIVA-II report<sup>6</sup> by substituting for  $u$  its law-of-the-wall value

$$u = \frac{u^*}{\kappa} \ln \frac{yu^*}{v_\ell} + B, \quad (4)$$

and then for the shear speed  $u^*$  its equilibrium boundary layer value

$$u^* = c_\mu^{1/4} K^{1/2}. \quad (5)$$

Thus, in the logarithmic region of an equilibrium turbulent boundary layer, the equations in KIVA-II and KIVA-3V give the same wall heat loss, but in many circumstances Eq. (5) is not satisfied. Then the new equations have been found to give better agreement with experimentally-measured wall heat loss.

#### E. Mixing-Controlled Turbulent Combustion Model

The input flag TRBCHEM is used to turn on the turbulent mixing-controlled chemistry option in subroutine CHEM.

The model is based on the eddy-dissipation model developed by Magnussen and Hjertager.<sup>32</sup> It produces a dependency of the turbulent flame speed on the square root of  $k$ , the turbulent kinetic energy, and with no dependence on  $\epsilon$ . The model has two free parameters, CAPA and CAPB. CAPA relates to the turbulent reaction rate. It varies with the operating conditions (e.g., temperature and density at ignition), and has the same dependences as the laminar flame speed. The range of CAPA is from near unity for heavy diesel fuels to 100 for hydrogen. CAPA is best chosen by experimental comparison, and then held constant for similar operating conditions. Because of its variability on fuels and combustion conditions, it is included as an input parameter in file ITAPE5. Note that because modifications of CAPA affect the flame speed and hence the flame temperature, CAPA also affects emission production rates. CAPB is set to 0.5 within the code and should not require modification.

## F. Optional RNG $k-\epsilon$ Turbulence Model

Besides the previous choices of a subgrid-scale and a standard  $k-\epsilon$  turbulence model, KIVA-3V offers the Renormalization Group (RNG) theory variant of  $k-\epsilon$  proposed by Han and Reitz.<sup>33,34</sup> This formulation includes the effects of compressibility using a rapid distortion analysis. Based on the rapid distortion regime that is applicable to engine flows, Han and Reitz have proposed a new relation between the integral length scale, which can be measured experimentally, and the macro length scale ( $k^{3/2}/\epsilon$ ), which can be computed numerically. In the KIVA-3V implementation, several of the turbulence model constants are redefined in subroutine RINPUT. Only the  $\epsilon$  equation in subroutine KESOLV requires modification by an additional term that accounts for the ratio of the turbulent to mean-strain time scale.

A new turbulence model switch, TURBSW, replaces the specification of TKESW in file ITAPE5. TURBSW = 0.0 turns off turbulence entirely, by setting TKESW = 0.0. TURBSW = 1.0 turns on the conventional  $k-\epsilon$  model. TURBSW = 2.0 turns on the RNG  $k-\epsilon$  model (these latter two set TKESW to 1.0). As in earlier KIVA versions, the input quantity SGSL, if greater than zero, specifies the SGS length scale.

The code modifications to incorporate the RNG option were contributed by Zhiyu Han.

## G. Fuel Library

**1. New Fuel Options.** The fuel library has been expanded by five new offerings: natural gas, dimethyl ether (DME), isopentane, helium, and argon.

**a. Natural gas.** The composition of natural gas varies significantly from one source to another.<sup>35,36</sup> This variability can cause difficulties in optimizing the performance of natural-gas-fueled engines used in over-the-road service. The primary constituent of natural gas is methane, which comprises anywhere from ~0.44 to ~0.99 of the mixture mass fraction, but most typically is in the range 0.80 to 0.85.

The KIVA-3V natural gas model is a mixture of methane, ethane, propane, n-butane, isobutane, n-pentane, and isopentane, and therefore contains no liquid constituents, such as n-hexane. The user is encouraged to set the mass fractions for each of the 7 constituents in the data statement FRACNG in subroutine FUEL to represent as closely as possible the gas being modeled. Individual mass fractions may be zero, but the seven mass fractions must sum to 1.0. The default values supplied in FRACNG are 0.85, 0.07, 0.04, 0.025, 0.005, 0.005, and 0.005. Subroutine FUEL will automatically calculate the correct molecular weight, heat of formation, and enthalpy for the specified mixture.

**b. Dimethyl Ether (DME).** In the past, DME was best known as a chemical step in the production of synthetic fuels, while at present it is most frequently used as a propellant in aerosol sprays as a replacement for ozone-depleting chlorofluorocarbons. Most recently, however, DME has attracted the attention of diesel engine researchers<sup>37</sup> because it is environmentally "green" even after it has been burnt. Exhaust from a DME-driven diesel engine contains no sulphur, almost no soot (all that can be measured comes from the motor oil), and only about 20% of the nitrogen oxides that diesel produces. Hydrocarbons are also dramatically reduced, and only CO<sub>2</sub> emissions remain at diesel levels.

The biggest difficulty with DME is its high vapor pressure and low viscosity. It is a gas at room temperature; thus its use as a liquid fuel requires pressurization. (The model in the KIVA-3V fuel library considers DME in its gaseous state). The efficiency of DME is comparable to diesel, but it contains less energy in a given weight. Although DME is easy to produce from methanol, this is not economically attractive, given the cost of methanol. Producing it from natural gas costs less, but requires far more chemistry and expensive equipment. However, it is possible that if DME were produced in bulk from natural gas, the price per unit of energy would be only a little higher than for diesel. A slightly impure DME, containing small amounts of water and methanol, is much cheaper to make, and works as well as neat DME.

**2. Tables of Surface Tension and Thermal Conductivity.** In KIVA-II and KIVA-3, the surface tension was treated as a linear fit between two data points: the surface tension value at 350K and a value of 0.0 at the critical temperature. An examination of actual surface tension curves, however, reveals that they depart noticeably from being a straight line. Accordingly, the linear fit procedure has been replaced by tabular interpolation of actual surface tension

values, specified at intervals of 10K in temperature in FUELIB, in the same manner that vapor pressure, liquid viscosity, etc. are treated.

FUELIB also contains a complete set of tables of liquid thermal conductivity, in ergs / (s cm degree K), which is required in the heat transfer calculation for evaporation of a liquid wall film. The thermal conductivity data was taken from several sources.<sup>38-40</sup>

## H. No-Hydro Option

The introduction of the valve model has brought with it the question of ensuring the integrity of the grid through the full engine cycle. Because of the multiple moving surfaces, and in particular the continuous rezoning required when the valves are canted, the possibility for inverted cells in complex grids has increased.

The no-hydro option provides a useful check to ensure that the entire grid and all moving surfaces will behave properly. It is obviously preferable to detect serious grid problems up front in a computer run that requires only minutes, rather than having an unpleasant surprise many hours into a full hydro run.

The input quantity NOHYDRO normally has a value of 0, but when NOHYDRO = 1, all hydro is bypassed. The run will have only piston and valve motion and grid rezoning. Subroutine CONVEX is called every cycle to check for non-convex cells, which are not desirable to have, but which may be nearly impossible to eliminate in some portions of the engine cycle. On the other hand, inverted cells are immediately fatal, so the no-hydro option also calls subroutine VOLUME every cycle to see if any cell volumes have turned negative.

## I. Low Mach-Number Flows

Although it is possible to run KIVA programs in the truly incompressible limit, doing so is totally impractical. For example, consider a duct with unit cell size, in which the flow velocity  $u_0 = 40$  cm/s and the sound speed  $c = 4 \times 10^4$  cm/s. This flow has a Mach number  $M_0 = u_0/c = 10^{-3}$ . A KIVA run with a time step on the order of a microsecond, EPSP =  $10^{-8}$ , and PERRMX =  $10^{-6}$ , might produce excellent results, but would require millions of cycles to run just a few seconds of problem time.

However, a workable approach is made possible by approximating full incompressibility through the use of the pressure gradient scaling (PGS) option. We define a scaled Mach number  $M^* = u_0/c^*$ , in which  $c^* = c/PGS$ . Then, we argue that a flow with an  $M^*$  two orders of magnitude greater than  $M_0$  will be acceptably close to incompressible, which suggests  $PGS = 100$ . Because the

sound speed  $c^*$  is now two orders of magnitude smaller than  $c$ , the time step can be in the millisecond range. In addition, the values of EPSP and PERRMX are not restricted to the tiny values mentioned above, but they need to be slightly smaller than in the standard program.

In this manner, KIVA has been successfully and efficiently applied to low speed laminar flows.<sup>41</sup> The results are for all practical purposes as good as those from a truly incompressible calculation. The observed gradients of pressure and density are negligible, while the flow features of interest are resolved.

The low-speed option is turned on by setting the input flag LOSPEED to 1.0, which will cause KIVA-3V to make several changes in the calculation:

- PGSSW is set to 1.0 in Subroutine RINPUT.
- PGS is initialized to 100.0 in subroutine RINPUT, and it is maintained at this value throughout the run in subroutines NEWCYC and PHASEB. (This should not be considered general, because either a larger or smaller value may scale better with the Mach number of the flow. Also, allowing the value of PGS to be calculated automatically in subroutine PHASEB may prove satisfactory.)
- EPSP is set to  $1.0 \times 10^{-5}$ , and PERRMX is set to 0.01 (in the main program KIVA). These are one order of magnitude smaller than the usual values of  $1.0 \times 10^{-4}$  and 0.10.
- DPREFIN, the reference pressure used in subroutine PSOLVE, is set to 0.0 in subroutine SETUP.

For low-speed flows in a duct that has pressure boundaries, it is advisable to specify a slightly lower pressure at the outlet. For example, if the driving pressure is one atmosphere ( $1.01326 \times 10^6$  dynes), an appropriate outlet pressure might be  $1.0 \times 10^6$  dynes. If DISTAMB is used, an appropriate value might be the duct length.

## J. High-Speed Flows

High speed flows are characterized by shocks, in which the transition from an undisturbed state to a compressed state becomes virtually instantaneous, and takes place over a very narrow region. From a computational point of view, there is a tendency for the solution to become discontinuous at the shock front. The traditional inclusion of an "artificial viscosity" term in compressible flow codes is intended primarily to allow automatic shock calculation, but also to add stability to the difference equations.

In the KIVA-3V high-speed flow option, the "artificial viscosity" is a bulk viscosity that is added to AMU in subroutine VISC. The coefficient of this term is an input quantity, ARTVIS, which is chosen so as to smear any shocks over the width of several zones. For most KIVA-3V applications (which are subsonic or only slightly transonic), ARTVIS should be 0.0, but in true high-speed flows, a value as high as 0.25 may be appropriate.

In addition, it is advisable to transport total energy, rather than specific internal energy in the advection phase, to ensure conservation of total energy. This is controlled by an input flag, ECNSRV; a value of 1.0 turns on the conservation of total energy in subroutines CCFLUX and MOMFLX.

Four test runs were made in a 10:1 shock tube: with and without artificial viscosity, and with and without the total energy conservation. All four cases ran nicely, indicating that the high-speed flow modifications are perhaps not necessary for shocks of this strength. The calculations used QSOU (quasi second-order upwind) advection.

One lack of generality in the artificial viscosity model is the choice of the length scale, named DELXSQ in subroutine VISC. Because the cells are hexahedral, there are several possible choices for the value of DELXSQ, as indicated in the program listing. Unfortunately, the best choice is problem dependent. Too large a value of DELXSQ can cause the program to fail with a temperature overflow.

## K. Soot Model

Because there have been several requests for it, KIVA-3V contains our soot model. This is an option that is turned on if the input flag ISOOT = 1. Subroutine SOOT dates back some years, and it remains to be tested here on more than a single-cell basis, as the original researchers have either left the Laboratory or have retired. The model is primarily based on that of V. F. Surovkin,<sup>42</sup> while the oxidation of soot particles follows the procedure of Haynes and Wagner,<sup>43</sup> with rate constants from Nagle and Strickland-Constable.<sup>44</sup>

The Surovkin model assumes that the formation and growth of radical nuclei and the formation and growth of soot particle nuclei are different stages of the same process, partially superimposed on one another, but that they occur through different mechanisms. The full process, from the moment when the hydrocarbon reaches the reaction temperature, to the separation of soot particles, consists of three stages: 1) the formation of radical nuclei, 2) the growth of the radical nuclei and their conversion on reaching a critical diameter into nuclei with a physical surface, and 3) the growth of the nuclei and their transformation

into carbon particles. The activation energies of the various stages of the process are based on experimental data.

The equation for the rate of formation of radical nuclei has contributions from both molecular decomposition and from the branching of the nucleus-forming process. However, there are two opposing terms in the equation to account for the destruction of radical nuclei by their mutual collision, and by collision with the surface of the growing particle nuclei.

The rate of growth of the radical nuclei is based on the assumption that at each active collision with a molecule of hydrocarbon, the radical-nucleus increases by the weight of the carbon part of the colliding molecule. This requires specification of the number of carbon atoms in the hydrocarbon molecule, an expression for the effective radius of the hydrocarbon molecule, and an expression for the total number of collisions between radical-nuclei and hydrocarbon molecules per unit time.

The rate of consumption of the hydrocarbon consists of the loss of hydrocarbon molecules as the result of active collisions with the surface of the radical-nuclei. The hydrocarbon is converted into radical-nuclei and particle-nuclei.

The eventual soot emission from a combustion device is usually much less than the amount actually generated within. This difference arises because of the combustion of soot particles in oxygen-containing regions beyond the soot-formation zone, and is an important process. The oxidation process for the soot particles is based on the formula of Nagle and Strickland-Constable, which uses empirical rate parameters, and is based on the concept that there are two types of sites on the carbon surface available for  $O_2$  attack. For the more reactive sites, the rate is controlled by the fraction of sites not covered by surface oxides, whereas in the less reactive sites, the reaction rate has a first-order dependence on the oxygen concentration.

The complete model also accounts for the creation and growth of soot particle nuclei, through the use of a fictitious soot-forming species, and particle destruction through particle-particle collisions. Finally, the cell-specific internal energies are reduced through radiative heat loss.

The implementation in the KIVA soot model requires three new cell arrays: the diameter of the growing fuel radical nuclei (DRN), the diameter of the growing soot particle nuclei (DNUC), and the density of the soot particle nuclei (CON0).

In addition, there are three new species arrays in SPD: the radical-nuclei (KRN), the particle-nuclei (KPN), and the fictitious species used for soot formation (KPN).

If the input flag ISOOT = 1, soot data follows on ITAPE5:

TCUTS: if cell temperature < TCUTS, bypass soot calculation  
DRNI: initial diameter of the growing fuel radical nuclei  
DNUCI: initial diameter of the growing soot particle nuclei  
DRNCR:  $d(r-n, cr)$ , critical diameter of the radical nuclei,  
chosen to obtain the correct induction time for the fuel  
RA: minimum radius of HC molecule; e.g.  $2e-8 = 2$  angstroms  
SURF: Surovakin's coefficient F, accounts for chemical  
branching in nucleus formation; observed experimentally  
RHORN: density of  $r-n$ ,  $= 2$  gms/(cm<sup>\*\*3</sup>) for carbon  
RHOPN: density of particle nuclei—has a similar value  
XNC: the number of carbon atoms per fuel hydrocarbon molecule  
EE1:  $e_1$ , activation energy for the formation of radical nuclei  
EE2:  $e_2$ , activation energy for the growth of radical nuclei  
EE3:  $e_3$ , activation energy for the growth of particle nuclei  
EE4:  $e_4$ , activation energy for  $r-n +$  soot interaction  
EKA, EKB, EKT, EKZ are the activation energies  $a, b, t, z$ ; and  
FKA, FKB, FKT, FKZ are the corresponding factors for the  
empirical rate constants  $ka, kb, kt, kz$ , in the oxidation  
of soot particles (Nagle and Strickland-Constable model)

Note: The eight activation energies are input in kcal/mole, and are converted to ergs/gm in subroutine RINPUT.

The following are suggested starting values for the above data:

TCUTS	1200.0		
DRNI	$2.248 \times 10^{-8}$		
DNUCI = DRNCR	$= 12.5 \times 10^{-8}$		
RA	$= 2.0 \times 10^{-8}$		
SURF	$= 2.0 \times 10^4$		
RHORN = RHOPN	= 2.0		
XNC	= 6.0 (if, e.g., C <sub>6</sub> H <sub>6</sub> )		
EE1 = 116.0	EE2 = 30.0	EE3 = 8.0	EE4 = 0.0
EKA = 30.0	EKB = 15.2	EKT = 97.0	EKZ = 4.1
FKA = 20.0	FKB = $4.46 \times 10^{-3}$	FKT = $1.51 \times 10^{-5}$	FKZ = 21.3

## VII. OUTPUT AND POST-PROCESSING

The K3POST post-processor (and the K3PREP grid generator) use CGS (Common Graphics System) software, which was developed at Los Alamos, for creating graphics. In addition, KIVA-3V can write output files for use by GMV (General Mesh Viewer), which was also developed at Los Alamos. These packages, described below, are in the public domain and may be downloaded from the World Wide Web.

A number of commercial vendors also offer graphics post-processing packages tailored for KIVA programs. For information on what is available, check out our Website and its links, described in Sec. I above.

### A. CGS (Common Graphics System) and Postscript Files

After a K3PREP or K3POST run, a package named PSCAN allows the graphics metafile produced by CGS to be viewed at the user's terminal. Hard copies of selected frames can be made on a laserjet printer or film, depending on the output devices connected to the system.

Postscript files can be obtained using CGS, using one of the following two approaches:

- Instead of calling GPLOT, call GPOST to make the metafile a Postscript file.
- One PSCAN option is to write selected frames to a new file, which PSCAN names NPLOT. For example, w10 will write the 10th frame to the new file. If, before doing this, you enter the command -ps immediately after you enter PSCAN, then the file NPLOT will be a Postscript file.

There are a number of versions of CGS and PSCAN available that are tailored for various platforms (Cray UNICOS, SGI, Sun, H-P, IBM/AIX, DEC, etc.) CGS and PSCAN can be downloaded from the World Wide Web by using the following URLs:

[http://www-c8.lanl.gov/graphics\\_vis/graphics\\_sw/cgs/cgs.html](http://www-c8.lanl.gov/graphics_vis/graphics_sw/cgs/cgs.html)

[http://www-c8.lanl.gov/graphics\\_vis/graphics\\_sw/pSCAN/pSCAN.html](http://www-c8.lanl.gov/graphics_vis/graphics_sw/pSCAN/pSCAN.html)

## B. K3POST: Optional IPOST File

With the increasing complexity of computing meshes, it became apparent that using an input file to select output options in K3POST would make the post-processor easier to use. If file IPOST is not present, K3POST will default the output to the values in the data statements in block data BDK3POST, as in the past. However, if file IPOST is present, it should contain the following data:

NAMEPOST	Identification line for the IPOST file, up to 80 chars.
XWINDOW	= 1.0 to display the plots in an X-Window on the user's workstation, = 0.0 otherwise.
NTRACTV	If NTRACTV = 1, plotting is interactive, and the user is queried for a 'y' (yes) or 'n' (no) for each dump read from ITAPE9. If NTRACTV = 0, all plotting is automatic.
MIRROR	= 1 for mirror image vector and contour plots in 2-D cylindrical runs, = 0 for right-side-only (i.e., true mesh) plots.
ICONT(26)	Set of binary flags to select contour plots. Data statements in block data BDK3POST and subroutine FULOUT indicate the sequence.
BANDTST	In remapping complex grids onto a uniform 2-D mesh, odd corners and isolated regions may be missed. If BANDTST = = 1.0, subroutine VFIND will make a second pass that picks these up.
PLOTY0	If PLOTY0 = 1.0, 2-D plots through the y = 0 plane of the KIVA-3V mesh will be drawn. This is appropriate when a symmetry plane or a flat plane passes through y = 0. If mesh is a 2-D sector, K3POST automatically creates only the y = 0 plots, including their mirror images if the flag MIRROR (see above) = 1.
SCALEY0	Scaling factor for the y = 0 velocity vector plots of subroutines PVPLOTY and VELPLTY.
REGCONTY	Min-max range to be considered in CONTURY plots, where 0.0 is global range over all regions, n.0 considers region IDREG = n only.
XBAND	Thickness of band on either side of XK plane in ZONPLTX.
YBAND	Thickness of band on either side of YK plane in ZONPLTY.

**ZBAND** Thickness of band on either side of ZK plane in ZONPLTZ.

(Having control of band thickness to be included in 2-D zone plots is a useful feature in complex meshes, as it allows the user to minimize the appearance of moth holes at one extreme, and the superimposition of nearby lines at the other extreme. For example, Fig. 3b was drawn with YBAND = 0.15, which omitted only two sides of one cell in the intake port. On the other hand, YBAND = 0.13 produced the best results for Fig. 3c.)

<b>NXPLOTS</b>	Number of simple 2-D plots of the mesh through specified XK planes, where are the XK planes.
<b>NYPLOTS</b>	Number of simple 2-D plots of the mesh through specified YK planes, where are the YK planes.
<b>YZONES(NYPLOTS)</b>	
<b>NZPLOTS</b>	Number of simple 2-D plots of the mesh through specified ZK planes, where are the ZK planes.
<b>ZZONES(NZPLOTS)</b>	
<b>NZSLICE</b>	Number of velocity vector plots across KIVA-3V z planes, where
<b>ZSLICE(NZSLICE)</b>	are the z slices. Note that this requires that the actual KIVA-3V z coordinates must lie across levels defined by ZSLICE.
<b>NVIEWS</b>	Number of views for velocity vector and contour plots, using remapped 2-d overlay meshes. For each view,
<b>XSLV,YSLV,ZSLV</b>	specify a slice in the x, y, or z physical plane. Only one of the three may be non-zero.
<b>NVHIDE</b>	Number of views of KIVA-3V mesh with hidden lines removed. For each view,
<b>XYROT,XZROT,Z0</b>	specify rotation of the viewing axis (in a counter-clockwise direction), and the z plane of the perspective projection.

### **C. Monitor Prints for IC Engines**

KIVA-3V subroutine MONITOR, which was contributed by Zhiyu Han, calculates a variety of quantities as a function of engine crank angle. Data is averaged over the cylinder (IDREG = 1) only. The subroutine is intended for IC

engine research, and will require modification for other applications. Information is written onto four separate files for post-processing:

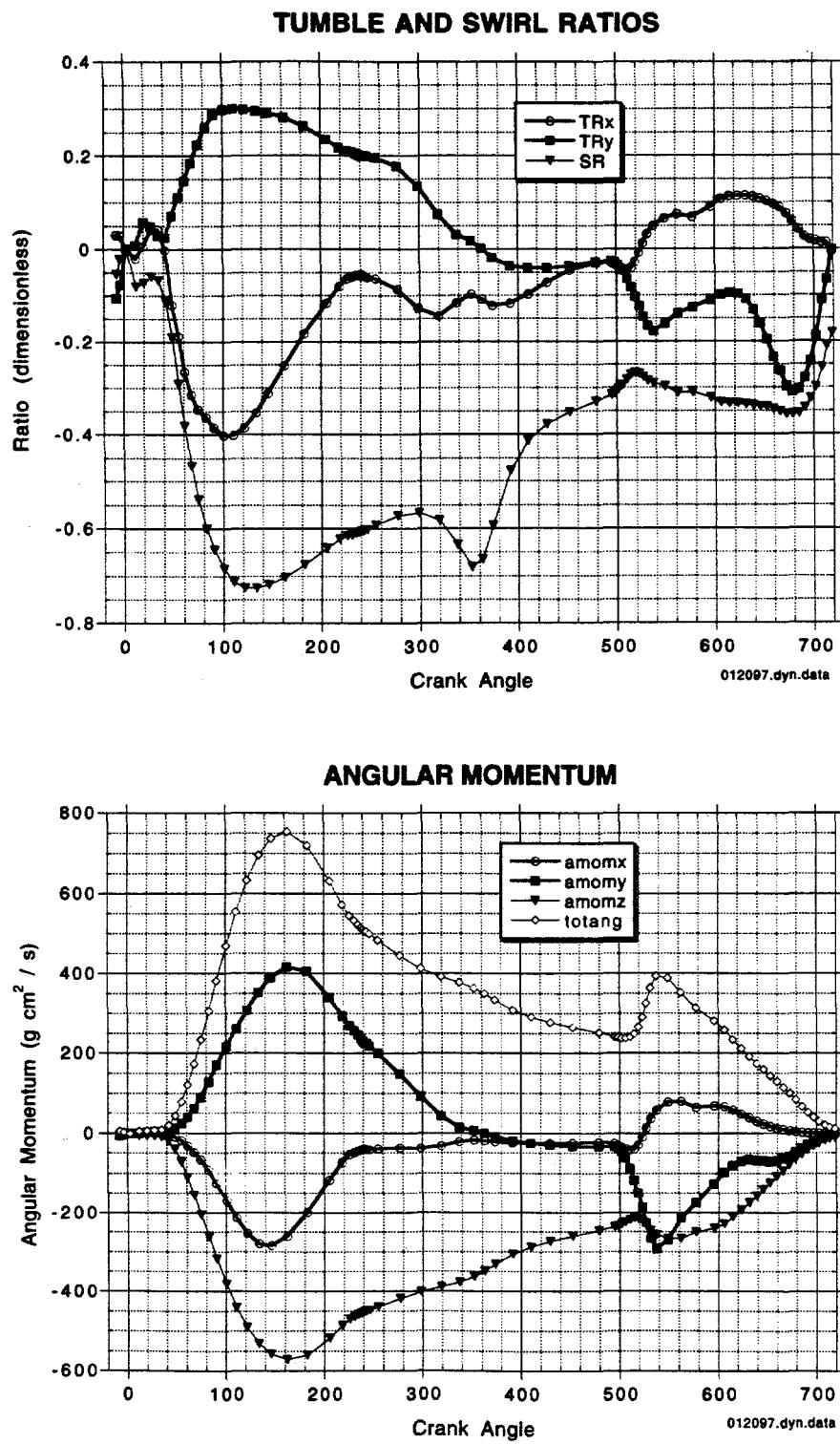
- 'dat.dynamic' contains cylinder dynamic data: crank angle, tumble ratios in both the x and y directions, swirl ratio, total angular momentum separately in the x, y, z directions and globally, and the total kinetic energy. The swirl and tumble ratios are relative to the instantaneous charge mass center in the cylinder. As an example, the two graphs in Fig. 23 were post-processed from the 'dat.dynamic' file from a 4-valve DOHC engine cold-flow calculation.
- 'dat.thermo' contains cylinder thermodynamic data: crank angle, average pressure, average temperature, average density, total volume, and total mass.
- 'dat.inject' contains cylinder fuel injection data, and is therefore intended for direct injection (DI) engines: crank angle, total fuel mass injected, total vapor mass, particle mass, vapor mass with equivalence ratio less than 0.5, between 0.5 and 1.5, and greater than 1.5. These are normalized by the instantaneous cylinder volume. Our usual assumptions are made that the fuel is species 1 and the oxidant is species 2, and that kinetic reaction 1 is the fuel oxidation reaction. Also included in the file is the instantaneous liquid fuel at or near the liner-piston interface. Liquid fuel in the layer with a thickness of 0.1 mm immediately adjacent to the walls is considered. This is intended for geometries that have a flat piston crown.
- 'dat.turb' contains cylinder turbulence data: crank angle, average  $k$ , average  $\epsilon$ , turbulence intensity, turbulence length scale, and turbulent viscosity.

The cyclic interval between writes to these four files is governed by the input quantity NCMON.

#### D. GMV (General Mesh Viewer)

A new graphics output option in KIVA-3V is the General Mesh Viewer (GMV) package, which was written at Los Alamos National Laboratory, and may be downloaded, with documentation, from the WorldWide Web. GMV is an easy-to-use 3-D scientific visualization tool designed to view simulation data from any type of structured or unstructured mesh. If the input flag named GMV = 1.0, files for post-processing by the GMV package will be written as part of the graphics output in subroutine FULOUT. GMV features include

- Read and display of any cell geometry,
- Display of nodes, cells, surfaces, and particle tracers,
- Isosurface and cutplane interpolation of grid data,



*Fig. 23. Examples of engine dynamic data from a 1600-rpm cold flow calculation of the asymmetric DOHC 4-valve engine example. The data are plotted from output file dat.dynamic, which was written by KIVA-3V subroutine MONITOR.*

- Ability to isolate a cell or groups of cells for closer analysis,
- Ability to query original cell or node data,
- A wide variety of animation options,
- A *flight* mode for interactive fly-throughs of the simulation,
- Many convenience features for high quality video production,
- A highly intuitive and well-behaved user interface, and
- Visualization of data from any current or future 3-D simulation.

Subroutine PLTGMV in KIVA-3V writes a sequentially-numbered output file tailored for GMV whenever subroutine FULOUT also writes to file OTAPE9. These files are named PLOTGMVxx, where xx counts from 00 to 99.

The GMV home page is located at

<http://www-xdiv.lanl.gov/XCM/gmv/GMVHome.html>

### VIII. OTHER CHANGES TO FILE ITAPE5

Other ITAPE5 additions and modifications that have not been described above are the following:

- IREZ: Several new options are available: IREZ = 3 is used to turn off accordioning of the squish region to ensure that the original  $\delta z$  spacing is preserved. IREZ  $\geq 4$  is used for selecting a **template** rezoner for grids with canted valves, where IREZ = 4 is used for quasi-symmetric pentroof cylinders, IREZ = 5 is used for 2-valve OHV geometries, and IREZ = 6 is used for geometries that combine both pentroof and wedge features.
- NCASPEC: If  $> 0$ , this is the number of crank angles listed in free-format in the line(s) that follow the NCASPEC line. For each crank angle in the list, subroutine FULOUT is called. This feature is useful when one wants output only at a selected number of specific crank angles.

The following lines have been deleted from ITAPE5:

- SAMPL. The SAMPL flag has been deleted, as we no longer have an operating system that supports this method of CPU-usage sampling.
- TVALVE. Individual valve temperatures now appear as part of the valve data, rendering this line obsolete.
- ANC4. We believe that most applications do not require the use of a node coupler. Accordingly, ANC4 has been moved from ITAPE5 to a data statement in the block data and has a standard value of zero. The purpose is to force the user to consider the need for node coupling.

## ACKNOWLEDGMENTS

I wish to give special thanks to my colleague Norman Johnson. Through our numerous and lengthy discussions, the implementation of the valve model in KIVA-3V was greatly enhanced. In addition, Norman has made the program and this report available in electronic form.

The liquid wall film model was created by Peter O'Rourke, and he provided algorithms for shaping valve ports and piston crowns.

Without the inspired work of Randy Hessel (now at Cummins Engine Company) to draw upon, the KIVA-3V valve model would not exist. Randy has generously offered useful advice and sound ideas through the many years that we have been collaborating, and he has served as a beta user of intermediate code versions.

Zhiyu Han at the University of Wisconsin-Madison supplied the RNG  $k-\epsilon$  turbulence option, along with a subroutine to monitor engine dynamic quantities, and has also been a beta user of the canted-valve feature.

Finally, I thank Margaret Findley for creating all the complex drawings in this report and for assisting in its final production. She has also managed the electronic distribution of the program files in a most professional manner.

## REFERENCES

1. D. C. Amsden and A. A. Amsden, "The KIVA Story: A Paradigm of Technology Transfer," *IEEE Trans. on Prof. Comm.* **36**, No. 4, pp. 190-195 (December 1993).
2. A. A. Amsden, T. D. Butler, P. J. O'Rourke, and J. D. Ramshaw, "KIVA: A Comprehensive Model for 2D and 3D Engine Simulations," SAE Technical Paper 850554 (1985).
3. 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).
4. 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).
5. 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 Technical Paper 872072 (1987).
6. 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).
7. P. J. O'Rourke and A. A. Amsden, "Three-Dimensional Numerical Simulations of the UPS-292 Stratified Charge Engine," SAE Technical Paper 870597 (1987).
8. A. A. Amsden, "KIVA-3: A KIVA Program with Block-Structured Mesh for Complex Geometries," Los Alamos National Laboratory report LA-12503-MS (March 1993).
9. 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 Technical Paper 920418 (1992).
10. Y.-X. Zhu, C. Savonen, N. L. Johnson, and A. A. Amsden, "Three-Dimensional Computations of the Scavenging Process in an Opposed-Piston Engine," SAE Technical Paper 941899 (1994).

11. F. E. Corcione, R. Rotondi, R. Gentili, and M. Migliaccio, "Modeling the Mixture Formation in a Small Direct-Injected Two-Stroke Spark-Ignition Engine," SAE Technical Paper 970364 (1997).
12. B. Dillies, A. Ducamin, L. Lebrere, and F. Neveu, "Direct Injection Diesel Engine Simulation: A Combined Numerical and Experimental Approach from Aerodynamics to Combustion," SAE Technical Paper 970880 (1997).
13. Z. Han, R. D. Reitz, J. Yang, and R. W. Anderson, "Effects of Injection Timing on Air-Fuel Mixing in a Direct-Injection Spark-Ignition Engine," SAE Technical Paper 970625 (1997).
14. Z. Han, L. Fan, and R. D. Reitz, "Multidimensional Modeling of Spray Atomization and Air-Fuel Mixing in a Direct-Injection Spark-Ignition Engine," SAE Technical Paper 970884 (1997).
15. S.-C. Kong and C.-W. Hong, "Multidimensional Intake Flow Modeling of a Four-Stroke Engine with Comparisons to Flow Velocity Measurements," SAE Technical Paper 970883 (1997).
16. J. Miok, K. Y. Huh, S. H. Noh, and K. H. Choi, "Numerical Prediction of Charge Distribution in a Lean Burn Direct-Injection Spark Ignition Engine," SAE Technical Paper 970626 (1997).
17. M. Pontoppidan, G. Gaviani, G. Bella, and V. Rocco, "Direct Fuel Injection—A Study of Injector Requirements for Different Mixture Preparation Concepts," SAE Technical Paper 970628 (1997).
18. F. X. Tanner, "Liquid Jet Atomization and Droplet Breakup Modeling of Non-Evaporating Diesel Fuel Sprays," SAE Technical Paper 970050 (1997).
19. S. L. Yang, R. Chen, M. C. Cline, H. L. Nguyen, and G. J. Micklow, "Numerical Simulation of a Low-Emission Gas Turbine Combustor Using KIVA-II," *International J. for Numerical Methods in Fluids*, **15**, 8, pp. 865–881 (October 1992).
20. M. C. Cline, J. M. Deur, G. J. Micklow, M. R. Harper, and K. P. Kundu, "Computation of the Flow Field in an Annular Gas Turbine Combustor," AIAA Paper 93-2074 (29th AIAA/SAE/ASME/ASEE Joint Propulsion Conference, Monterey, CA, June 28–30, 1993).
21. G. J. Micklow, S. Roychoudhury, H. L. Nguyen, and M. C. Cline, "Emissions Reduction by Varying the Swirler Airflow Split in Advanced Gas Turbine Combustors," *J. of Engineering for Gas Turbines and Power*, **115** (3), pp. 563–569 (July 1993).

22. S. L. Yang and M. C. Cline, "An Efficient Mapping Topology for Turbine Combustors with Inclined Slots/Staggered Holes," *J. of Propulsion and Power*, **11** (3), pp. 572–574 (May-June 1995).
23. M. C. Cline, G. J. Micklow, S. L. Yang, and H. L. Nguyen, "Numerical Analysis of the Flow Fields in a RQL Gas Turbine Combustor," *J. of Propulsion and Power*, **11** (5), pp. 894–898 (Sept.-Oct. 1995).
24. P.-K. Wu, M.-S. Chen, T. H. Chen, and A. S. Nejad, "Flowfields in a Side-Inlet Ducted Ramrocket with/without Swirler," AIAA Paper 95-2478 (1995).
25. P.-K. Wu, M.-H. Chen, R. A. Baurle, W. E. Eakins, T. H. Chen, and A. S. Nejad, "Effects of Incorporating a Cone Swirler into a Side-Inlet Ducted Ramrocket," *J. of Propulsion and Power*, **13** (1), pp. 162–164 (Jan.–Feb. 1997).
26. R. Taghavi, "CRI/TurboKiva Delivers the Power of Insight," *Cray Channels* **13** (4), pp. 26–27 (Winter 1992).
27. R. P. Hessel, "Numerical Simulation of Valved Intake Port and In-Cylinder Flows Using KIVA3," PhD Thesis, University of Wisconsin-Madison (1993).
28. R. P. Hessel and C. J. Rutland, "Intake Flow Modeling in a Four Stroke Diesel Using KIVA3," AIAA Paper 93-2952 (1993).
29. C. J. Rutland, C. M. Pieper, and R. P. Hessel, "Intake and Cylinder Flow Modeling with a Dual-Valve Port," SAE Technical Paper 930069 (1993).
30. P. J. O'Rourke and A. A. Amsden, "A Particle Numerical Model for Wall Film Dynamics in Port-Injected Engines," SAE Technical Paper 961961 (1996).
31. B. E. Launder and D. B. Spalding, *Computer Methods in Applied Mechanics and Engineering* **3**, 269 (1974).
32. B. F. Magnussen and B. H. Hjertager, "On Mathematical Modeling of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion." *Sixteenth Symposium (International) on Combustion*, The Combustion Institute, Pittsburgh (1977).
33. Z. Han and R. D. Reitz, "Turbulence Modeling of Internal Combustion Engines Using RNG k- $\epsilon$  Models," *Comb. Sci. Tech.* **106**, 207 (1995).

34. Z. Han, "Numerical Study of Air-Fuel Mixing in Direct-Injection Spark-Ignition and Diesel Engines," PhD Thesis, University of Wisconsin-Madison (1996).
35. *North American Combustion Handbook*, Second Edition (R. J. Reed, ed., North American Mfg. Co., Cleveland, Ohio, 1978).
36. *Marks' Standard Handbook for Mechanical Engineers* (T. Baumeister *et al.*, McGraw-Hill, New York, 1978).
37. S. C. Sorenson and S.-E. Mikkelsen, "Performance and Emissions of a 0.273 Liter Direct Injection Diesel Engine Fuelled with Neat Dimethyl Ether," SAE Technical Paper 950064 (1995).
38. N. B. Vargaftik, *Tables on the Thermophysical Properties of Liquids and Gases* (John Wiley & Sons, New York, 1975).
39. D. T. Jamieson, J. B. Irving, and J. S. Tudhope, *Liquid Thermal Conductivity, a Data Survey to 1973*, National Engineering Laboratory (Her Majesty's Stationery Office, Edinburgh, 1975).
40. *Handbook of Aviation Fuel Properties*, CRC Report No. 530 (Coordinating Research Council, Atlanta, Georgia, 1988).
41. C. J. Zusi, "Numerical Simulation of Laminar Flow in Wavy Wall Ducts," MSME Thesis, University of Nevada-Reno (1995).
42. V. F. Surovkin, "Analytical Description of the Processes of Nucleus-Formation and Growth of Particles of Carbon Black in the Thermal Decomposition of Aromatic Hydrocarbons in the Gas Phase," *Khimiya Tverdogo Topliva*, **10** (1), pp. 111–122 (1976).
43. B. S. Haynes and H. Gg. Wagner, "Soot Formation," *Prog. Energy Combust. Sci.*, **7**, pp. 229–273 (1981).
44. J. Nagle and R. F. Strickland-Constable, "Oxidation of Carbon Between 1000–2000°C," *Proc. of the Fifth Conf. on Carbon*, pp. 154–164 (Pergamon Press, London, 1962).

## **DISCLAIMER**

**Portions of this document may be illegible  
in electronic image products. Images are  
produced from the best available original  
document.**