

LA-13608-MS (May 1999)

*Approved for public release;
distribution is unlimited.*

Title: **KIVA-3V, RELEASE 2, IMPROVEMENTS TO
KIVA-3V**

Author(s): Anthony A. Amsden, T-3

Submitted to: *LAMS Report*

Los Alamos

NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. 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, RELEASE 2:
IMPROVEMENTS TO KIVA-3V**

by

Anthony A. Amsden

ABSTRACT

This report describes the changes made in the KIVA-3V computer program since its initial release version dated 24 March 1997. A variety of new features enhance the robustness, efficiency, and usefulness of the overall program for engine modeling. Among these are an automatic restart of the cycle with a reduced timestep in case of iteration limit or temperature overflow, which should greatly reduce the likelihood of having the code crash in mid run. A new option is the automatic deactivation of a port region when it is closed off from the engine cylinder and its reactivation when it again communicates with the cylinder. A number of corrections throughout the code improve accuracy, one of which also corrects the 2-D planar option to make it properly independent of the third dimension. Extensions to the particle-based liquid wall film model make the model somewhat more complete, although it is still considered a work-in-progress. In response to current research in fuel-injected engines, a split-injection option has been added. A new subroutine monitors the whereabouts of the liquid and gaseous phases of the fuel, and for combustion runs the energy balance data and emissions are monitored and printed. New features in the grid generator K3PREP and the graphics post-processor K3POST are also discussed.

CONTENTS

ABSTRACT	1
I. INTRODUCTION AND BACKGROUND	2
II. AUTOMATIC RESTART OF CYCLE WITH REDUCED TIME STEP.....	3
III. PORT DEACTIVATION AND REACTIVATION.....	4
IV. ACCURACY IMPROVEMENTS.....	6
A. Elimination of Diffusion Through Zero-Thickness Walls.....	6
B. Mass Conservation in Snapper Subroutines	6
C. Timestep Control on Number of Advective Flux Subcycles	8
D. UDOTN and PARTE terms in Subroutine LAWALL.....	8
E. 2-D Planar Grids.....	8
F. Fuel Library Corrections	9
G. Droplet Distortion from Sphericity	9
H. Correction of Miscellaneous Typos.....	10
V. EXTENSIONS TO THE WALL FILM MODEL.....	11
A. Splash Velocities of Secondary Droplets	11
B. Impingement Pressure Spreading	11
C. Particle Momentum and Energy.....	12
D. Gravitational Terms.....	13
E. Miscellaneous Corrections to the Wall Film Model.....	13
VI. OTHER NEW FEATURES	14
A. Split Injection	14
B. Expanded Monitoring for Engine Calculations.....	14
C. Particle Evaporation Cutoff in Subroutine EVAP	15
D. Particle Energy Coupling Above Valves in Subroutine PCOUP...16	
VII. DATA STORAGE.....	17
VIII. ADDITIONS TO FILE ITAPE5.....	22

IX.	ADDITIONS TO K3PREP	23
A.	Rounding of Runner Segments	23
B.	Valve Guides.....	24
C.	Cell-Face Boundary Conditions for Valve Stems	24
D.	Miscellaneous Changes and Notes.....	25
E.	Diesel Engine Grid with Vertical Valves and a Piston Bowl.....	26
X.	CHANGES TO K3POST.....	32
A.	Cutting Plane Rotation.....	32
B.	Plot Correction at Interface with Deactivated Region.....	32
	ACKNOWLEDGMENTS	33
	REFERENCES.....	34

I. INTRODUCTION AND BACKGROUND

Since the introduction of the original KIVA in 1985,¹⁻³ KIVA programs have become by far the most widely used CFD (computational fluid dynamics) programs for multidimensional combustion modeling. Worldwide interest grew with the 1989 introduction of KIVA-II^{4,5} and its improved numerical-solution algorithms. Although these two earlier versions lent themselves well to confined flows, they were quite inefficient for complex engine geometries or any application involving complicated boundary conditions. In KIVA-3⁶ (1993) a block-structured mesh was adopted, and the program became a practical tool for modeling geometries containing inlet and outlet ports in the cylinder wall. This development was driven by an interest in crankcase-scavenged 2-stroke engines in the early 1990s.

With the introduction of KIVA-3V⁷ in March 1997, an effective approach to modeling moving valves became available. No previous capability has been sacrificed along the way, and KIVA has matured to where it can be applied to full-engine-cycle calculations in virtually any 2- or 4-stroke design that has valves, ports in the cylinder wall, or even a combination of the two. KIVA-3V is now routinely applied to studies of both port fuel injection (PFI) and direct injection, spark-ignition (DISI) or gasoline direct-injection (GDI) engines, as well as large- and small-bore diesels, but the capabilities of the program also make it attractive for a variety of other applications that involve complex geometries, chemical reactions, liquid sprays, and wall films.

This second release of KIVA-3V makes available to the user community a variety of features and corrections that have been added to the program over the past two years. These changes were briefly outlined in the abstract above and are described in more detail in the following sections.

For those readers new to KIVA, the Los Alamos reports for KIVA-II,⁵ KIVA-3,⁶ and KIVA-3V⁷ are available electronically. The KIVA Website at <http://t3.lanl.gov/Kiva/Kiva.html> contains further information. In addition to providing links to these reports, the Website tells how to obtain KIVA-3V and how to subscribe to kiva-news and kiva-talk on the Internet; and gives links to several other installations that use KIVA programs.

II. AUTOMATIC RESTART OF CYCLE WITH REDUCED TIME STEP

Abnormal termination of a calculation has been one of the principal aggravations that KIVA users have had to contend with. In the case of a floating-point error or an out-of-bounds memory reference, the operating system simply ends the run with a message. Dynamic recovery is not an option.

More often, however, the problem is internal to the calculation and has been detected by KIVA itself. Typically there has been a temperature overflow, or an iteration limit has been exceeded. KIVA until now simply has terminated the calculation with a message, a call to subroutine FULOUT, and a copy of the drop file. Frequently, though, the user can recover by restarting from the last OTAPE8 file with a reduced time step to get the calculation past the point at which it failed.

I have overcome my reluctance to pay the penalty in computing storage and time and have automated the above procedure to attempt recovery after the following events: `tinvrt` or temperature overflow, big iteration limit exceeded, SIE driven negative in subroutine PCOUPPL, or the iteration limit exceeded in subroutine KESOLV, TSOLVE, VSOLVE, or YIT. This new feature promises to have a major impact, hopefully positive, on virtually all KIVA-3V users.

The recovery procedure works as follows. First, subroutine TIMENSAVE is called at the beginning of every cycle to save a current copy of selected common-block data. Then, if any one of the above failures occurs during the cycle, subroutine TIMENREST is called to restore the common-block data to the saved values. The time step is then halved and the cycle is repeated.

In my limited experience to date, halving the time step once has been sufficient for the calculation to recover. However, there is nothing to preclude the procedure from being repeated several times should the time step require even further reduction. The logic will not allow an endless loop; if the time step falls below its allowed minimum value, subroutine TIMSTP will terminate the run.

The whole business of temperature overflows and iteration limits being exceeded defies easy analysis. Often they may be attributed to computing numerical solutions of complex nonlinear equations with a time step that is just slightly too large, perhaps an acceptable explanation if there is nothing obviously unique about the cycle when the failure occurs. However, an overflow may be triggered if a port has just been reactivated and the current time step is much greater than it was when the port was deactivated earlier in the run. Badly distorted cells in the cylinder or failure to enforce the required degree of implicitness can also trigger failures, but these two causes will require separate consideration by the user.

III. PORT DEACTIVATION AND REACTIVATION

A new option is the automatic deactivation of a port region when it ceases to communicate with the cylinder. The deactivation feature can significantly reduce the calculation time in geometries that have long ports or runners that contain a large number of cells.

In a valved engine, deactivation occurs when the last valve in the port is closed by subroutine SNAPVTOP. If there are ports in the cylinder wall, deactivation occurs when the piston closes the last intake port or the last exhaust port in subroutine SNAPB or SNAPT. For any of these cases, entry DEACTIV in subroutine ACTIVATE is called. The port region is deactivated by setting all of the flags for the region to zero. Storage is sorted to minimize vector lengths by a call to subroutine SORT, followed by calls to subroutines VOLUME, APROJ, SETUPBC, BC, BCEPS, and GLOBAL. Deactivation is not allowed if the port region contains any spray parcels or has a fuel vapor mass greater than 1% of the total fuel mass injected.

Reactivation of a port is essentially the same process in reverse. In the case of valves, subroutine VALVE looks ahead from the current computational cycle. If it determines that a valve will open within 5 time steps of the current time, assuming the current time step will be maintained, it will reactivate the port by calling subroutine ACTIVATE. The idea is to give the flow field in the port a chance to adjust somewhat to the current pressure field before the valve opens. Reactivation will also take place immediately if subroutine NEWCYC determines that fuel injection within an intake port will commence in the current computational cycle. If the calculation is still in the setup phase, then subroutine VALVE has not yet been called, and therefore the feature that looks 5 time steps ahead is not yet in effect. In this case it is necessary to check in subroutine SNAPVTOP to see if the valve has just opened. For ports in the cylinder wall, the port region is reactivated when the first port opens in subroutines SNAPB or SNAPT.

To date, the only problems that I have encountered have been temperature overflows following the reactivation of the exhaust port in a valved engine. This is because the current time step is an order of magnitude greater than it was when the valve was closed and the port deactivated back in the beginning of the engine cycle. I have successfully dealt with this through the automatic restart of the computational cycle with a reduced time step, described in Sec. II above.

One might argue that the high velocities in the exhaust port should be set to zero when it is deactivated, on the basis that the conditions in the port will be completely different later in the cycle when the valve reopens. I have not yet addressed such a treatment, and in the current code the velocities, pressures, etc. are simply frozen at their values at the time of deactivation. Obviously, the more accurate mode of operation would be to keep intake and exhaust ports always

active, as they are in the original KIVA-3V. To give the user a choice in the matter, I have added the input flag DEACT to file ITAPE5. A value of 1.0 will permit the deactivation/ reactivation option, and a value of 0.0 will keep ports always active.

IV. ACCURACY IMPROVEMENTS

A. Elimination of Diffusion Through Zero-Thickness Walls

A small error in both the explicit and implicit diffusion of mass, temperature, k , and ϵ through closed valves has been corrected. This was discovered on 27 October 1997 in a closed-valve PFI calculation. A graph of the fuel vapor in the cylinder revealed that a minute amount of fuel vapor had begun to seep into the cylinder long before intake valve opening, commencing at the time that fuel vapor had found its way down the port to the valve seat.

The source of the error was that the cell-edge 1-4, 3-4, and 8-4 terms were not being set to zero around the periphery of each cell face that is part of the zero-thickness boundary around closed valves. Affected were subroutines YIT, RESY, EXDIF, REST, RESK, and RESE. Rather than modifying all these subroutines, however, it was much simpler to add one new do-loop at the end of subroutine BCDIFF to eliminate the diffusion, inasmuch as all six of these subroutines call BCDIFF.

B. Mass Conservation in Snapper Subroutines

The piston snapping routines SNAPB and SNAPT have assumed that the grid lines in the squish region are always perfectly vertical, i.e., aligned with the cylinder axis, in order to perfectly conserve mass. Similarly, the valve-snapping subroutines SNAPVFCE and SNAPVTOP have assumed that the grid lines above and below a valve are always aligned with the direction of valve travel.

It is generally not a problem to satisfy this criterion in a grid with vertical valves, but in a grid that has canted valves it becomes difficult if not impossible to satisfy the criterion in the rezoner, especially when the piston is near TDC (top dead center). Examples of such grids are as close at hand as the OHV (overhead valve) and DOHC (dual overhead camshaft) examples in the KIVA-3V report.⁷

Minor revisions were made in the density calculation in all four snapper subroutines on 27 June 1997 that eliminate or at least greatly reduce this source of mass nonconservation. The assumption in the original procedure was that, if a snapper made two cells out of one or one cell out of two, there was no change in the total volume involved. This is no longer assumed to be true. The logic is contained in new do-loops (do 508) added near the end of the four snapping subroutines after the new cell volumes are available.

The following description applies to a moving lower piston, but all of the valve and piston cases share the same approach, which makes use of the old and new cell volumes.

First, consider the case in which the piston is moving down and in which cell A above the piston is replaced by cells B and C in subroutine SNAPB. Before, one simply set the densities ρ (and similarly the species densities) of cells B and C equal to those of cell A, under the assumption that $V_B + V_C \equiv V_A$. Thus

$$\begin{aligned}\rho_B &= \rho_A, \\ \rho_C &= \rho_A.\end{aligned}$$

But now, these are considered to be only the initial densities of cells B and C, rather than the final densities. After the grid has been snapped and the new cell volumes V are available, the mass M is calculated for cells A, B, and C, where $M_A = \rho_A V_{A'}$ etc. The mass change δM is defined as

$$\delta M = M_A - (M_B + M_C),$$

which is used to adjust the masses of cells B and C,

$$\begin{aligned}M_B &= M_B + \delta M (V_B / (V_B + V_C)), \\ M_C &= M_C + \delta M (V_C / (V_B + V_C)),\end{aligned}$$

which in turn are used to calculate the final new densities:

$$\begin{aligned}\rho_B &= M_B / V_B, \\ \rho_C &= M_C / V_C.\end{aligned}$$

The initial species densities in cells B and C are then corrected by multiplying them by the ratio of the final to the initial density of the cells.

Second, consider the opposite case, in which the piston is moving up and in which cells A and B are replaced by cell C. As before, the initial densities (and similarly the species densities) of cell C are given by

$$\rho_C = (\rho_A V_A + \rho_B V_B) / (V_A + V_B).$$

Next, the mass M is calculated for cells A, B, and C, where $M_A = \rho_A V_{A'}$ etc. The mass change δM is defined as

$$\delta M = M_C - (M_A + M_B),$$

which is used to calculate the final new density

$$\rho_C = (M_C - \delta M) / V_C.$$

The initial species densities in cell C are then corrected by multiplying them by the ratio of the final to the initial density of the cell.

C. Timestep Control on Number of Advective Flux Subcycles

In most KIVA-3V applications, the time step is controlled most of the time by the rate-of-strain limit (RSTR). Typically, this time step is sufficiently small that the number of advective flux subcycles stays in single digits. Although an argument can be made that the time step should be reduced further to satisfy the convective limit, i. e., one flux subcycle per full calculational cycle, this is probably unnecessarily restrictive. On the other hand, in a calculation in which the number of flux subcycles remains consistently at 10 or greater, the coupling of the advection phase with the rest of the cycle is definitely questionable. Until recently, there has been no control of the time step based on the convective limit. I have recently added the quantity NFLUXMX (maximum number of flux subcycles) to subroutine TIMSTP to offer a control of the time step, and have given NFLUXMX a default value of 8 subcycles. Most calculations will not be greatly affected by this limit, but the user can raise or lower it if desired.

D. UDOTN and PARTE Terms in Subroutine LAWALL

On 8 December 1995 the quantity UDOTN was added to all 3 do-loops. The equation for UDOTN was in error until 14 November 1997, when I realized that it did not have the correct dimension of a velocity.

The background of the UDOTN term is as follows. On page 105 of the KIVA-II report,⁵ with reference to Eq. B-27 and Fig. B-1, it was stated that the normal velocity at points a, b, c, and d is negligible. For most purposes this is perhaps true, but we ascertained that the velocity component is sufficiently large that wall film particles have velocities that are not tangent to the wall. The term UDOTN was incorporated to subtract the normal component of shear stress from the UU, VV, and WW terms.

To correct this dimensional error, I moved the equations for AREA and RAREA to precede the equation for UDOTN, and multiplied UDOTN by RAREA. This procedure applies to all 3 do-loops in subroutine LAWALL.

E. 2-D Planar Grids

Although 2-D planar x-z grids have been an option in KIVA since the days of KIVA-II, the feature is now more clearly implemented. If the front and derriere faces of *every* cell are solid, the flag PLANE2D is set to 1.0 to control the front-

derriere treatment in subroutines BC, BCEPS, BCFC, BCRESEZ, CCFLUX, EXDIF, LAWALL, PINIT, REST, SETUP, SETUPBC, TIMSTP, and YSOLVE.

I discovered the error in subroutine LAWALL (described in Sec. IV.D above) when I was testing the flow through a 2-D converging duct to ensure that it was properly independent of the value of δy , the thickness of the grid in the third dimension. What I found was that the calculation was fine with small values, e.g., $\delta y \leq 1$, but as I increased the value, e.g., to 10^3 or 10^6 , the code blew up. With the LAWALL correction, the code is once again completely insensitive to the value of δy .

F. Fuel Library Corrections

In the model for a single-component mixture to represent natural gas in subroutine FUEL, my confusion of mass fractions with mole fractions resulted in errors in the molecular weight, heat of formation, and enthalpy for the mixture. Do-loops 30, 40, and 50 were rewritten 20 November 1997 to correct the errors.

The vapor pressure table for isooctane (2,2,4-Trimethylpentane) in block data FUELIB was adjusted slightly on 17 April 1998 to eliminate a sag in the data that became apparent when I plotted the curve of vapor pressure vs. temperature.

G. Droplet Distortion from Sphericity

The dynamic drag model in KIVA takes into account the distortion of airborne droplets due to the surrounding gas flow by using Taylor's analogy between a drop and a spring-mass system. However, in the calculation of the drag coefficient C_d , we have been treating the droplet as a rigid sphere, for which

$$C_d = 24/\text{Re} (1 + \text{Re}^{2/3}/6) \quad \text{Re} \leq 1000$$

$$C_d = 0.424 \quad \text{Re} > 1000.$$

It has been observed experimentally by many researchers that, when a liquid drop enters a gas stream with a sufficiently large Weber number, it deforms and is no longer spherical as it interacts with the gas. The drag coefficient of the drop should lie between the lower limit of a rigid sphere and the upper limit of a disk, which is 1.52. Liu, Mather, and Reitz⁸ proposed the expression

$$C_d = C_{d,\text{sphere}} [1.0 + 2.632 y],$$

where y is the drop distortion computed from the TAB model.⁹ In the limits of no distortion ($y = 0$) and maximum distortion ($y = 1$), the rigid sphere and disk drag coefficients are recovered, respectively. This adjustment to C_d was added to subroutines PMOM and PCOUPL on 13 August 1997.

H. Correction of Miscellaneous Typos

In subroutines DRDKE and DRDT, the fourth IF statement tests the I3 direction to determine whether the term AMUJP is to be set to 0.0. In both subroutines the quantity PERIODF should be corrected to PERIODD. This error has been present in KIVA-3, now KIVA-3V, since October 1993 when Cray-specific coding was replaced to create the workstation version. The error was reported by a user on 18 March 1998.

V. EXTENSIONS TO THE WALL FILM MODEL

A particle numerical model for the dynamics and evaporation of liquid wall films has been a part of KIVA-3V from the beginning. The basic model was described in 1996,¹⁰ and brief descriptions of the initial splash model and the model for valve seat clearing were included in the KIVA-3V report.⁷ The second-release version contains a number of extensions to the model, which are outlined below. A second SAE Technical Paper¹¹ that describes all the extensions to the model to date is in preparation. *It is important to note that the entire wall film model is still considered a work-in-progress and is expected to undergo further revisions and additions in the future. More comparison with experimental data will be required before the model can be considered validated.*

A. Splash Velocities of Secondary Droplets

When new spray droplets are formed as a result of splash, they are known as secondary droplets. Perhaps because of the newness of detailed multidimensional models of impinging sprays, little research has been done to provide information on the splash process, necessary for model validation. A great area of uncertainty remains in the specification of the velocity components for secondary droplets. The splash model in the second release of KIVA-3V has been expanded to allow for a distribution of velocities of the secondary droplets. On the basis of observed distributions, the secondary droplet velocities \mathbf{w} are now approximated by

$$\mathbf{w} = \omega' \mathbf{n} + v'(\cos \psi \mathbf{e}_t + \sin \psi \mathbf{e}_p) + 0.8 v_o \mathbf{e}_t,$$

where \mathbf{n} is the unit normal to the wall surface; \mathbf{e}_t is the unit vector tangent to the surface and in the plane of \mathbf{n} and the incident drop velocity; $\mathbf{e}_p = \mathbf{n} \times \mathbf{e}_i$; v_o is the component of the incident drop velocity tangent to the surface; and ω' , v' , and ψ are random variables. The quantity ω' is the normal component of the secondary droplet velocity, v' is the fluctuating component of the secondary droplet tangential velocity, and ψ is the angle that the fluctuating tangential velocity makes with the vector \mathbf{e}_t in the plane of the wall.

B. Impingement Pressure Spreading

The earlier version of the wall film model ignored the wall film pressure gradients that result from the normal momentum of an impinging spray. As a result, if a spray was directed perpendicular to a wall, the liquid remained confined to the impact area on the surface and never spread, simply accumulating to an increasing depth.

A model to remove this deficiency was added to KIVA-3V in the summer of 1997, and is embodied in a new subroutine named SPREAD. The approach makes use of the surface pressure differences between the cell face that the liquid is impinging upon and the four abutting surface faces. These differences are then used to accelerate wall film outward from an impingement site.

C. Particle Momentum and Energy

The calculation of the particle momentum has been expanded to include the effect that the momentum of an impacting droplet has on the momentum of film that is already on the wall, and vice versa. As part of this more accurate approach, the UFILM-, VFILM-, and WFILM- arrays were replaced by UMOM-, VMOM-, and WMOM- arrays on 17 November 1997, as it became more appropriate to work with the momenta.

If there is impingement in the present cycle, the pressure terms relating to impingement pressure spreading (see Sec. V.B above) are calculated and used to modify the momentum source term arrays before they are used in the film velocity calculation.

The new velocity of each film particle is based upon the wall shear stresses, the direction of gas flow relative to the wall, the mass source, the momentum source accumulated in subroutine SPLASH and modified in subroutine SPREAD, the momentum of the existing wall film, the mean film temperature, and the liquid viscosity at this temperature.

Also as part of the spray/wall interaction submodel, an energy term ETERM has been introduced in the calculation of the wall particle temperature. ETERM approximates the impingement source term in the particle energy equation, multiplied by the film thickness and divided by twice the film conductivity. When spray impinges on a wall film, the impingement mass mixes with pre-existing film mass. Thus, ETERM for particles already on the wall that have mixed with particles sticking to the wall during the current computational cycle is different from ETERM for particles that have just stuck to the wall and mixed with pre-existing wall film particles. The new flag JUSTHIT is used to distinguish between the two types of particles.

ETERM is subtracted in the equation for the particle surface temperature TPSURF, but we soon discovered that this could drive TPSURF negative. Analysis indicated that our traditional first guess for the new particle temperature, $TP_{NEW} = TP^n$, was no longer a good choice. A new equation for the first guess for TP_{NEW} , which eliminated the problem, uses ETERM as a positive quantity. The derivation of the new TP_{NEW} equation is based upon the observation that upon convergence, $TP_{SURF} \approx TP^{n+1}$. These changes were added 22 May 1997.

D. Gravitational Terms

The effects of gravity on a wall film are considered minor in applications to operating engines. However, for comparison of KIVA-3V results with bench experiments of impinging sprays and liquid film formation, the time scale is such that the effects may become nontrivial. To eliminate this area of uncertainty, gravitational terms were added to the wall film velocities on 14 April 1998.

E. Miscellaneous Corrections to the Wall Film Model

Several corrections have been made in the wall film model that are not directly related to the items discussed above.

- In the equation for DSIEP, the energy-coupling contribution in subroutine WALLFILM, the term HT was corrected to HTRANS on 25 April 1997.
- The particle mass term PARTE erroneously included a factor of FACSEC, which was removed 15 April 1997. PARTE is used in the calculation of the film thickness in subroutines LAWALL, PMOVTV, and SPLASH.
- A test was added to subroutine CLEAR on 2 April 1997 to ensure that a valve seat is cleared of all film when the valve closes. Previously, if the valve lift suddenly went to zero from a value that was greater than the remaining film thickness, the remaining film was left on the seat because there was no further opportunity to finish clearing it. When this situation arises, the added test sets the old value of the valve lift to 10^{-20} , which will then force the final clearing to occur.
- Tests were added 27 June 1997 to subroutine FILMSNAP to ensure that an adjustment of index I4P is made only if the particle is on the surface that is being snapped.

VI. OTHER NEW FEATURES

A. Split Injection

Today's increasingly sophisticated engine management systems are allowing designers to explore complex fuel-injection schedules. Whereas in the past a single fuel-injection pulse per engine cycle was the norm for both gasoline and diesel engines, a number of current designs utilize more than one injection pulse per engine cycle. Split injection is the designation for a schedule that has two or more distinct pulses. In some direct-injection gasoline engine designs, for example, fuel is injected twice—an early pulse during the intake stroke and a late pulse during the compression stroke. There are now some diesel injectors that deliver a sequence of pulses.

The fuel-injection model in KIVA-3V has been expanded to allow split injection in addition to the ordinary single injection. A new input quantity NUMINJ specifies the number of injection pulses per engine cycle, which can range from 1 to a maximum of LINJ, LINJ being a parameter in file comkiva.i that currently has a value of 10. For each pulse, a set of the six input quantities T1INJ, TDINJ, CA1INJ, CADINJ, TSPMAS, and TNPARC are specified. Thus each pulse has its own window, either in seconds or crank angle degrees, and its own liquid mass and number of spray parcels.

B. Expanded Monitoring for Engine Calculations

Over the past several years I have been involved in modeling studies of fuel-injected engines. As a diagnostic aid, I added a number of patches that calculated and wrote to output file OTAPE12 quantities related to the liquid and gaseous phases of the fuel and, for combustion runs, the energy balance data and emissions data. With the thought that this information might be useful to other engine researchers, I have integrated these diagnostics into the program.

Subroutine NEWCYC calls a new subroutine named FUELMON if there is any fuel in the system. FUELMON monitors both the liquid and gaseous phases of the fuel and considers the entire computing domain. It keeps track of the masses and current locations of wall film, entrained liquid and vapor. Four lines are written to the user's monitor and to file OTAPE12. Each line begins with a unique label that allows the user to edit the file and select a specific dataset for postprocessing with data-analysis graphics software:

Line 1: crank angle; total fuel mass in the system; liquid mass entrained in regions 1–3; vapor mass entrained in regions 1–3.

Line 2: crank angle; fuel film mass on the valve surfaces, by surface number, for up to 8 surfaces (4 valves).

Line 3: crank angle; fuel film mass on piston, liner, head, and walls of port regions 2 and 3.

Line 4: crank angle; entrained liquid mass lost out the left and right ports; film mass lost out the left and right ports; vapor mass lost out the left and right ports.

For combustion runs, the data printed by subroutine GLOBAL has been expanded to include energy balance data and emissions data:

The energy balance line contains the crank angle, the cumulative heat release from all chemical reactions, the cumulative change in total energy in the cylinder, the cumulative wall heat loss in the cylinder, and the cumulative pressure work done by the piston on the gas (all in ergs) as well as the current total mass of pre-mixed fuel.

The emissions line contains the crank angle and the amounts of CO₂, CO, and NO in the cylinder, in parts per million.

As in the original KIVA-3V, subroutine GLOBAL continues to print cumulative mass flux through open boundaries and cumulative mass flux through each valve.

In subroutine MONITOR, the cylinder fuel injection data written to file dat.inject has been revised from that of the original KIVA-3V. With the thought that subroutine FUELMON now comprehensively reports the fuel mass data, dat.inject now focuses on equivalence ratio information. There are now four rather than three sorting bins: vapor mass with equivalence ratio less than 0.01, between 0.01 and 0.5, between 0.5 and 1.5, and greater than 1.5. Printed are the crank angle, the total vapor mass in each of the four bins, and the total volume associated with each of the four bins.

C. Particle Evaporation Cutoff in Subroutine EVAP

A shortcoming of the numerical procedure in subroutine EVAP was encountered, which occurred when the mass that it wanted to evaporate in an iteration was minuscule. As a result, the corresponding number of iterations (NIT) was in the millions and the computer time became enormous. A test to eliminate this pathological case was added 23 April 1998. If the mass to be evaporated is less than 10^{-6} the mass of the parcel, the iteration is terminated. The circumstance is transient and should automatically resolve itself over succeeding cycles.

D. Particle Energy Coupling Above Valves in Subroutine PCOUPPL

Failures have occurred in port fuel-injection studies whereby, during coupling of particle specific internal energy to the gas field in cells immediately above the top of an intake valve, the specific internal energy of the cell is driven negative. When such failures have occurred, the volume of the cell has been quite small, a consequence of using the snapping technique to move the valves.

A test was added to subroutine PCOUPPL on 9 May 1997 that checks to see whether the proposed reduction of internal energy is greater than a quarter of the SIE of the cell. If so, the change is limited to that amount, and the remainder is taken out of the cell immediately above, which generally has a much larger volume and hence much more energy to offer than the affected cell.

Although this is clearly an ad hoc approach, it conserves specific internal energy and has eliminated the problem. As a backup, the automatic restart of the cycle with a reduced time step (see Sec. II) is available if the SIE is ever driven negative, from whatever cause, in subroutine PCOUPPL.

VII. DATA STORAGE

Storage has been increased by the requirements of the automatic restart of a cycle with a reduced time step, discussed in Sec. II. A new common block with array names beginning with "RE" contains beginning-of-cycle values of the necessary cell and vertex data. In addition, there are copies of most of the other common blocks.

There was a choice as to how to make copies of the common blocks. One choice would have been to continue to use the LOCR and LOCI functions that determine the lengths of real and integer arrays. This would require the creation and maintenance of complete duplicate copies of each common block, line by line, with the names slightly changed. For example, every name could be preceded by "RE." This approach would ensure that the duplicate commons matched the originals and were of the correct length.

The simpler approach that was chosen was to completely eliminate LOCR and LOCI and return to the use of parameter statements that specify the lengths of the common blocks. Each duplicate common block contains a single array that is given the dimension specified by the parameter statement. Changes were also required in subroutines BEGIN, SORT, TAPERD, and TAPEWR. The user must now remember, however, to adjust the length of the parameter statement when making changes to a common block.

Table 1 shows the allocation of the equivalenced arrays in the new release version of KIVA-3V. The differences between this and the table in Ref. 7 are principally in the wall film model, in which arrays with the name -FILM- have been replaced by arrays with the name -MOM-, and in the addition of the energy source terms for splashing droplets: PLQ, PFQ, and PBQ. These were discussed in Sec. V. The existing array RON is now also used in the four snapping subroutines as part of the mass conservation improvement discussed in Sec. IV.B. There has been no increase in the number of equivalenced arrays.

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

SUB-	SETUP	VISC	TIMSTP	NEWCYC	INJECT	PMOVRTV	BREAK	COLIDE	EVAP	LAWALL
ROUTINE	ADJPISTN			ACTIVATE	FRAN	FRAN	FRAN	FRAN		BC
\	ADJVALVE			FULOUT	PFIND	PFIND		REPACK		
\	APROJ, BC			MONITOR		SPLASH				
\	CONVEX			FUELMON		REPACK				
\	PISTON			TAPEWR						
ARRAY \	SETUPBC			TIMESAVE						
\	SORT.STATE									
\	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									TOTH	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						PLQ	*****	*****	*****	*****
E60						PFQ	*****	*****	*****	*****
E61						PBQ	*****	*****	*****	*****

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	CHEMEQ	PMOM	PCOUP	YSOLVE	(OPTIONAL)	EXDIF	PINIT
\	CLEAR	BCNODCPL		(OR	REPACK	BC	YIT	WRITE	BC	
\	REPACK	BC		CHMQGM)			RESY	E4	BCROT1	
\	SPREAD						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, PRESL	AUGMV					(CL)	(CL)	(CL)	*****
E5	TAUF, PRESF						(CLJ)	(CLJ)	(CLJ)	*****
E6	TAUB, PRESB						(CLK)	(CLK)	(CLK)	*****
E7	TAULX, TAULLX						(CFI)	(CFI)	(CFI)	*****
E8	TAULY, TAULLY						(CFJ)	(CFJ)	(CFJ)	*****
E9	TAULZ, TAULLZ						(CFK)	(CFK)	(CFK)	*****
E10	TAUF, TAUFFX						(CBI)	(CBI)	(CBI)	*****
E11	TAUFY, TAUFFY						(CBJ)	(CBJ)	(CBJ)	*****
E12	TAUFZ, TAUFFZ						(CBK)	(CBK)	(CBK)	*****
E13	TAUBX, TAUBBX						(RFSUM14)	(RFSUM14)	(RFSUM14)	*****
E14	TAUBY, TAUBBY						(RFSUM34)	(RFSUM34)	(RFSUM34)	*****
E15	TAUBZ, TAUBBZ						(RFSUM84)	(RFSUM84)	(RFSUM84)	*****
E16	PLM	DUDX	*****	*****	*****	*****	*****	*****	DUDX	*****
E17	PFM	DUDY	*****	*****	*****	*****	*****	*****	DUDY	*****
E18	PBM	DUDZ	*****	*****	*****	*****	*****	*****	DUDZ	*****
E19	TMEANL	DVDX	*****	*****	*****	*****	*****	*****	DVDX	*****
E20	TMEANF	DVDY	*****	*****	*****	*****	*****	*****	DVDY	*****
E21	PLX	DVDZ	*****	*****	*****	*****	*****	*****	DVDZ	*****
E22	PLY	DWDX	*****	*****	*****	*****	*****	*****	DWDX	*****
E23	PLZ	DWDY	*****	*****	*****	*****	*****	*****	DWDY	*****
E24	PFY	DWDZ	*****	*****	*****	*****	*****	*****	DWDZ	*****
E25	PFZ						PHID	*****	PHID	*****
E26	PBX						ENTHDF		DISPTIL	*****
E27	PBY						SPD14, DYP14		TEM14	
E28	PBZ						SPD34, DYP34		TEM34	
E29	HL				RU	RU	SPD84, DYP84		TEM84	
E30	HF				RV	RV	HISP		UTIL	*****
E31	HB				RW	RW	SPMTIL		VTIL	*****
E32	UMOML, TMNUML						ENTHTIL		WTIL	*****
E33	VMOML, TMDENL						DDY		RMVSU	*****
E34	WMOML						RES			
E35	UMOMF, TMNUMF						RESOLD			
E36	VMOMF, TMDENF						DRES			
E37	WMOMF						RDRDY			
E38	UMOMB, TMNUMB						DELTAY			
E39	VMOMB, TMDENB						DELYPH		TKE14	
E40	HTL	UB					XCEN		TKE34	PN
E41	HTF	VB					YCEN		TKE84	PHIP
E42	HTB	WB					ZCEN		EPS14	
E43	HYL						YSPM		EPS34	
E44	HYF						YSPD		EPS84	
E45	HYB								TT1MPH	
E46	*****	*****	*****	*****	DMTOT	DMTOT			TKE1MPH	
E47	*****	*****	*****	*****	*****	*****			EPS1MPH	
E48	*****	*****	*****	*****	DSIEP	DSIEP				
E49	*****	*****	*****	*****	DTKEP	DTKEP			CPC	*****
E50	*****	*****	*****	*****	*****	*****	*****	*****	RON, HTCTIL	*****
E51	*****	*****	*****	SIEN	*****	*****	*****	*****	SIEN, TTIL	*****
E52	TMEANB								TKETIL	*****
E53	*****	*****	*****	*****	*****	*****	*****	*****	EPSTIL	*****
E54	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
E55	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
E56	*****	UN	*****	*****	UN	*****	*****	*****	UN	*****
E57	*****	VN	*****	*****	VN	*****	*****	*****	VN	*****
E58	*****	WN	*****	*****	WN	*****	*****	*****	WN	*****
E59	PLQ									
E60	PFQ									
E61	PBQ									

Table 1. (continued)

SUB-ROUTINE	PGRAD (1.0)	VSOLVE	TSOLVE	PSOLVE	(OPTIONAL)	PGRAD (3.0)	PGRAD (1.0)	PHASEB	KESOLV	PACCEL
\	BCPGRAD	RESUJW	TINVRT	PGRAD (2.0)	READ	BCPGRAD	BCPGRAD		DRDKE, RESK	
\	BC	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	*****	*****	GAMMA, RGAMMA	RGAMMA	*****	*****	*****	RGAMMA	*****	*****
E3	*****	*****	*****	*****	*****	*****	*****	*****	EPS	*****
E4	*****	*****	(CLJ)	PL	(CLJ)	*****	*****	*****	(CLJ)	*****
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		PTM	*****	*****	*****	*****	*****	*****
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	TEM14	UALA	*****	*****	*****	*****	*****	*****
E28	*****	RESV	TEM34	UAFA	*****	*****	*****	VOLL	VOLL	*****
E29	*****	RESW	TEM84	UABA	*****	*****	*****	*****	*****	*****
E30	*****	*****	*****	*****	*****	UTIL	*****	*****	*****	*****
E31	*****	*****	*****	*****	*****	VTIL	*****	*****	*****	*****
E32	*****	*****	*****	*****	*****	WTIL	*****	*****	*****	*****
E33	RMVSU	RMVSU	*****	RMVSU	*****	RMVSU	RMVSU	*****	RDRDE	*****
E34	*****	RESUO	RES	RES	*****	*****	*****	*****	RES	*****
E35	*****	RESVO	RESOLD	RESOLD	*****	*****	*****	*****	RESOLD	*****
E36	*****	RESWO	DRES	DRES	*****	*****	*****	*****	DRES	*****
E37	*****	DRESU	RDRDT	RDRDP	*****	*****	*****	*****	RDRDK	*****
E38	*****	DRESV	DTEMP	DP	*****	*****	*****	*****	DELTKE, DEPS	*****
E39	*****	DRESW	TEMPHID	PPHIP	*****	*****	*****	*****	TKE14	*****
E40	PN	*****	PN	PN	*****	PN	PN	PN	TKE34	*****
E41	PHIP	*****	*****	PHIP	*****	PHIP	PHIP	*****	TKE84	*****
E42	*****	UB	*****	UB	*****	*****	*****	*****	EPS14	*****
E43	*****	VB	*****	VB	*****	*****	*****	*****	EPS34	*****
E44	*****	WB	*****	WB	*****	*****	*****	*****	EPS84	*****
E45	*****	DISSIP	DISSIP	*****	*****	*****	*****	DISSIP	DISSIP	*****
E46	*****	DUHAT	HTC	*****	*****	*****	*****	HTC	RTERMK	*****
E47	*****	DVHAT, VOLB	VOLB	VOLB	*****	*****	*****	VOLB	RTERME	*****
E48	*****	DWHAT	*****	*****	*****	*****	*****	RROVOLL	RROVOLL	*****
E49	*****	*****	CPC	*****	*****	*****	*****	*****	TKEPHID	*****
E50	*****	*****	HTCTIL	*****	*****	*****	*****	*****	EPSPHID	*****
E51	*****	*****	TTL	*****	*****	*****	*****	*****	*****	*****
E52	*****	*****	*****	*****	*****	*****	*****	*****	TKETIL	*****
E53	*****	*****	*****	*****	*****	*****	*****	*****	EPSTIL	*****
E54	*****	*****	*****	*****	*****	*****	*****	*****	TKEN	*****
E55	*****	*****	*****	*****	*****	*****	*****	*****	EPSN	*****
E56	*****	*****	*****	UN	*****	*****	*****	UN	*****	*****
E57	*****	*****	*****	VN	*****	*****	*****	VN	*****	*****
E58	*****	*****	*****	WN	*****	*****	*****	WN	*****	*****
E59	*****	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, BDNF			
E7				CFX, BNDD			
E8				CFY, BNDB			
E9				CFZ, BNDD			
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				ROSCL	WMOM		
E32				ROSIEV	SMOM		
E33				ROTKEV	FXV		
E34				ROSCLV	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		RON	
E51				CCIP			
E52				CCJM			
E53				CCJP			
E54				CCKM			
E55				CCKP			
E56				DCCL			
E57				DCCF			
E58				DCCB			
E59				ROSIEV			
E60				ROTKEV			
E61				ROSCLV			

VIII. ADDITIONS TO FILE ITAPE5

It is with some reluctance that I have added quantities to input file ITAPE5, but it was necessary in order to properly support two new KIVA-3V features described above:

1) DEACT, which controls port deactivation/reactivation, described in Sec. III, and

2) NUMINJ, the number of injection pulses, described in Sec. VI.A. There are NUMINJ sets of the following six quantities: T1INJ, TDINJ, CA1INJ, CADINJ, TSPMAS, and TNPARC. Note that there is a local rearrangement of the injection data on file ITAPE5 to make these six quantities contiguous.

IX. ADDITIONS TO K3PREP

Several new optional features have been added to the parametric grid generator K3PREP, which result in additional lines in the IPREP file.

A. Rounding of Runner Segments

The intake and exhaust regions in the four examples of valved engines in the KIVA-3V report⁷ are composed of valve ports and attached single-block runners. Shaping of the blocks is performed by subroutines VALVPORT, RUNNER, and SIAMESE. In all these examples, the runners are a single block that extends from a valve port to a pressure boundary, and represent the portions of the intake and exhaust passages that lie within the cylinder head casting. Although this block definition is adequate for the sample geometries, other engine designs require much more complex runners, with such features as long curved shapes, helical ports, and bifurcations. Also, it may be desirable to include a much greater portion of the induction system.

The creation of a helical intake port, as one example, requires a large number of blocks to achieve a smooth shape. Rounding the edges of these added runner segments is beyond the scope of subroutine RUNNER, and thus a new subroutine named ROUND has been added to address this need.

- NROUND (format A10,I5) is the number of runner segments to be rounded. If NROUND > 0, (NROUND) lines follow, each of which contains:
- N, NORIENT, RAD1, RAD2 (format 2I4,2F8.3), which says: Round the solid edges of runner extension block N. NORIENT specifies the logical orientation of flow through the segment, and RAD1 and RAD2 specify the radii of curvature of the solid edges. In the following chart L, R, F, D, B, and T have their usual KIVA meanings of left, right, front, derriere, bottom, and top:

NORIENT:	RAD1:	RAD2:
1 = L ↔ R	FB & FT	DB & DT
2 = F ↔ D	LB & LT	RB & RT
3 = B ↔ T	LF & RF	LD & RD

The NROUND command lines follow NSIAMESE and precede NPATCH in the IPREP file.

B. Valve Guides

Previously there was no provision in K3PREP to model valve guide intrusion into a valve port. Subroutine VALGUIDE has been added to shape and deactivate specified cells to represent valve guides.

- NVGUIDE (format A10,I5) is the number of valve guides to be modeled. If NVGUIDE > 0, (NVGUIDE) lines follow, each of which contains:
- N, NXYGUIDE, NZGUIDE, XSHIFT, YSHIFT, RADGUIDE, ZTOBOT (format 3I4,4F8.3) says: Create a valve guide in valve port N that is NXYGUIDE cells wide by NZGUIDE cells high, and is centered at (XSHIFT, YSHIFT), where (XSHIFT, YSHIFT) is the x and y translation of the blocks. The guide has a radius of RADGUIDE cm, and its lower edge is at ZTOBOT cm up from the edge of the valve seat (bottom of block N).

The NVGUIDE command lines follow NPENTXY and precede NVALVPORT in the IPREP file.

How accurately the user can represent a valve guide depends on the particular geometry. The resulting shapes of the cells around the guide are the most satisfactory when the base of the guide extends some distance into the port. However, engine designers try to make guides as nonintrusive as possible, balancing the requirement for good valve stem support with the need to minimize flow blockage. If the base of the valve guide intersects the outer radius of the port, or worse, if it lies above the port, it may be difficult or impossible to achieve satisfactory zoning because cell edges get collapsed on top of one another. This is one of the difficulties of working with a structured mesh. If an accurate representation of a valve guide produces an unacceptable mesh in the port, the user should either sacrifice some accuracy by moving the bottom of the guide down a bit, or forego the valve guide option entirely.

C. Cell-Face Boundary Conditions for Valve Stems

If you are generating a vertical-valve mesh with curved valve ports and are using the original IPREP for the vertical-valve example in Ref. 7 as a pattern, you will need to change the boundary condition for the left, right, derriere (and front if full-circle mesh) faces of the valve stems from 2.0 (SOLID) to 1.0 (MOVING), for blocks 23, 24, 25, and 39, 40, 41, along with specifying the appropriate IDFACE (4 or 2). Failure to do this will result in crooked valve stems, which arise in subroutine VALVPORT.

The original vertical valve mesh was created and in use a year before subroutine VALVPORT was written for the first canted-valve mesh. The mesh was to be used in a port fuel-injection study with our particle model for liquid

wall films. For the film to move properly on the intake valve stems, it was necessary to make the left, right, front, and derriere faces on the stems MOVING instead of SOLID. Note that KIVA-3V never actually moves the stem vertices, which ensures that the zoning never changes around the stem, but the current velocity of the valve is the wall velocity needed in the wall film model.

By the time I wrote subroutine VALVPORT, I was thinking in terms of valve stems being MOVING. Note that in the Do 260 loop of the subroutine there is an ELSEIF statement that includes a test on IDFACE(I4) \geq 2. The result of this test is that valve stems are left where they belong, but it also assumes that the valve stem is MOVING, with a correct IDFACE (2, 4, ...). An IDFACE of -1 will allow the (x,y) coordinates of the valve stem to change and result in its becoming crooked.

The cell-face boundary condition problem went undiscovered until I first curved the ports in the vertical-valve example.

D. Miscellaneous Changes and Notes

- A seventh column named IFIXED has been appended to the lines of block-reshaping commands read by subroutine RESHAPEB. Normally, IFIXED = 0; it is set to 1 only when reshaping a cylinder to match the outline of a bowl block below it. An example is shown in Sec. IX.E below.
- The earlier K3PREP could require user patches in subroutine RESHAPEB if a straight line was to be enforced across $x = 0$ or $y = 0$. The need for a patch has been eliminated by the addition of two flags in the IPREP file. If RESHAPE > 0, the two integer flags STRAIGHTX0 and STRAIGHTY0, each with format (A10,I5), must now be inserted at the end of the list of reshaping commands. If STRAIGHTX0 = 1, the mesh will have a straight line through $x = 0$, and if STRAIGHTY0 = 1, the mesh will have a straight line through $y = 0$.
- XBAND, YBAND, and ZBAND have been added to the IPREP file, following the PLOTMESH line. These have the same purpose as the quantities of the same name in K3POST (see pages 69–70 in Ref. 7), which is to control the thickness of the band considered in the plot subroutines ZONPLTX, ZONPLTY, and ZONPLTZ.
- Subroutine ZONHIDE now automatically makes a mirror-image plot across $y = 0$ for half-circle symmetry meshes, provided there is adequate storage ($NVERTS*2 \leq NV$) for the reflected vertex and cell-face data.

E. Diesel Engine Grid with Vertical Valves and a Piston Bowl

The set of sample grids with valves in Ref. 7 lacked an example with a true piston bowl. I have taken the vertical-valve example and added a Mexican hat bowl to create a prototype mesh for a direct-injection diesel engine. Several views are shown in Figs. 1 and 2. In addition to adding the bowl, I have used subroutine VALVPORT to shape the valve ports. Figure 3 is a drawing of the logical mesh that corresponds to this grid, and a plan view drawing is shown in Fig. 4. These latter two figures correspond to Figs. 6 and 9 in the KIVA-3V report.⁷ Fig. 5 shows the 19 points that define the bowl profile.

Besides adding another block to represent the bowl, it is necessary to embed the bowl outline up through the squish region through the range of valve travel. In a canted-valve geometry this can only be approximated, but the procedure is more straightforward in a vertical-valve geometry and can be expected to result in a robust grid.

My approach was to assign block number 1 to the bowl, which meant that the original set of blocks are now numbered 2–42 rather than 1–41. In addition, I added a final ghost block (block 43) for the purpose of carrying the bowl outline up through the squish region.

Note in the Fig. 2 plot across $z = 1.1557$, which is the level of the piston crown, that the circular shape of the bowl outline has been superimposed on the previously existing circular shapes of the valves, pockets, and ports. The (x,y) coordinates of the intersection points of the various overlapping circles were determined using the Law of Cosines and appear as specified LOXCY points in the IPREP file.

A comparison of the IPREP file with that for the earlier vertical-valve example shows a number of other differences:

- The (r,z) coordinates of the bowl outline in Fig. 5 have been added to the data for block 1. This is followed by a new data line ANGBOWL, which specifies the angles for the tent vertices for a concentric (r,z) piston bowl. In this particular geometry, 60 and 120 degrees produced a much more satisfactory grid than 45 and 135 degrees.
- The left, right, and derriere faces of the valve stems (blocks 24–26 and 40–42) are MOVING rather than SOLID and have IDFACE of 2 or 4. This is in accordance with the discussion in Sec. IX.C above.
- IFIXED, the seventh column of data appended to the lines of reshaping commands, has a value of 0 except for the final line, in which ghost block 43 is reshaping the cylinder to match the outline of the piston bowl below.

- STRAIGHTY0 = 1 to ensure that the mesh has a straight line through $y = 0$.
- RELAXB = 1 to relax the interior vertices of the piston bowl.

AN IMPORTANT NOTE CONCERNING THE GENERATION OF THIS GRID:

When I first developed this diesel engine grid with vertical valves and a piston bowl in August 1997, I had to make a number of patches in the then-current 1997 release version of K3PREP to get it to work. In December 1997 I created an interim new version of K3PREP that incorporated these patches. With the more powerful K3PREP, the input file IPREP could now supply all the necessary new information, as described above.

However, in the summer of 1998, I made a few further modifications to K3PREP based on a different bowl geometry that I was generating. Unfortunately, one new if-statement in subroutine RESHAPEB had the unintended effect of ruining the interior grid in the piston bowl of the diesel example. I discovered this on 22 April 1999 when I tried to generate it using the new release version of K3PREP (dated 1 October 1998) while writing this section of the report. Therefore, it is necessary to comment out line RESHAPEB.72, which reads:

```
if(fv(i4).eq.flbowl .or. fv(i8).eq.flbowl) ireshape(i4)=1
```

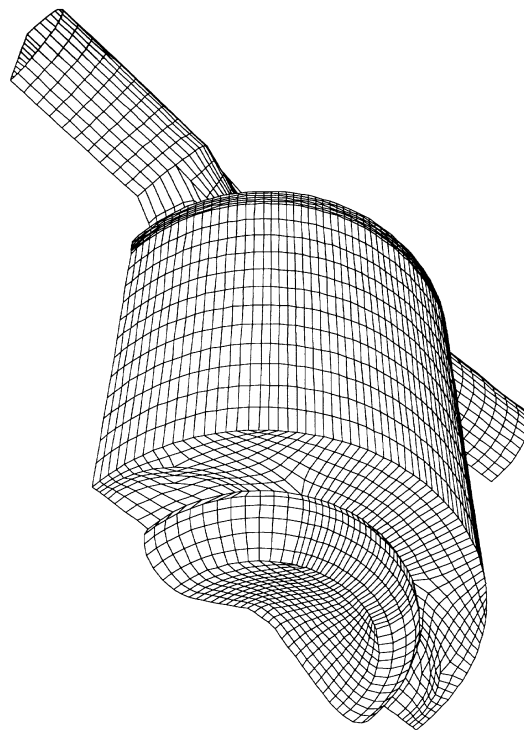
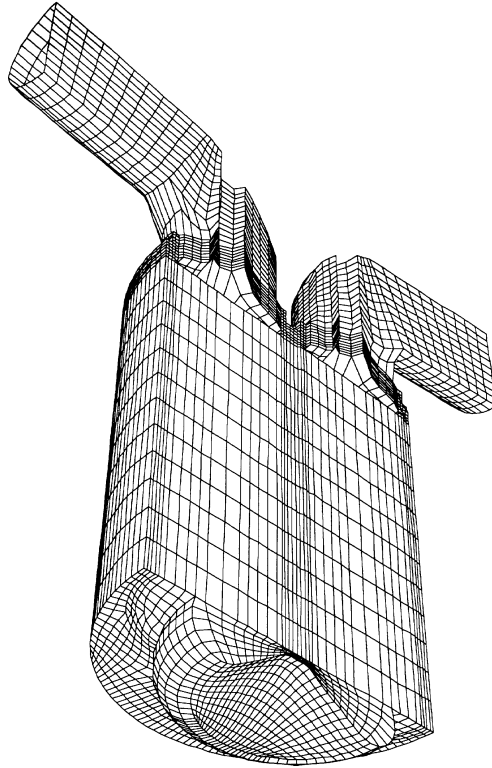


Fig. 1. Half-circle mesh for the direct-injection diesel engine example with two vertical valves and a Mexican hat piston bowl. The mesh contains 24,013 grid points.

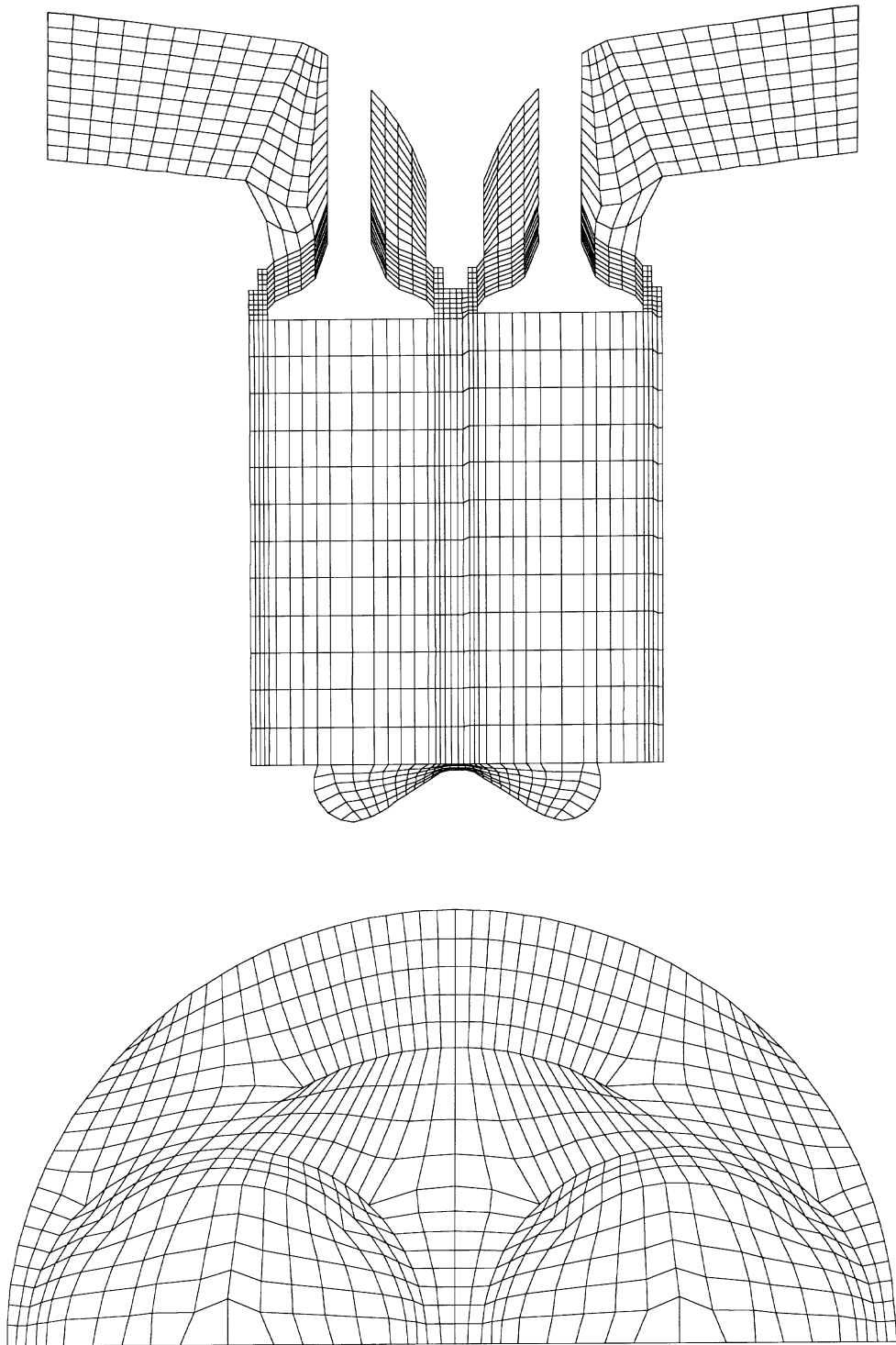


Fig. 2. Top: the diesel engine mesh through the $y = 0$ symmetry plane. Bottom: plan view through the squish region.

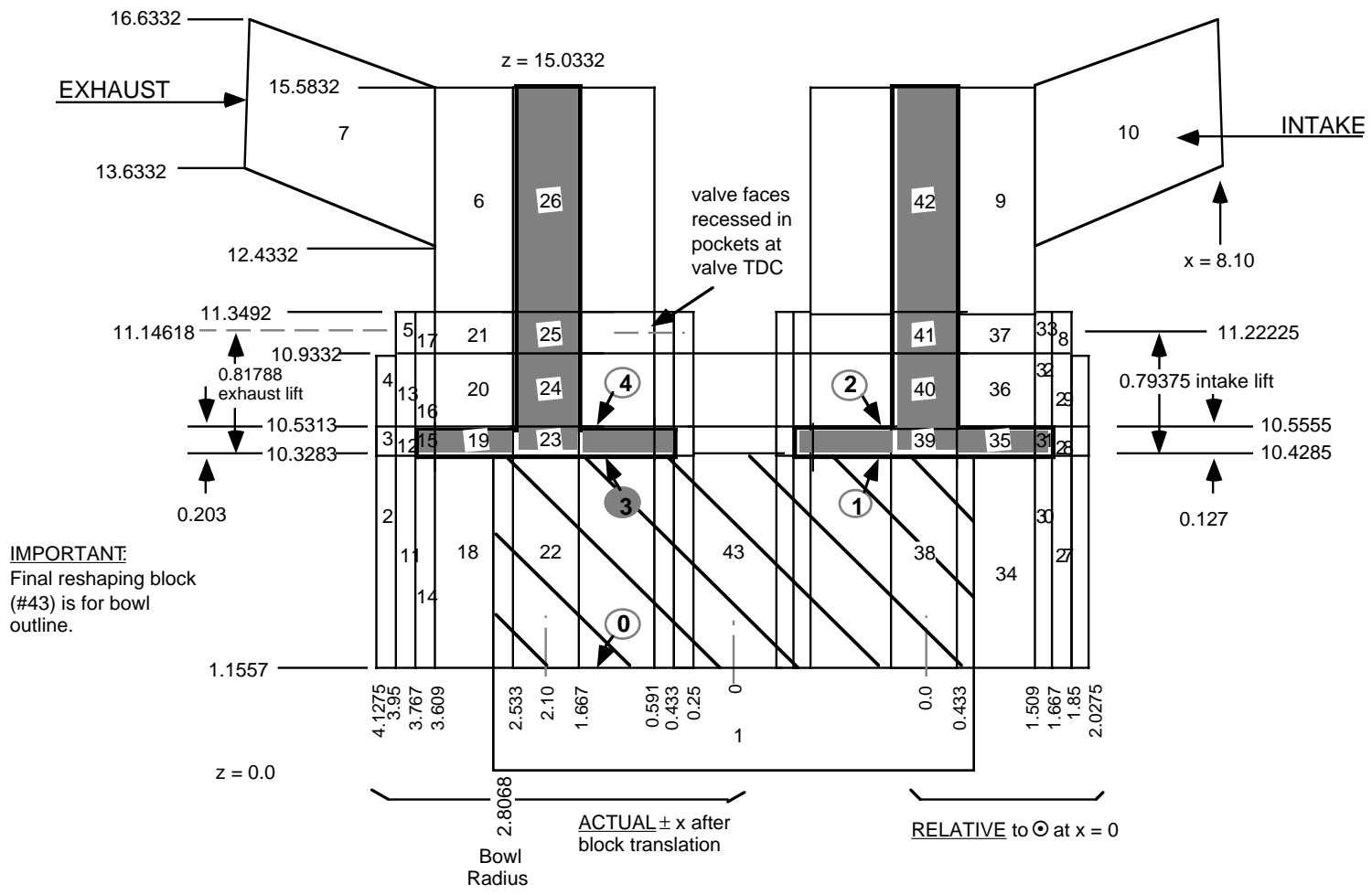


Fig. 3. The logical mesh for the diesel engine example, showing the numbering of the blocks for K3PREP, all necessary dimensions and, in circles, the numbering of the moving surfaces. The z dimensions have been increased by the bowl depth, 1.1557 cm, from those in the original vertical-valve example in Ref. 7.

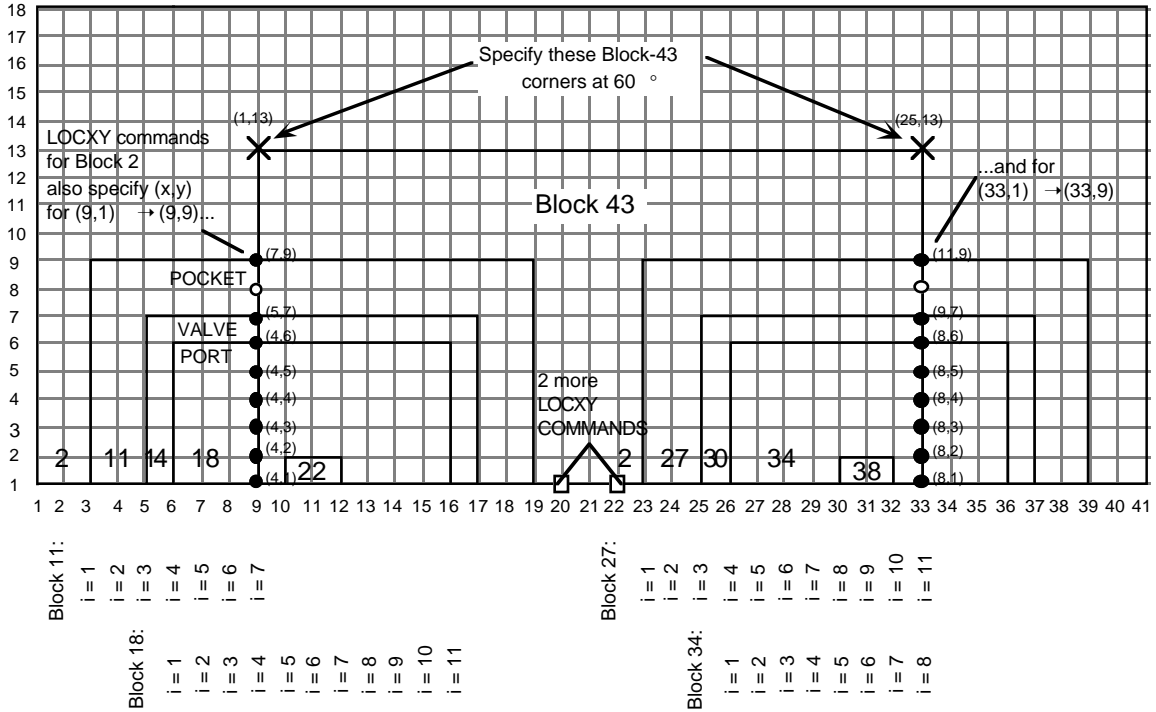


Fig. 4. Plan view of the logical mesh for the diesel engine example. The x values of the specified points for blocks 11 (1 point), 14 (1 point), 18 (6 points), 27 (1 point), 30 (1 point), and 34 (6 points) are relative to $x = 0$ at the center of the valve stem because the blocks have not been translated yet. The 20 specified points in block 2 are relative to $x = 0$ at the cylinder axis.

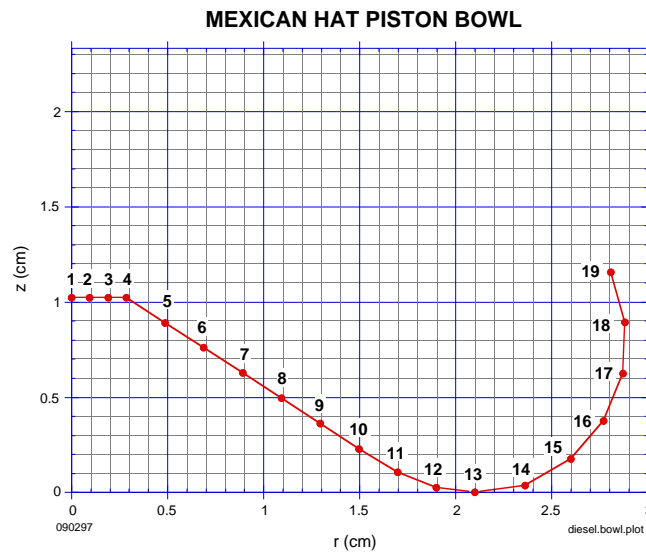


Fig. 5. The (r,z) coordinates that define the bowl outline for the diesel engine example.

X. CHANGES TO K3POST

A. Cutting Plane Rotation

The vector and contour plot command lines that specify XSLV, YSLV, and ZSLV have a fourth quantity ROTATEXY appended. ROTATEXY is an optional angle (in degrees) that specifies a temporary rotation of the mesh in (x,y) space. It is intended for XSLV or YSLV plots only, with the rotation angle and the offset XSLV or YSLV chosen so that the slice will pass at some selected vertical plane through the cylinder. This feature eliminates the restriction of a uniform x or y (with respect to the actual mesh). For example, a slice can now be drawn through the centers of two valves that have different diameters and different offsets from the true $x = 0$ or $y = 0$ planes.

B. Plot Correction at Interface with Deactivated Region

When an earlier version of K3POST is used to create plots of the mesh with hidden lines removed, plotting errors will result at the interface between the cylinder and the valve port if the valve port region is deactivated per Sec. III above. The tests in subroutine BBCON3 have been expanded in the current version of K3POST to eliminate this problem.

ACKNOWLEDGMENTS

I wish to thank Peter O'Rourke for his continuing collaboration on the wall film model and Norman Johnson for his time and patience in discussing many KIVA-3V-related topics. Norman has also maintained the KIVA information on our website and has made the reports of Refs. 5–7 available electronically. I would also like to acknowledge the many useful interactions that I have had with the worldwide KIVA user community and the inspiration that the community has been to me. These interactions have resulted in a number of improvements and corrections to KIVA-3V.

Finally, I thank Margaret Findley once again for creating a number of the figures and for her careful work in preparing my reports for publication.

REFERENCES

1. A. A. Amsden, T. D. Butler, P. J. O'Rourke, and J. D. Ramshaw, "KIVA: A Comprehensive Model for 2-D and 3-D Engine Simulations," SAE Technical Paper 850554 (1985).
2. 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).
3. 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).
4. 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).
5. 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).
6. 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).
7. A. A. Amsden, "KIVA-3V: A Block-Structured KIVA Program for Engines with Vertical or Canted Valves," Los Alamos National Laboratory report LA-13313-MS (July 1997).
8. A. B. Liu, D. Mather, and R. D. Reitz, "Modeling the Effects of Drop Drag and Breakup on Fuel Sprays," SAE Technical Paper 930072 (1993).
9. P. J. O'Rourke and A. A. Amsden, "The TAB Method for Numerical Calculation of Spray Droplet Breakup," SAE Technical Paper 872089 (1987).
10. 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).
11. P. J. O'Rourke and A. A. Amsden, "A Spray/Wall Interaction Submodel for the KIVA-3 Wall Film Model," SAE Technical Paper (in preparation).