

# **LEGIBILITY NOTICE**

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7408-ENG-38

TITLE: THE KIVA-II COMPUTER PROGRAM FOR TRANSIENT MULTIDIMENSIONAL CHEMICALLY REACTIVE FLOWS WITH SPRAYS

AUTHOR(S): Anthony A. Amsden, T. Daniel Butler, and Peter J. O'Rourke

SUBMITTED TO: To be presented at the 1987 Fuels and Lubricants Meeting, November 2-5, 1987, Toronto, Canada

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither 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. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by 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 United States Government or any agency thereof.

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 the contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

MASTER

Los Alamos Los Alamos National Laboratory Los Alamos, New Mexico 87545

*[Handwritten signature]*

**THE KIVA-II COMPUTER PROGRAM FOR TRANSIENT MULTIDIMENSIONAL  
CHEMICALLY REACTIVE FLOWS WITH SPRAYS**

**By**

**Anthony A. Amsden, T. Daniel Butler, and Peter J. O'Rourke  
Theoretical Division, Group T-3  
Los Alamos National Laboratory  
Los Alamos, New Mexico 87545**

## ABSTRACT

Since its public release in 1985, the KIVA computer program has been utilized for the time dependent analysis of chemically reacting flows with sprays in two and three space dimensions. This paper describes some of the improvements to the original version that have been made since that time. The new code called KIVA-II is planned for public release in early 1988. KIVA-II improves the earlier version in the accuracy and efficiency of the computational procedure, the accuracy of the physics submodels, and in versatility and ease of use. Numerical improvements include the use of the ICE solution procedure in place of the acoustic subcycling method and the implementation of a quasi-second-order-accurate convection scheme. Major extensions to the physical submodels include the inclusion of an optional k- $\epsilon$  turbulence model, and several additions to the spray model. We illustrate some of the new capabilities by means of example solutions.

## INTRODUCTION AND BACKGROUND

The in-cylinder dynamics of advanced internal combustion engines, such as the direct-injection stratified-charge (DISC) engine, involves a number of complex, closely coupled, physical and chemical processes. These include the transient three-dimensional dynamics of evaporating fuel sprays interacting with flowing multicomponent gases undergoing mixing, ignition, chemical reactions, and heat transfer. The KIVA code (1-3), has the ability to calculate such flows in engine cylinders with arbitrarily shaped piston geometries, including the effects of turbulence and wall heat transfer. In response to the needs of a large user community and to recent developments in the fields of numerical fluid dynamics and internal combustion engine modeling, we have implemented many improvements to KIVA since its public release in 1985. The changes are incorporated in a new version of the code, called KIVA-II, that will be released in early 1988. Indeed, KIVA-II builds on the capabilities of KIVA and is quite similar in structure. Current users of

KIVA will find the transition to KIVA-II to be straightforward.

An excerpt from Ref. (1) explains the basis under which KIVA was written: "Since KIVA was developed with applications to internal combustion engines in mind, it contains several features designed to facilitate such applications. However, the basic code structure is modular and quite general, and most of the major options (chemical reactions, sprays, etc.) can be individually activated or deactivated by setting appropriate values for the associated input switches. The code is therefore applicable to a wide variety of multidimensional problems in fluid dynamics, with or without chemical reactions or sprays." Indeed, KIVA has been used for numerous studies besides internal combustion engines, including cold flow analyses in complicated geometries, continuous spray combustors, Bunsen burner flames, nonreacting sprays, and hydrogen-oxygen flames propagating in long tubes, to name just a few. It is impractical to cite all such studies here because of the widespread distribution and use of the code in industry and universities. For internal combustion engines, besides the studies of the DISC engine that have been carried on at General Motors Research Laboratories, Princeton University, and Los Alamos, it has been used as the basis for numerical investigations of diesel engines (4-6) and to coal-fired diesels (7) as well.

From a historical perspective, KIVA-II is the latest in a series of multidimensional codes that we have produced since we began work on numerical simulations of internal combustion engines 12 years ago, under the sponsorship in what has become the Department of Energy ECUT program. All of them are multidimensional finite-difference codes that solve the transient equations of motion. The first of these was the RICE code (8). RICE was a two-dimensional Eulerian code that utilized rectangular computing zones for its mesh, eddy diffusivity to model the turbulence, Arrhenius kinetics with an arbitrary number of reactions and species to represent the chemical kinetics, and a partially implicit treatment to efficiently treat the acoustic terms for low Mach number flows. Bracco et al. at Princeton ably modified RICE and

produced the REC code (9), which included the effect of piston motion in the unresolved third dimension of the calculations. Another two-dimensional Eulerian code, APACHE (10), followed RICE. This had the capabilities of RICE and the generality of arbitrarily shaped cells. CONCHAS (11) followed APACHE and it likewise utilized arbitrarily shaped cells, but offered the feature of an arbitrary Lagrangian-Eulerian formulation that allowed the computing zones to follow the piston motion. In addition the turbulence effects were included in the calculations by use of a subgrid scale model. CONCHAS-SPRAY (12) replaced CONCHAS. As its name implies, it included a model for the spray dynamics, a statistical representation that accounted for a spectrum of droplet sizes and the effects of evaporation. The turbulence was included by means of a subgrid scale model that included a transport equation for turbulence kinetic energy, and a law-of-the-wall treatment for turbulent boundary layers. The chemistry was generalized to include both kinetic and equilibrium reactions. KIVA (1,2) then followed. In addition to retaining the capabilities of CONCHAS-SPRAY, it featured the ability to do either two- or three-dimensional problems with the same code. Furthermore, it had an expanded spray model that treated collisions and coalescence. An acoustic subcycling method was adopted to permit the efficient computation of low-Mach number flows.

The following section will give an overview of the changes since the release of KIVA. A selected set of these changes will then be described in more detail in subsequent sections.

## IMPROVEMENTS OVERVIEW

The improvements to KIVA are listed in Table I. These fall into four general categories: computational efficiency improvements, numerical accuracy improvements, new or improved physical submodels, and improvements in ease-of-use and versatility.

TABLE I

- I. Computational Efficiency Improvements
  - ICE method with conjugate residual iteration
  - Subcycled calculation of convection
  - Stochastic spray particle mover
- II. Numerical Accuracy Improvements
  - ICE method with conjugate residual iteration.
  - Optional quasi-second-order upwind convection scheme
  - Generalized mesh diffusion algorithm
  - Method for computing turbulent droplet dispersion when  $\Delta t$  exceeds turbulent correlation time
  - Convection of length scale in place of the turbulence dissipation rate  $\epsilon$
  - Simplified velocity boundary conditions

- Stochastic spray particle mover
- Spray particle splitter

### III. New or Improved Physical Submodels

- k- $\epsilon$  turbulence model
- Model for droplet aerodynamic breakup

### IV. Improvements in Ease-of-use and Versatility

- Nonflat cylinder head option
- Inflow/outflow boundaries
- Simplified velocity boundary conditions
- Alphabetized epilogue listing FORTRAN variables and their definitions
- Gravitational terms
- Eulerian and Lagrangian options
- Library of thermophysical properties of common hydrocarbons
- Initial Bessel function swirl profile
- Continuous or single-pulse spray injection

As a result of the computational efficiency improvements, KIVA-II is able to perform many calculations over a factor of two faster than KIVA. The numerical accuracy of the code has been improved in a great number of ways. Two new physical submodels have been added to the program. In addition to the subgrid scale turbulence model of KIVA, the user has the option to use a standard version of the k- $\epsilon$  turbulence model (13). An accompanying paper (14) in this conference details the new model for droplet aerodynamic breakup. This breakup model further improves one of the strongest physical submodels in the code, the spray model.

The improvements to the code's ease-of-use and versatility are numerous. Most of these changes allow the user to run a larger variety of problems using standard input. This has been in response to many users, both inside and outside the automotive engine design community, who are using the code to calculate steady and unsteady combustion problems, to explore alternative combustor designs and fuels, and to investigate fundamental equations in flame dynamics and structures. A new alphabetized epilogue allows the user to easily find the definitions and uses of FORTRAN variables within the code.

The major improvements since KIVA are the replacement of the acoustic subcycling method (15) with the ICE method (16), a subcycled convection algorithm, a generalized mesh diffusion algorithm, the addition of a droplet breakup model, the option to have inflow and outflow boundaries, a nonflat cylinder head option, and an option to use a quasi-second-order upwind scheme for convection. With the exception of the droplet breakup model (14), these improvements will now be described in more detail. In a separate section we tell of three improvements to the spray model: the stochastic spray particle mover, the spray particle splitter, and the method for computing turbulent droplet dispersion when  $\Delta t$  exceeds the turbulent correlation time.

## THE ICE METHOD WITH CONJUGATE RESIDUAL ITERATION

One of the biggest differences between KIVA and KIVA-II is in the method for calculating pressure wave propagation. Whereas KIVA uses an acoustic-subcycling (15) algorithm, KIVA-II uses the ICE method (16). There are two reasons for this change. First, our test calculations revealed some deficiencies of the acoustic subcycling method as it is implemented in KIVA. Second, a method called the conjugate residual (CR) method (17,18), has been found that solves efficiently the implicit equations of the ICE method. In this section we describe and illustrate by means of an example calculation the deficiencies of the acoustic-subcycling method and why the ICE method eliminates them. For more details concerning the implementation of the ICE method and a short introduction to the CR method, the reader is referred to Ref. 19.

The acoustic-subcycling method is used to improve the efficiency of compressible flow computer programs when they are applied to low Mach number problems. Many compressible flow codes require a computational time-step  $\Delta t$  that must satisfy the Courant sound speed restriction (20)

$$\frac{c\Delta t}{\Delta x} \leq 1, \quad (1)$$

where  $c$  is the speed of sound,  $\Delta t$  the time-step, and  $\Delta x$  the computational cell size. Thus, the acoustic mode is resolved by these codes. In low Mach number problems, this is inefficient because flow features of interest usually vary on a time scale much longer than the times for acoustic wave propagation, a phenomenon that is usually not of interest. The idea of acoustic subcycling is to difference those terms associated with the acoustic mode explicitly using a time-step  $\delta t$  that is a submultiple of the time-step  $\Delta t$  used for the remaining terms in the equations. The acoustic mode terms are calculated, or subcycled,  $\Delta t/\delta t$  times for each time-step, or cycle. The remaining terms are then incremented using  $\Delta t$ . The small time-step  $\delta t$  is still bound by the Courant condition (20). Often the large time-step  $\Delta t$  is determined by stability restrictions associated with explicit differencing of the convective terms:

$$\frac{u\Delta t}{\Delta x} \leq 1, \quad (2)$$

where  $u$  is the fluid velocity. Thus we have

$$\frac{\Delta t}{\delta t} = \frac{c}{u} = \frac{1}{M}, \quad (3)$$

where  $M$  is the Mach number. Equation (3) shows that when  $M \ll 1$  the acoustic subcycling method can be inefficient, unless it is combined with a method that scales up the Mach number (21), as is done in KIVA.

We describe two possible ways of implementing acoustic subcycling. Consider the

pressure and velocity equations of a single-component viscous fluid:

$$\frac{Dp}{Dt} + \rho c^2 \nabla \cdot u = \frac{\partial p}{\partial s} \bigg|_p \frac{Ds}{Dt}, \quad (4)$$

and

$$\frac{Du}{Dt} + 1/\rho \nabla p = \nabla \cdot \sigma + g. \quad (5)$$

In these equations,  $D/Dt$  is the substantial derivative,  $\rho$  the density,  $s$  the entropy,  $p$  the pressure,  $\sigma$  the viscous stress tensor, and  $g$  the acceleration due to gravity. Typically in low Mach number flows, the terms on the right-hand sides of Eqs. (4) and (5) are the slowly varying (SV) terms, while those on the left-hand sides are the acoustic mode (AM) terms. The SV terms are evaluated once each large time-step  $\Delta t$ . In one approach, which we label Method I, the computed values of the SV terms are multiplied by  $\Delta t$  and added to the pressures and velocities on the first subcycle. On subsequent subcycles of the current cycle, only the AM terms are calculated and affect the pressures and velocities. In an alternative approach, Method II, the values of the SV terms are calculated and stored. Each subcycle of the current cycle, these are multiplied by the small time-step  $\delta t$  and added to the pressures and velocities. Method II requires more storage but is more accurate.

Method I was used in the original version of KIVA, but one of its deficiencies soon became apparent. When an SV term nearly balances an AM term, this cannot be calculated by Method I if  $\Delta t > \delta t$ . For example, if the pressure is nearly spatially uniform and not varying in time, Eq. (4) implies that the divergence of the velocity field is determined by

$$\nabla \cdot u = \frac{1}{\rho c^2} \frac{\partial p}{\partial s} \bigg|_p \frac{Ds}{Dt}. \quad (6)$$

Thus, we have a balance between an SV term and an AM term. This balance cannot be calculated by Method I (unless  $\Delta t \approx \delta t$ ) because the effects of the SV terms are only felt on one subcycle of each computational cycle. For the remaining subcycles, the two AM terms are legislated to balance. This defect was corrected for Eq. (4) in an improved version of KIVA (2) by using Method II, but Method I is still used for Eq. (5).

One solution to this problem is to use Method II for differencing Eq. (5). This would require only minor modifications of KIVA, but a disadvantage of this solution is that three additional arrays would be needed -- one for each component of the SV term of Eq. (5).

Another approach is to use the ICE method, in which, effectively, the terms  $\rho c^2 \nabla \cdot u$  and  $1/\rho \nabla p$  in Eqs. (4) and (5) are differenced implicitly using the large time-step  $\Delta t$ . The implicit differencing is needed to stabilize the finite-difference approximation when the Courant condition is violated, and since all terms participate in the equation balance on each time-step, we can have a

balance between AM and SV terms, as is required, for example, when  $\nabla \cdot u$  is given by Eq. (6) in constant pressure flows.

A disadvantage of using implicit differencing equations has been that iterative procedures for their solution are often slow to converge. Recently, however, relatively fast and efficient solution procedures have been developed, and one of these, the conjugate residual (CR) method, is particularly attractive because of its low storage requirements and because the computer logic associated with it is vectorizable for use on computers of the Cray family.

To illustrate the deficiencies of Method I and how these are mitigated by the ICE method, we present results of calculations in which a slowly varying term, the spray source term in the momentum equation, Eq. (5), is balanced by the acoustic mode terms. In this idealized problem, a slug of liquid droplets moves axially in a tube of gas. The drops are uniformly distributed within the slug, which occupies one eighth of the length of the tube. The drops displace a negligible volume of the gas in the slug. Initially the gas is at rest, and the drop velocity is 4000 cm/s, which corresponds to a Mach number of 0.1 based on the gas sound speed. Periodic boundary conditions are imposed at the top and bottom of the tube, and hence, the gas accelerates as it acquires the momentum lost by the droplets on account of drag forces.

In the calculations, the tube was resolved with a mesh of 16 uniform cells in the axial direction and the Courant number  $c\Delta t/\Delta z$  was 10.0. Figure 1 gives velocity vector plots after the first cycle from the solutions obtained using the subcycling and iterating codes. The initial position of the slug of droplets is indicated in the figure. The plots from the subcycling code have large velocity variations. In contrast, the velocities obtained by the iterating code are nearly uniform in space, as would be the case if the flow were incompressible. Both codes conserve linear momentum, but both solutions have numerical errors. The plot of the subcycling code solution shows that the gas velocities at the bottom of the tube, where the droplets are located, are nearly zero. The momentum exchanged between the droplets and gas on the first cycle has propagated to the top of the tube. This unphysical behavior is caused by the method (described above) used to calculate the spray momentum source term, which is an SV term in the gas velocity equation. Since the spray is continuously giving up its momentum to the gas, the gas near the drops should be moving. On the other hand, the iterating code result is inaccurate because at these early times compressibility effects should be important, and the velocity field should be nonuniform. The nearly uniform velocity field of the iterating code solution is the result of implicit numerical errors that damp acoustic pressure waves (22).

In this example the errors of the subcycling version are more serious than those of the iterating code because the flow is nearly incompressible, and the subcycling code has introduced unphysical and slowly damped pressure waves.

## OPTIONAL QUASI-SECOND-ORDER UPWIND CONVECTION SCHEME

A pair of input coefficients in KIVA permitted the use of the interpolated donor cell differencing, which is second-order accurate in space and time on a uniform mesh, donor cell or upwind differencing, which is first-order accurate, or any amount of upwind differencing between these two extremes. This choice was made a priori for the entire mesh for every time-step during a particular computer run. Interpolated donor cell differencing introduces undesirable dispersion errors in regions of sharp gradients in the mesh. Donor cell differencing preserves monotonicity but is more diffusive, thereby smearing the gradients. A difference scheme for convection has the monotonicity property when the following is true: if convected quantity  $p$  is such that  $p_i^n$  lies between  $p_{i-1}^n$  and  $p_{i+1}^n$  in value and  $x_i^{n+1}$  lies between  $x_{i-1}^n$  and  $x_{i+1}^n$ , then  $p_i^{n+1}$  lies between  $p_{i-1}^n$  and  $p_{i+1}^n$ .

In KIVA-II, we have added another option that combines the best features of interpolated and donor cell differencing. In particular, this option provides a means of effectively calculating the two controlling coefficients in a space- time- and direction-dependent fashion so as to maximize the amount of interpolated donor cell differencing and minimize the donor cell while retaining the monotonicity property of the latter. When wavelengths of the computed solution are long compared to the mesh spacing, the scheme reduces to interpolated donor cell. When the computed solution has wavelengths comparable to the mesh spacing, the scheme reduces to donor cell, which strongly damps the shorter wavelength components and is monotone.

The method we use is a modification of the ideas proposed by Van Leer (23). Van Leer gives a necessary but not sufficient condition for monotonicity of a certain class of difference schemes for the convection equation. His condition gives a prescription for limiting the gradient of a convected quantity. By adopting a more aggressive gradient-limiter than he proposes, we have devised a scheme that has given us monotone results in all our test calculations.

To illustrate the properties of the old and new convection schemes, we calculate the convection of a square-shaped region of species 2 through a mesh of square cells otherwise occupied by species 1. The geometry and computational results of this extreme example are shown in Fig. 2. Species 2 is initially located in the lower left side of the computational region. Inside the square-shaped region, whose sides have length  $56x$ , the mass fraction of species 2 is 1.0 and outside it is 0.0. The region is convected at a  $45^\circ$  angle to the upper right by a uniform, steady velocity field. The problem is run to a time  $56x/u$ , where  $u$  is one component of the velocity. At this time, the exact solution is the translation of the square of material exactly the length of its side in each coordinate direction.

Contours of constant mass fraction from the interpolated donor cell solution are shown at the upper right. In the wake of the square they show

the dispersive ripples that are characteristic of the interpolated donor cell scheme. The maximum and minimum calculated mass fractions have the unphysical values 1.41 and -0.35. The donor cell solution is shown at the lower left. Because the donor cell scheme is monotone, there are no dispersive ripples, and the maximum and minimum values of 0.65 and 0.0 lie between the original maximum and minimum mass fractions. There has been considerable numerical diffusion of species 2, however, so that the original square shape is now nearly circular. Finally, the quasi-second-order upwind solution is shown at the lower right. The maximum and minimum mass fraction are 0.87 and 0.0, evidence of the monotonicity of the scheme. In addition, there has been considerably less numerical diffusion of material than in the donor cell calculation. To quantify the error in each solution we calculate

$$E = \sqrt{\frac{1}{25} \sum_{i,j} (1 - Y_{i,j})^2},$$

where  $Y_{i,j}$  is the mass fraction of material in cell (i,j) and the sum is over the twenty-five cells for which the exact solution is  $Y = 1.0$ . The values of  $E$  for the donor cell, interpolated donor cell, and quasi-second order upwind schemes were 0.532, 0.411, and 0.358, respectively.

One pays a price for the accuracy of the quasi-second-order upwind scheme; in test calculations, use of the method required fifteen to fifty percent more computer time, depending on the number of convective subcycles. For this reason we have included it as an option to the user. The decision to use quasi-second-order upwind differencing will depend on the cell Reynolds numbers in one's problem (20), and the purpose of the calculations. When all cell Reynolds numbers  $Res_x = \rho u \Delta x / \mu$  are less than two, then one should use the old convective differencing of KIVA. This is because the physical viscosity is so large that the numerical diffusion of even pure donor cell differencing will be dominated by physical diffusion. It would be a waste of computer time to use the new scheme. When  $Res_x \approx 2$ , then a possible approach is to use the old scheme for faster calculations whose purpose is to obtain qualitative flow features and trends and to use the new scheme when one wants to obtain the most accurate results possible for a given mesh spacing.

#### THE k-ε TURBULENCE MODEL

A standard k-ε turbulence model has been installed in KIVA-II. The option to use the subgrid scale (SGS) turbulence model has also been retained. The equations of the k-ε model are

$$\begin{aligned} \frac{\partial k}{\partial t} + \nabla \cdot (\rho u k) + \frac{2}{3} \rho k \nabla \cdot u \\ = \sigma \nabla^2 u + \nabla \cdot (\mu_T / \sigma_k \nabla k) - \rho \epsilon + W_k \end{aligned} \quad (7)$$

and

$$\begin{aligned} \frac{\partial \rho \epsilon}{\partial t} + \nabla \cdot (\rho u \epsilon) + \left( \frac{2}{3} C_{\epsilon_1} - C_{\epsilon_2} \right) \rho \epsilon \nabla \cdot u \\ = \frac{c}{k} C_{\epsilon_3} (\sigma \nabla^2 u) + \nabla \cdot (\mu_T / \sigma_\epsilon \nabla \epsilon) \\ - C_{\epsilon_2} \rho \epsilon^2 / k + C_\mu \frac{k}{k} W_\epsilon \end{aligned} \quad (8)$$

With the exception of the dissipation term, the turbulent kinetic energy equation is the same as the SGS turbulent kinetic energy equation. In particular, the forms of the momentum stress tensor  $\sigma$  and spray source term  $W_\epsilon$  are the same. An additional transport equation, Eq. (8), is now solved for the dissipation rate  $\epsilon$ . The turbulent viscosity of the k-ε model is given by

$$\mu_T = C_\mu \rho \frac{k^2}{\epsilon} \quad (9)$$

The quantities  $C_{\epsilon_1}$ ,  $C_{\epsilon_2}$ ,  $C_{\epsilon_3}$ ,  $C_\mu$ , and  $C_\mu$  are constants whose values are determined from experiments and some theoretical considerations. Standard values of these constants are often used in engine calculations.

The k-ε equations are solved together with the following standard boundary conditions:

$$\nabla k \cdot n_{\text{wall}} = 0 \quad (10)$$

and

$$\epsilon = 2.5 C_\mu^{3/4} \frac{k^{3/2}}{y} \quad (11)$$

The vector  $n_{\text{wall}}$  is the unit normal to the wall. The boundary condition on  $\epsilon$  is applied at the centers of computational cells adjacent to walls and whose centers are distance  $y$  from the wall.

The production, diffusion, and decay terms are differenced in Phase A of KIVA-II in an analogous fashion to the differencing of these terms in the SGS turbulence model. The compression ( $\nabla \cdot u$ ) terms are differenced in Phase B, and the convection terms are calculated in Phase C. In numerical comparisons with analytic solutions, we found it is more accurate to convect length scale  $L = k^{3/2}/\epsilon$  rather than  $\epsilon$ . This is because there are normally large gradients in  $\epsilon$  since  $\epsilon \sim 1/y$  near walls. In contrast,  $L \sim y$  near walls and  $L$  is thus a smoother variable than  $\epsilon$ .

#### THE SPRAY MODEL IMPROVEMENTS

In this section we describe three changes to the spray model that improve its numerical accuracy: the spray particle splitter, the stochastic spray particle mover, and an extension of the method for calculating turbulent droplet dispersion. When the mass associated with a computational spray particle exceeds a reference mass, which is twice the mass of an injected

particle, the splitter routine replaces that particle by two particles composed of drops with the identical properties, but with half the number of droplets of the original. The two new particles then follow different trajectories because of the method for calculating turbulent dispersion of droplets. This splitter is effectively a means of re-sampling from the droplet distribution function at downstream locations. It improves accuracy in calculations where there are frequent droplet coalescences, and because the method for calculating coalescences deletes computational particles, there may be large statistical sampling errors.

The stochastic particle mover allows one to use time-steps that are larger than the time-step of the Courant condition based on the droplet velocity. This condition is that a droplet can travel no more than one computational cell in one time-step and was necessary in KIVA so that droplets could exchange mass, momentum, and energy with all cells along their trajectory. With the stochastic particle mover, the computational particle position is still advanced by the formula

$$x_p^{n+1} = x_p^n + u_p^n \delta t,$$

but unlike the old algorithm, where the exchanges of mass, momentum, and energy were with the cell containing position  $x_p^{n+1}$ , the stochastic particle mover has these exchanges taking place in the cell containing the position  $x_p$ , where

$$\bar{x}_p = x_p^n + R(u_p^n \delta t)$$

and  $R$  is a random number equidistributed in the interval (0,1). Thus, on the average a particle exchanges mass, momentum, and energy with all cells along its trajectory, even though it may travel across many cells in one time-step.

The stochastic particle mover is useful in steady-state spray calculations or where there are quasi-steady regions in time-dependent calculations for which the droplet velocities greatly exceed the gas velocities. This is often true near a spray injector where droplets have large injection velocities and are still not in velocity-equilibrium with the gas. Farther downstream of the injector, where the droplet velocities nearly equal those of the gas, the condition  $u_p \delta t / \delta x < 1$  will be satisfied because this condition is enforced for the gas velocity.

The extension of the method for calculating turbulent droplet dispersion is intended to allow for situations where the computational time-step  $\delta t$  exceeds the droplet turbulent correlation time  $t_d$ . No allowance was made for these situations in the KIVA code, and as a result when  $\delta t > t_d$  large unphysical turbulent dispersion of droplets was calculated. The problem arises because a droplet "sees" more than one turbulent gas velocity  $u'$ , and as a result, receives more than one random change to its velocity in one time-step. In the new method, when  $\delta t > t_d$ , the particle position and velocity are updated using

$$\frac{u_p^{n+1} - u_p^n}{\delta t} = A_p^n (u_p^{n+1} - u_p^n) + g + \frac{w'}{\delta t}$$

and

$$\frac{x_p^{n+1} - x_p^n}{\delta t} = u_p^n + \frac{x'}{\delta t},$$

where  $u_p^{n+1}$  is the advanced-time gas velocity at the position of the particle,  $A_p^n$  is the particle drag function,  $g$  is the acceleration due to gravity, and  $x'$  and  $w'$  are turbulent position and velocity changes that are randomly chosen from certain probability distributions for these changes. These probability distributions, which are derived in Ref. 24, depend on the time-step  $\delta t$ , the turbulence correlation time  $t_d$ , and the droplet drag time  $1/A_p^n$ . Because of the large range of these times that we have seen in our spray calculations, no restrictive assumptions are made concerning the relative magnitudes of these times in KIVA-II.

## SUBCYCLED CONVECTION

In many applications, the time-step  $\Delta t$  in the original KIVA was limited by convection. This was particularly the case in high-swirl engine geometries, where for a considerable portion of the calculation the convective limit was far more restrictive than the diffusive limit.

To alleviate this situation, KIVA-II subcycles the pair of subroutines that calculate the fluxes of mass, momentum, and energy. This is accomplished by removing the convective limit  $\Delta t_{con}$  as a choice in the calculation of the new  $\Delta t$  each cycle. The number of fluxing subcycles NFLUXS is then calculated as the ratio  $\Delta t / \Delta t_{con}$  rounded up to the next integer, constrained to a minimum value of 1 and some maximum value in the range 5 to 10. A limit of 5 is recommended for transient calculations, such as internal combustion engines, while a limit of 10 is more efficient for steady-state calculations, where the flow field changes gradually over time. With NFLUXS determined, the subcycle time-step  $\Delta t_s$  is given by  $\Delta t_s = \Delta t / \text{NFLUXS}$ .

As an example of the efficiency that can be gained through subcycled convection, consider the KIVA calculations of the UPS-292-SC engine (25). Until spray and combustion commence near TDC, swirl velocities over 8000 cm/s controlled the time-step. The portion of the calculation from IVC at 116° BTDC to 39° BTDC originally required 2404 computational cycles in about 1h 24 min CPU time on a Cray X-MP. When repeated with the subcycled convection of KIVA-II, those dropped to 528 cycles in 46 min, a reduction to 22% as many computational cycles in 55% as much CPU time, running at the limit of 5 subcycles virtually throughout. Despite such a great difference in cycle and  $\Delta t$  history, the results do not differ significantly between the two runs.

## GENERALIZED DIFFUSION

The diffusion subroutine in the previous version of KIVA was designed to take advantage of the area projection arrays that are also used for calculating viscous stresses and pressure accelerations. This was efficient from a calculational standpoint, but would only give correct results in a truly orthogonal grid. While it is desirable to construct grids with as much orthogonality as practical, this is clearly impossible in modeling curvilinear geometries found in many applications, and for which the ALE concept is ideally suited.

Accurate calculation of diffusion requires the use of actual distances between vertices, rather than distances projected in the x, y, and z directions. To accomplish this, KIVA-II contains a generalized 19-point diffusion scheme that employs a set of geometric factors based on true distances.

To illustrate this difference, consider the nonorthogonal mesh shown in Fig. 3. If we impose an adiabatic wall condition and a temperature discontinuity, say  $T = 400$  K in the left half and  $T = 300$  K in the right half, the previous version of KIVA gives the temperature profile at some later time as shown by the x's in Fig. 4. In contrast, KIVA-II gives the profile shown by the circles, in close agreement with an approximate analytic solution for the heat diffusion, shown by the solid line. The agreement is excellent except at the left boundary, where the effect of different wall boundary conditions for the analytic solution is manifested. While the code is constrained to make  $dT/dx = 0$ , the analytic solution assumes that  $T = 400$  K is infinitely far away, and is asymptotically heading toward that value.

The generalized scheme in KIVA-II was further tested in a variety of multidimensional curvilinear geometries, with and without combustion. As indicated by the above heat diffusion test, the diffusion of mass, enthalpy and turbulence are likewise more accurately calculated for nonorthogonal grids in the new code.

## INFLOW AND OUTFLOW BOUNDARIES

Originally, KIVA was developed with applications to internal combustion engines specifically in mind, and we restricted the code to confined flows with a moving bottom boundary to simulate piston motion.

As the code became more widely used by ourselves and others, many diverse engine and nonengine applications emerged. Often, these applications have geometries that require the use of inflow and outflow boundaries. We considered code options that would allow any portion of any boundary to be rigid, inflow, or outflow. While this sounds attractive, the added complexity is not practical for a release code version. Accordingly, we offer some limited options that will guide users in adapting the code to their own specific needs.

In addition to retaining the basic confined-flow mesh from before, KIVA-II offers an optional inflow bottom boundary, and optional outflow boundaries along the top and/or right side of the

logical mesh. For an inflow bottom boundary, the user specifies the inflow velocity and densities of the various species. For an outflow boundary, or an appropriate flag requires specification.

An added feature for outflow boundaries that the user may specify is the point at which the ambient pressure is applied. This may be on the boundary itself, or some distance beyond. In the latter case, the boundary more readily absorbs acoustic waves, thus reducing any tendency for such waves to be reflected back in. This feature has proven useful in establishing steady-state flows, by reducing both the problem time and the average number of pressure iterations required per cycle.

## DOMED CYLINDER HEAD

Originally, KIVA applications focused on DISC and diesel engine designs, whose geometries are characterized by having a distinct bowl in the piston face, and a flat cylinder head. A mesh generator was included in the code that concentrated on the piston geometry, and simplified the task of defining its shape. Through the use of tabular input data, the user defines grid points along the piston silhouette, starting at the bowl axis and ending at the cylinder wall. The silhouette may contain a bowl or be flat. In a 3-mesh, the bowl shape is then rotated axially, assuming that the bowl is circular about its own axis. If desired, the bowl could be subsequently offset from the cylinder axis in a 3-D mesh.

While this generator has proven useful, it assumes a flat head. However, a number of designs for combustion chambers are distinguished by cylinder heads that are arched, or perhaps distinctly domed. To meet this need, the generator in KIVA-II has been extended to include an optional second table for nonflat heads.

Although we presently still constrain the piston bowl to be rotationally symmetric, we provide for head domes that may depart from rotational symmetry. An example is shown in Fig. 5, in which the dome is not only offset to one side but, when viewed from above, is seen to be partly circular and partly ellipsoidal. Again, optional tabular input data provide the necessary information for the generator to make this shape adjustment. At the bottom of the mesh in this example, the piston silhouette data are used in the previous manner. Here, we define a slightly domed piston top, which coincidentally matches the arch in the cylinder head surrounding the dome.

## NUMERICAL EXAMPLE OF A KIVA-II CALCULATION

In an earlier SAE paper (25), we presented an analysis of three-dimensional KIVA-II calculations of the spray, mixing, and combustion in the UPS-292 stratified charge engine for three different operating conditions. We reported that pressure histories compared well with experimental measurements at the highest load condition, and that computed wall heat losses were approximately two-thirds of the measured values.

for all test conditions. These earlier calculations, using a mesh of 10,000 computational cells, required approximately three hours of Cray X-MP computer time to run from IVC at 118° BTDC, out to 92° ATDC.

Subsequent to the publishing of that paper, we have rerun Case 3, using KIVA-II. This was the lowest-load case, and the one that exhibited the greatest discrepancy with experimental pressure data. This case had the latest injection of the three cases, and as a consequence the spray, mixing, and combustion events occurred closer to TDC than in the higher-load cases. We hypothesize that this case would therefore be the most sensitive of the three to any error in compression ratio. Accordingly, we have made slight adjustments to our approximation of the piston geometry, bringing our compression ratio down from 13.44 to 13.04, matching the experimental value of 13.0. In addition, we have increased our wall heat loss coefficient by 50%, as suggested by the earlier runs.

Figure 6 gives the computed and measured pressure histories from Case 3, and includes the curve from the new KIVA-II calculation. In addition to agreeing favorably with the experimental curve, the new result exhibits a pronounced drop in the rate of pressure rise during fuel injection, then a strong increase in the rate of rise, still during ignition. This effect is also apparent in the experimental curve, although less markedly. A possible explanation for this behavior is that during the injection period, fuel evaporation is cooling the gas, the rate of chemical heat release is small, and the rate of pressure rise decreases because the upward motion of the piston is slowing. Then after injection there is a large increase in the rate of chemical heat release.

One of the principal results from our earlier calculations remains unchanged, however, and that is the presence of hydrocarbons on the cylinder walls at 92° ATDC, illustrated by the temperature and fuel vapor mass fraction contour plots of Fig. 7. In all our calculations of the UPS engine, the combustion occurs in two phases: a short premixed burn during which a small fraction of the fuel is consumed, followed by a longer diffusion flame phase. The flames are not stationary, but are convected around the combustion chamber by the swirl, and into the squish region as the piston withdraws. If combustion is incomplete, unburned fuel will be left on the cylinder walls.

The rerun of Case 3 with KIVA-II required 1.77 hours on a Cray X-MP, in comparison with 3.32 hours for the earlier KIVA run. The reduction to nearly half the time is due mainly to subcycling the convective flux subroutines, which dramatically enhances efficiency in high-swirl applications such as the UPS engine. The rerun was made with partial donor-cell differencing, as in the original run. Tests indicate that the quasi-second-order upwind differencing option produces virtually the same results in this application, but is significantly slower. The conjugate residual pressure iteration in KIVA-II is about 10% slower than the acoustic subcycling of the earlier KIVA runs for the UPS application, at least at early times. This loss is far outweighed, however, by the more efficient convective fluxing in KIVA-II.

## ACKNOWLEDGMENTS

We are grateful to Edward G. Groff of General Motors Research Laboratories for supplying experimental data on the UPS-292 stratified charge engine. This work was supported by the United States Department of Energy, Office of Energy Research, Energy Conversion and Utilization Technologies Program.

## REFERENCES

1. 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).
2. 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 (1985).
3. 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 paper 850554 (October 1985).
4. M. A. Theobald, "A Numerical Simulation of Diesel Autoignition," Ph.D. Thesis, Massachusetts Institute of Technology (1986).
5. A. J. Brown, "A Stochastic Mixing Model for Predicting Emissions in a Direct Injection Diesel Engine," Ph.D. Thesis, Massachusetts Institute of Technology (1986).
6. W. K. Cheng and R. A. Gentry, "Effects of Charge Non-Uniformity on Diesel Heat Release Analysis," SAE Technical Paper 861668 (1986).
7. R. A. Gentry, B. J. Daly, and A. A. Amsden, "KIVA-COAL: A Modified Version of the KIVA Program for Calculating the Combustion Dynamics of a Coal-Water Slurry in a Diesel Engine Cylinder," Los Alamos National Laboratory report LA-11045-MS (August 1987).
8. W. C. Rivard, O. A. Farmer, and T. D. Butler, "LUCE: A Computer Program for Multicomponent Chemically Reactive Flows at All Speeds," Los Alamos Scientific Laboratory report LA-5812 (March 1975).

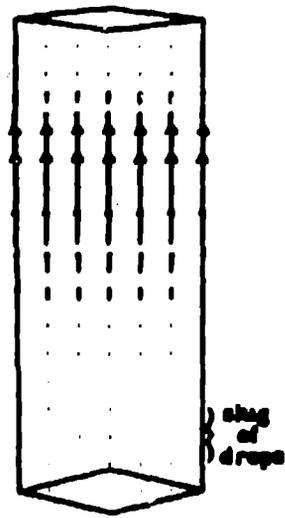
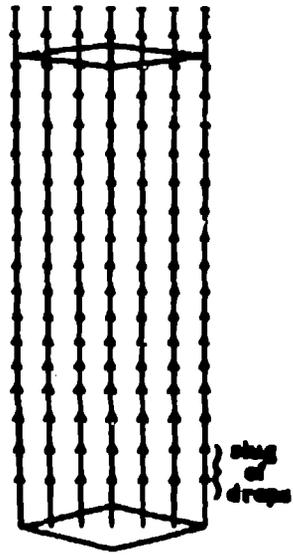
9. H. C. Gupta and S. A. Syed, "REC-P3 (Reciprocating Engine Combustion, Planar Geometry, Third Version): A Computer Program for Combustion in Reciprocating Engines," MAE Report No. 1431, Mechanical and Aerospace Engineering Department, Princeton University (1979).
10. J. D. Ramshaw and J. K. Dukowicz, "APACHE: A Generalized-Mesh Eulerian Computer Code for Multicomponent Chemically Reactive Fluid Flow," Los Alamos National Laboratory report LA-7427 (January 1979).
11. T. D. Butler, L. D. Cloutman, J. K. Dukowicz, and J. D. Ramshaw, "CONCHAS: An Arbitrary Lagrangian-Eulerian Computer Code for Multicomponent Chemically Reactive Fluid Flow at All Speeds," Los Alamos Scientific Laboratory report LA-8129-MS (November 1979).
12. L. D. Cloutman, J. K. Dukowicz, J. D. Ramshaw, and A. A. Amsden, "CONCHAS-SPRAY: A Computer Code for Reactive Flows with Fuel Sprays," Los Alamos National Laboratory report LA-9294-MS (May 1982).
13. B. E. Launder and D. B. Spalding, Mathematical Models of Turbulence Academic Press, New York (1972).
14. P. J. O'Rourke and A. A. Amsden, "The TAB Method for Numerical Calculation of Spray Droplet Breakup," submitted to 1987 International Fuels and Lubricants Meeting and Exposition, Toronto.
15. L. C. Hasselman, "TDC - A Computer Code for Calculating Chemically Reacting Hydrodynamic Flows in Two Dimensions," Lawrence Livermore Laboratory report UCRL-62931 (May 1980).
16. F. H. Harlow and A. A. Amsden, *J. Comput. Phys.* **3**, 80 (1968).
17. R. Chandra, "Conjugate Gradient Methods for Partial Differential Equations," Ph.D. Thesis, Yale University, 1978.
18. H. C. Elman, "Iterative Methods for Large, Sparse, Non-symmetric Systems of Linear Equations," Ph.D. Thesis, Yale University, 1982. Also available as Technical Report #229 from Yale University.
19. P. J. O'Rourke and A. A. Amsden, "Implementation of a Conjugate Residual Iteration in the KIVA Computer Program," Los Alamos National Laboratory report LA-10849-MS (October 1986).
20. P. J. Roache, Computational Fluid Dynamics, Hermosa Publishers Albuquerque, New Mexico (1982).
21. J. D. Ramshaw, P. J. O'Rourke, and L. R. Stein, *J. Comput. Phys.* **59**, 361 (1985).
22. P. J. O'Rourke, "The Acoustic Mode in Numerical Calculations of Subsonic Combustion," in Combustion and Nonlinear Phenomena, eds. P. Clavin, B. Larrouturou and P. Pelce, Les Editions de Physique France (1985).
23. B. van Leer, *J. Comput. Phys.* **32**, 101 (1979).
24. P. J. O'Rourke, "Statistical Properties and Numerical Implementation of a Model for Droplet Dispersion in a Turbulent Gas," submitted to *J. Comput. Phys.*
25. P. J. O'Rourke and A. A. Amsden, "Three-Dimensional Numerical Simulations of the UPS-292-SC Engine," presented at the SAE International Congress and Exposition, Detroit, Michigan, February 23-27, 1987. SAE Technical Paper 870597.

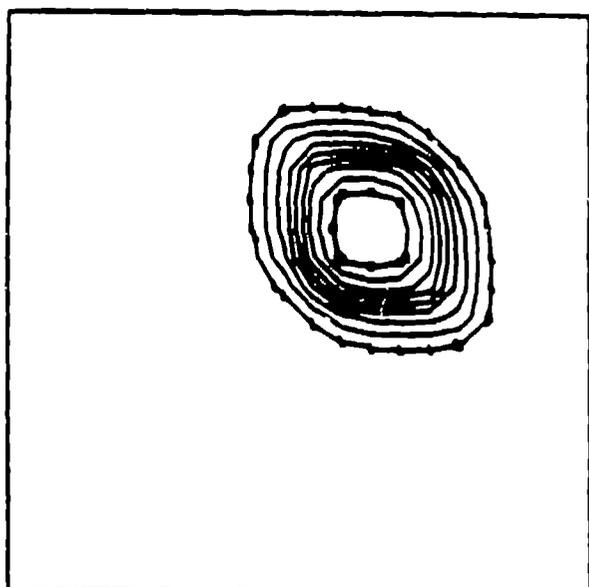
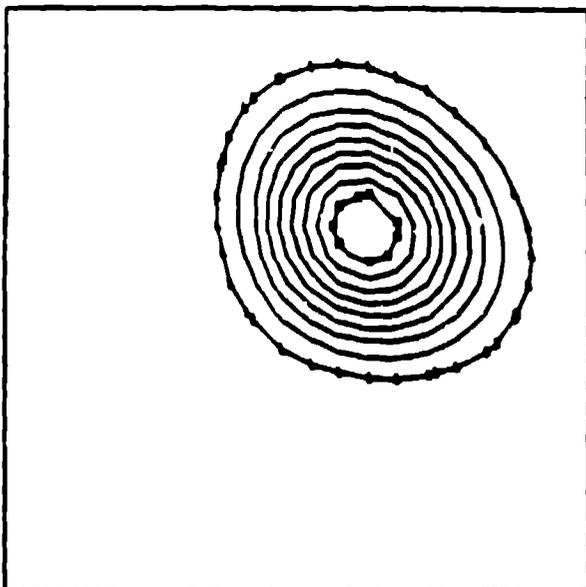
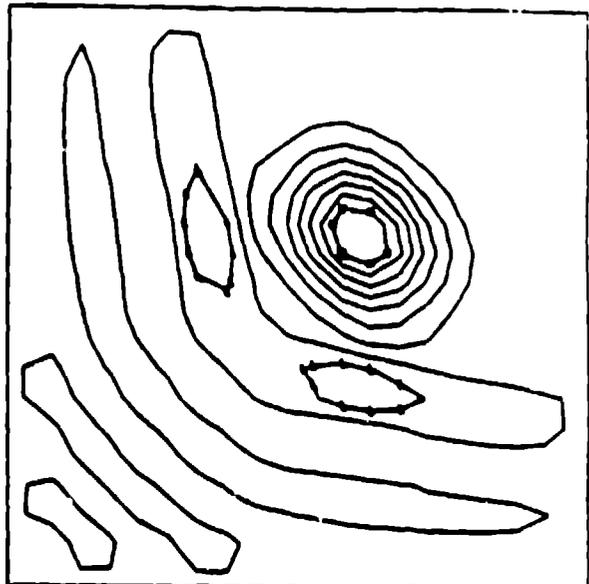
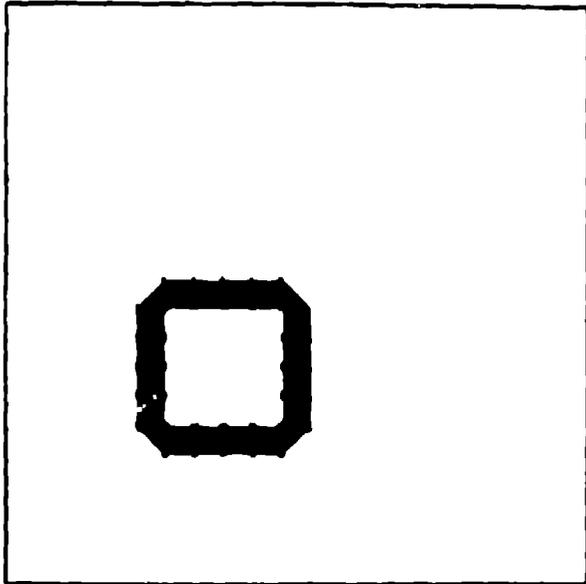
#### FIGURES

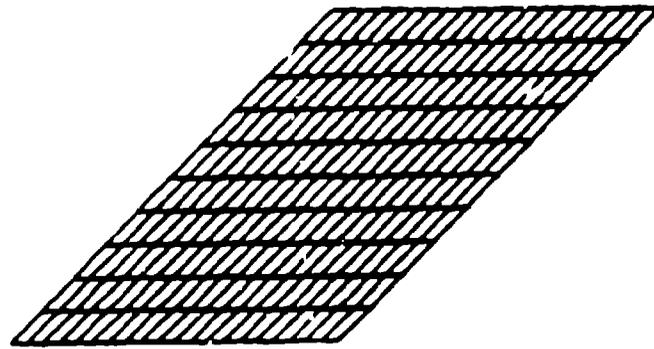
- Fig. 1. Velocity vector plots from cycle 1 of the iterating- and subcycling-code solutions to the droplet slug problem.
- Fig. 2. In the convection test, a block of one fluid is convected to the upper right through another fluid by a uniform, steady velocity field. The results are shown at a later time for three different convection schemes, described in the text.
- Fig. 3. A nonorthogonal grid tilted at 45°, as used for the heat diffusion test described in the text.
- Fig. 4. The generalized diffusion scheme in KIVA-II compares favorably with the analytic solution for the heat diffusion test, performing well in a nonorthogonal mesh.
- Fig. 5. A KIVA-II mesh with an arched and domed head and an arched piston face. The overhead view illustrates the semi-ellipsoidal shape of the dome.
- Fig. 6. Pressure histories from the UPS-292 stratified charge engine, test condition Case 3. The baseline case was run with the earlier version of KIVA, the corrected case was run with KIVA-II.
- Fig. 7. Temperature and fuel vapor mass fraction contour plots at the end of the KIVA-II run of UPS engine Case 3.

**ITERATIVE CODE**

**SUBCYCLING CODE**







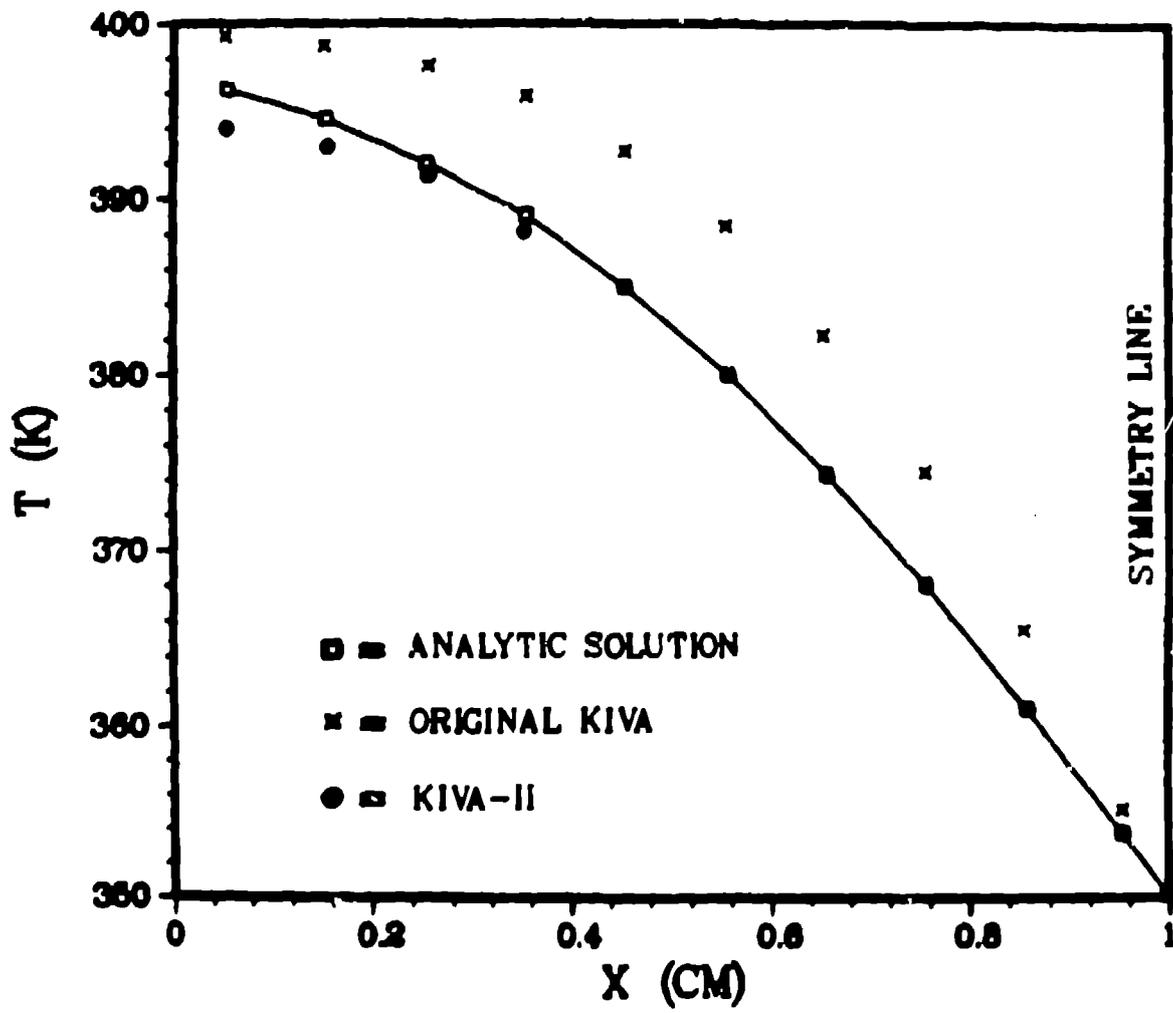


FIG. 4

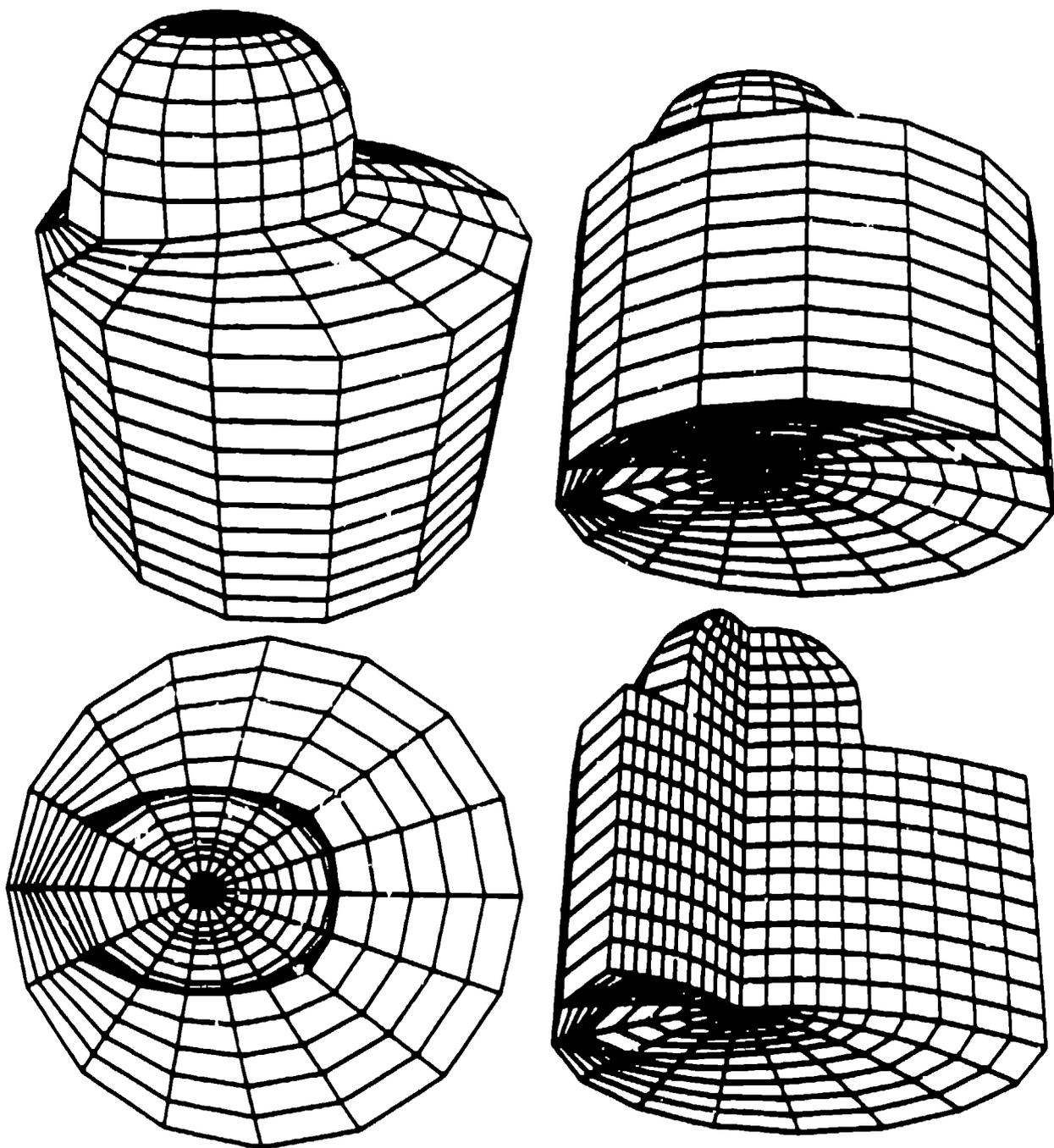
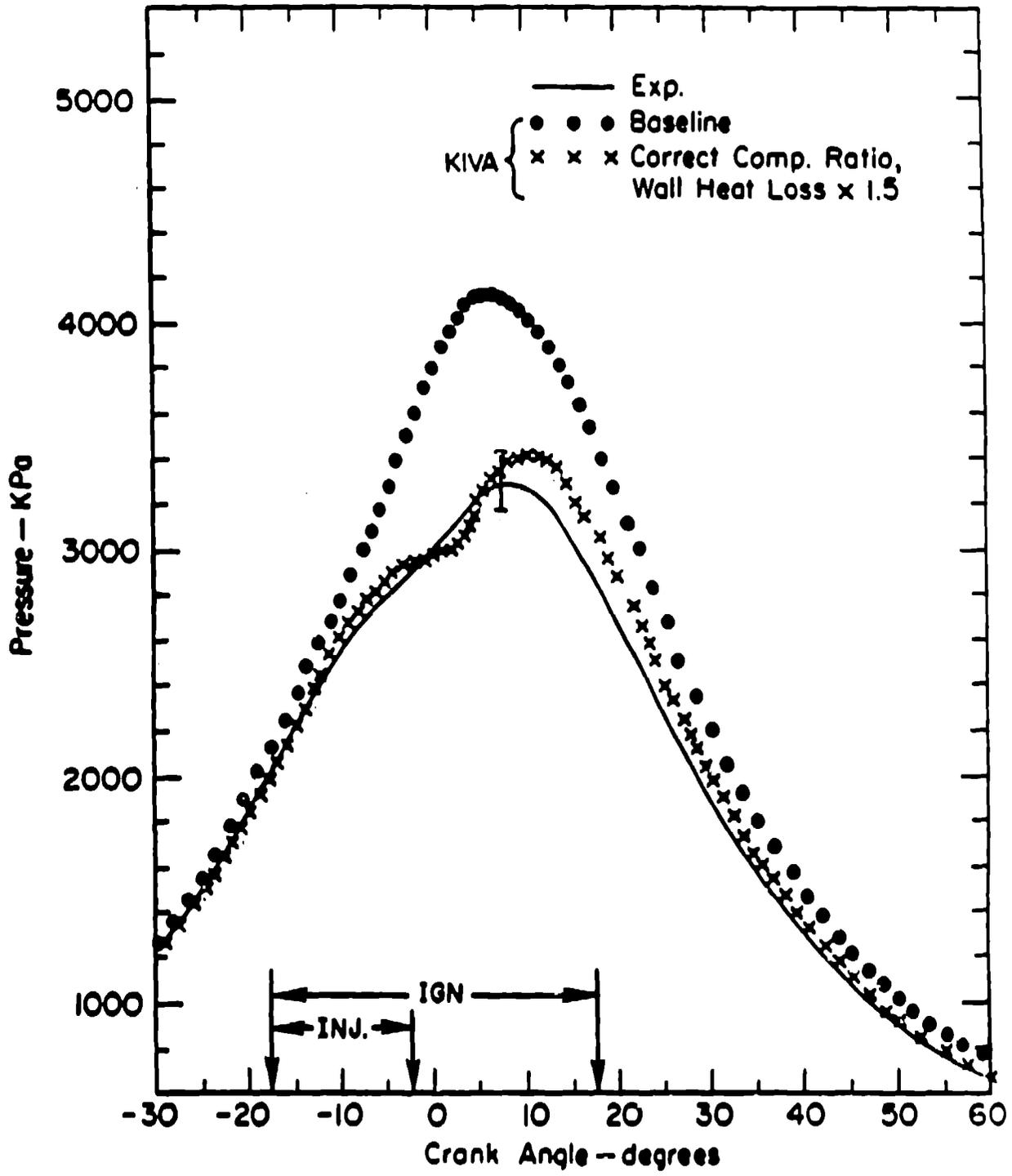


FIG. 5

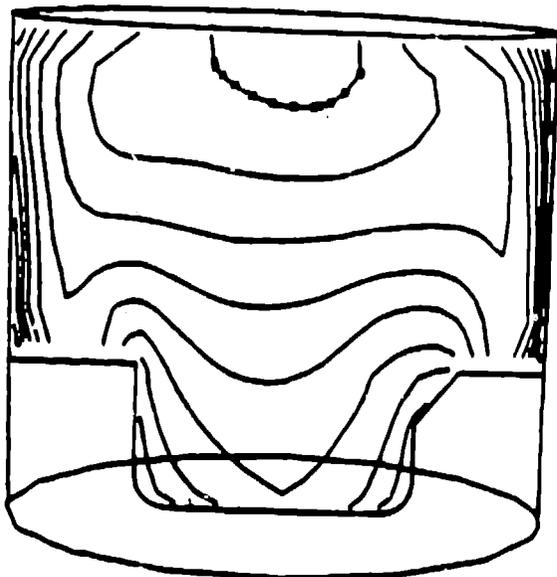
Case 3



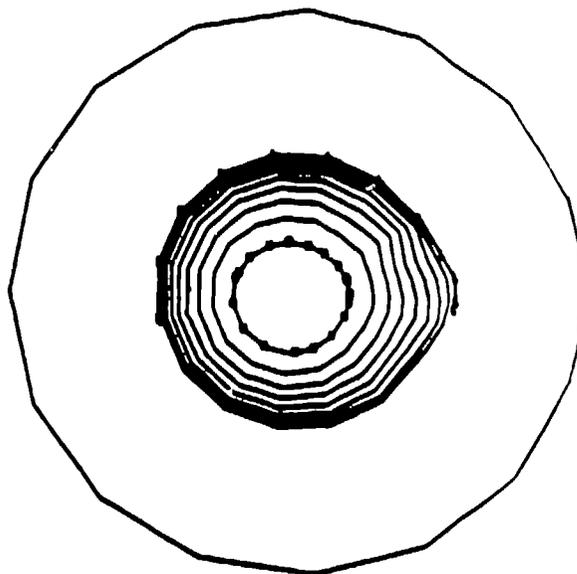
$j = 1 \& 10$

$k = 11$

TEMPERATURE

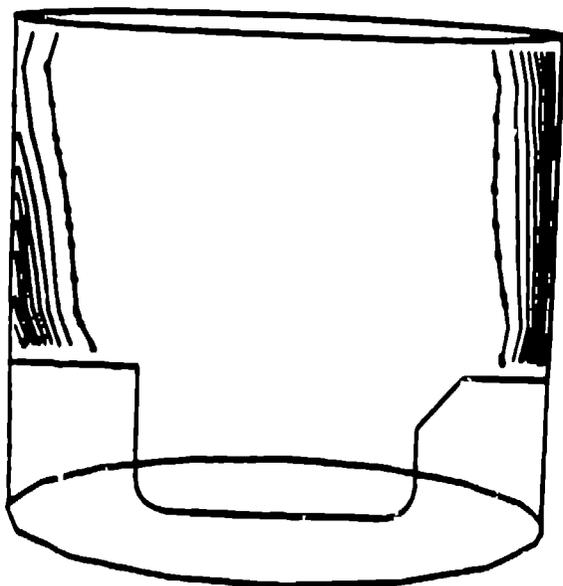


$H = 862 \text{ K} , L = 441 \text{ K}$

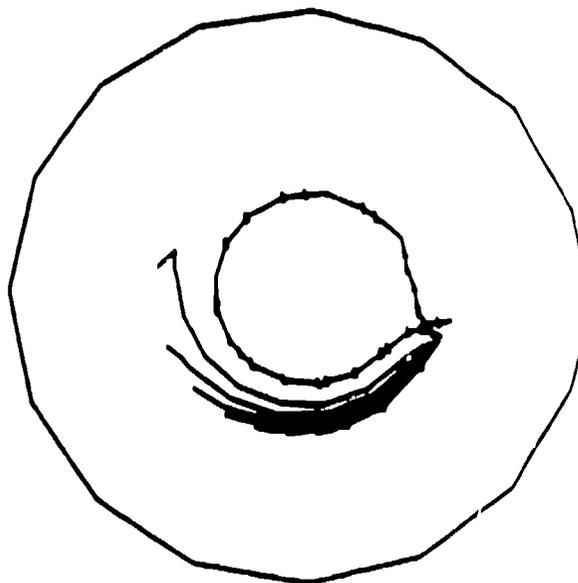


$H = 636 \text{ K} , L = 516 \text{ K}$

MASS FRACTION  $C_6H_6$



$H = 4.4 \times 10^{-6} , L = 4.96 \times 10^{-6}$



$H = 2.82 \times 10^{-6} , L = 3.13 \times 10^{-6}$