EGG-WM--9040

DE91 006178

COMPUTER MODELING OF FLUID FLOW AND COMBUSTION
IN THE ISV CONFINEMENT HOOD

Richard W. Johnson
Seungho Paik

September 1990

Idaho National Engineering Laboratory
EG&G Idaho, Inc.
Idaho Falls, ID 83415

Prepared for the
U.S. Department of Energy
Idaho Operations Office
Under DOE Contract No. DE-AC07-76ID01570

MASTER

db
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

COMPUTER MODELING OF FLUID FLOW AND COMBUSTION
IN THE ISV CONFINEMENT HOOD
EGG-WM-9040

Approved By:

S. K. Merrill, Manager
In Situ Vitrification

Date: 10/30/90

Prepared By:

R. W. Johnson

Date: 9/24/90

Reviewed By:

C. E. Slater
Technical Reviewer

Date: 9/24/90

ABSTRACT

Safety and suitability objectives for the application of the In Situ Vitrification (ISV) technology at the INEL require that the physical processes involved in ISV be modeled to determine their operational behavior. The mathematical models that have been determined to address the modeling needs adequately for the ISV analysis package are detailed elsewhere. The present report is concerned with the models required for simulating the reacting flow that occurs in the ISV confinement hood. An experimental code named COYOTE has been secured that appears adequate to model the combustion in the confinement hood.

The COYOTE code is a two-dimensional, transient, compressible, Eulerian, gas dynamics code for modeling reactive flows. It recognizes nonuniform Cartesian and cylindrical geometry and is based on the ICE (Implicit Continuous-fluid Eulerian) family of solution methods. It includes models for chemical reactions based on chemical kinetics as well as equilibrium chemistry. The mathematical models contained in COYOTE, their discrete analogs, the solution procedure, code structure and some test problems are presented in the report.

TABLE OF CONTENTS

ABSTRACT	ii
LIST OF FIGURES	iv
LIST OF TABLES	v
NOMENCLATURE	vi
1. INTRODUCTION	1
2. MATHEMATICAL MODELS	4
2.1 Governing Equations	4
2.2 Boundary Conditions	8
3. NUMERICAL SOLUTION METHODS	11
3.1 Differential Form of the Fluid Dynamic Equations	11
3.2 Discrete Model Equations	13
3.3 The Numerical Solution Procedure	20
3.4 The Numerical Boundary Conditions	25
3.5 Time Step Constraints	25
4. CODE DESCRIPTION	28
4.1 Overall COYOTE Flow Chart	28
4.2 Flow Control and Description of Subroutines	28
5. VERIFICATION TESTS AND SAMPLE PROBLEM	35
5.1 Test for Successful Porting of Code	35
5.2 Poiseuille Flow	37
5.3 Potential Corner Flow	39
5.4 Unsteady Heat Conduction	46
5.5 Kinetic Chemistry Reaction	55
5.6 Simulation of Flow and Combustion in the ISV Hood	57
6. CONCLUSIONS	60
7. REFERENCES	61

LIST OF FIGURES

1.	General mesh arrangement showing code indices and ghost cells surrounding the computational domain	10
2.	Arrangement of finite difference variables in a typical mesh cell. Indices used in the code are shown in parentheses	10
3.	Overall flow diagram for COYOTE reacting flow code	29
4.	COYOTE tree diagram showing how subprograms are called in the code relative to each other. A dashed line indicates an abnormal abort. The dotted line indicates that the subroutine contains only comment information	34
5.	Schematic of the transfer test problem built into COYOTE. The problem is a shock tube with air initially contained on each side of an imaginary membrane at the conditions shown. At time $t = 0$, the membrane is removed and a pressure shock wave allowed to propagate	36
6.	Schematic of the flow domain for the second test problem: Poiseuille flow	38
7.	Streamwise velocity profile for the Poiseuille channel flow near the end of the computational domain	40
8.	Schematic of the potential corner flow problem. The left and bottom walls are free-slip while the flow enters from the top and exits to the right. Analytically determined streamlines corresponding to the numerical solution grid are shown	41
9.	Schematic of the domain for the unsteady conduction test problem	48
10.	Initial distribution of temperature as a function of x for $y=b/2$ for the unsteady conduction test problem	49
11.	Distribution of temperature as a function of x for $y=b/2$ at time $t=0.001$ seconds for the unsteady conduction test problem .	50
12.	Distribution of temperature as a function of x for $y=b/2$ at time $t=0.002$ seconds for the unsteady conduction test problem .	51
13.	Distribution of temperature as a function of x for $y=b/2$ at time $t=0.003$ seconds for the unsteady conduction test problem .	52
14.	Distribution of temperature as a function of x for $y=b/2$ at time $t=0.004$ seconds for the unsteady conduction test problem .	53

15.	Distribution of temperature as a function of x for $y=b/2$ at time $t=0.005$ seconds for the unsteady conduction test problem .	54
16.	Numerical results for COYOTE compared to the analytical solution for the transient molecular concentration for the kinetic dissociation reaction of a molecule	56
17.	Schematic of the base problem for the three animations recorded on videocassette. The problem is an axisymmetric large scale ISV confinement hood with off-gases entering from a localized region and concentric supply and exhaust ports located at the top center.	58

LIST OF TABLES

1.	Numerical solution steps in COYOTE	20
2.	Numerical boundary conditions found in COYOTE (using the FORTRAN indexing of the code) where ϕ represents ρ , ρ^k , and U .	24
3.	Values for the u -velocity component for the potential corner flow test problem	43
4.	Values for the v -velocity component for the potential corner flow test problem	44
5.	Percentage error in the horizontal pressure gradient, dp/dx , for the potential corner flow test problem	45
6.	Percentage error in the vertical pressure gradient, dp/dy , for the potential corner flow test problem	45

NOMENCLATURE

<u>Symbol</u>	<u>Description of Symbol</u>
a,b,c,d,e	coefficients
A	coefficient
c	molar concentration
c_p	heat capacity at constant pressure
D	diffusion coefficient
E	activation energy
f_i	body force vector
g	gravitational acceleration
G	rate of generation of species
h	heat transfer coefficient
H	specific enthalpy
j_i	diffusional mass flux vector
k	thermal conductivity
k'	reaction rate coefficient
K_c	concentration equilibrium coefficient
m	mass
n_k^{ij}	vector normal to surface Γ_{ij} , pointing from Ω_i to Ω_j
p	pressure
q_i	heat flux vector
Q	chemical heat source
r	reaction rate
R	universal gas constant
t	time
T	temperature
u_i	velocity vector
U	internal energy
W	molecular weight
x_i	position vector
y_i	mole fraction of species i
Z	generic chemical symbol

Greek

δ_{ij}	unit tensor
Δ	increment
ϵ	emissivity
ζ	coefficient
η	heat of formation
μ	dynamic viscosity
μ_2	second coefficient of viscosity
ρ	density
σ_{ij}	stress tensor
Σ	pressure iteration parameter
τ_{ij}	deviatoric stress tensor
ω	boundary mass fluxes
Ω	3D region

Mathematical

∇	del operator
$\rightleftharpoons$	forward and backward reactions
$\cup$	union (combined with)
:	double tensorial contraction

Subscripts

b	backward (reaction)
f	forward (reaction)
i	vector quantity
ij	2nd rank tensor
r	reaction r

Superscripts

c convective component
k species
n time level
R radiation component
s surface (ground)
T matrix transpose

Overlines

^ approximate value
• time derivative

1. INTRODUCTION

Application of In Situ Vittrification (ISV) technology to the INEL for remediation of buried wastes requires that safety and process suitability concerns be adequately addressed. It has been determined that the key physical processes involved in ISV must be simulated to help resolve the issues of safety and process applicability¹. One of these processes is the flow and combustion of off-gases from the melt as they enter the ISV confinement hood. For this process, it is necessary to model the mass and momentum transport, chemical kinetics, thermodynamics, possible effects of turbulence, and potential for explosions¹ of off-gases in the confinement hood. The various models that address these physical phenomena are detailed in Reference 2.

Two combustion codes that have been made available for use at the INEL and have undergone initial screening² are the COYOTE³ and KIVA⁴ codes. COYOTE is an experimental code, still under development, written to provide a vehicle for testing new methods and models for reacting flows. It is simpler to work with than the large, 3D, general mesh code KIVA, which is designed to simulate the complex reacting flow in an internal combustion cylinder. COYOTE uses methods and models from KIVA and other earlier codes (SOLA-ICE⁵) that have been in use for a number of years. Since KIVA violates some of the screening requirements described in Reference 2, the preliminary choice for simulating flow and combustion in the ISV confinement hood is the COYOTE code. COYOTE appears to meet most of the process modeling requirements², although it is less mature and not as well documented as KIVA.

As the preliminary choice to model the flow and combustion in the ISV confinement hood, COYOTE has been examined and tested against some simple verification problems to help determine its suitability for the ISV program objectives. The objectives of the present report are to detail the models, numerical methods, and solution techniques used in COYOTE for modeling flow and combustion; additional objectives include presenting the results of verification tests and the configuration for a sample ISV hood simulation

performed with COYOTE. While a description of the models and numerics used in COYOTE is given in Reference 3, those that pertain to flow and combustion in the ISV confinement hood are reproduced here to incorporate ISV analysis nomenclature¹ and for reader convenience.

The solution method used in COYOTE belongs to a family of solution methods dubbed the ICE (Implicit Continuous-fluid Eulerian) family of methods. The ICE method is actually semi-implicit, since the pressure and momentum divergence are determined implicitly while all other quantities are determined explicitly. This group of methods is designed to handle a wide range of flow speeds from incompressible to compressible flow.

The numerical discretization scheme in COYOTE is based on finite-differences and provides for any combination of centered-differencing, upwinding, or interpolated donor-cell differencing (a convection determined mix of centered-differencing and upwinding) for the convection terms in the Navier-Stokes, energy, and species diffusion equations. The marker and cell (MAC) method (staggered grid) is used to enhance numerical stability.

COYOTE is a multicomponent flow code that can handle as many species as needed. Thermodynamic properties are either determined from the JANAF^{6,7} tables or from an equation of state. The ideal gas law equation of state is currently built into the code as well as the JANAF data for a dozen common reactive species. The chemistry models built into COYOTE allow treatment using standard chemical kinetics for relatively slow reactions and equilibrium chemistry for relatively fast reactions.

The species transport equation employs Fick's law and further assumes that the diffusion coefficient is a constant for all species interactions. The energy equation includes the effects of heat flux from conduction and mass diffusion, but not from radiation. However, radiation heat transfer to the confinement hood environment from the off-gases entering the hood may be important and will be considered for addition to the combustion code in future development of the code.

Boundary conditions are menu-selectable from a standard set. The original set has been enhanced, however, to accommodate the inlet and exhaust ports on the ISV confinement hood.

This document describes the models and numerical methods incorporated in COYOTE that are pertinent to combustion in the ISV confinement hood. Section 2 describes the physical and chemical models that are contained in the code. Section 3 outlines the numerical methods used to represent and solve the physical models. Section 4 details the structure and logistics of the code. Finally, Section 5 reports verification tests made to verify some of the COYOTE models and the configuration of a sample combustion problem in the ISV hood which is animated and available on videocassette.

2. MATHEMATICAL MODELS

The mathematical models employed in COYOTE to represent flow, energy transfer, and chemical reactions are presented in this section. The boundary conditions that are available for the dependent model variables are also described.

2.1 Governing Equations

The flow, heat transfer, and combustion processes modeled in COYOTE are based on the compressible Navier-Stokes equations for a mixture of gases in Eulerian coordinates³.

The conservation of mass or mass transport for each species k is represented by

$$\frac{\partial \rho^k}{\partial t} + \nabla \cdot (\rho^k \mathbf{u}_i) = -\nabla \cdot \mathbf{j}_i^k + G^k \quad (1)$$

where ρ^k is the species density, $\mathbf{u}_i$ is the average mixture velocity, $\mathbf{j}_i^k$ is the diffusional mass flux vector of species k , and G^k is the rate of change of species k due to chemical reaction. The diffusional flux is a complicated function which in general depends on pressure gradient, temperature gradient, body forces and the concentration gradients of other species present in the system⁸. However, since the only body force present is gravity, the body force term is identically zero while the dependencies on pressure and temperature gradients are neglected. The expression used for the diffusional flux in COYOTE is a simplified version of Fick's Law:

$$\mathbf{j}_i^k = -\rho D \nabla \left(\frac{\rho^k}{\rho} \right) \quad (2)$$

where D is the diffusion coefficient (assumed to be independent of species) and ρ is the total density.

Upon summing (1) over all species present, the continuity equation is obtained:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}_i) = 0. \quad (3)$$

The momentum transport (Navier-Stokes) equations employed in COYOTE are represented by

$$\frac{\partial \rho \mathbf{u}_i}{\partial t} + \nabla \cdot (\rho \mathbf{u}_i \mathbf{u}_j) = \rho \mathbf{g}_i - \nabla p - \nabla \cdot \tau_{ij} \quad (4)$$

where p is the pressure, and $\mathbf{g}_i$ the acceleration due to gravity. The deviatoric stress tensor τ_{ij} is given by

$$\tau_{ij} = -\mu [\nabla \mathbf{u}_i + (\nabla \mathbf{u}_i)^T] - \mu_2 (\nabla \cdot \mathbf{u}_i) \delta_{ij} \quad (5)$$

where μ is the dynamic viscosity, μ_2 the second coefficient of viscosity, and δ_{ij} the unit tensor. Either a constant value for the kinematic viscosity ν_0 may be input by the user ($\mu_{ij} = \rho_{ij} \nu_0$) or the user may choose to use the Sutherland formula³ for the temperature variation for air, $\mu = 1.457 \times 10^{-5} \times [T^{3/2}/(T + 110)]$, where T is in Kelvins.

COYOTE incorporates two turbulence models, ie. the $k-\epsilon$ and a sub-grid scale (SGS) eddy viscosity models. Since it has not been determined that turbulence will need to be modeled, a discussion of the above models will not be given. Reference 3 gives an account of the two turbulence models as well as the turbulent boundary conditions available in COYOTE.

Energy conservation is expressed in terms of the specific internal energy as

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho u_i U) = -p \nabla \cdot u_i - \tau_{ij} \cdot \nabla u_i - \nabla \cdot q_i + \sum_k \eta^k G^k \quad (6)$$

where U is the internal energy and η^k is the heat of formation for species k . The heat flux vector q_i in the energy equation (6) is approximated by

$$q_i = -k \nabla T + \sum_k H^k j_i^k \quad (7)$$

where k is the mixture thermal conductivity, T is the temperature, and H^k is the specific enthalpy of species k . The contribution of radiation heat transfer to the energy equation is neglected in COYOTE.

The equation of state is based on the ideal gas law and given by the sum of partial pressures for each species:

$$p = \sum_k \frac{R \rho^k T}{W^k} \quad (8)$$

where R is the universal gas constant, W^k is the molecular weight of species k , and T is the absolute temperature.

The mixture internal energy is given by

$$\rho U = \sum_k \rho^k U^k(T) \quad (9)$$

where the species internal energy U^k is linearly interpolated from data provided in the JANAF^{6,7} tables for species enthalpy.

Chemical reactions occurring during the combustion process are represented by

where the symbol $\rightleftharpoons$ indicates both forward and backward reactions, Z^k represents the chemical symbol for species k , and a_r^k and b_r^k are the stoichiometric coefficients for the r th reaction. The rate of change G^k of species k appearing in the species continuity equation is represented by

$$G^k = W^k \sum_r (b_r^k - a_r^k) r_r \quad (11)$$

where r_r is the rate of reaction of the r th reaction.

The reaction may be either an equilibrium reaction where the forward rate of reaction is equal to the backward rate, or a kinetic reaction where the rates are not equal. In the latter case, the net reaction rate is given by

$$r_r = k'_{fr} \prod_k \left(\frac{\rho^k}{W^k} \right)^{a_r^{k'}} - k'_{br} \prod_k \left(\frac{\rho^k}{W^k} \right)^{b_r^{k'}} \quad (12)$$

where k'_{fr} and k'_{br} are forward and backward reaction coefficients for reaction r and the exponents $a_r^{k'}$ and $b_r^{k'}$ are reaction orders. A generalized Arrhenius form is used for the forward and backward rate coefficients:

$$k'_{fr} = A_{fr} T^{\zeta_{fr}} e^{(-E_{fr}/RT)} \quad (13)$$

$$k'_{br} = A_{br} T^{\zeta_{br}} e^{(-E_{br}/RT)} \quad (14)$$

where E_{fr} and E_{br} are forward and backward activation energies, respectively, and A_{fr} , A_{br} , ζ_{fr} , and ζ_{br} are empirical constants. For equilibrium reactions, the net reaction rate is zero. Equation (12) then becomes

$$\frac{k'_{fr}}{k'_{br}} = \prod_k \left(\frac{\rho^k}{W^k} \right)^{b_r^{k'} - a_r^{k'}} = K_{cr}(T) \quad (15)$$

where $K_{cr}(T)$ is the concentration equilibrium constant for reaction r and is represented by the following temperature dependent relation

$$K_{cr}(T) = \exp \left[a_r \ln T_A + \frac{b_r}{T_A} + c_r + d_r T_A + e_r T_A^2 \right] \quad (16)$$

where a_r , b_r , c_r , d_r , and e_r are empirical coefficients related to reaction r and are input by the user. Also, $T_A = T/1000K$.

2.2 Boundary Conditions

The computational domain that is recognized by COYOTE is limited to Cartesian and cylindrical (axisymmetric) geometry and, hence, has a left, a right, a top, and a bottom boundary. (See Figure 1). Boundary conditions are therefore required for the velocity, species density, total density, and specific internal energy on each of the four boundaries. Additionally, a specified boundary condition is allowed for the pressure on the top and bottom boundaries. Let the following quantities be defined:

- u_n = velocity normal to a boundary,
- u_p = velocity parallel to a boundary,
- n = direction normal to a boundary, and
- ϕ = generic variable representing density ρ , species density ρ^k , and specific internal energy U .

The boundary conditions that are available in COYOTE are

1. Free-slip boundary:
 $u_n = 0, u_p = u_p, \partial\phi/\partial n = 0$
2. No-slip boundary
 $u_n = 0, u_p = 0, \partial\phi/\partial n = 0$

3. Continuative flow

$$u_{n, \text{ outside }} = u_{n, \text{ inside }}, u_{p, \text{ outside }} = u_{p, \text{ inside }}, \partial\phi/\partial n = 0$$

4. Periodic flow

$$u_n = u_{n, \text{ opposite wall }}, u_p = u_{p, \text{ opposite wall }}, \phi = \phi_{\text{opposite wall}}$$

5. Specified pressure

$$p = p_{\text{specified}}, \partial u_n / \partial n = 0, \partial u_p / \partial n = 0, \partial\phi / \partial n = 0$$

6. Specified velocity

$$u_n = u_{\text{specified}}, u_p = 0, \partial\phi / \partial n = 0$$

7. Combined boundary condition

Any one of the above boundary conditions can be specified for a particular segment of the boundary. (This option has been added for the ISV confinement hood application to allow for air supply inflow, exhaust outflow, and solid boundary on the top of the hood).

Boundary conditions 1-4 are available on all four boundaries. Conditions 5 and 6 are only available on the top and bottom boundaries and 7 is only available on the top boundary.

The implementation of the mathematical models and boundary conditions, plus the solution procedure for solving the coupled conservation equations is detailed in the following section.

Figure 1. General mesh arrangement showing code indices and ghost cells surrounding the computational domain.

Figure 2. Arrangement of finite difference variables in a typical mesh cell. Indices used in the code are shown in parentheses.

3. NUMERICAL SOLUTION METHODS

The model equations are given first in differential form below and then in discrete finite-difference form. The nature of the ICE solution method is given next. Finally, the discrete boundary conditions and stability limits for the time step are given.

3.1 Differential Form of the Fluid Dynamic Equations

COYOTE solves the two dimensional Eulerian flow equations in Cartesian and cylindrical coordinates. Before presenting the discrete forms of the flow equations, they are presented in differential form for ease of comparison with the discrete forms. The species continuity equation is

$$\begin{aligned} \frac{\partial \rho^k}{\partial t} + \frac{1}{r} \frac{\partial (r \rho^k u)}{\partial x} + \frac{\partial (\rho^k v)}{\partial z} - G^k + \frac{1}{r} \frac{\partial}{\partial x} \left[r \rho D \frac{\partial}{\partial x} \left(\frac{\rho^k}{\rho} \right) \right] \\ + \frac{\partial}{\partial z} \left[\rho D \frac{\partial}{\partial z} \left(\frac{\rho^k}{\rho} \right) \right] \end{aligned} \quad (17)$$

where x is the horizontal or radial coordinate, z the vertical or axial coordinate, and u and v are velocity components in the x and z directions, respectively. The parameter r is set to one for Cartesian coordinates and to x (the radial distance) for cylindrical coordinates.

The radial and axial momentum equations are given by

$$\begin{aligned} \rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial z} \right) = \rho g_x - \frac{\partial p}{\partial x} + \frac{1}{r} \frac{\partial}{\partial x} \left(2r\mu \frac{\partial u}{\partial x} \right) \\ - \frac{\partial}{\partial x} \left[\frac{2\mu}{3} \left(\frac{1}{r} \frac{\partial ru}{\partial x} + \frac{\partial v}{\partial z} \right) \right] + \frac{\partial}{\partial z} \left[\mu \left(\frac{\partial u}{\partial z} + \frac{\partial v}{\partial x} \right) \right] - \frac{2\xi\mu u}{r^2} \end{aligned} \quad (18a)$$

and

$$\begin{aligned} \rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial z} \right) = \rho g_z - \frac{\partial p}{\partial z} + \frac{1}{r} \frac{\partial}{\partial x} \left[r\mu \left(\frac{\partial u}{\partial z} + \frac{\partial v}{\partial x} \right) \right] \\ + \frac{\partial}{\partial z} \left[2\mu \frac{\partial v}{\partial z} - \frac{2\mu}{3} \left(\frac{1}{r} \frac{\partial ru}{\partial x} + \frac{\partial v}{\partial z} \right) \right] \end{aligned} \quad (18b)$$

where the second coefficient of viscosity has been taken as $\mu_2 = -2\mu/3$. The parameter ξ is assigned the value of zero and one for Cartesian and cylindrical geometry, respectively. The value of r is set the same as for the continuity equation.

The conservation of specific internal energy is given by

$$\begin{aligned} \frac{\partial \rho U}{\partial t} + \frac{1}{r} \frac{\partial r \rho u U}{\partial x} + \frac{\partial \rho v U}{\partial z} = -p \left(\frac{1}{r} \frac{\partial ru}{\partial x} + \frac{\partial v}{\partial z} \right) + \left[\frac{1}{r} \frac{\partial}{\partial x} \left(rk \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) \right] \\ + \mu \left[2 \left(\frac{\partial u}{\partial x} \right)^2 + 2 \left(\frac{\partial v}{\partial z} \right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial v}{\partial x} \right)^2 + \frac{2\xi u^2}{x^2} - \frac{2}{3} \left(\frac{1}{r} \frac{\partial ru}{\partial x} + \frac{\partial v}{\partial z} \right)^2 \right] \\ + \sum_k \left\{ \eta^k G^k + \frac{1}{r} \frac{\partial}{\partial x} \left[r H^k \rho D \frac{\partial}{\partial x} \left(\frac{\rho^k}{\rho} \right) \right] + \frac{\partial}{\partial z} \left[H^k \rho D \frac{\partial}{\partial z} \left(\frac{\rho^k}{\rho} \right) \right] \right\}. \end{aligned} \quad (19)$$

The finite-difference discretization of the above differential equations is given in the following section.

3.2 Discrete Model Equations

In order to discretize fully the differential model equations, a computational domain layout and labeling system is needed. Figure 1 illustrates the computational domain used in COYOTE. As shown, there is a layer of cells surrounding the computational domain which is used only for convenience in applying the boundary conditions. The mesh cells may be nonuniform in size. Obstacles may be defined in the mesh by blocking out given cells with a cell flag array. The layout and labeling for each cell is shown in Figure 2. The labeling that is used for the discrete equations herein is given along with the actual FORTRAN indexing in parentheses. (FORTRAN recognizes only integer indexing). The u-velocity component is located at the center of the vertical faces, while the v component is located at the center of the horizontal faces. (The concept of vertical and horizontal orientations is solely for reference to Figure 2). All other dependent variables are located at the node within the cell. The coordinates of the cell nodes are denoted as (x_i, z_j) . Hence, for a nonuniform mesh, the velocities are not located midway between nodes. However, in order to maintain first order accuracy on a nonuniform mesh, it is necessary to define an average position of two cell centers

$$\bar{x}_{i-1/2} = 0.5(x_{i-1} + x_i) \quad (20)$$

to be used in place of cell edges in some instances. The mesh in the z direction is defined similarly.

In writing the discrete equations, the superscript n is used for time level. For convenience, only those terms that are to be evaluated at the new time level (n+1) are noted; all other terms are at the current time level (n). The discrete representation for the species transport equation (17) is

$$\begin{aligned}
\rho_{i,j}^{k,n+1} - \hat{\rho}_{i,j}^k - \Delta t & \left\{ \frac{1}{r_i(\bar{x}_{i+1/2} - \bar{x}_{i-1/2})} \left[\bar{r}_{i+1/2} C_{i+1/2,j}^x \left(\frac{\rho^k}{\rho} \right) - \bar{r}_{i-1/2} C_{i-1/2,j}^x \left(\frac{\rho^k}{\rho} \right) \right] \right. \\
& + \frac{1}{\bar{z}_{j+1/2} - \bar{z}_{j-1/2}} \left[C_{i,j+1/2}^z \left(\frac{\rho^k}{\rho} \right) - C_{i,j-1/2}^z \left(\frac{\rho^k}{\rho} \right) \right] \Bigg\} \\
& + \frac{\Delta t}{2 Sc} \left[D_i^1 (\mu_{i+1,j} + \mu_{i,j}) \left(\frac{\rho_{i+1,j}^k}{\rho_{i+1,j}} - \frac{\rho_{i,j}^k}{\rho_{i,j}} \right) - D_i^2 (\mu_{i,j} + \mu_{i-1,j}) \left(\frac{\rho_{i,j}^k}{\rho_{i,j}} - \frac{\rho_{i-1,j}^k}{\rho_{i-1,j}} \right) \right. \\
& + D_j^3 (\mu_{i,j+1} + \mu_{i,j}) \left(\frac{\rho_{i,j+1}^k}{\rho_{i,j+1}} - \frac{\rho_{i,j}^k}{\rho_{i,j}} \right) - D_j^4 (\mu_{i,j} + \mu_{i,j-1}) \left(\frac{\rho_{i,j}^k}{\rho_{i,j}} - \frac{\rho_{i,j-1}^k}{\rho_{i,j-1}} \right) \Bigg] \quad (21)
\end{aligned}$$

where Sc is the Schmidt number and the cell-face centered convective fluxes for a generic quantity Q are given by

$$\begin{aligned}
C_{i+1/2,j}^x(Q) &= 0.5 \left\{ (\rho_{i,j} Q_{i,j} + \rho_{i+1,j} Q_{i+1,j}) u_{i+1/2,j}^{n+1} \right. \\
& + \left[\alpha |u_{i+1/2,j}^{n+1}| + \frac{\beta (u_{i+1/2,j}^{n+1})^2 \Delta t}{x_{i+1} - x_i} \right] (\rho_{i,j} Q_{i,j} - \rho_{i+1,j} Q_{i+1,j}) \Bigg\} \quad (22)
\end{aligned}$$

and

$$\begin{aligned}
C_{i,j+1/2}^z(Q) &= 0.5 \left\{ (\rho_{i,j} Q_{i,j} + \rho_{i,j+1} Q_{i,j+1}) v_{i,j+1/2}^{n+1} \right. \\
& + \left[\alpha |v_{i,j+1/2}^{n+1}| + \frac{\beta (v_{i,j+1/2}^{n+1})^2 \Delta t}{z_{j+1} - z_j} \right] (\rho_{i,j} Q_{i,j} - \rho_{i,j+1} Q_{i,j+1}) \Bigg\} \quad (23)
\end{aligned}$$

These fluxes are linear combinations of donor cell differencing (α), interpolated donor cell differencing (β), and centered differencing ($1 - \alpha - \beta$). Reference 9 describes these schemes in detail. The diffusion coefficients in the species transport equation (21) are given by

$$D_i^1 = \frac{\bar{r}_{i+1/2}}{r_i(x_{i+1} - x_i)(\bar{x}_{i+1/2} - \bar{x}_{i-1/2})}, \quad (24)$$

$$D_i^2 = \frac{\bar{r}_{i-1/2}}{r_i(x_i - x_{i-1})(\bar{x}_{i+1/2} - \bar{x}_{i-1/2})}, \quad (25)$$

$$D_j^3 = \frac{1}{(z_{j+1} - z_j)(\bar{z}_{j+1/2} - \bar{z}_{j-1/2})}, \quad (26)$$

and

$$D_j^4 = \frac{1}{(z_j - z_{j-1})(\bar{z}_{j+1/2} - \bar{z}_{j-1/2})}. \quad (27)$$

The explicit discretization of the x-direction or radial momentum equation (18a) in COYOTE is given by

$$\begin{aligned} \hat{u}_{i-1/2,j} - u_{i-1/2,j} + \Delta t \left[g_x - \frac{2(P_{i,j} - P_{i-1,j})}{(\rho_{i,j} + \rho_{i-1,j})(x_i - x_{i-1})} - \left(u \frac{\partial u}{\partial x} \right)_{i-1/2,j} - \left(v \frac{\partial u}{\partial z} \right)_{i-1/2,j} \right] \\ + \frac{2 \Delta t}{\rho_{i,j} + \rho_{i-1,j}} \left\{ \frac{2}{r_{i-1/2}(x_i - x_{i-1})} \left(r_i \mu_{i,j} \frac{u_{i+1/2,j} - u_{i-1/2,j}}{x_{i+1/2} - x_{i-1/2}} - r_{i-1} \mu_{i-1,j} \frac{u_{i-1/2,j} - u_{i-3/2,j}}{x_{i-1/2} - x_{i-3/2}} \right) \right. \\ - \frac{2}{3(x_i - x_{i-1})} [\mu_{i,j}(\nabla \cdot u)_{i,j} - \mu_{i-1,j}(\nabla \cdot u)_{i-1,j}] - \xi u_{i-1/2,j} \frac{\mu_{i,j} + \mu_{i-1,j}}{r_{i-1/2}^2} \\ \left. + \frac{1}{z_{j+1/2} - z_{j-1/2}} [(S_{xz})_{i-1/2,j+1/2} - (S_{xz})_{i-1/2,j-1/2}] \right\}. \end{aligned} \quad (28)$$

The quantity $\hat{u}$ is the explicit approximation of the u-component of velocity and is based on the pressure gradient at time level n. Since it is desired to base u implicitly on the n+1 level for pressure, this approximation must be corrected. This correction occurs during the application of the ICE method pressure iteration phase where the pressure is determined for the n+1 time level. The solution method is outlined in Section 3.3 below.

The radial convective velocity gradient appearing in (28) is given by

$$\left(u \frac{\partial u}{\partial x}\right)_{i-1/2,j} = \frac{\beta \Delta t u_{i-1/2,j}^2}{2(x_i - x_{i-1})} \left(\frac{u_{i-1/2,j} - u_{i-3/2,j}}{\Delta x_{i-1}} - \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x_i} \right) \quad (29)$$

$$+ u_{i-1/2,j} \frac{\Delta x_{i-1} [1 - \alpha \operatorname{sgn}(u_{i-1/2,j})] \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x_i} + \Delta x_i [1 + \alpha \operatorname{sgn}(u_{i-1/2,j})] \frac{u_{i-1/2,j} - u_{i-3/2,j}}{\Delta x_{i-1}}}{\Delta x_i + \Delta x_{i-1} + \alpha \operatorname{sgn}(u_{i-1/2,j}) (\Delta x_i - \Delta x_{i-1})}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and $\operatorname{sgn}(u)$ is the sign (+ or -) of u. The user-selectable combination of upwind, interpolated donor cell and centered differencing are incorporated into the convective term discretizations similarly as for the species conservation equation (21). The discrete representation for the axial convection term is analogous to that for the radial one. The staggered nature of the mesh requires that the following quantities be computed for the axial convection:

$$v_{i-1/2,j}^{\text{int.}} = \frac{\Delta x_i (v_{i-1,j-1/2} + v_{i-1,j+1/2}) + \Delta x_{i-1} (v_{i,j-1/2} + v_{i,j+1/2})}{2(\Delta x_i + \Delta x_{i-1})} \quad (30)$$

and

$$\left(\frac{\partial u}{\partial z}\right)_{i-1,j+1/2} = \frac{u_{i-1/2,j+1} - u_{i-1/2,j}}{\Delta z_{j+1/2}} \quad (31)$$

The discrete equation for the axial convection term is given as

$$\begin{aligned} \left(v \frac{\partial u}{\partial z}\right)_{i-1/2,j} &= \frac{\beta \Delta t (v_{i-1/2,j}^{\text{int}})^2}{2(\bar{z}_{j+1/2} - \bar{z}_{j-1/2})} \left(\frac{u_{i-1/2,j} - u_{i-1/2,j-1}}{\Delta z_{j-1/2}} - \frac{u_{i-1/2,j+1} - u_{i-1/2,j}}{\Delta z_{j+1/2}} \right) \\ &+ v_{i-1/2,j}^{\text{int}} \frac{\Delta z_{j+1/2} [1 + \alpha \operatorname{sgn}(v_{i-1/2,j}^{\text{int}})] \left(\frac{\partial u}{\partial z}\right)_{i-1/2,j-1/2} + \Delta z_{j-1/2} [1 - \alpha \operatorname{sgn}(v_{i-1/2,j}^{\text{int}})] \left(\frac{\partial u}{\partial z}\right)_{i-1/2,j+1/2}}{\Delta z_{j+1/2} + \Delta z_{j-1/2} + \alpha \operatorname{sgn}(v_{i-1/2,j}^{\text{int}}) (\Delta z_{j+1/2} - \Delta z_{j-1/2})} \end{aligned} \quad (32)$$

The cell-centered velocity divergence appearing in the radial momentum equation is approximated as follows:

$$(\nabla \cdot u_1)_{i,j} = \frac{r_{i+1/2} u_{i+1/2,j} - r_{i-1/2} u_{i-1/2,j}}{r_i (x_{i+1/2} - x_{i-1/2})} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{z_{j+1/2} - z_{j-1/2}} \quad (33)$$

The xz component of the stress tensor in (28), calculated at cell corners, is given as

$$(S_{xz})_{i-1/2,j+1/2} = \mu_{i-1/2,j+1/2} \left(\frac{u_{i-1/2,j+1} - u_{i-1/2,j}}{z_{j+1} - z_j} + \frac{v_{i,j+1/2} - v_{i-1,j+1/2}}{x_i - x_{i-1}} \right) \quad (34)$$

The cell-corner viscosities are calculated by averaging the four surrounding cell-centered viscosities. This completes the full specification of the radial momentum equation.

The explicit axial momentum equation is analogous to the radial equation and is given as

$$\begin{aligned}
 \hat{v}_{i,j-1/2} - v_{i,j-1/2} + \Delta t \left[g_z - \frac{2(P_{i,j} - P_{i,j-1})}{(\rho_{i,j} + \rho_{i,j-1})(z_j - z_{j-1})} - \left(u \frac{\partial v}{\partial x} \right)_{i,j-1/2} - \left(v \frac{\partial v}{\partial z} \right)_{i,j-1/2} \right] \\
 + \frac{2\Delta t}{(\rho_{i,j} + \rho_{i,j-1})\Delta z_{j-1/2}} \left[\mu_{i,j} \left(\frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta z_j} - \frac{1}{3}(\nabla \cdot u_i)_{i,j} \right) \right. \\
 \left. - \mu_{i,j-1} \left(\frac{v_{i,j-1/2} - v_{i,j-3/2}}{\Delta z_{j-1}} - \frac{1}{3}(\nabla \cdot u_i)_{i,j-1} \right) \right] \\
 + \frac{2\Delta t}{r_i(\bar{x}_{i+1/2} - \bar{x}_{i-1/2})(\rho_{i,j} + \rho_{i,j-1})} \left[\bar{r}_{i+1/2}(S_{xz})_{i+1/2,j-1/2} - \bar{r}_{i-1/2}(S_{xz})_{i-1/2,j-1/2} \right].
 \end{aligned} \tag{35}$$

The discrete representation of the energy equation (19) is written using the convective flux operators, Equations (22) and (23), and the diffusion coefficients defined by Equations (24)-(27) such that

$$\begin{aligned}
& \rho_{i,j}^{n+1} U_{i,j}^{n+1} - \rho_{i,j} \hat{U}_{i,j} + \Delta t \left[D_1^1 (k_{i+1,j} + k_{i,j}) (T_{i+1,j} - T_{i,j}) - D_1^2 (k_{i,j} + k_{i-1,j}) (T_{i,j} - T_{i-1,j}) \right. \\
& + D_j^3 (k_{i,j+1} + k_{i,j}) (T_{i,j+1} - T_{i,j}) - D_j^4 (k_{i,j} + k_{i,j-1}) (T_{i,j} - T_{i,j-1}) \left. \right] \\
& - \Delta t \left\{ \frac{1}{r_i (\bar{x}_{i+1/2} - \bar{x}_{i-1/2})} \left[\bar{r}_{i+1/2} C_{i+1/2,j}^x(U) - \bar{r}_{i-1/2} C_{i-1/2,j}^x(U) \right] \right. \\
& + \frac{1}{\bar{z}_{j+1/2} - \bar{z}_{j-1/2}} \left[C_{i,j+1/2}^z(U) - C_{i,j-1/2}^z(U) \right] + p_{i,j}^{n+1} (\nabla \cdot u_1)_{i,j}^{n+1} \left. \right\} \\
& + \mu_{i,j} \Delta t \left\{ 2 \left(\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x_i} \right)^2 + 2 \left(\frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta z_j} \right)^2 \right. \\
& + \left[\frac{u_{i+1/2,j+1} + u_{i-1/2,j+1} - u_{i+1/2,j-1} - u_{i-1/2,j-1}}{2(z_{j+1} - z_{j-1})} \right. \\
& + \left. \left. \frac{v_{i+1,j+1/2} + v_{i+1,j-1/2} - v_{i-1,j+1/2} - v_{i-1,j-1/2}}{2(x_{i+1} - x_{i-1})} \right]^2 \right. \\
& + \left. \frac{\xi (u_{i+1/2,j} + u_{i-1/2,j})^2}{2x_i^2} - \frac{2}{3} \left(\frac{r_{i+1/2} u_{i+1/2,j} - r_{i-1/2} u_{i-1/2,j}}{r_i \Delta x_i} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta z_j} \right)^2 \right\}.
\end{aligned}
\tag{36}$$

3.3 The Numerical Solution Procedure

The solution procedure in COYOTE is divided into four phases³. Table 1 outlines the steps in the COYOTE solution procedure. After initialization, the first phase computes new transport coefficients and the new time step. COYOTE has the option to automatically reset the time step each time to prevent instability. Next, the chemical kinetic and equilibrium reactions, Equations (12) and (15), are solved to produce new

Table 1. Numerical Solution Steps in COYOTE³.

Solution Phase	Solution Step	Operation
0	1.	Initialize all variables.
1	2.	Compute transport coefficients.
1	3.	Compute new time step.
1	4.	Compute species and energy changes due to chemistry.
1	5.	Calculate explicit approximations to velocities.
1	6.	Apply boundary conditions, heat transfer.
2	7.	Iterate to obtain pressure; update u, and v.
3	8.	Compute species transport; update p.
3	9.	Solve internal energy equation (except chemistry and boundary layer).
4	10.	Update old time arrays.
4	11.	Apply equation of state to reset p, T, c_v , and W.
4	12.	Compute new SOR relaxation factors.
4	13.	Write output.
4	13.	Quit or go to new time step.

time values for the generation rates for each species. After the generation rates are computed, the species densities and the internal energy are updated as follows:

$$\hat{\rho}_{i,j}^k = \rho_{i,j}^k + \Delta t G^{k,n} \quad (37)$$

and

$$\hat{U}_{i,j} = U_{i,j} + \Delta t \sum_k \eta^k G^{k,n} . \quad (38)$$

The circumflex symbols over the density and internal energy indicate intermediate values since the other contributions to ρ and U have not yet been computed.

The next step in phase one is to compute the explicit velocity approximations using Equations (28) and (35) from the previous section. Finally, the boundary conditions are applied to account for drag, etc.

The second phase involves applying the Implicit Continuous-fluid Eulerian (ICE) method to obtain values for the overall density, internal energy, the pressure, and the velocity components. The ICE family of solution methods includes a number of variations. The particular method used in COYOTE is outlined here. A basic feature of the ICE family of solution methods is that flows ranging from incompressible to compressible may be simulated using the method. This is important for ISV since the flow will range from low-speed to the compressible range.

A partially implicit Lagrangian solution is calculated by solving simultaneously the following set of equations for the starred variables:

$$\rho_{i,j}^* = \frac{\rho_{i,j}}{1 + \Delta t (\nabla \cdot \mathbf{u}_i^*)_{i,j}} , \quad (39)$$

$$U_{i,j}^* = \hat{U}_{i,j} - \frac{\Delta t p_{i,j}^*}{\rho_{i,j}} (\nabla \cdot u_i^*)_{i,j}, \quad (40)$$

$$p_{i,j}^* = f(\rho_{i,j}^*, U_{i,j}^*), \quad (41)$$

$$u_{i+1/2,j}^* = \hat{u}_{i+1/2,j} - \frac{\Delta t}{\rho_{i+1/2,j} (x_{i+1} - x_i)} (p_{i+1,j}^* - p_{i,j}^*), \quad (42)$$

$$v_{i,j+1/2}^* = \hat{v}_{i,j+1/2} - \frac{\Delta t}{\rho_{i,j+1/2} (z_{j+1} - z_j)} (p_{i,j+1}^* - p_{i,j}^*), \quad (43)$$

and

$$(\nabla \cdot u_i^*)_{i,j} = \frac{r_{i+1/2} u_{i+1/2,j}^* - r_{i-1/2} u_{i-1/2,j}^*}{r_i (x_{i+1/2} - x_{i-1/2})} + \frac{v_{i,j+1/2}^* - v_{i,j-1/2}^*}{z_{j+1/2} - z_{j-1/2}}. \quad (44)$$

where $f(\rho, U)$ represents the gas equation of state.

This phase is also called the pressure iteration, since iterations must be made to obtain the solutions to Equations (39)-(44). The starred variables in these equations represent values that are not at either time level (n or $n+1$), but at some intermediate level since ρ and U must still be updated using the full discrete equations (21) and (36). Hence, Equations (39)-(44) represent 'Lagrangian' approximations to the complete Eulerian equations.

The iteration is begun by defining a new parameter $\Sigma(p)$ as follows:

$$\Sigma = p_{i,j}^* - f(\rho_{i,j}^*, U_{i,j}^*). \quad (45)$$

At the beginning of the iteration, $\Sigma(p)$ is nonzero since the pressure doesn't match the value returned from the equation of state. It is the objective of the iteration to reduce Σ to some small value set by the user. This is accomplished by using successive over-relaxation (SOR) of Newton's method. This can be written as

$$p_{i,j}^{v+1} = p_{i,j}^v - \frac{\Sigma(p^v)}{\frac{\partial \Sigma(p^v)}{\partial p}} \quad (46)$$

where v is the iteration count. The sensitivity term in the denominator on the rhs of (46) is calculated once and then used throughout the iteration phase. The term is computed by employing (45) and perturbing the pressure. The approximate effects to the values for p , U , and the velocity divergence are then computed. This information is then used together with (45) to obtain a value for the sensitivity³. The second stage of the pressure iteration involves obtaining updates to the divergence, density and internal energy based on the current value of pressure and velocity components. Equation (46) is then used to update the pressure, after which the velocities are updated completing the iteration. If the pressure differences are small compared to the previous pressure values for each node, the iteration is terminated and phase three is reached.

Phase three consists of computing the full species transport and the internal energy equation (except for the chemistry which is computed in phase one). Phase four involves miscellaneous tasks as shown in Table 1.

Table 2. Numerical boundary conditions found in COYOTE, using the FORTRAN indexing of the code, where ϕ represents p , p^k , and U .

BOUNDARY CONDITION TYPE	TOP BOUNDARY (i=1,nx2)	BOTTOM BOUNDARY (i=1,nx2)	RIGHT BOUNDARY (j=1,ny2)	LEFT BOUNDARY (j=1,ny2)
1. free slip	$v(i,ny2) = 0$ $u(i,ny2) = u(i,ny1)$ $v(i,ny3) = -v(i,ny1)$ $\phi(i,ny2) = \phi(i,ny1)$	$v(i,2) = 0$ $u(i,1) = u(i,2)$ $v(i,1) = -v(i,3)$ $\phi(i,1) = \phi(i,2)$	$u(nx2,j) = 0$ $v(nx2,j) = v(nx1,j)$ $u(nx3,j) = -u(nx1,j)$ $\phi(nx2,j) = \phi(nx1,j)$	$u(2,j) = 0$ $v(1,j) = v(2,j)$ $u(1,j) = -u(3,j)$ $\phi(1,j) = \phi(2,j)$
2. no slip	$v(i,ny2) = 0$ $u(i,ny2) = -u(i,ny1)$ $v(i,ny3) = -v(i,ny1)$ $\phi(i,ny2) = \phi(i,ny1)$	$v(i,2) = 0$ $u(i,1) = -u(i,2)$ $v(i,1) = -v(i,3)$ $\phi(i,1) = \phi(i,2)$	$u(nx2,j) = 0$ $v(nx2,j) = -v(nx1,j)$ $u(nx3,j) = -u(nx1,j)$ $\phi(nx2,j) = \phi(nx1,j)$	$u(2,j) = 0$ $v(1,j) = -v(2,j)$ $u(1,j) = -u(3,j)$ $\phi(1,j) = \phi(2,j)$
3. continuative	$v(i,ny2) = v(i,ny1)$ $u(i,ny2) = u(i,ny1)$ $v(i,ny3) = v(i,ny2)$ $\phi(i,ny2) = \phi(i,ny1)$	$v(i,2) = v(i,3)$ $u(i,1) = u(i,2)$ $v(i,1) = v(i,2)$ $\phi(i,1) = \phi(i,2)$	$u(nx2,j) = u(nx1,j)$ $v(nx2,j) = v(nx1,j)$ $u(nx3,j) = u(nx2,j)$ $\phi(nx2,j) = \phi(nx1,j)$	$u(2,j) = u(3,j)$ $v(1,j) = v(2,j)$ $u(1,j) = u(2,j)$ $\phi(1,j) = \phi(2,j)$
4. periodic	$v(i,ny2) = v(i,2)$ $u(i,ny2) = u(i,2)$ $v(i,ny3) = v(i,3)$ $\phi(i,ny2) = \phi(i,2)$	$v(i,2) = v(i,ny2)$ $u(i,1) = u(i,ny1)$ $v(i,1) = v(i,ny1)$ $\phi(i,1) = \phi(i,ny1)$	$u(nx2,j) = u(2,j)$ $v(nx2,j) = v(2,j)$ $u(nx3,j) = u(3,j)$ $\phi(nx2,j) = \phi(2,j)$	$u(2,j) = u(nx2,j)$ $v(1,j) = v(nx1,j)$ $u(1,j) = u(nx1,j)$ $\phi(1,j) = \phi(nx1,j)$
5. specified press.	$v(i,ny2) = \max[v(i,ny2), 0]$ $v(i,ny3) = v(i,ny2)$ $u(i,ny2) = u(i,ny1)$ $p(i,ny2) = p_{spec.}$ $\phi(i,ny2) = \phi(i,ny1)$	$v(i,2) = \max[v(i,2), 0]$ $v(i,1) = v(i,2)$ $u(i,1) = u(i,2)$ $p(i,1) = p_{spec.}$ $\phi(i,1) = \phi(i,2)$	N/A	N/A
6. specified u_i	$u(i,ny2) = 0$ $v(i,ny1) = v_{spec.}$ $v(i,ny2) = v(i,ny1)$ $\phi(i,ny2) = \phi(i,ny1)$	$u(i,1) = 0$ $v(i,2) = v_{spec.}$ $v(i,1) = v(i,2)$ $\phi(i,1) = \phi(i,2)$	N/A	N/A

3.4 The Numerical Boundary Conditions

Figures 1 and 2 illustrate the computational domain and the layout of each cell within the entire domain, respectively. The cells go from $i=1$ to $nx2$ in the x direction, and from $j=1$ to $ny2$ in the z direction. Indices $nx3$ and $ny3$ are used in the code to indicate velocity components that lie on the top and right boundaries of the ghost cells for completeness. The boundary conditions are set both by specifying those variables that lie exactly on the boundary (the line dividing the real domain and the ghost cells) and by setting the variables in the ghost cells to appropriate values depending on the boundary condition desired. The boundary conditions given in Section 2.2 above are written explicitly, using the actual FORTRAN indexing in the code, in Table 2. Here, ϕ represents the total density ρ , the density for each species ρ^k , and the specific internal energy U .

3.5 Time Step Constraints

In order to preserve the stability of the solution procedure, there must be time step constraints applied. Although an exact stability analysis cannot be applied to the nonlinear reacting flow problem, simplified model equations can be analyzed to give broad stability guidelines.

In order to prevent fluid moving more than one cell at a time, convective time limits for each direction are applied. That is, the x -direction convective limit is

$$\Delta t < 0.2 \left[\frac{\Delta x_{\min}}{u_{\max}} \right], \quad (47)$$

while the y-direction limit is

$$\Delta t < 0.2 \left[\frac{\Delta y_{\min}}{v_{\max}} \right]. \quad (48)$$

The factor 0.2 is used to prevent the density from becoming negative since it is theoretically possible for fluid to be convecting out each cell wall at the same rate, leaving a 'negative' mass in the cell.

A von Neumann stability analysis gives a limit to the time step size based on the diffusion of momentum and energy. The momentum diffusion time limit is

$$\Delta t < 0.2 \min_{i,j} \left[\frac{\rho}{2\mu_{\text{effective}}} \frac{\Delta x_i^2 \Delta y_j^2}{\Delta x_i^2 + \Delta y_j^2} \right], \quad (49)$$

while the energy diffusion limit is

$$\Delta t < 0.2 \min_{i,j} \left[\frac{\rho C_v}{2k_{\text{effective}}} \frac{\Delta x_i^2 \Delta y_j^2}{\Delta x_i^2 + \Delta y_j^2} \right]. \quad (50)$$

Again, the factor 0.2 has been used and the minima in Equations (49) and (50) are over the entire mesh.

A further time limit based on the chemical reaction rates for the kinetic reactions is applied. This limit restricts the fractional amount of internal energy change due to chemical heat release to 20 percent for each time step in order preserve accuracy.

COYOTE has the option of automatically resetting the time step each cycle (new time step) by choosing the most restrictive of the above limits or allowing the user to set the step to a value that will remain the same throughout the run. The user can run one cycle and allow the code to compute the time step limits to see what the maximum step should be and then use that in order to maintain a constant step for the main run.

4. CODE DESCRIPTION

This section describes the overall structure and flow control of the COYOTE code. Each subroutine presently being used for fluid flow and combustion in the ISV confinement hood is listed and its function described. A flow control tree is presented to show how the flow is controlled between subroutines.

4.1 Overall COYOTE Flow Chart

Table 1 in Section 3.3 lists the procedural steps followed by COYOTE in developing the numerical solutions to the model equations. Figure 3 provides similar information in chart form with additional information concerning the logical flow.

4.2 Flow Control and Description of Subroutines

COYOTE is written in partial subroutine form with each subroutine performing a separate logical task. The main program retains the task, however, of computing the discrete momentum and energy equations and performing the pressure iteration described earlier in Section 3.3 as well as other tasks. A list of the subroutines in COYOTE and their functions follows.

1. bcc

This subroutine sets the boundary conditions for the total density, the individual species densities, the internal energy and pressures for the specified pressure boundary conditions.

2. bcv

This subroutine sets the boundary conditions for the velocity components.

Figure 3. Overall flow diagram for COYOTE reacting flow code.

3. chem

This subroutine computes the changes in species densities and internal energies due to kinetic chemical reactions. (Whether a reaction is kinetic or equilibrium must be set by the user. Relatively fast reactions are usually assumed to be equilibrium, while slow ones must follow the kinetic reaction model).

4. chemeq

This subroutine calculates the changes in the densities and internal energy for equilibrium chemical reactions.

5. chemeqm

This subroutine contains the Meintjes-Morgan quick equilibrium chemistry package.

6. coyote - main program

The main program of COYOTE reads the input deck, computes the explicit approximations for the velocity components, performs the pressure iteration for the ICE method, computes the internal energy, resets the current variables into old time arrays, and computes the SOR relaxation factors. It also writes to the output file, graphics dump and restart dump files, and makes calls to the other subroutines to control the overall algorithm.

7. decomp

This subroutine triangularizes a numerical solution array.

8. difop

This subroutine defines the arrays that specify the geometry and size of the numerical mesh.

9. dumper

This subroutine performs the graphic dump.

10. epilog

This subroutine contains a brief variable dictionary and some equations of state data for several chemicals. Epilog is actually just a comment section.

11. exitldc

This subroutine is used for normal and abnormal exits and indicates where the exit occurred. The routine 'exit' called by exitldc is a system routine.

12. prcon

This subroutine creates contour plots on a line printer.

13. rinput

This block data subprogram contains the original input deck. The code has been changed now such that the input is read in by the main program which means that this subprogram is a default data input deck. (rinput has been left unchanged from the original). Also, rinput contains the JANAF^{6,7} table data for the enthalpies of the species present in the problem. Presently it contains data for methane, O₂, N₂, CO₂, H₂O, H, H₂, O, N, OH, CO, and NO.

14. robhood

This subroutine prevents elements of a dependent variable array that should all be positive (such as the internal energy) from remaining negative if one should happen to go negative during the solution process. Quantities of the dependant variable are 'stolen' from neighboring cells and added to the cell where a negative value has occurred until it becomes positive.

15. setup

This subroutine initializes the run by converting the JANAF table enthalpies to internal energies in the cgs system of units, calls difop, sets up the kinetic and equilibrium reaction data, prints out initial

values, calls the restart subroutine (tapin), and performs other miscellaneous tasks.

16. sgefs

This subroutine decides whether to call decomp to triangularize a matrix or solve to solve it.

17. solve

This subroutine solves a linear system of equations $Ax = b$.

18. species

This subroutine solves the species transport equation.

19. state

This subroutine determines the thermodynamic state based on temperature either by calling the JANAF tables or using an equation of state. Currently, the ideal gas equation of state is used.

20. tapin

This subroutine reads in the restart file.

21. tapout

This subroutine writes the restart file.

22. tcs

This subroutine computes the effective viscosities and thermal conductivities for the mesh.

23. timestp

This subroutine automatically resets the time step if desired by the user based on convection, diffusion and chemistry limits.

COYOTE contains other subroutines which currently are not being used for ISV. They are blowby (allows leakage), lwall (computes the law of the wall for

turbulent flow), markr (computes marker particle paths), swirl (computes a swirl or circumferential velocity component), and turb (contains two turbulence models).

Figure 4 is a flow tree for COYOTE showing how the flow is controlled between subroutines.

Figure 4. COYOTE tree diagram showing how subprograms are called in the code relative to each other. A dashed line indicates an abnormal abort. The dotted line indicates that the subroutine contains only comment information.

5. VERIFICATION TESTS AND SAMPLE PROBLEM

This section reports the results of five verification test runs using COYOTE. The first test is one that is built into the program and is designed to check that the code has successfully been ported to a new facility and/or machine. Its input is built into the block data subprogram 'rinput.' The other four verification tests are designed to check that specific terms in the governing equations have been coded properly in COYOTE. The configuration for a sample ISV hood scenario is also given. Three variations of the problem are detailed. The results for the three cases have been animated and are available on videocassette.

5.1 Test for Successful Porting of Code

A test problem is built into the block data rinput in COYOTE to be run at a new facility after the code is ported to check for problems that may have occurred during the transfer of the code. The problem is basically a one-dimensional shock tube problem with a grid of 192×5 cells. Since there is a layer of fictitious cells around the computational domain, the actual grid is 190×3 cells. Figure 5 is a schematic of the test problem showing initial conditions. There is only one species present: air. The domain is enclosed on four sides by solid (free-slip) walls. An imaginary membrane positioned midway down the tube divides air at one pressure from air at another pressure. The initial temperature is uniform throughout at 340 K. The initial velocity is zero everywhere and the viscosity for the air is computed using the built-in Sutherland formula (see Section 2.1). The state properties for air are determined using the ideal gas law which yields the initial pressures shown in Figure 5 for the given densities and temperature. The initial internal energy is a function of temperature only due to the ideal gas assumption. At time $t = 0$, the membrane is removed and a pressure shock begins to propagate. The test problem is only run for 10^{-4} seconds with intermediate and final results provided in tabular form.

Figure 5. Schematic of the transfer test problem built into COYOTE. The problem is a shock tube with air initially contained on each side of an imaginary membrane at the conditions shown. At time $t=0$, the membrane is removed and a pressure shock wave allowed to propagate.

The results of the test run performed at the INEL using the CRAY XMP/216 have been compared to the tabulated results sent with the copy of COYOTE. The INEL results are found to be the same as those sent with the code to within machine round-off error, indicating that the code has been ported successfully to the INEL system.

5.2 Poiseuille Flow

A standard test flow problem that is designed to check the diffusion terms in the Navier-Stokes equations is Poiseuille flow in a simple geometry. Poiseuille flow is fully-developed confined laminar incompressible flow. For simple geometries, there exist analytical solutions for this type of flow since the troublesome convection terms are zero. This problem also checks the boundary conditions for no-slip walls. The geometry used for this test problem is flow in a straight channel (parallel flat walls that extend to infinity in two directions).

Figure 6 illustrates the flow domain and conditions for the Poiseuille flow test problem. The boundary conditions on the side walls are standard no-slip conditions while the inlet is uniform flow in the streamwise direction and the outflow is specified as continuative. The size of the flow domain is 2.5 cm long by 0.05 cm wide and is divided into a 50 x 10 cell mesh. The hydraulic diameter is defined as one-half the channel width, making the flow domain 100 hydraulic diameters long. This is more than long enough to achieve fully-developed flow. The kinematic viscosity of the air flowing in the channel is given as $0.1568 \text{ cm}^2/\text{sec}$. The inlet flow velocity is 1500 cm/sec, giving the Reynolds number of the flow as $Re = u (1/2 \text{ width})/\nu = 239$, well within the laminar flow range. The Mach number is well within the incompressible range at 0.044. (Experience with the code has indicated that below a Mach number of about 0.01, the solution algorithm becomes unstable and problems are encountered). The density is held constant at $1.177 \times 10^{-3} \text{ g/cm}^3$. Finally, the convective scheme parameters α and β are set to 0.9 and 0.1, respectively.

Figure 6. Schematic of the flow domain for the second test problem: Poiseuille flow.

The results for the streamwise velocity profile across the channel are plotted in Figure 7 and compared with the analytical solution at the same points. As shown, the agreement is very good with only very slight disagreement near the walls. This is due primarily to the fact that the no-slip boundary conditions in COYOTE are low-order accurate. Additional accuracy, however, seems unnecessary for application to the ISV confinement hood.

5.3 Potential Corner Flow

The third test flow problem is designed to test the convection and pressure gradient terms in the Navier-Stokes equations. An analytical solution exists for potential (inviscid, irrotational) flow in a sector¹⁰. A right angle is chosen for the sector, such that flow enters from the top, and exits to the right side, with a wall on the left and at the bottom. Figure 8 illustrates the potential corner flow domain and boundary conditions. Streamlines located to correspond to the numerical simulation at the inflow and outflow boundaries are shown. Since the theoretical flow is inviscid, the wall boundary conditions are set to free-slip. The top inlet and right-side outlet flow conditions are set to the analytical solution values. Since this flow is incompressible, only the pressure gradients are of interest. COYOTE, however, computes and uses the absolute pressure. Therefore, in making comparisons with the true solution, only the pressure gradients are used.

The numerical solution uses a uniform 12 x 12 extended grid domain (10 x 10 actual flow domain). The results for the numerical simulation are tabulated in Tables 3 - 6. Table 3 shows computed results for the u-velocity components. The u-velocities that are located exactly on the outflow boundaries and are set to analytical solution values are those for $i = 12$, for all j . The u-velocities for $j = 12$, for all i , lie in the ghost cells above the inflow boundary and are set to the analytical solution values, also. The computed u-velocities for each i value should be equal to the values at the top of each column ($j = 12$). As can be seen, the largest error is 0.05%.

Figure 7. Streamwise velocity profile for the Poiseuille channel flow near the end of the computational domain.

Figure 8. Schematic of the potential corner flow problem. The left and bottom walls are free-slip while the flow enters from the top and exits to the right. Analytically determined streamlines corresponding to the numerical solution grid are shown.

The results for the v-velocity components are given in Table 4. Here, the values at the inflow boundary, $j = 12$ for all i , are set to the analytical values, as are the v-velocities in the ghost cell column at the outflow boundary, $i = 12$ for all j . The v-velocities in each row should be the same as for the right-most column, $i = 12$. As shown, the largest error is also 0.05%.

Tables 5 and 6 show the percentage errors in the pressure gradients for the horizontal and vertical directions, respectively. As can be seen, the percentage errors are very low except on the right boundary for $\partial p / \partial x$ and at the top boundary for $\partial p / \partial y$. These boundary errors result from the fact that the boundary conditions in COYOTE are not designed to handle a flow where the pressure gradient is changing at the boundary. The corner flow test, nevertheless, does verify that the convective flow terms and pressure gradient terms in the momentum equations are implemented correctly.

Table 3. Values for the u-velocity component for the potential corner flow test problem.

i =	1	2	3	4	5	6	y =
j							
12	0.000E+00	0.000E+00	1.000E+02	2.000E+02	3.000E+02	4.000E+02	1.050E+00
11	0.000E+00	0.000E+00	9.999E+01	2.000E+02	3.000E+02	4.000E+02	9.500E-01
10	0.000E+00	0.000E+00	9.998E+01	2.000E+02	2.999E+02	3.999E+02	8.500E-01
9	0.000E+00	0.000E+00	9.997E+01	1.999E+02	2.999E+02	3.999E+02	7.500E-01
8	0.000E+00	0.000E+00	9.997E+01	1.999E+02	2.999E+02	3.999E+02	6.500E-01
7	0.000E+00	0.000E+00	9.996E+01	1.999E+02	2.999E+02	3.999E+02	5.500E-01
6	0.000E+00	0.000E+00	9.996E+01	1.999E+02	2.999E+02	3.999E+02	4.500E-01
5	0.000E+00	0.000E+00	9.996E+01	1.999E+02	2.999E+02	3.998E+02	3.500E-01
4	0.000E+00	0.000E+00	9.995E+01	1.999E+02	2.999E+02	3.998E+02	2.500E-01
3	0.000E+00	0.000E+00	9.995E+01	1.999E+02	2.999E+02	3.998E+02	1.500E-01
2	0.000E+00	0.000E+00	9.995E+01	1.999E+02	2.999E+02	3.998E+02	5.000E-02
1	1.500E+02	-7.959E-05	9.995E+01	1.999E+02	2.999E+02	3.998E+02	-5.000E-02
x=	-0.100	0.000	0.100	0.200	0.300	0.400	
i =	7	8	9	10	11	12	y =
j							
12	5.000E+02	6.000E+02	7.000E+02	8.000E+02	9.000E+02	1.000E+03	1.050E+00
11	5.000E+02	6.000E+02	7.000E+02	8.000E+02	9.000E+02	1.000E+03	9.500E-01
10	4.999E+02	5.999E+02	6.999E+02	7.999E+02	9.000E+02	1.000E+03	8.500E-01
9	4.999E+02	5.999E+02	6.999E+02	7.999E+02	9.000E+02	1.000E+03	7.500E-01
8	4.999E+02	5.999E+02	6.999E+02	7.999E+02	8.999E+02	1.000E+03	6.500E-01
7	4.998E+02	5.998E+02	6.999E+02	7.999E+02	8.999E+02	1.000E+03	5.500E-01
6	4.998E+02	5.998E+02	6.998E+02	7.999E+02	8.999E+02	1.000E+03	4.500E-01
5	4.998E+02	5.998E+02	6.998E+02	7.999E+02	8.999E+02	1.000E+03	3.500E-01
4	4.998E+02	5.998E+02	6.998E+02	7.998E+02	8.999E+02	1.000E+03	2.500E-01
3	4.998E+02	5.998E+02	6.998E+02	7.998E+02	8.999E+02	1.000E+03	1.500E-01
2	4.998E+02	5.998E+02	6.998E+02	7.998E+02	8.999E+02	1.000E+03	5.000E-02
1	4.998E+02	5.998E+02	6.998E+02	7.998E+02	8.999E+02	1.000E+03	-5.000E-02
x=	0.500	0.600	0.700	0.800	0.900	1.000	

Table 4. Values for the v-velocity component for the potential corner flow test problem.

i =	1	2	3	4	5	6	y =
j							
13	-1.100E+03	-1.100E+03	-1.100E+03	-1.100E+03	-1.100E+03	-1.100E+03	1.100E+00
12	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03	1.000E+00
11	-9.000E+02	-9.000E+02	-9.000E+02	-9.000E+02	-9.000E+02	-9.000E+02	9.000E-01
10	-7.999E+02	-7.999E+02	-7.999E+02	-7.999E+02	-7.999E+02	-7.999E+02	8.000E-01
9	-6.999E+02	-6.999E+02	-6.999E+02	-6.999E+02	-6.999E+02	-6.999E+02	7.000E-01
8	-5.998E+02	-5.998E+02	-5.998E+02	-5.998E+02	-5.999E+02	-5.999E+02	6.000E-01
7	-4.998E+02	-4.998E+02	-4.998E+02	-4.998E+02	-4.998E+02	-4.999E+02	5.000E-01
6	-3.998E+02	-3.998E+02	-3.998E+02	-3.999E+02	-3.999E+02	-3.999E+02	4.000E-01
5	-2.999E+02	-2.999E+02	-2.999E+02	-2.999E+02	-2.999E+02	-2.999E+02	3.000E-01
4	-1.999E+02	-1.999E+02	-1.999E+02	-1.999E+02	-1.999E+02	-1.999E+02	2.000E-01
3	-9.995E+01	-9.995E+01	-9.995E+01	-9.995E+01	-9.996E+01	-9.996E+01	1.000E-01
2	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	6.000E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.000E-01
x=	-0.050	0.050	0.150	0.250	0.350	0.450	
i =	7	8	9	10	11	12	y =
j							
13	-1.100E+03	-1.100E+03	-1.100E+03	-1.100E+03	-1.100E+03	-1.100E+03	1.100E+00
12	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03	1.000E+00
11	-9.000E+02	-9.000E+02	-9.000E+02	-9.001E+02	-9.001E+02	-9.000E+02	9.000E-01
10	-7.999E+02	-8.000E+02	-8.000E+02	-8.000E+02	-8.000E+02	-8.000E+02	8.000E-01
9	-6.999E+02	-6.999E+02	-6.999E+02	-7.000E+02	-7.000E+02	-7.000E+02	7.000E-01
8	-5.999E+02	-5.999E+02	-5.999E+02	-5.999E+02	-6.000E+02	-6.000E+02	6.000E-01
7	-4.999E+02	-4.999E+02	-4.999E+02	-4.999E+02	-5.000E+02	-5.000E+02	5.000E-01
6	-3.999E+02	-3.999E+02	-3.999E+02	-3.999E+02	-4.000E+02	-4.000E+02	4.000E-01
5	-2.999E+02	-2.999E+02	-2.999E+02	-3.000E+02	-3.000E+02	-3.000E+02	3.000E-01
4	-1.999E+02	-1.999E+02	-2.000E+02	-2.000E+02	-2.000E+02	-2.000E+02	2.000E-01
3	-9.996E+01	-9.997E+01	-9.997E+01	-9.998E+01	-9.999E+01	-1.000E+02	1.000E-01
2	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.000E-01
x=	0.550	0.650	0.750	0.850	0.950	1.050	

Table 5. Percentage error in the horizontal pressure gradient, dp/dx , for the potential corner flow test problem.

i -	3	4	5	6	7	8	y -
j							
12	1.024E+00	1.228E+00	1.290E+00	1.313E+00	1.317E+00	1.310E+00	1.050E+00
11	7.279E-02	6.896E-02	6.032E-02	4.719E-02	2.988E-02	9.014E-03	9.500E-01
10	7.367E-02	6.904E-02	5.984E-02	4.613E-02	2.795E-02	5.168E-03	8.500E-01
9	7.519E-02	7.026E-02	6.071E-02	4.657E-02	2.783E-02	4.483E-03	7.500E-01
8	7.839E-02	7.328E-02	6.348E-02	4.898E-02	2.979E-02	5.958E-03	6.500E-01
7	8.251E-02	7.726E-02	6.724E-02	5.241E-02	3.278E-02	8.361E-03	5.500E-01
6	8.687E-02	8.150E-02	7.126E-02	5.614E-02	3.609E-02	1.107E-02	4.500E-01
5	9.099E-02	8.547E-02	7.504E-02	5.965E-02	3.923E-02	1.369E-02	3.500E-01
4	9.446E-02	8.880E-02	7.820E-02	6.258E-02	4.186E-02	1.591E-02	2.500E-01
3	9.699E-02	9.122E-02	8.047E-02	6.468E-02	4.375E-02	1.751E-02	1.500E-01
x=	0.100	0.200	0.300	0.400	0.500	0.600	

i -	9	10	11	y -
j				
12	1.271E+00	3.393E+00	-1.520E+02	1.050E+00
11	-1.388E-02	-3.719E-02	-1.563E+02	9.500E-01
10	-2.327E-02	-6.414E-02	-4.694E+01	8.500E-01
9	-2.354E-02	-5.674E-02	-4.689E+01	7.500E-01
8	-2.243E-02	-5.605E-02	-4.689E+01	6.500E-01
7	-2.084E-02	-5.598E-02	-4.688E+01	5.500E-01
6	-1.895E-02	-5.542E-02	-4.687E+01	4.500E-01
5	-1.705E-02	-5.458E-02	-4.686E+01	3.500E-01
4	-1.542E-02	-5.377E-02	-4.686E+01	2.500E-01
3	-1.424E-02	-5.321E-02	-4.686E+01	1.500E-01
x=	0.700	0.800	0.900	

Table 6. Percentage error in the vertical pressure gradient, dp/dy , for the potential corner flow test problem.

i -	3	4	5	6	7	8	y -
j							
11	-2.679E+01	-2.679E+01	-2.680E+01	-2.680E+01	-2.680E+01	-2.680E+01	9.000E-01
10	-1.032E-01	-1.036E-01	-1.039E-01	-1.041E-01	-1.041E-01	-1.036E-01	8.000E-01
9	-4.904E-02	-4.991E-02	-5.110E-02	-5.249E-02	-5.390E-02	-5.517E-02	7.000E-01
8	-1.211E-02	-1.344E-02	-1.532E-02	-1.761E-02	-2.011E-02	-2.252E-02	6.000E-01
7	1.868E-02	1.698E-02	1.457E-02	1.158E-02	8.276E-03	5.020E-03	5.000E-01
6	4.406E-02	4.207E-02	3.924E-02	3.573E-02	3.181E-02	2.789E-02	4.000E-01
5	6.424E-02	6.202E-02	5.886E-02	5.495E-02	5.056E-02	4.613E-02	3.000E-01
4	7.934E-02	7.692E-02	7.351E-02	6.930E-02	6.457E-02	5.977E-02	2.000E-01
3	8.941E-02	8.685E-02	8.326E-02	7.883E-02	7.388E-02	6.886E-02	1.000E-01
x=	0.150	0.250	0.350	0.450	0.550	0.650	

i -	9	10	11	12	y -
j					
11	-2.680E+01	-2.678E+01	-1.361E+02	-1.216E+02	9.000E-01
10	-1.034E-01	-1.108E-01	-1.505E-01	-7.937E+00	8.000E-01
9	-5.629E-02	-5.707E-02	-5.344E-02	-1.853E+00	7.000E-01
8	-2.437E-02	-2.447E-02	-2.481E-02	-1.543E+00	6.000E-01
7	2.382E-03	1.491E-03	-1.204E-03	-1.520E+00	5.000E-01
6	2.457E-02	2.291E-02	1.927E-02	-1.520E+00	4.000E-01
5	4.232E-02	4.020E-02	3.643E-02	-1.537E+00	3.000E-01
4	5.565E-02	5.343E-02	5.024E-02	-1.591E+00	2.000E-01
3	6.456E-02	6.264E-02	6.120E-02	-1.520E+00	1.000E-01
x=	0.750	0.850	0.950	1.050	

5.4 Unsteady Heat Conduction

This test problem is designed to check the diffusion terms in the energy equation. The test problem is a two-dimensional, unsteady, initial and boundary value heat conduction problem that has an analytical solution. The problem domain, shown in Figure 9, is a rectangular plate with uniform, isotropic properties. An initial distribution of temperature is imposed on the plate, which then decays with time as the energy conducts out from the plate. The describing equation for the problem is given by

$$\frac{1}{\alpha} \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \quad (51)$$

where the initial condition is $T(x,y,0)=xy$ for $0 < x < a$ and $0 < y < b$ and the boundaries are held constant at $T_w=300K$. The analytical solution is given in terms of a double Fourier series as¹¹

$$T(x,y,t) = T_w + \frac{4ab}{\pi} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\cos(m\pi) \cos(n\pi)}{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \exp(-\lambda_{mn}^2 \alpha t) \quad (52)$$

where the eigenvalues are given by

$$\lambda_{mn}^2 = \left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2. \quad (53)$$

The computational mesh is 10 x 20 cells within the 0.1 cm by 0.2 cm domain. The convection, pressure-divergence and dissipation terms in the energy equation (19) are made zero by setting the velocities and pressure to zero. Since there is only one specie in the problem, the chemistry source and diffusion terms are zero, as well. A time step of 1×10^{-5} seconds is used to ensure stability. The results for temperature as a function of x for a constant value of y ($y=b/2$) are plotted for various values of time in Figures 10-16. (Figure 10 gives the initial condition for the problem). As can be seen, agreement between the exact and numerical solutions improves with increasing time values. The discrepancies at $t=0.001$ and 0.002 seconds are due to the truncation of the Fourier series present in the 'exact' solution.

✓ The truncation has a larger effect for smaller values of t , as expected. The energy diffusion terms thus appear to be correctly coded and functioning properly.

Figure 9. Schematic of the domain for the unsteady conduction test problem.

Figure 10. Initial distribution of temperature as a function of x for $y=b/2$ for the unsteady conduction test problem.

Figure 11. Distribution of temperature as a function of x for $y=b/2$ at time $t=0.001$ seconds for the unsteady conduction test problem.

Figure 12. Distribution of temperature as a function of x for $y=b/2$ at time $t=0.002$ seconds for the unsteady conduction test problem.

ENERGY EQUATION TEST EXACT vs. CALCULATED

Figure 13. Distribution of temperature as a function of x for $y=b/2$ at time $t=0.003$ seconds for the unsteady conduction test problem.

ENERGY EQUATION TEST EXACT .vs. CALCULATED

Figure 14. Distribution of temperature as a function of x for $y=b/2$ at time $t=0.004$ seconds for the unsteady conduction test problem.

ENERGY EQUATION TEST EXACT .vs. CALCULATED

Figure 15. Distribution of temperature as a function of x for $y=b/2$ at time $t=0.005$ seconds for the unsteady conduction test problem.

5.5 Kinetic Chemistry Reaction

The purpose of the kinetic chemistry reaction test problem is to check the kinetic chemistry model in COYOTE. Choosing both forward and backward reaction rate coefficients k_{fr} and k_{br} , analytical solutions are easily obtainable for the transient dissociation only and recombination only reactions for a two-atom molecule. The analytical solution for the dissociation reaction for the molecule is given by

$$[A](t) = [A]_0 \exp(-k_{fr}t) \quad (54)$$

where $[A]$ is the unsteady molecular concentration, $[A]_0$ is the initial concentration of the biatomic molecule and t is time. The COYOTE results for the time varying species density of the molecule is plotted in Figure 16 against the analytical solution for the first 0.01 seconds. The agreement is seen to be excellent. The results for the recombination reaction (not shown) indicate similar agreement.

Figure 16. Numerical results for COYOTE compared to the analytical solution for the transient molecular concentration for the kinetic dissociation reaction of a molecule.

5.6 Simulation of Flow and Combustion in the ISV Hood

Numerical simulations of a possible transient inflow of off-gases into the ISV confinement hood with combustion have been made. Interpretation of the results of such simulations, which consist of thousands of lines of numerical data output, is greatly facilitated by generating animation of the transient. Such an animation has been created for each of three numerical simulations and each has been recorded on VHS videocassette. The three simulations represent a single hood geometry, but varying inlet air supply rate. This section describes the configurations of the problem portrayed on the videocassette.

A schematic of the problem with its associated boundary conditions is illustrated in Figure 17. The cylindrical axisymmetric geometry is chosen since COYOTE is presently limited to two-dimensional rectangular or cylindrical coordinates. The axisymmetric case chosen allows appropriate representation of a concentric pipe configuration for the air supply and exhaust ports such as is used for the intermediate-scale ISV tests. The place chosen for the point of off-gas inflow from the ground corresponds to an electrode position for an electrode separation of 4 meters. It has been observed that off-gases frequently enter the hood from near the electrodes.

The conditions chosen for the solid boundaries are free-slip for the momentum equations and constant temperature for the energy equation. Free-slip is chosen since it is believed that the wall boundary layers have little effect on the flow in the hood. The choice of constant wall temperature reflects the limited boundary conditions presently available in COYOTE. While the energy boundary conditions are not appropriate for simulating hood wall temperatures, they seem adequate for examining the flow and combustion of the off-gases in the hood.

The three flows simulated are transient, compressible, laminar and reactive. All simulations apply ignition near the entrance point of the off-

Figure 17. Schematic of the base problem for the three animations recorded on videocassette. The problem is an axisymmetric large scale ISV confinement hood with off-gases entering from a localized region and concentric supply and exhaust ports located at the top center.

gases. The off-gases consist of 50% H₂ and 50% CO by mole fraction. The chemical reactions considered are the following

The boundary temperatures for the combustion case are set to 2000K while the off-gases are set to 1000K at the inflow point. The pressure of the exhaust port is held at one atmosphere for all cases. The influx of off-gases is based on an ISV engineering design study¹² and is given as 416 SCFM for each case. The three cases correspond to three values used for the inlet air flow required for the complete combustion of the off-gases. They are

Case 1: 1000 CFM (at 1 atm and 300K)

Case 2: 3000 CFM (at 1 atm and 300K)

Case 3: 8000 CFM (at 1 atm and 300K).

The conclusions drawn from this study are the following:

1. Combustion occurs along the melt surface,
2. The air inflow forms a recirculating zone between the off-gas inlet and the hood wall,
3. Complete combustion occurs in each case, and
4. Different hood geometries and configurations must be tested for different conditions to find optimal or suitable designs.

6. CONCLUSIONS

The model equations that represent the mass, momentum and energy transfer and chemical reactions in a reacting fluid flow have been presented. The methods used in numerically discretizing the differential model equations as they are employed in the COYOTE reacting flow code are detailed. The solution method called ICE that is used in COYOTE is outlined. Finally, the results of five test problems and one sample ISV hood problem are reported. The conclusions reached regarding the test problems are:

1. the COYOTE code has successfully been ported to the INEL CRAY XMP,
2. the axial momentum diffusion, the axial and radial convection and pressure gradient terms, the energy diffusion and the chemical kinetic model are correctly programmed into the code, and
3. COYOTE meets the requirements that have been identified so far, except for modeling of the explosion potential, for simulating flow and combustion in the ISV confinement hood.

In view of the above conclusions and since a number of enhancements have been made to the version of COYOTE that was acquired by the INEL for use for ISV analysis, it will be known henceforth as COYOTE-ISV.

7. REFERENCES

1. MacKinnon, R. J. et al, "Modeling Requirements for In Situ Vitrification," EGG-WM-9035, July, 1990.
2. MacKinnon, R. J. et al, "In Situ Vitrification Model Development and Implementation Plan," (Draft), March 23, 1990.
3. Cloutman, L. D., "COYOTE: A Computer Program for 2-D Reactive Flow Simulations," LLNL Report UCRL-ID-103611, April, 1990.
4. Amsden, A. A. , O'Rourke, P. J., and Butler, T. D., "KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays," LANL Report LA-11560-MS, UC-96, May, 1989.
5. Cloutman, L. D. , Hirt, C. W. , and Romero, N. C., "SOLA-ICE: A Numerical Solution Algorithm for Transient Compressible Fluid Flows," LASL Report LA-6239, UC-34, July, 1976.
6. Stull, D.R. and Prophet, H., JANAF Thermochemical Tables, 2nd ed., U.S. Department of Commerce/National Bureau of Standards, NSRDS-NBS 37, June, 1971.
7. Chase, M.W., Curnutt, J.L., Hu, A.T., Prophet, H., Syverud, A.N., and Walker, L.C., 'JANAF thermochemical table, 1974 supplement', J. Phys. Chem., Ref., Data, 3, 311, 1974.
8. Williams, F. A., Combustion Theory, 2nd Ed., Addison-Wesley Publishing Company, Inc., Redwood City, CA, 1988.
9. Ramshaw, J. D., private communication, 1988.
10. Currie, I. G., Fundamentals of Fluid Mechanics, McGraw-Hill Book Company, New York, 76-78, 1974.
11. Powers, D. L., Boundary Value Problems, 2nd ed., Academic Press, New York, 1979.
12. Slater, C. E., Johnson, R. W., and MacKinnon, R. J., "Off-Gas Release Rate Calculations for the Large Scale Test in Pit 9," Engineering Design File, EDF Serial Number ERP-ISV-038, Idaho National Engineering Laboratory, July, 1990.

END

DATE FILMED

02 / 26 / 91