ED21/G. Wilhold

CFD Research Corporation

3325 Triana Blvd. ● Huntsville, Alabama 35805 ● Tel: (205) 536-6576 ● FAX: (205) 536-6590

CFDRC

COMBUSTION CHAMBER ANALYSIS CODE

Final Report

by

A.J. Przekwas, Y.G. Lai, A. Krishnan, R.K. Avva, and M.G. Giridharan

May 1993

CFDRC Report: 4105/6

for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Alabama 35812

Contract Number: NAS8-37824

NASA CORs: G. Wilhold and K. Tucker

PREFACE

Under the NASA MSFC contract NAS8-37824, CFDRC has developed an advanced CFD code: REFLEQS (REactive FLow EQuation Solver). REFLEQS has been designed for turbulent flow and heat transfer problems with and without chemical reactions. Particular attention has been given to the needs of predicting flow environment and efficiency of combustion in liquid rocket engines, using LOX/H_2 or LOX/Hydrocarbon propellant systems. The results of this study are described in the following three volumes:

1: Final Report;
2: REFLEQS-Validation and Applications Manual; and
3: REFLEQS - User's Manual.

The final report includes descriptions of the mathematical basis of REFLEQS.

The authors wish to acknowledge and thank all those who have contributed to this work. In particular, thanks are due to:

- Mr. Kevin Tucker and Dr. Paul McConnaughey of NASA MSFC/ED32 for their guidance and constructive suggestions;
- Robert Williams, Robert Garcia, Joe Ruf and Ted Benjamin of MSFC/ED32, and Klaus Gross, John Hutt and Ho Trinh of MSFC/EP55 for their feedback on code performance;
- Mr. Jack Hyde of Aerojet Corporation for his feedback on code performance;
- Dr. Ashok K. Singhal (President & Technical Director of CFDRC) for overall guidance in code development and project work;
- Mr. Clifford E. Smith, Milind V. Talpallikar, and Mark L. Ratcliff of CFDRC for testing and validating various versions of the code; and
- Ms. Jennifer L. Swann for preparing the typescripts of the reports.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

1. INTRODUCTION

1.1 Background

Until recently, liquid rocket thrust chambers were designed using methods of characteristics and boundary layer approximations. In the mid 1980's, new prediction methodologies, based on solutions of Navier-Stokes equations were proposed by Liang (1986), and Przekwas *et al* (1984, 1986). Computational Fluid Dynamics (CFD) found its way in practical rocket engine design due to several initiatives coordinated by NASA Marshall Space Flight Center and annually reported in the Proceedings of "Computational Fluid Dynamics Workshops".

This report presents the results of the three-year effort (1989-1992), sponsored by NASA MSFC, to develop an advanced CFD code for predicting two-phase, reactive flows in liquid rocket thrust chambers. The starting point for this project was the existing REFLEQS code of CFDRC.

1.2 Project Objectives and Approach

The overall objective of this project was to develop a 3D CFD code that could be used as a tool in designing and predicting flow environment in LOX/H_2 and LOX/Hydrocarbon liquid propellant thrust chambers. The code should be capable of performing steady-state and transient analysis of single-phase and two-phase reactive flows on non-orthogonal grids.

A building block approach was to be used in which the most advanced, methodology would be first implemented in the 2D code, tested and then extended to the 3D code. Code testing, verification/validation and documentation efforts were planned to be performed concurrently.

Three major activities were planned for this project: development of the new generation REFLEQS code, code tests, and validation, and computational studies supporting STME engine development.

The starting version of the REFLEQS code utilized staggered grid arrangement, first order numerical schemes, interactive SIMPLEC algorithm and weak form of conservation equations (see Yang *et al*, 1992). The planned direction and selected code modifications for this project included the following:

a. employ accurate differencing schemes (second-order accurate);
b. incorporate efficient solution algorithms;
c. incorporate options for Cartesian, cylindrical, and Body-Fitted Coordinate (BFC) systems;
d. incorporate advanced turbulence models with boundary layer treatment;
e. develop Eulerian-Lagrangian spray dynamics and combustion models, accounting for turbulence modulation effects for evaporating systems;
f. incorporate available finite-rate reduced mechanism combustion models;
g. incorporate turbulence-combustion interaction models; and
h. include thermal radiation model.

Early in the project, it was decided to incorporate state-of-the-art numerical methodology by implementing non-staggered grid arrangement, higher-order TVD-type numerical schemes, non-iterative PISO solution procedure, and strong form of conservation equations expressed in terms of Cartesian velocity components.

Several new methodologies prior to this project had not been thoroughly tested, (*e.g.* PISO algorithm for compressible flows, Hautmann 4-step reduced chemistry for fuel rich conditions, *etc*.). As a result, fundamental development, testing, and validation efforts were directed to the development of new, robust fluid dynamic solvers. As a result, most code subroutines are completely reprogrammed.

Parallel to the code development, active, continuous support was provided for analyzing flows and heat transfer problems during the STME development effort, *e.g.*, NLS manifold study, STME nozzle cooling, vehicle base heating, *etc*.

1.3 Salient Features of the Code

This section outlines the salient features of the REFLEQS code, specifically designed for solving turbulent flow and heat transfer problems with and without chemical reaction. It consists of two fully compatible codes: REFLEQS-2D and REFLEQS-3D, and a common preprocessor, REFLEQSP.

The codes are written in modular form, using FORTRAN 77 language. The provided pre-processor allows the user to specify the problem in a compact input file without modifying the main code. Several users can be supported with a single version of the code. Geometric flexibility implemented into the code allows the user to simulate flows in complex geometric configurations, multiple flow inlets and exits, and with internal solid objects. Selected capabilities, techniques, and physical models are summarized below.

1.3.1 Numerical Techniques

The following are numerical techniques of the REFLEQS code.

a. Finite-volume approach of solving Favre-averaged Navier-Stokes equations.
b. Cartesian, polar, and non-orthogonal Body-Fitted Coordinates (BFC).
c. Colocated (non-staggered) grid.
d. Strong conservative form of momentum equations with Cartesian components as dependent variables.
e. Stationary or rotating coordinate systems.
f. Pressure-based methodology for incompressible and compressible flows.
g. Blockage technique for flows with internal solid objects.
h. Four differencing schemes: upwind, central (with damping terms), MUSCL, and third-order Osher-Chakravarthy.
i. Steady-state and transient (second-order time-accurate) analysis capability.
j. Fully implicit and conservative formulation.
k. Pressure-based solution algorithms, including SIMPLE, a variant of SIMPLEC, and non-iterative PISO algorithm for all flow speeds.

l. Symmetric Whole-Field linear equation solver and Conjugate Gradient Squared solver.

m. Multi-component flows with heat and mass transfer.

n. Eulerian-Lagrangian approach for two-phase flows.

1.3.2 Physical Models

The following are the physical models for the REFLEQS code.

a. JANNAF Property Tables for gaseous species.

b. Equation of State for arbitrary composition of ideal gas.

c. Four turbulence models:

  1. Standard k-ε model (Launder and Spalding, 1974);
  2. Extended k-ε model (Y.S. Chen and Kim, 1987);
  3. Multiple Time Scale Model (Kim and C.P. Chen, 1988); and
  4. Low-Reynolds Number k-ε Model (Chien, 1982); this provides the capability of **resolving boundary layers.**

d. Standard and extended wall functions.

e. Instantaneous reaction (diffusion controlled combustion) model.

f. Equilibrium chemistry model (accelerated JANNAF method).

g. One step finite rate chemistry models and two-step reduced mechanism models for O_2/H_2 and O_2/Hydrocarbon propellants.

h. Evaporating reactive spray dynamics model.

i. Discrete ordinate thermal radiation model.

1.3.3 Code Organization

Figure 1-1 illustrates the overall flowchart of the code. A common preprocessor (developed under CFDRC's R&D funds) supports both 2D and 3D codes. It provides the flexibility of restarting problems, reading grids from external files, (*e.g.*, EAGLE format or PLOT3D format), screening user's input file for simple errors and incorporates some degree of intelligence by analyzing input consistency. Results can be processed with NASA's PLOT3D or FAST graphical postprocessors, or more conveniently with CFDRC's CFD-VIEW scientific data visualization package.

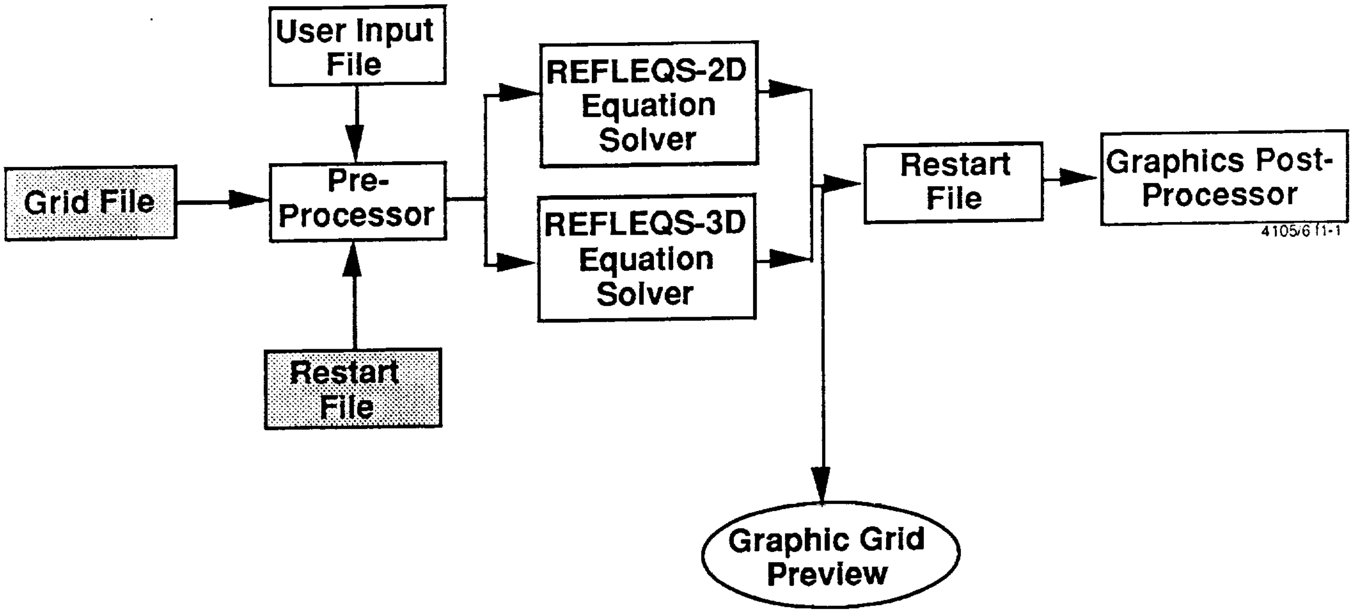


Figure 1-1. REFLEQS Code Organization

1.3.4 Code Testing

The REFLEQS code has been checked out for a large number of problems by using the following four-step validation procedure:

a. Check-out problems (with known solutions);
b. Benchmark problems (with benchmark quality data);
c. Validation problems (with detailed experiment data); and
d. Field problems (with global performance data).

Results of the computational studies can be found in the volume entitled, "REFLEQS - Validation and Applications Manual".

1.4 Outline of the Report

Section 2 provides a general description of the governing partial differential equations embodied in the REFLEQS codes. Section 3 describes the physical models available.

The geometry and computational grid topologies are explained in Section 4 while the control volume discretization techniques are discussed in Section 5. Details including the similarities and differences of the solution algorithms, SIMPLE, SIMPLEC, and PISO, are described in Section 6.

A separate section (Section 7) is dedicated to boundary conditions, providing detailed explanation for each boundary condition type available to the user. Section 8 presents the overall solution procedure and linear equation solvers used in the code. Conclusions are presented in Section 9 and recommendations for further development and applications of REFLEQS are given in Section 10.

2. BASIC GOVERNING EQUATIONS

In this section, the basic governing equations for single-phase fluid flow are presented. These equations are derived from the conservation laws of mass, momentum, and energy which can be found in most fluid mechanics textbooks. Since a conservative finite-volume method is used in REFLEQS, all the governing equations are expressed in conservative forms.

In Cartesian tensor form, the mass conservation and momentum conservation can be expressed as

$$\begin{gathered}\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial x_j}\left(\rho\, u_j\right)=0 \\ \frac{\partial \rho u_i}{\partial t}+\frac{\partial}{\partial x_j}\left(\rho u_j u_i\right)=-\frac{\partial p}{\partial x_i}+\frac{\partial \tau_{ij}}{\partial x_j}+\rho f_i\end{gathered} \tag{2.1}$$

where u_i is the *ith* Cartesian component of the instantaneous velocity, ρ is the fluid density, p is the instantaneous static pressure, τ_{ij} is the viscous stress tensor, f_i is the body force. For the Newtonian fluid, τ_{ij} can be related to the velocity through

$$\tau_{ij}=\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{2}{3}\,\mu\left(\frac{\partial u_k}{\partial x_k}\right)\delta_{ij} \tag{2.2}$$

where μ is the fluid dynamic viscosity and δ_{ij} is the Kronecker delta.

The energy equation can take several forms and different forms are good for different classes of problems. Two forms are used in the REFLEQS code: static enthalpy; and stagnation enthalpy. The static enthalpy form of the energy equation can be expressed as:

$$\frac{\partial \rho h}{\partial t}+\frac{\partial}{\partial x_j}\left(\rho u_j h\right)=-\frac{\partial q_j}{\partial x_j}+\frac{\partial p}{\partial t}+u_j\frac{\partial p}{\partial x_j}-\tau_{ij}\frac{\partial u_i}{\partial x_j} \tag{2.3}$$

Note that the above equation is not strictly conservative by its nature and is recommended for use for incompressible and low Mach number flows. The total enthalpy form of the energy equation, on the other hand, is fully conservative and is recommended for high speed compressible flows. The total enthalpy H, defined as $H = h + v^2/2$, is governed by the following equation

$$\frac{\partial \rho H}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho\, u_j H\right] = -\frac{\partial q_j}{\partial x_j} + \frac{\partial p}{\partial t} + \frac{\partial}{\partial x_i}\left(\tau_{ij} u_j\right) \tag{2.4}$$

In Equations (2.3) and (2.4), q_j is the j-component of the heat flux. By using Fourier Law, the heat flux is calculated by

$$q_j = -k\frac{\partial T}{\partial x_j} \tag{2.5}$$

where k is the thermal conductivity.

The species conservation equation is written as:

$$\frac{\partial \rho Y_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho u_j Y_i\right] = \frac{\partial}{\partial x_j}\left(\rho D \frac{\partial Y_i}{\partial x_j}\right) + \dot{W} \tag{2.6}$$

where Y_i is the mass fraction of species i, D is the mass diffusivity, and $\dot{W}$ represents source/sink terms due to chemical reactions.

For turbulent flows, the diffusion terms in the above equations are replaced by the effective diffusion due to turbulence. Section 3 discusses the calculation of the effective diffusion (or eddy diffusivities) for various turbulence models.

3. PHYSICAL MODELS

3.1 Turbulence Models

In this section, the theoretical framework behind turbulence modeling is outlined followed by a detailed description of the various turbulence models available in the REFLEQS code. Section 3.1.1 introduces Favre time-averaging and the Boussinesq eddy viscosity concept. Section 3.1.2-3.1.7 describe the turbulence models implemented in the REFLEQS code.

3.1.1 Favre Averaged Navier-Stokes Equations

The basic governing fluid dynamics equations have already been introduced in Section 2. These equations are, in general, applicable to Newtonian fluid flow under steady or transient, incompressible or compressible, laminar, transitional or turbulent conditions. The nonlinearity of the Navier-Stokes equations, coupled with the complexity of the boundary conditions, makes it impossible to obtain analytical solutions for all but a limited number of flows of engineering interest. Hence one is forced to resort to approximate or numerical methods. Even though a wide variety of numerical techniques can be applied to solve the Navier-Stokes equations for laminar flows, Direct Numerical Simulation (DNS) of turbulent flows is feasible only at very low Reynolds numbers. Turbulent flows are inherently unsteady and they contain a wide range of time and length scales, and resolution of these scales requires very short time steps and fine grids. The CPU and memory requirements for direct simulations are too large even for the fastest and largest present day computers.

As most engineering applications only require time-mean quantities, the Navier-Stokes equations are usually averaged over time or ensemble of statistically equivalent flows to yield averaged equations. In the averaging process, a flow quantity ϕ is decomposed in to mean and fluctuating parts. The following two types of averaging are generally used.

Reynolds (or time) Averaging: $\phi = \bar{\phi} + \phi'$ where $\phi = (1/T)\int_{t_o}^{t_o+T} \phi \, dt$

Favre (or density) Averaging: $\phi = \tilde{\phi} + \phi''$ where $\tilde{\phi} = \overline{\rho\phi}/\bar{\rho}$

Note that overbar denotes Reynolds averaging while tilde denotes Favre averaging. T should be large compared the fluctuation time scale so that mean quantities are stationary over a number of samples. It also must be borne in mind that the mean quantities can vary in time on a scale much larger than T.

Applying the Favre averaging procedure to Navier-Stokes equations (Equation 2.1), one obtains, the Favre-averaged Navier-Stokes (FANS) equations given below. (For detailed derivation, see Cebeci and Smith, 1974.)

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{\rho}\, \tilde{u}_j\right) = 0$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\, \tilde{u}_j\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\, \tilde{u}_i \tilde{u}_j\right) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\bar{\mu}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \tilde{u}_m}{\partial x_m}\delta_{ij}\right)\right] + \frac{\partial}{\partial x_j}\left(-\bar{\rho}\, \widetilde{u_i'' u_j''}\right) \qquad (3.1)$$

The FANS equations contain less information than the full NS equations, but have additional unknown terms $-\bar{\rho}\, \widetilde{u_i'' u_j''}$ called the Reynolds stresses. These correlations between the fluctuating components arise in the averaging process, and need to be modeled to achieve closure of the FANS equations.

All the turbulence models available in REFLEQS employ the generalized Boussinesq eddy viscosity concept in which the Reynolds stress $-\bar{\rho}\, \widetilde{u_i'' u_j''}$ is treated as a linear function of the mean strain rate

$$-\bar{\rho}\, \widetilde{u_i'' u_j''} = \mu_t\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial xi} - \frac{2}{3}\frac{\partial \tilde{u}_m}{\partial x_m}\delta_{ij}\right) - \frac{2}{3}\bar{\rho} k\, \delta_{ij} \qquad (3.2)$$

Here k is half the trace of the Reynolds stress tensor and μ_t is the turbulent eddy viscosity. Following the kinetic theory of gases, the eddy viscosity is modeled as the product of a velocity scale q and a length scale ℓ.

$$\mu_t = C \bar{\rho} q \ell \tag{3.3}$$

where C is a constant of proportionality. Various turbulence models differ in the way q and ℓ are estimated. In the following description of models, the overbar form μ and ρ, and tilde for $u, v, etc.$ are dropped for convenience.

3.1.2 Baldwin-Lomax Model

This belongs to the class of algebraic turbulence models because the velocity and length scales are obtained from algebraic relations. It is also commonly referred to as a mixing-length model because it employs Prandtl's mixing-length hypothesis in modeling length and velocity scales.

Baldwin and Lomax (1978) developed this model primarily for wall-bounded flows. Like the mixing-length model of Cebeci and Smith (1974), it employs different expressions for μ_t in the inner and outer parts of the boundary layer.

$$\mu_t = \begin{Bmatrix} \mu_{t\,inner} & for\ y \le y_{crossover} \\ \mu_{t\,outer} & for\ y \ge y_{crossover} \end{Bmatrix} \tag{3.4}$$

In the inner layer, Prandtl's mixing-length model and the Van Driest's damping function are used to estimate the length scale,

$$\ell = \kappa y \left[1 - exp\left(-y^+ / A^+\right)\right] \tag{3.5}$$

where A^+ is the Van Driest's damping constant. y^+, the distance from the wall in wall units, is defined as

$$y^+ = y u_\tau / \nu, \quad u_\tau = \sqrt{\tau_w / \rho} \tag{3.6}$$

In the preceding expression u_τ is commonly known as the friction velocity with τ_w being the shear stress at the wall.

The velocity scale q is modeled as the product of ℓ and the root mean square vorticity,

$$|\omega| = \left[\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\right)^2\right]^{1/2} \tag{3.7}$$

Using the preceding expressions, the eddy viscosity in the inner layer is obtained as

$$\mu_{t\,inner} = \rho\, \ell^2 |\omega| \tag{3.8}$$

The outer layer eddy viscosity is determined from the following expression:

$$\mu_{t\,outer} = K\, C_{cp}\, \rho\, F_{wake} F_{kleb}(y) \tag{3.9}$$

where K is the Clauser constant, C_{cp} is an additional constant, and

$$F_{wake} = min\left\{y_{max} F_{max},\, C_{wk}\, y_{max} \frac{U_{dif}^2}{F_{max}}\right\} \tag{3.10}$$

The quantities y_{max} and F_{max} are determined from the function

$$F(y) = y|\omega|\left[1 - exp\left(-y^+/A^+\right)\right] \tag{3.11}$$

The quantity F_{max} is the maximum value of $F(y)$ that occurs within the boundary layer and y_{max} is the value of y at which the maximum occurs.

$F_{kleb}(y)$ is the Klebanoff intermittency factor given by

$$F_{kleb}(y) = \left[1 + 5.5\left(\frac{C_{kleb}y}{y_{max}}\right)^6\right]^{-1} \qquad (3.12)$$

The quantity U_{dif} is the difference between the maximum and minimum total velocity in the boundary layer (*i.e.*, at a fixed x station)

$$U_{dif} = \sqrt{\left(u^2 + v^2 + w^2\right)_{max}} - \sqrt{\left(u^2 + v^2 + w^2\right)_{min}} \qquad (3.13)$$

The second term in U_{dif} is zero for stationary walls.

The values used for the constants appearing in the preceding expressions are

$$A^+ = 26 \quad C_{cp} = 1.6 \quad C_{kleb} = 0.3 \quad C_{wk} = 0.25 \quad \kappa = 0.4 \quad K = 0.0168$$

3.1.3 Standard k-ε Model

Several versions of k-ε models are in use today, but the model employed in REFLEQS is based on Launder and Spalding (1974). This model employs two partial differential equations to estimate the velocity and length scales and hence commonly known as the two-equation model. These equations are the k- and ε-equations which govern the transport of the turbulent kinetic energy (TKE) and its dissipation rate respectively. The modeled equations are

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}\left(\rho\, u_j k\right) = \rho P - \rho\varepsilon + \frac{\partial}{\partial x_j}\left[\frac{\mu + \mu_t}{\sigma_k}\frac{\partial k}{\partial x_j}\right]$$

$$\frac{\partial}{\partial t}(\rho\,\varepsilon) + \frac{\partial}{\partial x_j}\left(\rho\, u_j\,\varepsilon\right) = C_{\varepsilon_1}\frac{\rho\, P\,\varepsilon}{k} - C_{\varepsilon_2}\frac{\rho\,\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\left[\frac{\mu + \mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_j}\right] \qquad (3.14)$$

with the production P defined as

$$P = \nu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_m}{\partial x_m} \delta_{ij} \right) \frac{\partial u_i}{\partial x_j} - \frac{2}{3} k \frac{\partial u_m}{\partial x_m} \qquad (3.15)$$

The square root of k is taken to be the velocity scale while the length scale is obtained from

$$\ell = \frac{C_\mu^{3/4} k^{3/2}}{\varepsilon} \qquad (3.16)$$

The expression for eddy viscosity is

$$\nu_t = \frac{C_\mu k^2}{\varepsilon} \qquad (3.17)$$

The five constants used in the model are:

$$C_\mu = 0.09\ ;\ C_{\varepsilon_1} = 1.44\ ;\ C_{\varepsilon_2} = 1.92\ ;\ \sigma_k = 1.0\ ;\ \sigma_\varepsilon = 1.3$$

Wall functions are used for this model. A description of wall functions is provided in Section 7.4.

3.1.4 Extended k-ε Model

This model was developed by Chen and Kim (1987) by modifying the standard k-ε model to make the dissipation rate more responsive to the mean strain rate than the standard k-ε model. The k-equation retains the same form as that of the standard k-ε model. In addition to the dissipation rate time scale, *k/ε,* a production range time scale, *k/P* is introduced in the ε-equation as shown below.

$$\frac{\partial}{\partial t}(\rho\, \varepsilon) + \frac{\partial}{\partial x_j}\left(\rho\, u_j\, \varepsilon\right) = C_{\varepsilon_1} \frac{\rho\, P\, \varepsilon}{k} - C_{\varepsilon_2} \frac{\rho\, \varepsilon^2}{k} + C_{\varepsilon_3} \frac{\rho\, P^2}{k} + \frac{\partial}{\partial x_j}\left[\frac{\mu_f \mu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_j} \right] \qquad (3.18)$$

The net effect of this formulation is to enhance the development of ε when the mean strain is strong and to suppress the generation of ε when the mean strain is weak. The model constants are:

$$C_\mu = 0.09\ ;\ C_{\varepsilon_1} = 1.15\ ;\ C_{\varepsilon_2} = 1.9\ ;\ C_{\varepsilon_3} = 0.25\ ;\ \sigma_k = 0.75\ ;\ \sigma_\varepsilon = 1.15$$

3.1.5 Multiple Time-Scale Model

Most turbulence models, including the standard k-ε model, assume one time scale for both the production and the dissipation rates of turbulence. However, experiments and the Direct Numerical Simulation (DNS) of turbulence have shown that most of the turbulence production occurs at large scales (energy carrying eddies) and is cascaded to smaller scales (dissipative eddies) where most of the dissipation takes place. Several authors (*e.g.*, Hanjalic *et al.*, 1980; Fabris *et al.*, 1981) have suggested partitioning of the energy spectrum into production range k_p and dissipation range k_t, the sum of the two quantities being k. Kim and Chen (1988) have developed a multiple time-scale turbulence model based on variable partitioning of turbulent kinetic energy. The partitioning is determined as a part of the solution and depends on local turbulence intensity, production, energy transfer and dissipation rate. The partition is moved into higher wave numbers when production is high, and to low wave numbers when production vanishes. The model uses a transport equation for each k_p, k_t, ε_p and ε_t, however these equations differ from each other only in the source terms.

$$\frac{\partial}{\partial t}\left(\rho k_p\right) + \frac{\partial}{\partial x_j}\left(\rho\, u_j k_p\right) = \rho P - \rho\varepsilon_p + \frac{\partial}{\partial x_j}\left(\frac{\mu + \mu_t}{\sigma_{kp}}\frac{\partial k_p}{\partial x_j}\right)$$

$$\frac{\partial}{\partial t}\left(\rho k_t\right) + \frac{\partial}{\partial x_j}\left(\rho u_j k_t\right) = \rho\, \varepsilon_p - \rho\, \varepsilon_t + \frac{\partial}{\partial x_j}\left(\frac{\mu + \mu_t}{\sigma_{kt}}\frac{\partial k_t}{\partial x_j}\right)$$

$$\frac{\partial}{\partial t}\left(\rho\varepsilon_p\right) + \frac{\partial}{\partial x_j}\left(\rho u_j \varepsilon_p\right) = C_{\varepsilon p_1}\frac{\rho P^2}{kp} + C_{\varepsilon p_2}\frac{\rho P \varepsilon_p}{kp} - C_{\varepsilon p_3}\frac{\rho\, \varepsilon_p^2}{kp} + \frac{\partial}{\partial x_j}\left(\frac{\mu + \mu_t}{\sigma\varepsilon_p}\frac{\partial \varepsilon_p}{\partial x_j}\right)$$

$$\frac{\partial}{\partial t}\left(\rho\varepsilon_t\right)+\frac{\partial}{\partial x_j}\left(\rho u_j\varepsilon_t\right)=C_{\varepsilon t_1}\frac{\rho\varepsilon P^2}{kt}+C_{\varepsilon t_2}\frac{\rho P\varepsilon_p\,\varepsilon_t}{kt}-C_{\varepsilon t_3}\frac{\rho\,\varepsilon_t^2}{kt}+\frac{\partial}{\partial x_j}\left(\frac{\mu+\mu_t}{\sigma\varepsilon_t}\frac{\partial\varepsilon_t}{\partial x_j}\right) \qquad (3.19)$$

The eddy viscosity is defined as

$$v_t = C_\mu\ k^2/_{\varepsilon p} \qquad (3.20)$$

where k, the total TKE, is calculated as

$$k = k_p + k_t \qquad (3.21)$$

The model constants are:

$C_\mu = 0.09;\ \sigma_{kp} = \sigma_{kt} = 0.75;\ \sigma_{\varepsilon p} = \sigma_{\varepsilon t} = 1.15$

$C_{\varepsilon p1} = 0.21;\ C_{\varepsilon p2} = 1.24;\ C_{\varepsilon p3} = 1.84; C_{\varepsilon t1} = 0.29;\ C_{\varepsilon t2} = 1.28;\ C_{\varepsilon t3} = 1.66$

3.1.6 Low Reynolds Number Model

All three models described above, viz. the Standard k-ε model, the extended k-ε model, and the multiple time scale model, are of high-Reynolds number form and therefore require the used of wall functions. However, the commonly used wall functions may not be accurate in flows with large separation, suction, blowing, heat transfer, relaminarization, *etc*. This difficulty associated with wall functions can be reduced by the use of low-Reynolds number k-ε models which permit integration of momentum and k-ε equations all the way to the wall. The k-ε equations are modified as shown below to include the effect of molecular viscosity in the near wall regions.

$$
\begin{gathered}
\frac{\partial}{\partial t}(\rho\, k)+\frac{\partial}{\partial x_j}\left(\rho\, u_j\, k\right)=\frac{\partial}{\partial x_j}\left[\frac{\mu+\mu_t}{\sigma_k}\frac{\partial k}{\partial x_j}\right]+\rho\left(P-\varepsilon-D\right)\\
\frac{\partial}{\partial t}(\rho\,\varepsilon)+\frac{\partial}{\partial x_j}\left(\rho\, u_j\,\varepsilon\right)=\frac{\partial}{\partial x_j}\left[\frac{\mu+\mu_t}{\sigma_\varepsilon}\frac{\partial\varepsilon}{\partial x_j}\right]+C_{\varepsilon_1}f_1\frac{\rho\, P\,\varepsilon}{k}-C_{\varepsilon_2}f_2\frac{\rho\varepsilon^2}{k}+E\\
\mu_t=C_\mu f_\mu\frac{\rho k^2}{e}
\end{gathered}
\tag{3.22}
$$

Several versions low-Reynolds number k-ε models are available today. After a careful analysis of the comparative studies done by Patel *et al.* (1985), Brankovic and Stowers (1988), Avva *et al* (1990), the low-Re model of Chien (1982) was selected for implementation in REFLEQS. The model parameters appearing in the preceding equations are:

$$
\begin{gathered}
C_\mu=0.09\,;\; C_{\varepsilon_1}=1.35\,;\; C_{\varepsilon_2}=1.8\,;\sigma_k=1.0\,;\;\sigma_\varepsilon=1.3\\
f_\mu=1-exp\left(-0.0115\,y^+\right),f_1=1.0\,;\; f_2=1-0.22\,exp\left[-\left(R_r/6\right)^2\right]\\
D=2\nu k/y^2\,;\; E=-2\nu\left(\varepsilon/y^2\right)exp\;\left(-0.5\,y^+\right)
\end{gathered}
\tag{3.23}
$$

3.2 Combustion Models

The combustion models implemented in the REFLEQS code include: instantaneous, equilibrium, and finite rate chemistry models. Previously, the finite-rate model in the REFLEQS code was restricted to one-step or multi-step irreversible reactions such that a particular fuel species may appear as a reactant in one reaction step only. In this study, efforts were made to extend the chemistry modeling capability of the REFLEQS code to include dependent, multi-step reactions in which the same fuel (or oxidizer) species can exist in more than one step. The cases of H_2-Air and Hydrocarbon-Air combustions are considered.

3.2.1 2-Step H_2-O_2 Reaction Model

The combustion model for H_2-O_2 reaction is based on the 2-step model of Rogers and Chinitz (1983).

$$H_2 + O_2 = 2OH \tag{3.24a}$$

$$H_2 + 2OH = 2H_2O \tag{3.24b}$$

Further simplifications are made by assuming that the first step, Equation (3.24a), is in equilibrium, and the second step, Equation (3.24b), is irreversible. The implementation of this reaction mechanism into the REFLEQS code is discussed below.

<u>Basic Equations:</u> The basic approach of REFLEQS in solving chemical species, is to solve the mixture fractions and the "fuel" species of a finite-rate reaction first, and then derive the mass fractions (or concentrations) of all species according to element conservation. For this 2-step H_2-O_2 reaction of mixed equilibrium chemistry and finite-rate chemistry, the basic equations are the tranport equations of conserved mixture fractions

$$F\,(f_i) = 0\,, \quad i = 1, 2, ..., NCMPS - 1 \tag{3.25}$$

where *NCMPS* is the number of mixture compositions. The transport equation of fuel species $[H_2]$ in Equation (3.24b) is given as:

$$F\left(\left[H_2\right]\right) = -K_{f,H_2O}\left[H_2\right]\left[OH\right]^2 \tag{3.26}$$

The extra relations include equilibrium Equation of (3.24a), *i.e.*,

$$K_{OH} = \frac{[OH]^2}{\left[H_2\right]\left[O_2\right]} \tag{3.27}$$

and element conservation

$$[OH] + 2\left[H_2\right] + 2\left[H_2O\right] = [H] \tag{3.28a}$$

$$[OH] + 2[O_2] + [H_2O] = [O] \tag{3.28b}$$

In the above Equations (3.25) through (3.28a,b), F( ) denotes the fluid transport operator and, f_i is the mixture fraction to be solved for NCMPS-1 compositions (with $f_{NCMPS} \equiv 1 - \sum_{i=1}^{NCMPS-1} f_i$). The equilibrium constant K,OH for Equation (3.24a) is expressed as

$$K_{OH} = exp\left(\frac{-\Delta G}{RT}\right), \Delta G = 2G_{OH} - \left(G_{H_2} + G_{O_2}\right) \tag{3.29}$$

where G is the Gibbs free energy at T.

Solution Procedure: Once the mixture fraction f_i is solved, the mole values of elements $[H]$, and $[O]$ can be derived as

$$[H] = [OH]^{(0)} + 2\left[H_2\right]^{(0)} + \left[H_2O\right]^{(0)} \tag{3.30a}$$

$$[O] = [OH]^{(0)} + 2\left[O_2\right]^{(0)} + \left[H_2O\right]^{(0)} \tag{3.30b}$$

where $[\ ]^{(0)}$ is the un-reacted species concentration from mixture fraction f_i.

With $[H_2]$ solved from Equation (3.26), the species $[O_2]$, $[OH]$, $[H_2O]$ are to be obtained from Equation (3.30b) plus

$$[OH] + 2[H_2O] = [H] - 2[H_2] \tag{3.31}$$

$$[OH]^2 = K_{OH}[H_2][O_2] \tag{3.32}$$

After algebraic manipulations, the Equations (3.30b), (3.31), and (3.32) are solved in different cases of $[H_2]$ values as:

(i) $[H_2] > 0$, then

$$[OH] = \frac{1}{2}\left(\sqrt{b^2 + 4bR_1} - b\right) \tag{3.33a}$$

$$[O_2] = [OH]^2 / a \tag{3.33b}$$

(ii) $[H_2] = 0$, then

$$[OH] = 0 \tag{3.34a}$$

$$[O_2] = 1/4\, R_1 \tag{3.34b}$$

and in both cases

$$[H_2O] = [O] - 2[O_2] - [OH] \tag{3.35}$$

In the above expressions,

$$b = 1/4a = 1/4\, K_{OH}[H_2] \tag{3.36}$$

$$R_1 = max\left\{o, 2[O] - [H] + 2[H_2]\right\} \tag{3.37}$$

3.2.2 2-Step CH_4 - O_2 Reaction Model

The 2-step CH_4 - O_2 reaction model of Westbrook and Dryer (1981) is employed and simplified by assuming only forward reaction is present.

$$CH_4 + \frac{3}{2}O_2 \overset{k_{1f}}{\rightarrow} CO + 2H_2O \tag{3.38a}$$

$$\frac{1}{2}O_2 + CO \overset{k_{2f}}{\rightarrow} CO_2 \tag{3.38b}$$

where the reaction rates of Equations (3.38a and 3.38b) are given by Westbrook and Dryer (1981), (in cm-g-s unit) respectively as

$$\dot{\omega}_1 = 6.7 \times 10^{12} e^{\frac{-48.4}{RT}} \left[CH_4\right]^{0.2} \left[O_2\right]^{1.3} \tag{3.39a}$$

$$\dot{\omega}_2 = 10^{14.6} e^{\frac{-40}{RT}} [CO] \left[H_2O\right]^{0.5} \left[O_2\right]^{0.25} \tag{3.39b}$$

An algorithm similar to that for the 2-step H_2-O_2 mechanism is used. The transport equations to be solved are Equation (3.25) for mixture fraction *fi* and those for "fuel" species as

$$F\left(\left[CH_4\right]\right) = -\dot{\omega}_1 \tag{3.40}$$

$$F\left(\left[O_2\right]\right) = -1.5\dot{\omega}_1 - 0.5\dot{\omega}_2 \tag{3.41}$$

Other species can be derived from the element conservation relations.

$$\left[CH_4\right] + [CO] + \left[CO_2\right] = [C] \tag{3.42a}$$

$$4\left[CH_4\right] + 2\left[H_2O\right] = [H] \tag{3.42b}$$

$$2\left[O_2\right] + [CO] + 2\left[CO_2\right] + \left[H_2O\right] = [O] \tag{3.42c}$$

With elements $[C]$, $[H]$, and $[O]$ obtained from mixture fraction *fi*, and $[CH_4]$, $[O_2]$ solved from Equations (3.39), (3.41), and (3.42) yields

$$\left[H_2O\right] = \frac{1}{2}[H] - 2\left[CH_4\right] \tag{3.43}$$

$$\left[CO_2\right] = [O] - [C] + \left[CH_4\right] - 2\left[O_2\right] + \left[H_2O\right] \tag{3.44}$$

$$[CO] = [C] - \left[CH_4\right] - \left[CO_2\right] \tag{3.45}$$

3.2.3 Chemical Equilibrium Model

The equilibrium model proposed by Meintjes and Morgan (1989) is described below. This method is based on calculating the element potentials or the element activities.

The chemical potential of any species "*j*" can be written as:

$$\mu_j = \mu_j^o + RTln\left(\frac{p_j}{p_o}\right) \qquad (3.46)$$

where u_j = chemical potential, μ_j^o = standard state Gibbs function, p_o = atm · pr and p_j = partial pressure of species "*j*".

$$p_j = \frac{n_j}{n} p \qquad (3.47)$$

where n_j/n is the mole fraction of species "*j*", p is the total pressure of mixture and p_o is atm pressure.

The chemical potential "μ_j" can be written as:

$$\mu_j = \sum_{i=1}^{I} a_{ij}\mu_i \qquad (3.48)$$

where μ_i is the chemical potential of atomic element *i* and a_{ij} is the number of moles of element "*i*" per mole of species "*j*". Substituting into Equation (3.46),

$$\frac{1}{RT}\sum_{i=1}^{I} a_{ij}\mu_i = \frac{\mu_j^o}{RT} + ln\left(\frac{n_j}{n}\frac{p}{p_o}\right) \qquad (3.49)$$

From the equation of state,

$$\frac{p}{n} = \frac{RT}{V} \qquad (3.50)$$

Therefore,

$$\frac{1}{RT}\sum_{i=1}^{I} a_{ij}\,\mu_i = ln\left(\frac{RT}{p_o}\right) + ln\left(\frac{n_j}{V}\right) + \frac{\mu_j^o}{RT} \qquad (3.51)$$

Defining molar concentration, $c_j = n_j/v$ and

Rewriting Equation (3.51),

$$ln\, c_j = -\frac{\mu_j^o}{RT} - ln\left(\frac{RT}{p_o}\right) + \sum_{i=1}^{I} a_{ij}\,\lambda_i \qquad (3.52)$$

where $c_j = n_j/\mathrm{v}$ and $\lambda_i = \mu_i/RT$.

Equation (3.52) can be written as:

$$ln\, c_j = ln\, R_j + \sum_{i=1} a_{ij}\, ln\, Z_i \qquad (3.53)$$

where

$$R_j = \frac{p_o}{RT}\, exp\left(-\frac{\mu_j^o}{RT}\right) \qquad (3.54)$$

and $Z_i = e^{\lambda i}$ (an element variable)

Equation (3.53) is simplified as:

$$c_j = R_j \prod_{i=1}^{I} (Z_i)^{a_{ij}}\, j = 1, ...J \qquad (3.55)$$

Since atoms of each element are conserved,

$$\sum_{j=1}^{I} a_{ij} c_j = b_i , i = 1, \dots I \tag{3.56}$$

where b_i is the total number of moles of element "i" in all species.

Equations (3.55) and (3.56) form the basis of this formulation. Substituting Equation (3.55) into (3.56):

$$\sum_{j=1}^{J} \left\{ a_{ij} R_j \prod_{m=1}^{I} \left(Z_m\right)^{a_{mj}} \right\} - b_i = 0 , i = 1, \dots I \tag{3.57}$$

Thus there are "i" non-linear algebraic equations for the "I" unknown values of Z_i. A good guess is used to obtain the initial values of Z_i and the subsequent values are obtained by solving this system. The values of c_j are calculated from Equation (3.55). The temperature is calculated from "c_j" and then the new values of R_j are calculated and again the system is solved for new values of Z_i and so on until convergence is achieved.

3.3 Turbulence-Combustion Interaction

In many combustion systems, reactants enter in separate streams, with fuel and oxidizer non-premixed. The resulting combustion process of diffusion flames, in the presence of flow fluctuation due to turbulence, is very complicated. Generally, the "fast-chemistry" assumption is made to simplify the problem of turbulence-combustion interaction. Under this assumption, the chemical reactions are considered to have very high production rates, so that the reactions are completed as soon as the reactants are mixed. In other words, the reaction time is assumed to be negligibly short in comparison with the turbulent mixing time. In the limit of fast chemistry, the instantaneous species concentrations and temperature (with given enthalpy) are functions only of conserved scalar concentrations (*e.g.*, mixture fractions) at that instant. The averaged thermodynamic variable can be obtained from the knowledge of statistics of those scalars, which can be expressed by the probability-density function (pdf). A further approximation is made by assuming a form of the pdf that can be evaluated from a few moments of the conserved scalar.

In this study, the conserved-scalar approach based on the fast-chemistry assumption is adopted to account for the effects of turbulence-combustion interaction. The pdf of the mixture fraction is assumed to be a tophat function.

3.3.1 Conserved Scalar Equations

For the problem of diffusion controlled combustion, the effect of mass diffusivity differences among different species are negligible in most turbulent flows at moderate to high Reynolds number (see Kuo, 1986). Therefore, the simplest and best choice for the conserved scalar varaible is the mixture fraction.

To determine the pdf, which describes the statistics of mixture fraction, the mean value and variance of the mixture fraction are needed. These can be solved from the Favre-averaged transport equations suggested by Jones and Whitelaw (1982), as

$$\bar{\rho}\,\tilde{u}_j \frac{\partial \tilde{f}}{\partial x_j} = \frac{\partial}{\partial x_j}\left\{\frac{\mu_t}{\sigma_t}\frac{\partial \tilde{f}}{\partial x_j}\right\} \tag{3.58}$$

$$\bar{\rho}\,\tilde{u}_j \frac{\partial \tilde{g}}{\partial x_j} = \frac{\partial}{\partial x_j}\left\{\frac{\mu_t}{\sigma_t}\frac{\partial \tilde{g}}{\partial x_j}\right\} + 2\,\frac{\mu_t}{\sigma_t}\left(\frac{\partial \tilde{f}}{\partial x_j}\right)^2 - C_D \frac{\bar{\rho}\varepsilon}{k}\,\tilde{g} \tag{3.59}$$

where $\tilde{f}$ and $\tilde{g}$ $(g = f''^2)$ are the density-weighted mean and variances of mixture fraction f, respectively, and C_D has the value of 2.

3.3.2 Top-Hat Pdf Model

The "top-hat" pdf with possible delta function at 0 and 1 is assumed as

$$p(f) = \begin{cases} P_0 \ , \ f = 0 \\ c \ , \quad 0 < a \leq f \leq b < 1 \\ P_1 \ , \ f = 1 \end{cases} \tag{3.60}$$

The specific expression of *p(f)* (*i.e.*, the values of parameters, P_0, P_1, *a, b, c*) can be determined by the following moments.

$$\int_0^1 p(f)\, df = 1 \tag{3.61a}$$

$$\int_0^1 f p(f)\, df = \tilde{f} \tag{3.61b}$$

$$\int_0^1 (f - \tilde{f})^2 p(f) df = \tilde{g} \tag{3.61c}$$

which leads to the explicit form for Equation (3.61) as

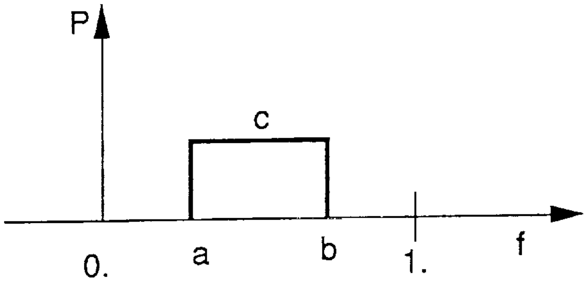
$$P_0 + P_1 + c(b - a) = 1 \tag{3.62a}$$

$$P_1 + \frac{1}{2} c\left(b^2 - a^2\right) = \tilde{f} \tag{3.62b}$$

$$P_1 + \frac{1}{3} c\left(b^3 - a^3\right) = \tilde{g} + \tilde{f}^2 \tag{3.62c}$$

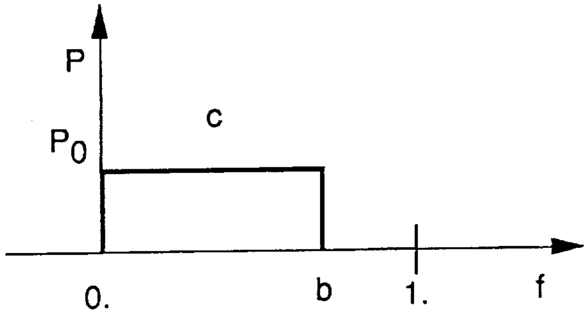
The expression of p(f) for the mean value $\tilde{f} \leq 0.5$ can be derived for three different cases from Equations (3.62a-c) as

(i) $3\tilde{g} < \tilde{f}^2$



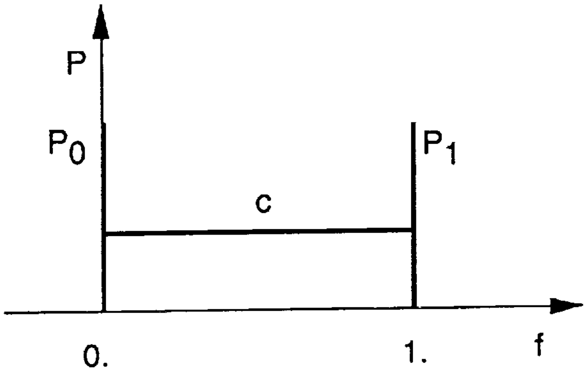
$$\begin{cases} P_0 = P_1 = 0 \\ a = \tilde{f} - \sqrt{3\tilde{g}} \\ b = \tilde{f} + \sqrt{3\tilde{g}} \\ c = \dfrac{1}{2\sqrt{3\tilde{g}}} \end{cases} \qquad (3.63)$$

(ii) $\tilde{f}^2 \le 3\tilde{g} < 2\tilde{f} - 3\tilde{f}^2$



$$\begin{cases} P_0 = \dfrac{3\tilde{g} - \tilde{f}^2}{3\left(\tilde{f}^2 + \tilde{g}\right)} \\ P_1 = 0 \\ a = 0 \\ b = \dfrac{3\left(\tilde{f}^2 + \tilde{g}\right)}{2\tilde{f}} \\ c = \dfrac{8\tilde{f}^3}{9\left(\tilde{f}^2 + \tilde{g}\right)^2} \end{cases} \qquad (3.64)$$

(iii) $3\tilde{g} \ge 2\tilde{f} - 3\tilde{f}^2$



$$\begin{cases} P_0 = \left(1 - \tilde{f}\right) + 3\tilde{f}\left(1 - \tilde{f}\right) + 3\tilde{g} \\ P_1 = \tilde{f} + 3\tilde{f}\left(1 - \tilde{f}\right) + 3\tilde{g} \\ a = 0 \\ b = 1 \\ c = 6\tilde{f}\left(1 - \tilde{f}\right) - 6\tilde{g} \end{cases} \qquad (3.65)$$

Owing to the symmetrical properties of $p(f,\tilde{f},\tilde{g})$ and $p(1-f,\tilde{f},\tilde{g})$ and $\tilde{f}=0.5$ for $\tilde{f}\in(0,1)$, for $\tilde{f}>0.5$, P(f) can be expressed simply by obtaining the parameters as

$$a|_{\tilde{f}} = 1-b|_{(1-\tilde{f})} \tag{3.66a}$$

$$b|_{\tilde{f}} = 1-a|_{(1-\tilde{f})} \tag{3.66b}$$

$$c|_{\tilde{f}} = c|_{(1-\tilde{f})} \tag{3.66c}$$

$$P_0|_{\tilde{f}} = P_1|_{(1-\tilde{f})} \tag{3.66d}$$

$$P_1|_{\tilde{f}} = P_0|_{(1-\tilde{f})} \tag{3.66e}$$

where $[\,]|_{\tilde{f}}$ denotes the parameter for $\tilde{f}>0.5$, and $[\,]|_{(1-\tilde{f})}$ is the parameters given in Equations (3.63-3.65) with $\tilde{f}$ substituted for $(1-\tilde{f})$.

3.3.3 Mean Values of Thermodynamic Variables

Based on the fast-chemistry assumption, the thermodynamic variables such as species mass fractions (or concentrations), temperature, and density, are functions only of the mixture fraction. The mean values of these variables, which are of interest, are obtained by averaging the instantaneous values at the given mixture fraction, f, with the pdf $P(f)$ over the sample domain $f\in(0,1)$ *i.e.*

$$\tilde{Y}_i = \int_0^1 Y_i(f)\,P(f)\,df \tag{3.67a}$$

$$\tilde{T} = \int_0^1 T_i(f)\,P(f)\,df \tag{3.67b}$$

$$\tilde{\rho}^{-1} = \int_0^1 \frac{P(f)}{\rho(f)} \, df \tag{3.67c}$$

in which Y_i is the species mass fraction, T, ρ are temperature and density, respectively.

The integration of Equations (3.67a-c) are approximated by numerical quadrature with limited discrete samples.

3.4 Spray Model

The droplet equations of mass, momentum and energy are solved in a Lagrangian frame of reference moving with the droplets. These solutions are used to calculate the source/sink terms for the corresponding gas phase equations.

The equation of motion for the droplet is written as (Crowe *et al.* (1977)):

$$m_d \frac{d\mathbf{v}}{dt} = C_D \rho (\mathbf{U} - \mathbf{v}) |\mathbf{U} - \mathbf{v}| \frac{Ad}{2} - V_d \nabla p + m_d \mathbf{g} \tag{3.68}$$

where m_d is the mass of the droplet and $\mathbf{v} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}$ its velocity vector. u, v and w are the Cartesian velocity components. C_D is the drag coefficient and ρ, $\mathbf{U}$ and p are the density, velocity and pressure of the gas, respectively. A_d is the droplet surface area and V_d is the droplet volume. For a spherical droplet, $A_d = \pi d^2$ and $V_d = \pi\, d^3/6$ where d is the droplet diameter. $\mathbf{g}$ is the gravity vector. Equation (3.68) accounts for the acceleration/deceleration of the droplet due to combined effects of drag with gas, local pressure gradients and body forces such as gravity.

The drag coefficient, C_D, for the droplet is based on the local Reynolds number evaluated as:

$$\mathrm{Re} = \frac{\rho|\mathrm{U} - \mathrm{v}|\, d}{\mu} \tag{3.69}$$

where μ is the dynamic viscosity of the gas. The following correlations have been found to be valid for different ranges of Reynolds numbers (Crowe *et al*, 1977):

$$\begin{aligned} C_D &= \frac{24}{\mathrm{Re}} && \text{for Re} < 1 \\ C_D &= \frac{24}{\mathrm{Re}}[1 + 0.15\mathrm{Re}^{0.687}] && \text{for } 1 < \text{Re} < 10^3 \\ C_D &= 0.44 && \text{for Re} > 10^3 \end{aligned} \tag{3.70}$$

Substituting for C_D into Equation (3.68),

$$\frac{d\mathbf{v}}{dt} = \frac{18\mu f}{\rho_d d^2}(\mathrm{U} - \mathrm{v}) - \frac{\nabla p}{\rho_d} + \mathbf{g} \tag{3.71}$$

where ρ_d is the droplet liquid density and $f = C_D Re/24$. The droplet locations are determined from the velocities as:

$$\frac{d\mathbf{R}}{dt} = \mathbf{v} \tag{3.72}$$

where **R** is the position vector of the droplet.

The mass conservation equation for the droplet is given by:

$$\frac{dm_d}{dt} = -\,\mathrm{Sh}\,(\rho D)\,\pi d\,(x_v - x_\infty) \tag{3.73}$$

where Sh is the Sherwood number and D is the mass diffusion coefficient for the gas. x_v and x_∞ are the vapor mass fractions at the droplet surface and the free stream, respectively. In this model, x_v is calculated from the liquid saturation pressure. The Sherwood number is obtained from the following expression (Crowe *et al*, 1977):

$$\mathrm{Sh} = 2 + 0.6\mathrm{Re}^{0.5}\, S_c^{\ 0.333} \tag{3.74}$$

where S_c is the Schmidt number. Since $m_d = \rho_d \pi d^3/6$, the mass equation for the droplet is rewritten in terms of its diameter:

$$\frac{d(d)}{dt} = -2\mathrm{Sh}\,(\rho D)\,\frac{x_v - x_\infty}{\rho_d d} \tag{3.75}$$

The energy equation for the droplet is written as:

$$m_d C_d \frac{dT_d}{dt} = \dot{q} + L\left(\frac{dm_d}{dt}\right) \tag{3.76}$$

where $\dot{q}$ is the sensible heat transferred to the droplet and T_d is the droplet temperature. L is the latent heat of vaporization for the droplet fluid and C_d is the specific heat of the droplet. $\dot{q}$ is calculated as:

$$\dot{q} = \mathrm{Nu}\pi kd\ \ (T_g - T_d) \tag{3.77}$$

where k and T_g are the thermal conductivity and temperature of the gas, respectively. The Nusselt number, Nu, is obtained from the following correlation:

$$\mathrm{Nu} = 2 + 0.6\mathrm{Re}^{0.5}\,\mathrm{Pr}^{0.333} \tag{3.78}$$

where Pr is the Prandtl number. Substituting the above expressions into Equation (3.76), the energy equation is rewritten as:

$$\frac{dT_d}{dt} = \frac{T_g - T_d}{\theta} - \frac{Q_L}{\theta} \tag{3.79}$$

where

$$Q_L = \frac{L\,\mathrm{Sh}\,(\rho D)\left(x_v - x_\infty\right)}{\mathrm{Nu}k} \tag{3.80}$$

$$\theta = \frac{\rho_d d^2 C_d}{6\mathrm{Nu}k} \tag{3.81}$$

If the droplet temperature is below its boiling point, Equation (3.75) is used as the mass conservation equation. Once the droplet attains its boiling point, the mass conservation equation is obtained by setting the left hand side of Equation (3.76) to zero, *i.e.*, the droplet temperature cannot exceed its boiling point. Therefore,

$$\frac{dm_d}{dt} = -\frac{\dot{q}}{L} \tag{3.82}$$

This equation implies that all the heat transferred from the gas is used to vaporize the droplet.

The solutions to the droplet equations provide sources/sink terms to the gas phase conservation equations. The sources/sinks are assigned to the particular cell (in the Eulerian gas phase) in which the droplet is located. The calculation of the source/sinks is as follows.

The mass of the droplet is continuously monitored along its trajectory. The source/sink term for a cell is given as:

$$\Delta\dot{m} = \dot{n}\rho_d\pi \sum_i \frac{\left(d_i^3\right)_{\mathrm{in}} - \left(d_i^3\right)_{\mathrm{out}}}{6} \tag{3.83}$$

the summation takes place over all the droplets located in the given cell. The subscripts "in" and "out" refer to cell inlet and outlet conditions. $\dot{n}$ is the number density of droplets in a given parcel. Similarly the momentum source/sink term is evaluated as:

$$\Delta\dot{\mathbf{M}} = \dot{n}\rho_d\pi\sum_i \frac{\left(\mathbf{v}_i d_i^3\right)_{\mathrm{in}} - \left(\mathbf{v}_i d_i^3\right)_{\mathrm{out}}}{6} \tag{3.84}$$

This includes the effects of frictional interactions with the gas. The momentum source term is a vector. The energy source/sink term is given as:

$$\Delta\dot{E} = \Delta\dot{m}h_v - \dot{n}\sum_i \dot{q}_i \tag{3.85}$$

where h_v is the enthalpy of the vapor at the vaporization temperature and $\dot{q}_i$ is the heat transfer from the gas to the droplet (calculated according to Equation (3.77)).

3.5 Radiation Model

A large number of numerical techniques are available for solving the equations governing the transfer of thermal radiation. The most widely used among them are listed below.

a. Hottel's Zone method
b. Flux method
c. P-N approximation method
d. Monte-Carlo method
e. Discrete-transfer method
f. Discrete-ordinate method

Hottel's zone method was the first general numerical procedure developed to handle the radiative transfer equation. It has been widely used to estimate the radiation transfer in the absence of detailed knowledge of the flow and reaction. This method involves the use of exchange factors which needs to be worked out in advance for complex geometries. The disadvantage of this method is that it is very uneconomical and not suited for complex configurations with strongly participating media. The flux methods involve differential approximation to the radiative transfer equation. This method is computationally efficient and for this reason it has been widely used in general combustion problems. However, this method suffers from inadequate accuracy and the difficulty to extend to complex geometries. Compared to this method, the P-N approximation method is more accurate.

Among all the existing numerical treatments for radiative heat transfer, the Monte-Carlo method is the most accurate method. In this method, the randomly chosen energy releases are tracked through the combustion chamber for their lifetimes. Unfortunately, this method is computationally expensive. The discrete-transfer method, on the other hand, is economical and can be applied in a straightforward manner to complex geometries. This method, however, cannot treat the scattering by the medium containing droplets and soot particles and hence is not suitable for combustor applications.

After comprehensive review of all the methods, it has been found that the discrete-ordinate is well suited for radiation prediction in rocket combustion chambers. There are several advantages of this method. Firstly, the discrete-ordinate method can be easily extended for polar and complex geometries. Secondly, the evaluation of the in-scattering term is relatively simple. Also, the boundary conditions can be imposed more accurately at inlets and exits. Finally, this method is accurate and can be easily coupled with CFD codes based on the control volume approach. The governing equations and the solution procedure for this method are described below.

3.5.1 <u>Governing Equations</u>

The integro-differential radiative heat transfer equation for an emitting-absorbing and scattering gray medium can be written as (Fiveland, 1984)

$$(\Omega \cdot \nabla) I(r, \Omega) = -(k + \sigma) I(r, \Omega) + k I_b(r) + \frac{\sigma}{4\Pi} \int_{\Omega' = 4\Pi} I(r, \Omega') \phi(\Omega' \rightarrow \Omega) d\Omega' \qquad (3.86)$$

where *Ω* is the direction of propagation of the radiation beam, *I* is the radiation intensity which is a function of both position *(r)* and direction (*Ω*), *k* and *σ* are the absorption and scattering coefficients respectively. I_b is the intensity of black body radiation at the temperature of the medium, ϕ is the phase function of the energy transfer from the incoming Ω' direction to the outgoing direction Ω. The term on the left hand side represents the gradient of the intensity in the specified direction Ω. The three terms on the right hand side represent the changes in intensity due to absorption and out-scattering, emission and in-scattering respectively.

The boundary condition for solving the above equation may be written as

$$I(r,\Omega) = \varepsilon\, I_b(r) + \frac{\rho}{\pi}\int_{n\cdot\Omega'<0} \left|n\cdot\Omega'\right| I(r,\Omega')\, d\Omega' \qquad (3.87)$$

where I is the intensity of radiant energy leaving a surface at a boundary location, ε is the surface emissivity, ρ is the surface reflectivity, and n is the unit normal vector at the boundary location.

In the discrete-ordinate method, Equations (3.86) and (3.87) are replaced by a discrete set of equations for a finite number of ordinate directions. The integral terms on the right hand side of the above equations are approximated by summation over each ordinate. The discrete-ordinate equations may then be written as,

$$\mu_m \frac{\partial I_m}{\partial x} + \xi_m \frac{\partial I_m}{\partial y} = -(k+\sigma)\, I_m + kI_b + \frac{\sigma}{4\Pi}\sum_{m'} w_{m'}\, \phi_{m'm}\, I_{m'}\,;\; m = 1, \ldots M \qquad (3.88)$$

For cylindrical polar geometries, the discrete-ordinate equation may be written as,

$$\frac{\xi_m}{r}\frac{\partial (r\, I_m)}{\partial r} - \frac{1}{r}\frac{\partial (\eta_m I_m)}{\partial \phi} + \mu_m \frac{\partial I_m}{\partial z} = -(k+\sigma)\, I_m + kI_b + \frac{\sigma}{4\Pi}\sum_{m'} w_{m'}\, \phi_{m'm}\, I_{m'}\,;\; m = 1, \ldots M \qquad (3.89)$$

where the second term on the left hand side represents the angular intensity redistribution. It accounts for the change in the direction cosines as a beam travels in a straight line.

The different boundary conditions required to solve this equation are given as,

Wall Boundary

$$I_m = \varepsilon I_b + \frac{\rho}{\Pi} \sum_{m'} w_{m'} |\mu_{m'}| I_{m'} \tag{3.90}$$

Symmetry Boundary

$$I_m = I_{m'} ; \mu_{m'} = -\mu_m \tag{3.91}$$

Inlet and Exit Boundary

$$I_m = 0 \tag{3.92}$$

In the above equations, m and m' denote the outgoing and incoming directions, respectively. For a particular direction, denoted by m, the values μ and ξ are the direction cosines along the x and y directions. These equations represent m coupled partial differential equations for the m intensities, I_m. In the present code, only the S_4 approximation, *i.e.*, 12 ordinate directions are considered. Table 3-1 lists the values of the direction cosines and the associated weights.

Table 3-1.
Direction Cosines and Their Associated Weights for the S_4 Approximation

Direction Number	Direction Cosines			Weight
	μ_m	ξ_m	η_m	w_m
1	-0.908248	-0.295876	.295876	π/3
2	-0.908248	0.295876	.295876	π/3
3	-0.295876	-0.908248	.295876	π/3
4	-0.295876	-0.295876	.908248	π/3
5	-0.295876	0.295876	.908248	π/3
6	-0.295876	0.908248	.295876	π/3
7	0.295876	-0.908248	.908248	π/3
8	0.295876	-0.295876	.908248	π/3
9	0.295876	0.295876	.908248	π/3
10	0.295876	0.908248	.295876	π/3
11	0.908248	-0.295876	.295876	π/3
12	0.908248	0.295876	.295876	π/3

4105/6 t3-1

Equation (3.88) can be cast in a fully conservative form for a general BFC coordinate system as

$$\frac{\partial}{\partial x'}\left[\mu_m I_m J F_{1x} + \xi_m I_m J F_{1y}\right] + \frac{\partial}{\partial y'}\left[\mu_m I_m J F_{2x} + \xi_m I_m J F_{2y}\right]$$
$$= J\left[-(k+\delta)\, I_m + k\, I_b + \frac{\sigma}{4\Pi} \sum w_{m'}\, \phi_{m'm}\, I_{m'}\right] \tag{3.93}$$

where x' and y' are the local grid coordinate directions, J is the Jacobian of the coordinate transformation and

$$F_{1x} = \frac{\partial x'}{\partial x}$$
$$F_{1y} = \frac{\partial x'}{\partial y}$$

$$F_{2x} = \frac{\partial y'}{\partial x}$$
$$F_{2y} = \frac{\partial y'}{\partial y} \tag{3.94}$$

Equation (3.89) can be integrated over the control volume, shown in Figure 3-1, to obtain

$$\mu_m\left(A_e I_{m_e} - A_w I_{m_w}\right) + \xi_m\left(A_n I_{m_n} - A_s I_{m_s}\right) + \left(A_n - A_s\right) \frac{\left[\gamma_{m+1/2} I_{P,\,m+1/2} - \gamma_{m-1/2} I_{P,m-1/2}\right]}{w_m}$$
$$= -(k+\sigma)\, \forall\, I_{mP} + k \forall\, I_{bp} + \forall \frac{\sigma}{4\Pi} \sum w_{m'}\, \phi_{m'm}\, I_{m'p} \tag{3.95}$$

where the coefficients γ are evaluated from the recurrence relation

$$\gamma_{m+1/2} = \gamma_{m-1/2} - \xi_m w_m \tag{3.96}$$

In the above equation A denotes the cell face area and $\forall$ denotes the cell volume and subscripts w, e, s, n, P denote the west, east, south, north, and cell center, respectively. Figure 3-1 shows a sketch of the control volume.

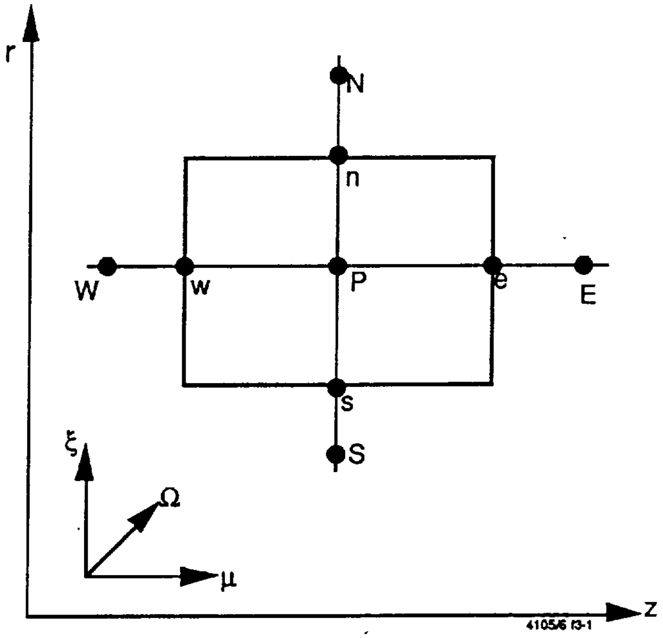


Figure 3-1. Sketch of the Control Volume

The in-scattering term on the right hand side of this equation contains the phase function ϕ which can be prescribed for both isotropic and anisotropic scattering by the medium. For a linear anisotropic scattering, the phase function may be written as,

$$\phi_{m'm} = 1 \cdot 0 + a_o\left[\mu_m \mu_{m'} + \xi_m \xi_{m'} + \eta_m \eta_{m'}\right] \tag{3.97}$$

where a_o is the asymmetry factor that lies between -1 and 1. The values -1,0,1 denote backward, isotropic and forward scattering, respectively.

Equations (3.95) involves 5 intensities out of which 2 intensities are known from the boundary conditions. The other cell face intensities can be eliminated in terms of the cell center intensity by a spatially weighted approximation. For example, if the east and north cell face intensities are unknown, they can be eliminated in terms of cell center intensity by using

$$I_{m_n} = \frac{I_{m_p} - (1-\alpha)\, I_{m_s}}{\alpha} \tag{3.98a}$$

$$I_{m_e} = \frac{I_{m_p} - (1-\alpha)\, I_{m_w}}{\alpha} \tag{3.98b}$$

where α is the weighting factor. $\alpha=1$ yields a upwind differencing scheme and $\alpha=1/2$ yields a central differencing or the "diamond differencing" scheme. Substituting Equation (3.98) into the integrated discrete-ordinate equation results in a algebraic equation for the cell center intensity,

$$I_{m_p} = \frac{\mu_m A\, I_{m_w} + \xi_m B\, I_{m_s} + S_1 {}_{-} S_2}{\mu_m A_N + \xi_m A_S + \alpha\,(k+\sigma)\,\forall} \tag{3.99}$$

where

$$A = A_e(1-\alpha) + A_w\,\alpha$$
$$B = A_n(1-\alpha) + A_S\,\alpha$$
$$S_1 = \alpha\, k \forall\, I_{bp}$$
$$S_2 = \alpha \forall \frac{\sigma}{4\Pi}\, w_{m'}\, \phi_{m'm}\, I_{m'P}$$

Equation (3.99) is appropriate when both the direction cosines are positive and the direction of integration proceeds in a direction of increasing x and y. For negative direction cosine, the direction of integration is reversed and the integration sweep is started from the appropriate corner of the domain. Thus, for each set of direction cosines, the intensities at all the cell centers are obtained by marching in the appropriate direction. This procedure is repeated for all the ordinate directions. The in-scattering term is evaluated explicitely by using the previous iteration values, thus decoupling the m ordinate equations.

Assuming local radiative equilibrium, the source term for each computational cell *i.e.* the net radiative heat flux, is given by,

$$S = \forall\, k \sum_{m} I_m w_m - \forall\, 4\Pi\, k\, I_b \tag{3.100}$$

This source term is added to the gas phase energy equation. The gas phase equations and the radiative transfer equations are solved in an iterative manner.

4. GEOMETRY AND ITS COMPUTATIONAL REPRESENTATION

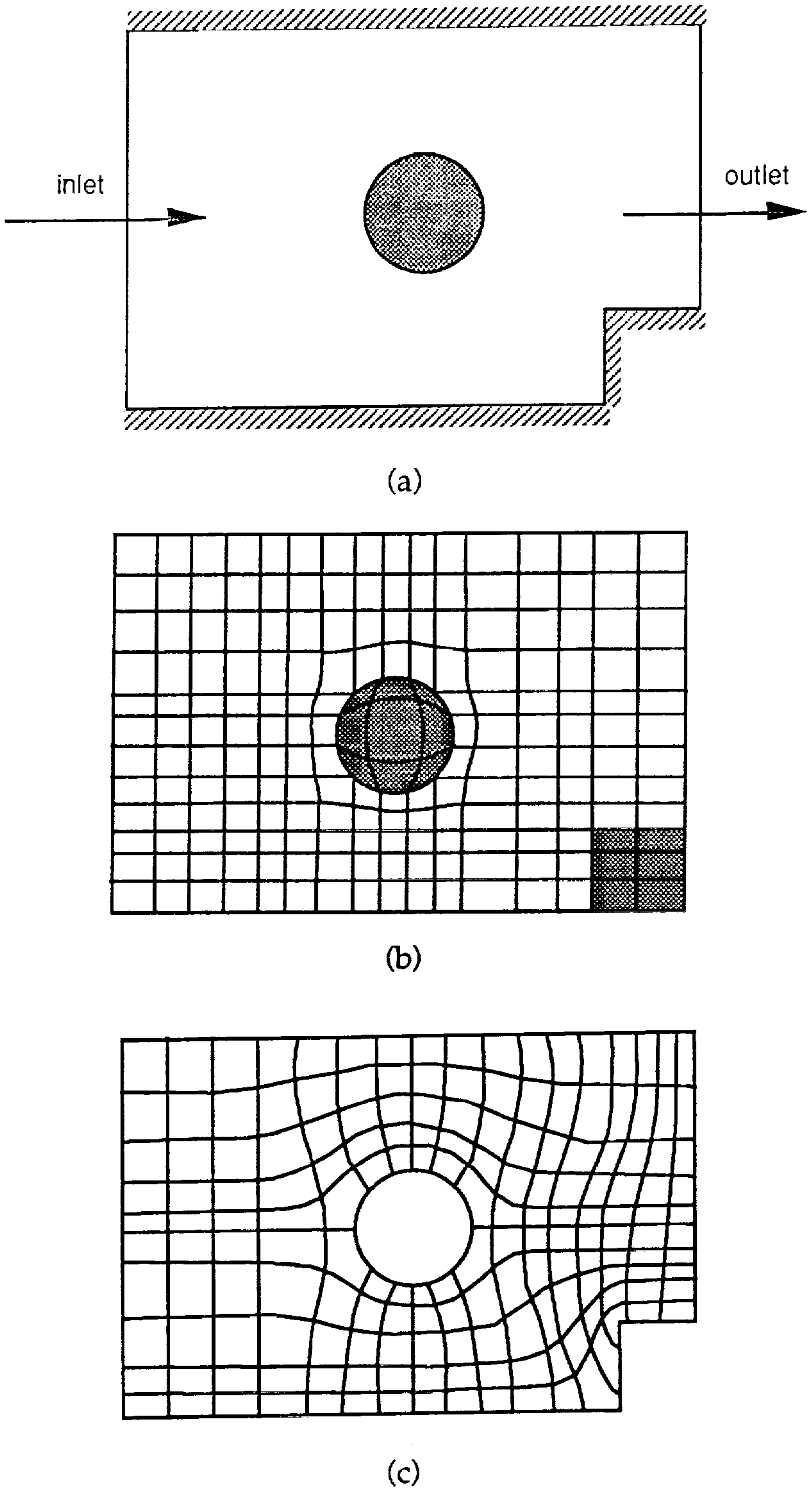
The REFLEQS code uses a single domain structured grid approach. One important feature of the code is the use of a non-orthogonal body-fitted grid with internal blockages. As a result, many flow problems with complex geometries can be handled. In this section, a brief description of the geometry, the Body-Fitted Coordinate (BFC) system, and the internal blockage concepts are presented.

4.1 Geometry and Grid

The REFLEQS code can be used to solve any fluid flow problems with complex geometries as long as the flow domain is covered with a structured grid. A grid is a discrete representation of the flow geometry and computational domain. A grid is considered to be structured if there exists three grid lines (for a 3D problem) to identify three distinctive directions and any face of a control volume is on two grid lines. In other words, a single (I,J,K) index can be used to identify a cell or a point of a structured grid. Since the REFLEQS code allows internal blockages, the above mentioned structured grid is able to cover most flow problems despite its geometric limitations. An example of a two-dimensional flow problem is shown in Figure 4-1a and two possible grid layouts, one with blockage and another without blockage, are displayed in Figures 4-1b and 4-1c.

4.2 Body-Fitted-Coordinate (BFC) System

In the REFLEQS code, three different coordinate systems are used: the Cartesian grid system; 2D axisymmetric polar system; and an arbitrary Body-Fitted-Coordinate system. The Cartesian and polar systems are specialized grid systems for problems with simple geometries and they could offer some savings of the storage and execution time. In general, the REFLEQS code works with the BFC system and on a BFC grid. Figure 4-1b and 4-1c are just two simple examples of such a grid system.



(a)

(b)

(c)

Figure 4-1. A Sample Two-Dimensional Flow with Two Possible Grids with and without Blockages

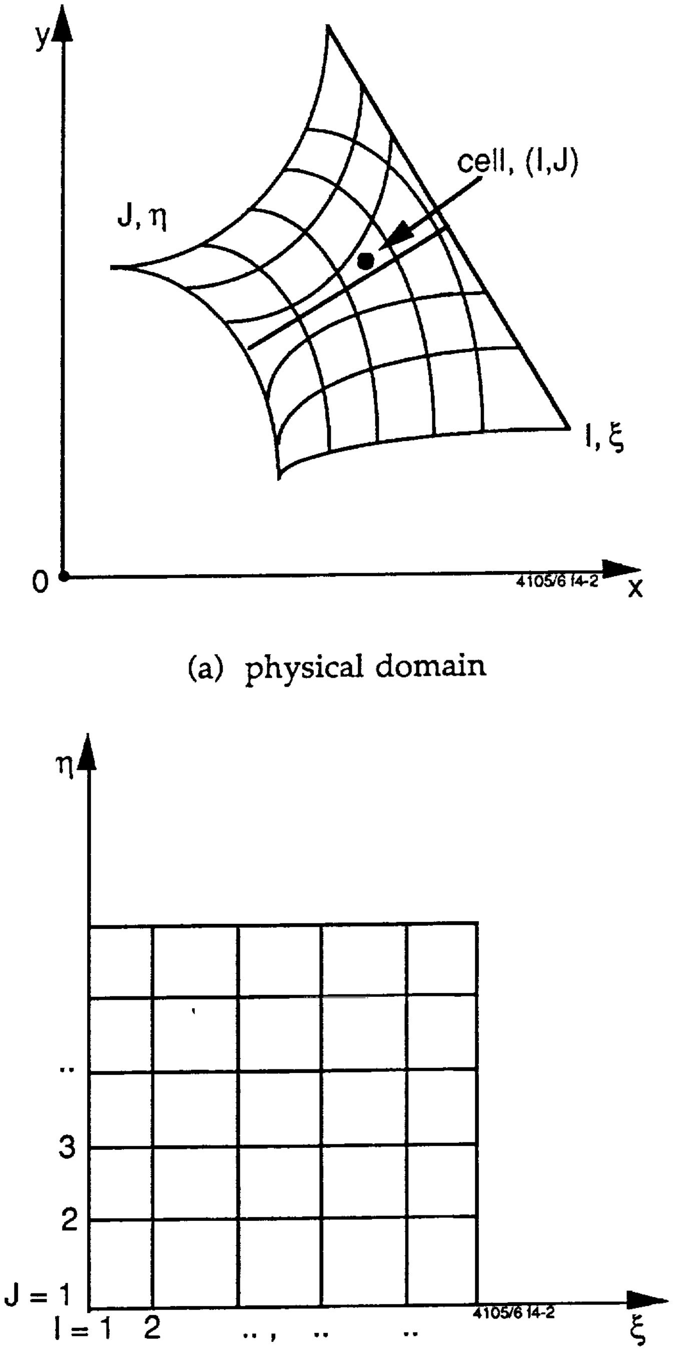
Mathematically, a BFC system can be viewed as a coordinate transformation from physical domain to computational domain as illustrated in Figure 4-2 for a 2D problem. The ξ and η coordinate lines and the I and J indices have a direct correlation. In order to identify each point on the physical domain, a base Cartesian coordinate system is always used as shown in Figure 4-2a. As a result, the transformation can be expressed as

$$\xi = \xi(x, y, z), \eta = \eta(x, y, z), \zeta = \zeta(x, y, z) \text{ or } \begin{aligned} x &= X(\xi, \eta, \zeta) \\ y &= Y(\xi, \eta, \zeta) \\ z &= Z(\xi, \eta, \zeta) \end{aligned} \qquad (4.1)$$

Therefore, each point on a physical domain can be identified by a triplet (x, y, z). This triplet is a function of ξ, η and ζ or is associated with a particular (i, j, k) index. Therefore, the REFLEQS code works on a BFC grid system expressed on a base Cartesian coordinates. Note that a physical domain is always transformed to a rectangular domain as shown in Figure 4-2.

For future reference, a brief introduction of the BFC coordinate system is given below. A comprehensive description of BFC coordinate system can be found in the work of Thompson *et al*, (1985). For derivation convenience, (ξ_1 ξ_2, ξ_3) is used to replace (ξ, η, ζ). In a BFC coordinate system, (ξ_1, ξ_2, ξ_3), the covariant base vectors are defined as

$$\vec{\varepsilon}_j = \frac{\partial \vec{r}}{\partial \xi_j} ; j = 1, 2, 3 \qquad (4.2)$$



(a) physical domain

(b) computational domain

Figure 4-2. An Illustration of the Transformation from Physical to Computational Domain

where $\vec{r}$ is the displacement vector and is equal to $x\,\vec{i} + y\,\vec{j} + z\,\vec{k}$, and contravariant base vectors are defined as

$$\vec{\varepsilon}^{\,1} = \frac{\vec{\varepsilon}_2 \times \vec{\varepsilon}_3}{J}\,;\; \vec{\varepsilon}^{\,2} = \frac{\vec{\varepsilon}_3 \times \vec{\varepsilon}_1}{J}\,;\; \vec{\varepsilon}^{\,3} = \frac{\vec{\varepsilon}_1 \times \vec{\varepsilon}_2}{J} \tag{4.3}$$

where J is the Jacobian defined as $J = \left(\vec{\varepsilon}_1 \times \vec{\varepsilon}_2\right) \cdot \vec{\varepsilon}_3$. Basically, $\vec{\varepsilon}_j$ is a base vector along ξ_j coordinate line and $\vec{\varepsilon}^{\,j}$ is a base vector normal to the surface formed by $\vec{\varepsilon}_i$ and $\vec{\varepsilon}_k (i \neq j \neq k)$. But note that $\vec{\varepsilon}_j$ and $\vec{\varepsilon}^{\,j}$ are not unit bases and unit bases can be defined as

$$\vec{e}_j = \frac{\vec{\varepsilon}_j}{h_j}\,, h_j = \left|\vec{\varepsilon}_j\right| = \sqrt{\left(\frac{\partial x}{\partial \xi_1}\right)^2 + \left(\frac{\partial y}{\partial \xi_2}\right)^2 + \left(\frac{\partial z}{\partial \xi_3}\right)^2} \tag{4.4}$$

and

$$\vec{e}^{\,j} = \frac{\vec{\varepsilon}^{\,j}}{\left|\vec{\varepsilon}^{\,j}\right|} \quad \textit{(no summation on j)} \tag{4.5}$$

where h_j is usually called scale factors.

It is easy to show that the covariant and contravariant base vectors satisfy the basic relationship

$$\vec{\varepsilon}_i \cdot \vec{\varepsilon}^{\,j} = \delta_{ij} \tag{4.6}$$

where δ_{ij} is the Kronecker delta.

In the BFC system, the gradient, divergence, curl, and Laplacian operators can be expressed in conservative form as:

Gradient: $$\nabla f = \frac{1}{J} \frac{\partial}{\partial \xi_k}\left(J\,\vec{\varepsilon}^{\,k} f\right)$$

Divergence: $$\nabla \cdot \vec{V} = \frac{1}{J} \frac{\partial}{\partial \xi_k}\left(J\,\vec{\varepsilon}^{\,k} \cdot \vec{V}\right)$$

Curl: $$\nabla \times \vec{V} = \frac{1}{J} \frac{\partial}{\partial \xi^k} \left(J \vec{\varepsilon}^k \times \vec{V} \right)$$

Laplacian: $$\nabla^2 f = \frac{1}{J} \frac{\partial^2}{\partial \xi^i \partial \xi^j} \left[J f \vec{\varepsilon}^i \cdot \vec{\varepsilon}^j \right]$$

The geometric meaning of the above quantities can be easily explained on a 2D local point P curvilinear coordinates shown in Figure 4-3. For a BFC grid, the Jacobian is simply the volume of each cell.

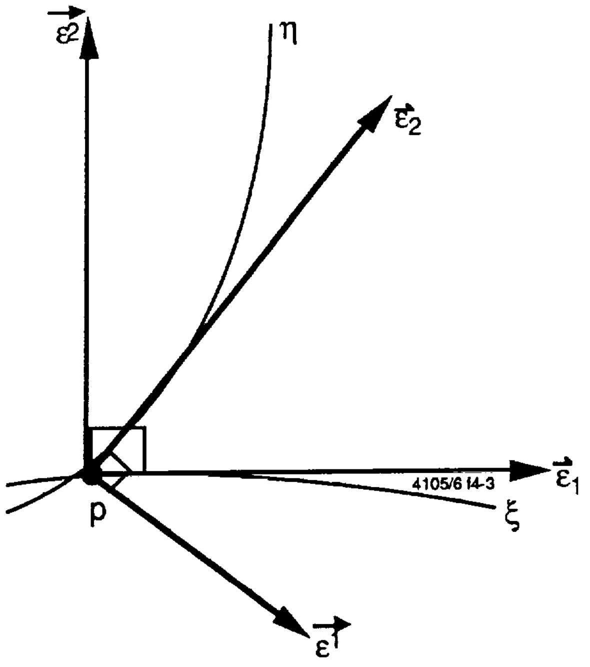


Figure 4-3. Geometrical Meaning of Covariant and Contravariant Bases for a 2D BFC System

4.3 Blockage Concept

In many engineering problems the boundaries of the domain are very complicated and there can be internal flow obstacles. To facilitate the use and generation of a single domain structured grid, the REFLEQS code employs the so called "blockage" concept. With blockage capability, several parts of the flow domain can be real solids or imaginary dead regions which are blocked-off. Such blocked-off regions are illustrated in Figure 4-4 for a 2D flow where both real and imaginary blockage is used. As it can be seen, a structured grid is very difficult to generate without blockage capability and blockage offers convenience in grid generation. The blocked-off regions will participate in the grid generation and numerical solution process but will have solid wall properties. The numerical treatment of the blockages is not discussed here and interested readers should consult the book by Patankar (1980).

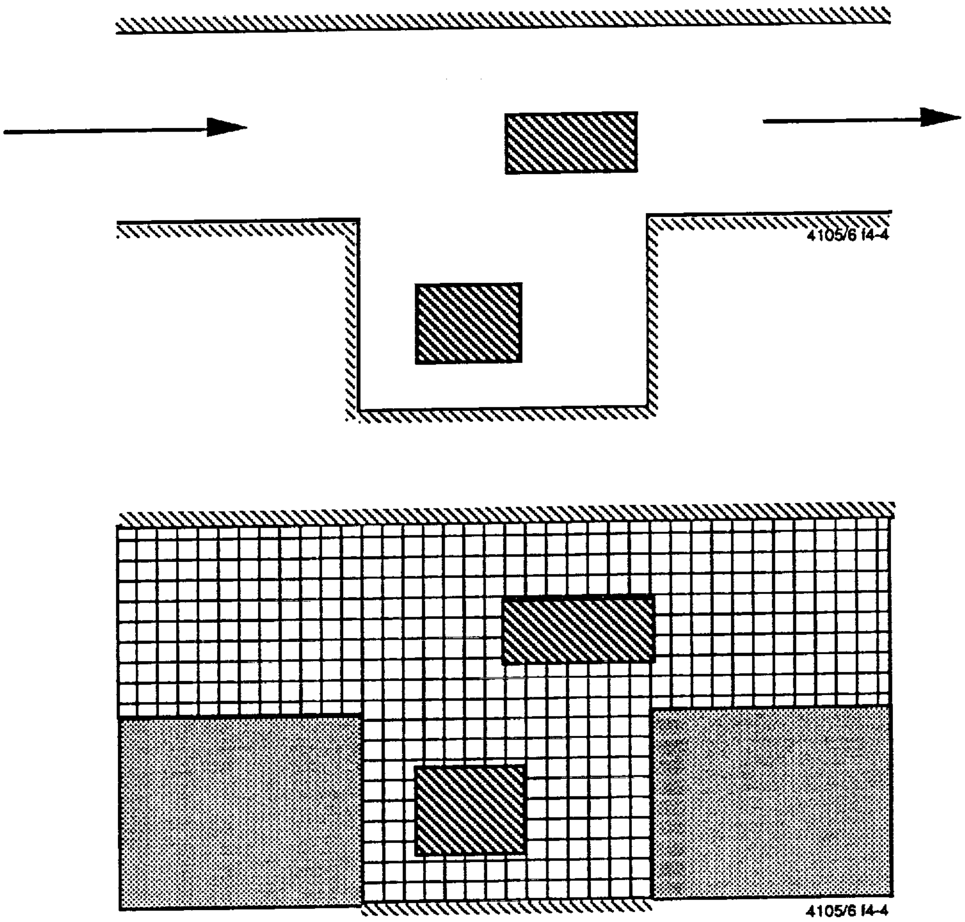


Figure 4-4. Illustration of the Use of Blocked-Off Regions

5. DISCRETIZATION METHODS

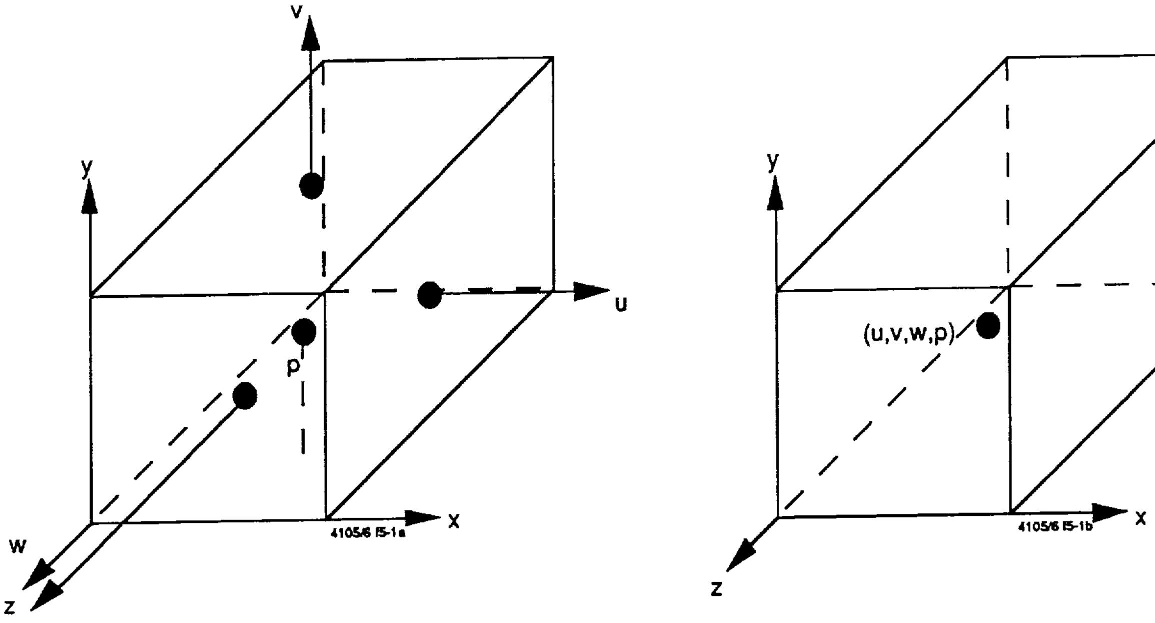
The fluid flows are governed by several physical conservation laws and these laws can be written as Partial Differential Equations (PDE's) as presented in Section 2. A numerical method to solve these PDE's consists of the discretization of the PDE's on a given grid, formation of corresponding linearized algebraic equations, and the solution of the algebraic equations. This way, a final set of discrete numbers on a grid is obtained which represents the numerical solution of the PDE's. In this section, the discretization of the governing equations is presented. The finite-volume approach is adopted due to its attractive capability of preserving the conservation property. In the following, the storage locations of the dependent variables are first discussed. The discretization process then follows in more detail.

5.1 Staggered versus Colocated Grid Approach

The staggered grid approach was widely used for incompressible flow simulations in the past. With the staggered grid approach, proposed by Harlow and Welch (1965), the velocity components are stored at positions between the pressure nodes as illustrated in Figure 5-1a. Such an approach ensures the pressure is readily available for momentum equations and velocity components are available for the continuity equation without interpolation. As a result, a proper pressure coupling is guaranteed and the well-known checkerboard instability is prevented. However, the disadvantages of the staggered grid approach are well known and the following are just two examples.

- It is not easy to extend the staggered grid approach to non-orthogonal curvilinear grids. Several proposed extensions are extremely complex to apply and can cause loss of accuracy.
- Many state-of-the-art CFD methodologies are difficult to apply with a staggered grid approach such as multigrid, local grid refinement, and multi-zoning.

The current state-of-the-art approach is based on the colocated grid (or non-staggered grid) proposed by Rhie and Chow (1983) and later by Peric (1985). The grid arrangement of this approach is illustrated in Figure 5-1b. This colocated grid approach has many advantages over the staggered grid approach and is used in the REFLEQS code. This approach evaluates the cell face velocity using momentum equations and consequently the cell face velocity is directly linked to a third pressure derivative term. This linkage ensures that the checkerboard instability is eliminated. With a colocated grid approach and Cartesian velocity as dependent variables, many coding complications are avoided and more accurate solutions can be obtained, particularly for viscous flows.



(a) (b)

Figure 5-1. Illustration of Variable Storage Locations for: (a) Staggered; and (b) Colocated Grid

5.2 A General Convection-Diffusion Equation

It is noted that all governing equations, except continuity equation, can be expressed in a general form as

$$\frac{\partial \rho\phi}{\partial t} + \frac{\partial \rho u\phi}{\partial x} + \frac{\partial \rho v\phi}{\partial y} + \frac{\partial \rho w\phi}{\partial z} = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial \phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma \frac{\partial \phi}{\partial z}\right) + S_\phi \qquad (5.1)$$

where ϕ can stand for Cartesian velocity components, total enthalpy, turbulence kinetic energy, mass concentration, *etc.* Γ is the effective diffusivity and S_ϕ is the source term. Therefore, the above equation, in convection-diffusion form, will be considered for discretization. The continuity equation will be discussed later in Section 5.7.

First of all, Equation (5.1) needs to be transformed to BFC coordinates using the new independent variable $\xi(x, y, z)$, $\eta(x, y, z)$, and $\zeta(x, y, z)$. Without detailed derivation, it is sufficient to write down the transformed equation in (ξ, η, ζ) coordinates as follows:

$$\frac{1}{J}\frac{\partial}{\partial t}(J\rho\phi) + \frac{1}{J}\frac{\partial}{\partial \xi_k}\left\{J\rho\left(\vec{V}\cdot\vec{\varepsilon}^{\,k}\right)\phi\right\} = \frac{1}{J}\frac{\partial}{\partial \xi_k}\left[\Gamma J g^{jk}\frac{\partial \phi}{\partial \xi_j}\right] + S_\phi \tag{5.2}$$

with

$$g^{jk} = \vec{\varepsilon}^{\,j}\cdot\vec{\varepsilon}^{\,k} \tag{5.3}$$

The discretization involves an integration of Equation (5.2) over a control volume as shown in Figure 5-2. That is:

$$\iiint_V [\mathit{Equation\ (5.2)}]\cdot J\; d\,\xi\, d\,\eta\, d\,\zeta \tag{5.4}$$

Note that $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ has been chosen on every control volume with the REFLEQS code. This is purely for computational convenience. In the next several sections, individual terms of the above integration will be discussed.

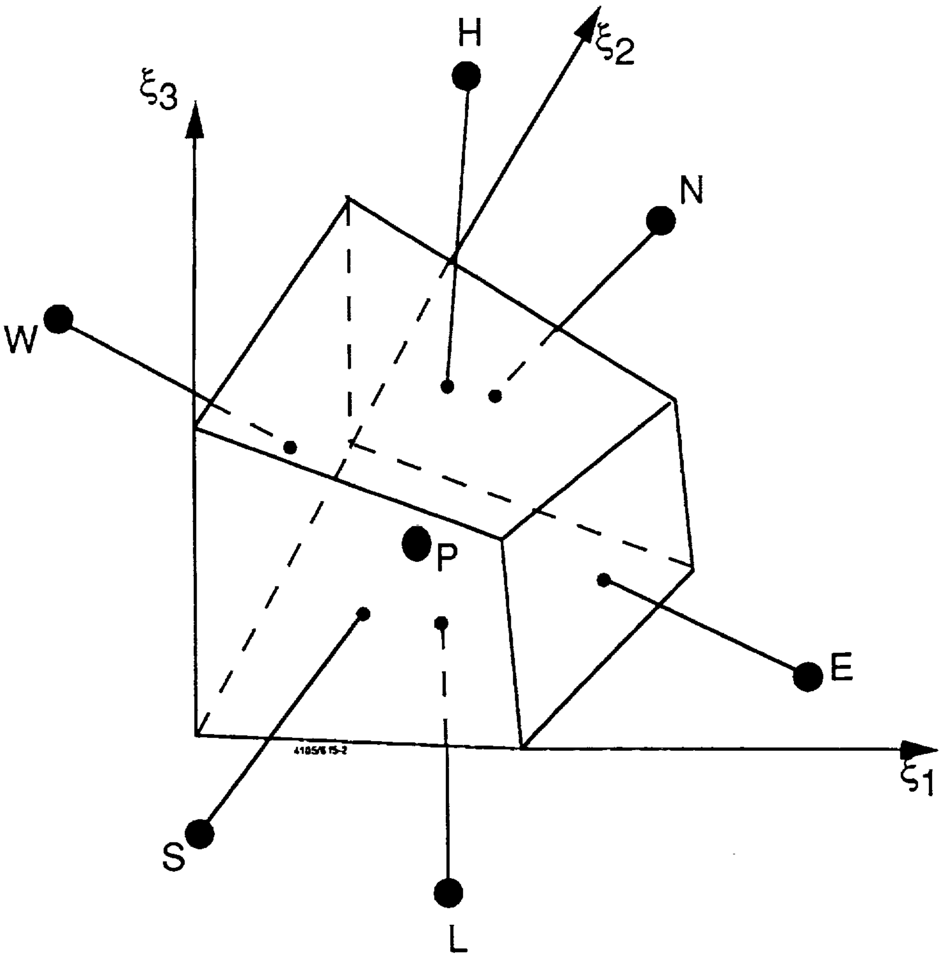


Figure 5-2. The Labeling Scheme of a Control Volume

5.3 Transient Term

Consider the discretization of the transient term of Equation (5.2), *i.e.*,

$$T = \oiiint_{\forall} \frac{1}{J}\frac{\partial}{\partial t}(J\,\rho\,\phi)\cdot J\,d\xi\,d\eta\,d\zeta = \frac{\partial}{\partial t}\oiiint_{\forall}(J\,\rho\phi)\,d\xi\,d\eta\,d\zeta = \frac{\rho\phi\forall - \rho^{o}\phi^{o}\forall^{o}}{\Delta t} \qquad (5.5)$$

In the above, superscript "*o*" stands for a value at an old time, variables without "*o*" superscipt are at the new time, and $\forall$ stands for the volume. Note that the above discretization holds true for both Euler first-order and Crank-Nicholson second-order schemes.

5.4 Convection Term and Different Convection Schemes

By defining a physcial contravariant velocity component U^k such that $\vec{V} = U^k \vec{e}_k$, the convection term can be rewritten as

$$C = \frac{1}{J}\frac{\partial}{\partial \xi_k}\left[J \rho\left(\vec{V} \cdot \vec{\varepsilon}_k\right)\phi\right] = \frac{1}{J}\frac{\partial}{\partial \xi}\left[\frac{J}{h_1}\rho U^1 \phi\right] + \frac{1}{J}\frac{\partial}{\partial \eta}\left[\frac{J}{h_2}\rho U^2 \phi\right] + \frac{1}{J}\frac{\partial}{\partial \zeta}\left[\frac{J}{h_3}\rho U^3 \phi\right] \tag{5.6}$$

Integration of the above term over a control volume gives,

$$C = C_e - C_w + C_n - C_s + C_h - C_l = G_e\phi_e - G_w\phi_w + G_n\phi_n - G_s\phi_s + G_h\phi_h - G_l\phi_l \tag{5.7}$$

with G_e defined, for example, as

$$G_e = \left(\frac{J}{h_1}\rho\, U^1\right)_e \tag{5.8}$$

where$G$'s represent the mass flux through a face of the control volume. The subscripts, $e, w, n, s, h,$ and l represent the values evaluated at east, west, north, south, high, and low faces as shown in Figure 5-2. The evaluation of the mass fluxes G, will be described in Section 5.7 and the evaluation of ϕ values at control volume faces are described next.

Without loss of generality, the evaluation of ϕ_e is discussed on a uniform grid as shown in Figure 5-3. The following various schemes can be used for this purpose:

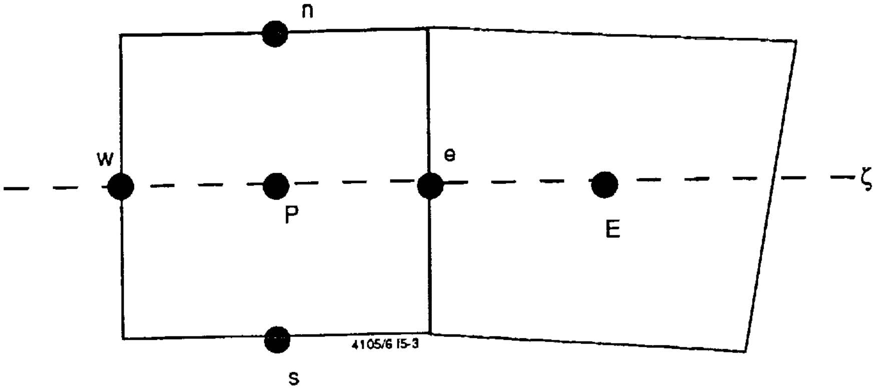


Figure 5-3. An Illustration of the Cell Face Value Evaluation

5.4.1 First-Order Upwind Scheme

First-order upwind approach will evaluate ϕ_e using either ϕ_P or ϕ_E depending on the flow direction at point e. Mathematically, it can be expressed as

$$C_e^{UP} = \begin{cases} G_e \phi_P \; if \; G_e > 0 \\ G_e \phi_E \; if \; G_e < 0 \end{cases} \tag{5.9}$$

or in more convenient form as

$$C_e^{UP} = G_e \frac{\phi_E + \phi_P}{2} - \left| G_e \right| \frac{\phi_E - \phi_P}{2} \tag{5.10}$$

This scheme has first-order accuracy and is one of the most stable schemes.

5.4.2 Central Difference Scheme

Pure central difference approach will evaluate ϕ_e by averaging the values at point P and E. That is

$$C_e^{CN} = G_e \frac{\phi_E + \phi_P}{2} \tag{5.11}$$

It is widely known that pure central difference scheme may cause unphysical oscillations. For most problems, some damping (or artificial viscosity) is needed for stability. In practice, central difference scheme with damping is constructed as

$$C_e = d_e C_e^{UP} + \left(1 - d_e\right) C_e^{CN} \tag{5.12}$$

with d_e, the damping coefficient, representing the fraction of Upwind scheme used.

Equation (5.12) can be re-written as

$$C_e = G_e \frac{\phi_P + \phi_E}{2} - d_e \left| G_e \right| \frac{\phi_E - \phi_P}{2} \tag{5.13}$$

Comparing Equation (5.10) with Equation (5.11), it is clear why the scheme is called central difference plus damping. $d_e = 0$ yields the pure central scheme, while $d_e = 1.0$ results in the upwind scheme. In this report, central difference scheme often refers to pure central plus the damping term as given above.

5.4.3 Smart Scheme with Minmod Limiter

It is observed that the damping coefficient is constant for a central difference scheme. There are many flow situations where damping is needed only in certain limited regions. Therefore, a smart scheme is designed to adaptively calculate damping coefficient depending on the local flow property. The minmod limiter can be used to obtain damping coefficient d_e as

$$d_e = minimode \left(1, \gamma_e\right) \tag{5.14}$$

with

$$\gamma_e = \frac{\phi_{E-U} - \phi_{P-U}}{\phi_E - \phi_P} \quad and\ U = sign\left(G_e\right) \tag{5.15}$$

The MINMOD function is defined as: *MINMOD* (α, β) = *sign* (α) *max* $[o, min\ (|\alpha|, \beta)]$

This scheme will reduce to upwind scheme if there exists local extrema such as across shocks ($\gamma_e < 0$), to central difference scheme when $\gamma_e \geq 1$, and to second-order upwind scheme for $0 < \gamma_e < 1$.

5.4.4 Other High-Order Schemes

For compressible flows with shocks, several high-order schemes with limiters are very accurate for shock capturing. Three such schemes are introduced in the following.

Osher-Chakravarthy Scheme: Osher-Chakravarthy scheme evaluates the damping coefficient as

$$d_e = \frac{1-\eta}{2} minmod\left(\beta, \gamma_e\right) + \frac{1+\eta}{2} minmod\left(1, \beta\gamma_e\right) \tag{5.16}$$

with $\eta = \frac{1}{3}$ and $\beta = \frac{3-\eta}{1-\eta}$

Roe's Superbee Scheme: Roe's Superbee scheme is used to obtain the damping coefficient as

$$d_e = Max\left[o, min\left(2\gamma, 1\right), min\left(\gamma, 2\right)\right] \tag{5.17}$$

van Leer's MUSCL Scheme: van Leer's MUSCL scheme is used to obtain the damping coefficient as

$$d_e = \frac{|\gamma_e| + \gamma_e}{(1 + \gamma_e)} \tag{5.18}$$

5.5 Diffusion Terms

The diffusion term in Equation (5.2) can be split into two parts: main diffusion (j = k); and cross diffusion (j ≠ k). Let's consider main diffusion first, *i.e.*,

$$D_M^k = \frac{1}{J}\frac{\partial}{\partial \xi_k}\left[\Gamma J g^{kk} \frac{\partial \phi}{\partial \xi_k}\right], \quad k = 1, 2, or\, 3 \tag{5.19}$$

Without loss of generality, k = 1 term will be used for derivation. Integration of D_M^1 term over a control volume, as shown in Figure 5-2, leads to

$$\iiint D_M^1 J\, d\xi d\eta d\zeta = \left[\Gamma J g^{11} \frac{\partial \phi}{\partial \xi}\right]_e - \left[\Gamma J g^{11} \frac{\partial \phi}{\partial \xi}\right]_w \tag{5.20}$$

It is easy to show that

$$J g^{11} = \frac{A^2}{J} = \frac{A}{h_1 \sin\, \alpha_1} \tag{5.21}$$

where A is the area of the control volume face along the ξ direction and α_1 represents the angle between vector $\vec{e}_1$ and the plane formed by $\vec{e}_2$ and $\vec{e}_3$. By defining

$$D_\xi = \frac{\Gamma \cdot A}{h_1 \sin \alpha_1} \tag{5.22}$$

we have

$$\iiint D_M^1 J\, d\xi d\eta d\zeta = D_\xi^e\left(\phi_E - \phi_P\right) - D_\xi^w\left(\phi_P - \phi_W\right) = -\left(D_\xi^e + D_\xi^w\right)\phi_P + D_\xi^e\,\phi_E + D_\xi^w\,\phi_W \quad (5.23)$$

Therefore, D_ξ, the main diffusion coefficient needs to be evaluated at the faces of each control volume.

The cross diffusion term is written as

$$D_c^{jk} = \frac{1}{J}\frac{\partial}{\partial \xi_k}\left[J\,\Gamma\, g^{jk}\frac{\partial \phi}{\partial \xi_j}\right],\ j \neq k \quad (5.24)$$

Consider D_c^{21} for illustration purpose:

$$\iiint D_c^{21} J\, d\xi d\eta d\zeta = \left(\Gamma J g^{21}\frac{\partial \phi}{\partial \eta}\right)_e - \left(\Gamma J g^{21}\frac{\partial \phi}{\delta \eta}\right)_w \quad (5.25)$$

By defining $D^{21} = \frac{\Gamma}{4} J g^{21}$ and assuming $\phi_{ne} = \frac{1}{4}\left(\phi_P + \phi_E + \phi_N + \phi_{NE}\right)$, *etc.*, we have

$$\iiint D_c^{21} J\, d\xi d\eta d\zeta = D_e^{21}\left(\phi_N + \phi_{NE} - \phi_S - \phi_{SE}\right) - D_w^{21}\left(\phi_N + \phi_{NW} - \phi_S - \phi_{SW}\right) \quad (5.26)$$

Other cross terms can be similarly discretized.

5.6 Final Linear Equation Set

From Subsections 5.3 to 5.5, the general convection-diffusion Equation (5.2) has been discretized term by term on a control volume. Assembling all the terms together, the final linear equation is resulted and can be written in general form as:

$$
\begin{aligned}
A_P\phi_P = {} & A_W\phi_W + A_E\phi_E + A_S\phi_S + A_N\phi_N + A_L\phi_L + A_H\phi_H \\
& + A_{SW}\phi_{SW} + A_{SE}\phi_{SE} + A_{NW}\phi_{NW} + A_{NE}\phi_{NE} \\
& + A_{LS}\phi_{LS} + A_{LN}\phi_{LN} + A_{HS}\phi_{HS} + A_{HN}\phi_{HN} \\
& + A_{WL}\phi_{WL} + A_{WH}\phi_{WH} + A_{EL}\phi_{EL} + A_{EH}\phi_{EH} \\
& + S^{\phi}
\end{aligned}
\qquad (5.27)
$$

5.7 Discretization of Mass Conservation and Mass Flux Evaluation

As mentioned before, mass conservation equation is a special one which can not be written as the general convection-diffusion form (Equation (5.2)). Moreover, in a pressure-based method mass conservation is used to determine pressure field. For this purpose, the mass conservation equation needs special attention. The general mass conservation equation can be transformed into BFC coordinates as

$$
\frac{1}{J}\frac{\partial J\rho}{\partial t} + \frac{1}{J}\frac{\partial}{\partial \xi_k}\left(J\rho\, \vec{V}\cdot\vec{\varepsilon}^{\,k}\right) = 0 \qquad (5.28)
$$

Therefore, integration over a control volume yields

$$
\frac{\rho\forall - \rho^{o}\forall^{o}}{\Delta t} + G_e - G_w + G_n - G_s + G_h - G_l = 0 \qquad (5.29)
$$

With $G_e = \left(\frac{J}{h_1}\,\rho\, U^1\right)_e$ for example. Note that G's are the mass flux through a control volume face as mentioned in Section 5.4.

Our next task is to evaluate mass flux G's at control volume faces such that the checkerboard instability is eliminated for a colocated grid arrangement. This is achieved by averaging momentum equation to the cell faces and relating the cell face velocity directly to the local pressure gradient. This procedure is presented next.

The discretized x-momentum equation can be expressed at cell P as

$$\left[a_p^u + \left(\frac{\rho\forall}{\Delta t}\right)_p\right] u_p = \sum_{nb} a_{nb}^u u_{nb} - \forall_p \left(\frac{\partial p}{\partial x}\right)_p + \left(\frac{\rho\forall}{\Delta t}\right)_p u_p^o + S_u^u \tag{5.30}$$

where *nb* refers to all the neighboring cells and the transient term is explicitly expressed to ensure that the momentum equation at cell face includes the effect of a time step. Dividing the above equations by a_p^u yields

$$\left(1 + cd_p^u\right) u_p = \sum_{nb} \frac{a_{nb}^u}{a_p^u} u_{nb} - d_p^u \left(\frac{\partial p}{\partial x}\right)_p + cd_p^u u_p^o + \frac{S_u^u}{a_p^u} \tag{5.31}$$

where

$$c = \frac{\rho}{\Delta t} \quad , \quad d_p^u = \frac{\forall}{a_p^u} \tag{5.32}$$

The above momentum equation is at P-cell but in reality it applies to every point. Therefore, at cell face f, for example, we have

$$\left(1 + cd_f^u\right) u_f = \left(\sum_{nb} \frac{a_{nb}^u}{a_p^u} u_{nb}\right)_f - d_f^u \left(\frac{\partial p}{\partial x}\right)_f + cd_f^u u_f^o + \left(\frac{S_u^u}{a_p^u}\right)_f \tag{5.33}$$

Since we do not know how to evaluate $\left(a_{nb}^u\right)_f$, for example, the following quantities at f will be obtained by averaging the same quantity from two neighboring nodal points.

$$d_f^u \, , \; \left(\sum_{nb} \frac{a_{nb}^u}{a_p^u} u_{nb}\right)_f , \; \left(\frac{S_u^u}{a_p^u}\right)_f \tag{5.34}$$

Therefore, by using Equation (5.31) we have

$$\left[\sum_{nb} \frac{a_{nb}^u}{a_p^u} u_{nb} + \frac{S_u^u}{a_p^u}\right]_f = \overline{\left(\sum_{nb} \frac{a_{nb}^u}{a_p^u} u_{nb} + \frac{S_u^u}{a_p^u}\right)} = \overline{\left(\left(1 + cd_p^u\right) u_p + d_p^u \left(\frac{\partial p}{\partial x}\right)_p - cd_p^u \, u_p^o\right)} \tag{5.35}$$

where overbar means average from point P and E. Hence Equation (5.33) can be written as

$$\left(1 + cd_f^u\right) u_f = \overline{\left(1 + cd_p^u\right) u_p + d_p^u \left(\frac{\partial p}{\partial x}\right)_p - cd_p^u \, u_p^o} - \overline{d_p^u} \left(\frac{\partial p}{\partial x}\right)_f + cd_f^u \, u_f^o \tag{5.36}$$

and the above equation is further reduced to

$$u_f = \overline{u_p} + d^u \left[\overline{\left(\frac{\partial p}{\partial x}\right)_p} - \left(\frac{\partial p}{\partial x}\right)_f\right] + cd^u \left(u_f^o - \overline{u_p^o}\right) \tag{5.37}$$

with

$$d^u = \frac{\overline{d_p^u}}{1 - c\overline{d_p^u}} \tag{5.38}$$

Now it is clear from the above equations that the cell face velocity is obtained from an average of the two neighboring point velocity plus a second-order pressure correction and a second-order previous time velocity correction. The pressure term serves as the mechanism to avoid the checkerboard problem and the previous time velocity serves as a mechanism to obtain a time-independent steady solution.

The y-component and z-component velocities can be similarly evaluated and the contravariant velocity component at east face can be thus evaluated as:

$$U_f^1 = u_f \cdot F_{1x} + v_f \cdot F_{1y} + w_f \cdot F_{1z} \tag{5.39}$$

with

$$F_{1x} = h_1 \vec{\varepsilon^1} \cdot \vec{i} \; ; \; F_{1y} = h_1 \vec{\varepsilon^1} \cdot \vec{j} \; ; \; F_{1z} = h_1 \vec{\varepsilon^1} \cdot \vec{k} \tag{5.40}$$

6. VELOCITY PRESSURE COUPLING

In Section 5, the mass conservation equation and other governing equations are discretized on an arbitrary control volume. It is noted that both pressure and Cartesian velocity components appear in the mass conservation equation and the main variable is not clearly identifiable. It is the task of this section to reformulate the mass conservation equation such that the pressure increment (or pressure correction) is the main variable. As a result, the velocity and pressure fields are linked together. This approach is classified as a pressure-based or pressure correction method.

For ease of presentation, only incompressible flow and Cartesian grids are used. Extension to BFC and compressible flows are straightforward. The discretized equations for mass and momentum conservation can be re-written as

$$\sum_f u_f = 0$$

$$\tilde{A}^i_p u_i = \sum_{nb} A^i_{nb} u_{i,nb} - D^i \nabla_i p + \hat{S} \qquad (6.1)$$

where $\sum_f$ stands for the summation over all faces, $\sum_{nb}$ is the summation over all the neighboring cells.

In the following, three velocity-pressure coupling algorithms are described which are available from the REFLEQS code.

6.1 SIMPLE Algorithm

The SIMPLE algorithms (Patankar *et al* , 1972) relies on using a guessed value for pressure p^*, and using the momentum equation can be used to obtain the velocity,

u_i'. If the pressure has an increment p', the velocity components will also have a corresponding increment u_i'. This increment can be written as

$$\tilde{A}_p^i u_i' = \sum_{nb} A_{nb}^i u_{i,nb}' - D^i \nabla_i p' \tag{6.2}$$

In the SIMPLE algorithm $u_{i,nt}'$ are neglected and the above increment equation is approximated as

$$\tilde{A}_p^i u_i' = -D^i \nabla_i p' \quad \text{or} \quad u_i' = -\frac{D^i}{\tilde{A}_p^i} \nabla_i p' \tag{6.3}$$

Therefore, at cell faces, the increment is

$$u_f' = -\overline{\left(\frac{D^i}{\tilde{A}_p^i}\right)} \nabla_f p' \tag{6.4}$$

We would like to enforce the new velocity,

$$u_f = u_f^* + u_f' \tag{6.5}$$

to satisfy the mass conservation, *i.e.*,

$$-\sum_f u_f' = \sum_f u_f^* \tag{6.6}$$

or

$$\sum_f \overline{\left(\frac{D^i}{\tilde{A}_p^i}\right)} \nabla_f p' = \sum_f u_f^* \tag{6.7}$$

Note that the right hand side of the above equation is simply the mass imbalance. The above pressure correction equation is a Poisson equation and any linear equation solver can be used to solve it. After p' is obtained, a new pressure and a new mass conserved face velocity u_f are obtained using Equations (6.4) and (6.5).

6.2 SIMPLEC Algorithm

SIMPLEC algorithm (van Doormal and Raithby, 1984) differs from SIMPLE in the approximation of the momentum increment equation. With SIMPLEC algorithm, it is assumed that

$$\sum_{nb} A^i_{nb} u'_{i,nb} = \sum_{nb} A^i_{nb} u'_i \tag{6.8}$$

As a result, the increment momentum Equation (6.2) becomes,

$$\left(\tilde{A}^i_p - \sum_{nb} A^i_{nb} \right) u'_i = -D^i \nabla_i p' \tag{6.9}$$

Since $\tilde{A}^i_p$ is always larger than $\sum_{nb} A^i_{nt}$, the coefficient

$$\hat{A}^i_p = \tilde{A}^i_p - \sum_{nb} A^i_{nb} \tag{6.10}$$

is always positive.

With $\hat{A}^i_p$ replacing $\tilde{A}^i_p$ the same derivation performed for SIMPLE can be used and the pressure correction equation is also given by Equation (6.7). It is obvious that the modification from SIMPLE to SIMPLEC is very small. However, in most problems, the efficiency improvement is significant.

6.3 PISO Algorithm

PISO is a Predictor-Corrector algorithm and is more mathematically sound compared with SIMPLE and SIMPLEC since no assumptions are made to the momentum increment equation. The algorithm is described as follows: with a guessed pressure, p^*, the momentum equation is first solved to get u^*_i. This is the predictor step and can be written as

Predictor:

$$\tilde{A}^i_p u^*_i = \sum_{nb} A^i_{nb} u^*_{i,nb} - D^i \nabla_i p^* + \tilde{S}_i \tag{6.11}$$

Next, two correctors are performed. First-corrector will provide a new u^{**}_i and p^{**} such that

$$\begin{cases} \sum_f u^{**}_f = 0 \\ \tilde{A}^i_p u^{**}_i = \sum_{nb} A^i_{nb} u^*_{i,nb} - D^i \nabla_i p^{**} + \tilde{S}_i \end{cases} \tag{6.12}$$

and second-corrector will get the velocity u_f and pressure, p, such that

$$\begin{cases} \sum_f u_f = 0 \\ \tilde{A}^i_p u_i = \sum_{nb} A^i_{nb} u^{**}_{i,nb} - D^i \nabla_i p + \tilde{S}_i \end{cases} \tag{6.13}$$

The similarity between PISO and SIMPLE is obvious if the new momentum equation is subtracted from the old momentum equation. That is:

First corrector: $$\tilde{A}^i_p u_i^{*\prime} = -D^i \nabla_{,i} p^{\prime *} \text{ with } \begin{cases} u_i^{\prime *} = u_i^{**} - u_i^{*} \\ p^{\prime *} = p^{**} - p^{*} \end{cases} \qquad (6.14)$$

Second corrector: $$\tilde{A}^i_p u_i^{\prime} = \Sigma A^i_{nb} u^{*\prime}_{i,nb} - D^i \nabla_{,i} p^{\prime} \text{ with } u_i^{\prime} = u_i - u_i^{**} ; p^{\prime} = p - p^{**} \qquad (6.15)$$

Since $U^{*\prime}_{i,nb}$ is known at second-corrector, the above increment equation has the similar form of SIMPLE and SIMPLEC. However, it is clear that PISO does not make any approximation regarding the momentum increment equation. Also it is worth mentioning that the number of correctors does not have to be two and more correctors can be performed.

7. BOUNDARY CONDITIONS

In order to completely specify the mathematical problem, it is necessary to supply the conditions at the boundaries of the solution domain for all the dependent variables. These conditions are usually either specification of the value of the dependent variable, or the value of the associated flux, or a combination between the two.

REFLEQS offers a variety of boundary conditions which include:

a. Inlet: where all the dependent variables are fixed except pressure;

b. Exit: where all the dependent variables are extrapolated from the interior except pressure. Pressure can be fixed (FIXP), extrapolated (EXTRAP), or both (FIXPEX);

c. Symmetry: where the normal gradient of all the variables are equal to zero except the velocity component normal to the boundary which is set to zero;

d. Solid Wall: where the velocity is fixed, pressure is extrapolated, and the value or the flux is known for other dependent variables; and

f. Periodic Boundary Condition: where the two boundaries exchange the information.

For each dependent variable, at all boundaries, appropriate modifications of the discretized equation at the "near boundary" nodes is required. For most of the dependent variables, the boundary conditions are implemented by modifying the source terms SU_ϕ and SP_ϕ:

$$\phi_p = \frac{\sum\limits_{nb} A_{nb}\,\phi_{nb} + SU_\phi}{\sum\limits_{nb} A_{nb} - SP_\phi} \qquad (7.1)$$

A brief introduction of the above boundary conditions are provided below.

7.1 Inlet with Specified Mass Flow Rate (FIXM)

At an inlet, the amount of incoming mass flow rate $\dot{m}_{IN}$ through the boundary cell face and the incoming ϕ_B property should be specified. In this case, the finite-difference coefficient connecting the boundary node to its neighboring internal node is set to zero and then the SU_ϕ and SP_ϕ source terms are modified as:

$$SU_\phi = SU_\phi + \dot{m}_{IN}\,\phi_B \tag{7.2}$$

$$SP_\phi = SP_\phi - \dot{m}_{IN} \tag{7.3}$$

The pressure correction equation is not modified in this manner. A link coefficient with the boundary node is set to zero for p'-equations.

7.2 Exit Boundary

At the exit, the fluid flows out of the calculation domain, and the information of most dependent variables is not available a priori. Fortunately, the information there is often not important if the convection is the dominant activity. Therefore, all the dependent variables except pressure can be extrapolated. This is equivalent to setting the boundary link coefficients to zero. As for pressure, it is fixed at the exit for most incompressible or subsonic Mach number flows (this is called FIXP boundary condition). However, for supersonic flows, pressure should be extrapolated at the exit (EXTRAP boundary condition). REFLEQS offers a third boundary condition, FIXPEX, to take care of the situation where the exit flow condition supersonic or subsonic is not known a priori.

7.3 Symmetry Condition

At symmetry boundary, a zero mass flux through the boundary is assumed. At the same time, the normal derivative of all the scalar variables is set to zero.

7.4 Solid Wall Boundary

For laminar flows or turbulent flows without using wall functions, the no-slip assumption is made at the solid walls. As a result, the mass flux through a wall is set to zero. On the other hand, the pressure at the wall is not known, a priori, and extrapolation from the internal cells is used. For other dependent variables, a constant value, a constant flux, or a combination of the two is needed at the wall. For example, for the energy equation, either constant temperature or adiabatic condition can be assumed. For turbulent flows with wall function boundary condition, a special procedure is carried out.

The wall functions described by Launder and Spalding (1974), are derived from experimental and analytical knowledge of the one-dimensional Couette flow which exists near the wall. A semi-empirical universal function of non-dimensional distance normal to the wall, y^+, is:

$$y^+ \equiv \frac{\rho \; \delta y \cdot u_\tau}{\mu} \tag{7.4}$$

In the above definition, δy is the distance normal to the wall and u_τ is the "friction velocity" given by:

$$u_\tau = \left(\frac{\tau_w}{\rho} \right)^{1/2} \tag{7.5}$$

In the internal sublayer *($y^+ > 11.63$)* the velocity variation may be described by a logarithmic relationship *i.e.*:

$$u = \frac{u_\tau}{\kappa} \ell n \left(E y^+ \right) \tag{7.6}$$

where $E = 9.70$and $\kappa = 0.4034$ are experimentally determined constants.

In both the viscous *($y^+ \leq 11.63$)* and internal *($y^+ > 11.63$)* sublayers, the shear stress is calculated from the product of effective viscosity μ_{eff} and normal velocity gradient $\partial u / \partial y$, *i.e.*:

$$\tau_w = \mu_{eff} \frac{\partial u}{\partial y} \qquad (7.7)$$

where

$$\mu_{eff} = \begin{cases} \mu \ \text{ for } y^+ \leq 11.63 \\ \mu_{turb} \text{ for } y^+ > 11.63 \end{cases} \qquad (7.8)$$

Near the wall, the transport equation for the turbulent kinetic energy, *k*, reduces to a balance between the local production and dissipation of *k* to give:

$$\mu_t \left(\frac{\partial u}{\partial y} \right)^2 = \rho \varepsilon \qquad (7.9)$$

The velocity gradient may be replaced from Equation (7.7) and the dissipation rate from:

$$\mu_t = C_\mu \, \rho \, \frac{k^2}{\varepsilon} \qquad (7.10)$$

to give:

$$\tau_w = C_\mu^{1/2} \, \rho \, k = \rho \, u_\tau^2 \qquad (7.11)$$

Hence, it follows from Equation (7.5)

$$\tau_w = \frac{\rho\, C_\mu^{1/4}\, k^{1/2}\, \bar{u}}{\frac{1}{\kappa}\,\ell n\left(Ey^{+}\right)} \tag{7.12}$$

7.5 Periodic Boundary

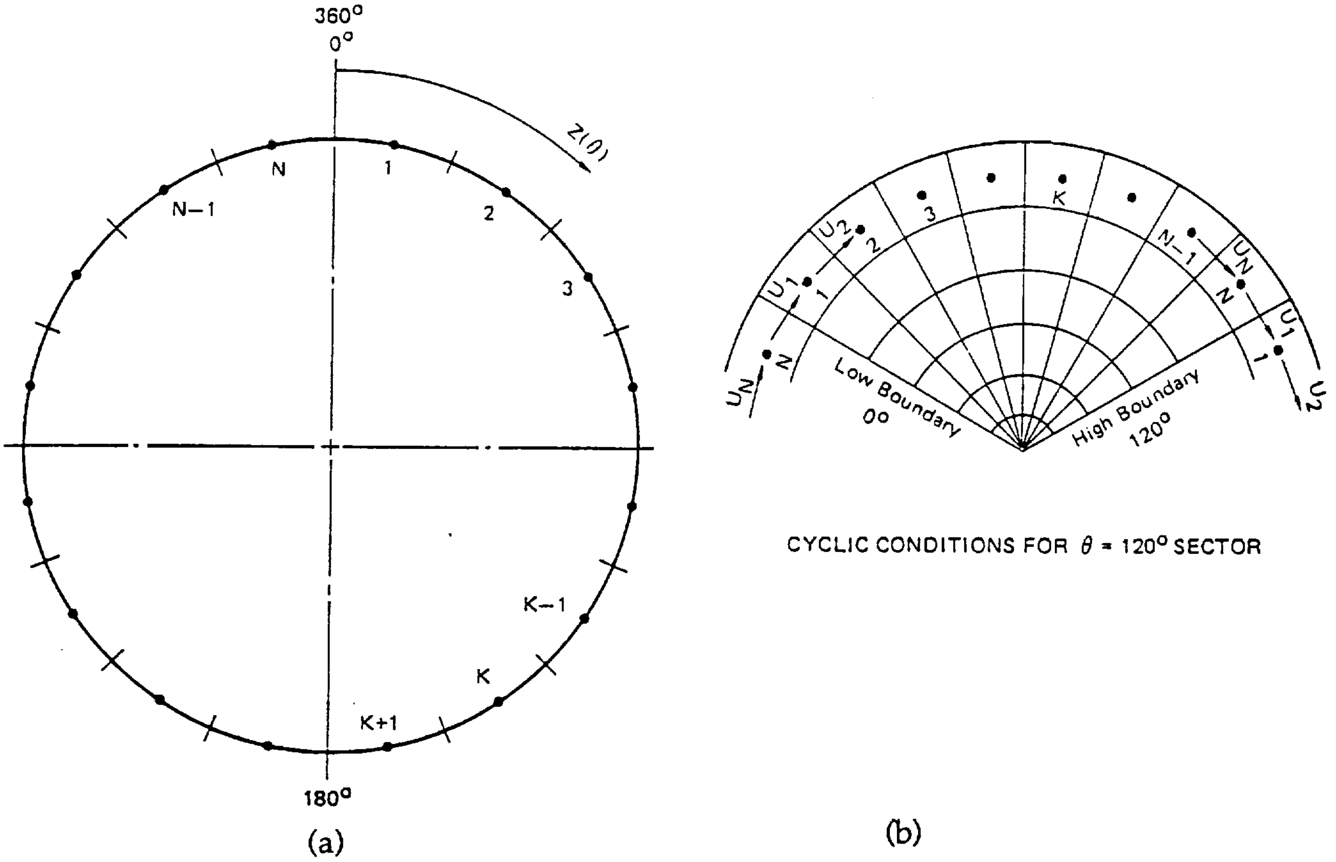
The periodic (cyclic) boundary conditions appear in the circumferential direction if the two ends of the calculation domain in the z-direction join up with one another. This can occur in a polar-coordinate direction in which the whole angular extent from 0 to 360° is to be considered (Figure 7-1a) or when "repetition" is present in the flow pattern in the angular coordinate direction (Figure 7-1b).

The general rule is that whenever identical conditions are to be expected at z = o and z = last z, the finite flux, equal on both sides, is to be expected at that surface, then the boundaries are cyclic.

In this circumstance, the boundary conditions can be specified as follows:

$$\phi_{LB} = \phi_N \;,\; \phi_{HB} = \phi_1 \;,\; \text{and}\;\; C_1^z = C_{HB}^z \tag{7.13}$$

where indices *HB* and *LB* denote high boundary (k = N) and low boundary (k = 1), C^z is the convective flux in the z-direction.



CYCLIC CONDITIONS FOR θ = 120° SECTOR

Figure 7-1. Grid Notation for Periodic Boundary Conditions in the Z-Direction

8. SOLUTION METHOD

In the previous sections, the discretization method, the derivation of the pressure-correction equation, and the implementation of the boundary conditions have been described. The task of this section is to assemble all the elements together and describe how the discretized governing equations are solved.

8.1 Overall Solution Process

The Navier-Stokes equations are a set of highly non-linear and coupled differential equations. In the discretization of the governing equations, it is seen that the coupling among different dependent variables is one iteration delayed and the non-linear terms are linearized. Therefore the above approach is usually called an iterative method. Due to the de-coupling procedure, the final linear equation set for each governing equation is well defined and can be solved separately using any linear equation solver. This approach is called segregated or equation-by-equation approach. In the following, the overall solution process of the segregated method is described to solve transient Navier-Stokes equations.

a. Prescribe an initial fluid flow field $\{\phi^{o}\}$ at the beginning of the computation (t^{o}).

b. Make one increment in time $(t^{n+1} = t^{n} + \Delta t)$ and we try to get solution $\{\phi^{n+1}\}$. For this purpose, an iterative method is used. Therefore, set $\{\phi^{n+1}\}_k = \{\phi^{n}\}$.

c. With known $\{\phi^{n+1}\}_k$, all the link coefficients A_{nb} are evaluated.

d. The momentum equations are solved first to obtain $\left\{u_i^{n+1}\right\}_{k+1}^{*}$ using the known pressure $\left\{p^{n+1}\right\}_k$. This velocity field will not satisfy the mass conservation.

e. Pressure correction is solved to obtain the new pressure $\left\{p^{n+1}\right\}_{k+1}$ and new velocities are obtained as $\left\{u_i^{n+1}\right\}_{k+1}$. This new velocity field may satisfy the continuity equation but not satisfy momentum equation.

f. Any other scalar equations, ϕ, such as energy, turbulence, combustion equations are solved to get $\left\{\phi^{n+1}\right\}_{k+1}$.

g. Steps c-f are repeated until all equations are satisfied or a predetermined convergence criterion is met. At this point, the solution at n+1 time step is calculated.

h. Steps b-g can be used to obtain a transient solution at any time locations.

8.2 Linear Equation Solvers

In the overal solution process, one important component is to solve the linear equation set resulted from each governing equations. This linear equation solver usually consumes a large fraction of cpu time and may require large amounts of memory. Therefore, the selection of linear equation solvers is always a trade-off among execution time, storage and accuracy needed. A direct solver such as Guassian Elimination is an accurate method but requires prohibitive storage and execution time. It is rarely a useful solver for segregated iterative approach. The best candidate for solving sparse banded matrix like ours is the iteration based solver. In the 60's and 70's Alternating Direction Implicit (ADI) method was very popular. Despite the fact that it requires very small storage and cpu time per iteration, ADI method becomes prohibitively slow to solve the large system. Therefore, ADI is also not a good candidate. In REFLEQS, two advanced solvers are developed. One is the Whole Field Solver (WFS) which is an enhanced Stone's Strongly Implicit Procedure (SIP). The WFS, unlike SIP, is free from any adjustable α- parameter and is symmetric. Another solver is the Conjugate Gradient Squared (CGS) solver which may provide an accurate solution for difficult linear system with reasonable storage and execution time cost. The two solvers are described in the following sections.

8.2.1 Whole Field Solver

Figure 8-1 illustrates the backsubstitution node arrangement and the nomenclature for the WFS discussion.

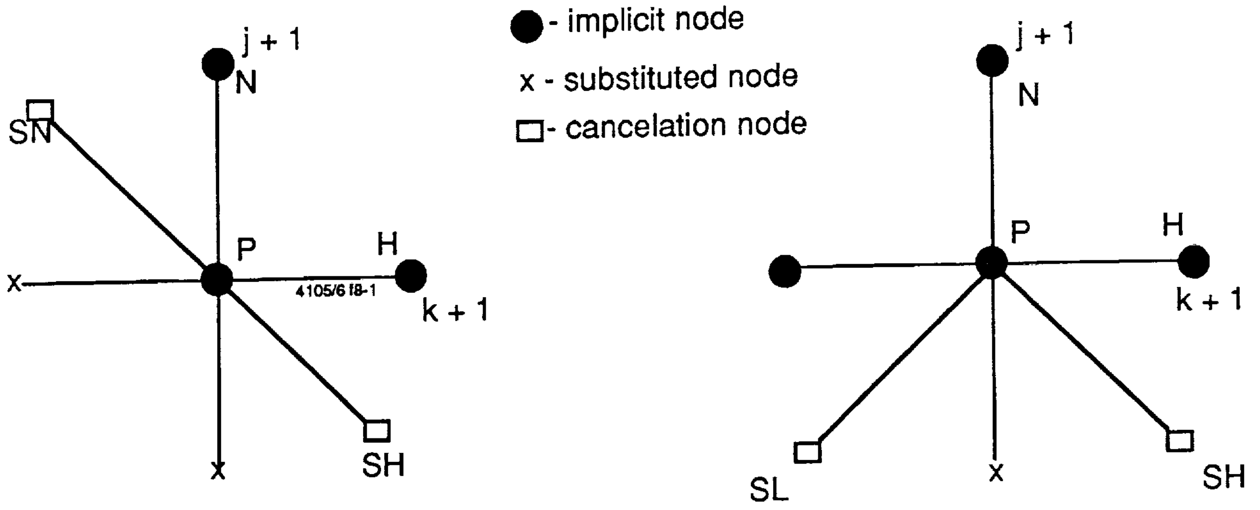


Figure 8-1. Grid Node Nomenclature for the WFS Discussion

With the i,j,k indices omitted the finite-difference (FD) equation can be expressed as:

$$a_p\phi = a_E\phi_{i+1} + a_W\phi_{i-1} + a_N\phi_{j+1} + a_S\phi_{j-1} + a_H\phi_{k+1} + a_L\phi_{k-1} + SU \qquad (8.1)$$

The difference between WFS and SIP solvers is the solution algorithm of the z-symmetric 3D WFS for the system of the FD equations can be summarized as follows.

a. In SIP only 3 points are treated implicitly in backsubstitution (P, H, N) while in WFS (P, L, H, N).

b. Nonsymmetric arrangement of cancellation nodes is used in SIP while in WFS (SH, SN) symmetric arrangement is retained, (SL, SH).

c. No extra memory storage is needed in WFS compared to SIP.

8.2.2 Conjugate Gradient Squared Solver

Conjugate gradient type solvers have become very popular in recent years since they often provide a faster convergence rate. They also have many other advantages (vectorization, no user specified parameters, *etc.*) over classic iterative methods. Among various nonsymmetric and nonpositive definite conjugate gradient solvers, the CGS algorithm has been found to yield faster convergence and is simpler to implement in the CFD code. The CGS algorithm applied to the system $AX = b$ is expressed as follows:

Initialization (n=0), $\quad r_o = b - AX_o$

$$q_o = p_{-1} = 0; \ \rho_{-1} = 1$$

$$\begin{aligned}
&\rho_n = \tilde{r}_o^{\,T} r_n; \ \beta_n = \frac{\rho_n}{\rho_{n-1}} \\
&u_n = r_n + \beta_n q_n \\
&p_n = u_n + \beta_n \left(q_n + \beta_n p_{n-1} \right) \\
&v_n = Ap_n \\
&\sigma_n = \tilde{r}_o^{\,T} v_n; \ \alpha_n = \frac{\rho_n}{\sigma_n}
\end{aligned}$$

Interaction (n>0),

$$\begin{aligned}
&q_{n+1} = u_n - \alpha_n v_n \\
&r_{n+1} = r_{n - \alpha_n} A\left(u_n + q_{n+1}\right) \\
&x_{n+1} = x_n + \alpha_n \left(u_n + q_{n+1} \right)
\end{aligned}$$

The convergence rate of conjugate gradient algorithms depends on the spectral radius of the coefficient matrix and can be effectively accelerated by preconditioning the system. To a matrix, the linear system $AX = b$ is transformed into an equivalent one whose coefficient matrix has a lower spectral condition number. This is obtained by using an approximate matrix p, and this transformed system is solvable by a direct method, and applying the CGS algorithm to the equivalent system $\tilde{A}X = p^{-1} AX = p^{-1} b = b\cdot$ In practice, each time in a CGS iteration one has to compute a matrix-vector product of the form $y = \tilde{A}X$, with $\tilde{A} = p^{-1}A$. The vector y is actually computed by solving a linear system of the form $py = AX$. Incomplete Cholesky decomposition is used as a preconditioner in the code. It has been found that preconditioned CGS is very efficient and robust.

9. CONCLUSIONS

As a result of this study, an advanced CFD-combustion code, REFLEQS, has been developed for analyzing two-phase spray flows in rocket combustors. State-of-the-art numerical schemes and solution algorithms were incorporated into the code for accurate flow simulations in complex geometries. Colocated grid arrangement, strong conservation of momentum equations and high order numerical schemes were used to ensure good solution accuracy.

To simulate flow and combustion in liquid rocket engines, several physical models were incorporated into the code including:

- turbulence models (Baldwin-Lomax, standard k-ε, extended k-ε, low Reynolds number, and multiple-scale models);
- H_2/O_2 and hydrocarbon/O_2 combustion models (equilibrium, and one-step and reduced mechanism models);
- PDF based turbulence combustion interaction models;
- two-phase Eulerian-Lagrangian spray dynamics model; and
- discrete ordinate thermal radiation model.

The code was systematically and extensively verified against a large number of test problems to ensure solution invariance to coordinate orientation, grid skewness, boundary condition order, initial conditions, *etc.* Results of the testing, validation, and application studies are presented in the REFLEQS: Valiation and Applications Manual. The code has also been successfully applied for several liquid rocket engine configurations including SSME, AMCC, STME, and OMV. It has also been used for several turbomachinery applications, including NASA MSFC's Bearing Seals and Materials Tester (BSMT) flow problems.

10. RECOMMENDATIONS

Based on the experience of work during this project and assessment of current trends and future needs, several improvements are recommended for REFLEQS. The suggested enhancements are classified in the following three categories:

a. Physical Model Enhancement;
b. Numerical Method Improvement; and
c. Code Validation and Applications.

10.1 Physical Model Enhancement

To resolve complex spray combustion dynamics in rocket engines, the following physical models need further enhancement:

Multi-Step, Finite-Rate, Chemical Kinetics Model: The chemical equilibrium analysis has been widely used to predict species and temperature distributions in the combustion chamber and the nozzle. However, this analysis is valid only under low Mach number and high temperature conditions for which the chemical time scale is much smaller than the flow time scale. In regions of low temperature (near cooled walls or regions where the mixture ratio is far from stoichiometric) or high speed flow (*i.e.*, in the nozzle region), the equilibrium assumption is no longer valid and the effect of finite rate chemistry must be taken into account. The model should be formulated in the general form and demonstrated for established reaction mechanisms (such as 7-step O_2-H_2 and multi-step C_nH_m - O_2).

Supercritical Spray Model: The properties of fluids change drastically near the critical point. Most rocket engine thrust chambers operate under supercritical conditions. Advanced models, such as those being developed at Pennsylvania State University and University of Tennessee Space Institute (UTSI), should be incorporated to include the effects of supercritical phenomena on the vaporization process under these conditions.

Coupled Atomization and Spray Dynamics Model: Existing spray dynamics models require droplet size distribution and spatial resolution for droplet formation rate.

This information, assumed a priori, has great influence on spray dynamics results. Recently, advanced atomization models, based on Jet Embedding and VOF methods, have been developed (Giridharan *et al*, 1991 and Giridharan *et al*, 1992). The most suitable models for analysis of single injector and multi-injector interactions should be incorporated and numerically optimized for shear coaxial, shear/swirl, and impinging injectors.

Conjugate Heat Transfer Model: A conjugate heat transfer model should be implemented to assess the effects of heat transfer between various engine components (such as the heat flux across the injector face to the incoming cryogenic propellants). The conjugate heat transfer model needs to be incorporated in conjunction with the multi-domain capability.

10.2 Numerical Algorithm Improvements

Current trends in design methodology indicate a shift from analyzing component performance to a system of component interactions. Multi-domain, multi-media simulation with a large number of cells (10^6) are needed. To achieve this, the following enhancements are recommended for incorporation into the code.

Multi-Domain Capability: The analysis of interior and exterior flow in rocket engines can be greatly optimized by the use of multi-domain capability. It allows efficient grid generation and limits the number of blocked computational cells in the domain. It is also useful for grid refinement in localized regions of high gradients. A general purpose multi-domain capability should be incorporated into the code.

Adaptive Gridding Capability: This feature allows the grid to adapt itself to the solution of the problem. Therefore, the grid points are automatically clustered in regions of high flow gradients and become relatively coarser in regions with small gradients. This facilitates the efficient utilization of the computational cells in the flow domain. A moving grid capability should be used to implement the adaptive gridding feature into the code.

Convergence Enhancement Capability: Advanced methods such as multi-grid and coupled-variable conjugate gradient solvers should be implemented to improve the computational efficiency of solving large systems of linear and nonlinear equations. In addition, a second-order convergence rate scheme such as the Newton iteration method should be implemented and assessed. These features are especially useful to accelerate convergence of fine grid solutions.

10.3 Code Validation and Applications

The REFLEQS code can be applied to a wide variety of, subsonic, and supersonic flows, with and without heat transfer and combustion.

Even though the code has been extensively checked out for a large number of problems (as described in the REFLEQS: Validation and Applications Manual), additional detailed validation studies need to be performed for cases where reliable experimental data are available. More emphasis should be placed on 3D validation efforts and on flows coupled with complex physics (spray, combustion, radiation, turbulence, *etc.*).

11. REFERENCES

Avva, R.K., Smith, C.E., and Singhal, A.K., (1990), "Comparative Study of High and Low Reynolds Number Versions of k-ε Models," AIAA-90-0246, Aerospace Sciences Meeting.

Baldwin, B.S. and Lomax, H. (1978), "Thin Layer Approximation and Algebraic Model for Separated Flows," AIAA Paper 78-257.

Brankovic, A. and Stowers, S.T. (1988), ``Review of Low-Reynolds Number Turbulence Models for Complex Internal Separated Flows," AIAA-88-3006.

Cebeci, T. and Smith, A.M.O. (1974), *Analysis of Turbulent Boundary Layers,* Academic Press, New York.

Chen, Y-S. and Kim. S-W. (1987), ``Computation of Turbulent Flows Using an Extended k-ε Turbulence Closure Model," NASA CR-179204, NASA Marshall Space Flight Center, Huntsville, AL.

Chien, K. Y. (1982), "Predictions of Channel and Boundary-Layer Flows with Low-Reynolds-Number Turbulence Model," AIAA J., Vol. 20, pp. 33-38.

Crowe, C.T., Sharma, M.P., and Stock, D.E., "The Partical-Source-in-Cell (PSI-CELL) Model for Gas-Droplet Flows, *J. Fluids. Eng.*, pp. 325-332, 1977.

Fabris, G., Harsha, P.T., and Edelman, R.B., (1981), "Multiple-Scale Turbulence Modeling of Boundary Layer Flows for Scramjet Applications," NASA CR-3433.

Fiveland, W.A., "Discrete-Ordinates Solutions of the Radiative Transport Equation for Rectangular Enclosures," *J. of Heat Transfer*, vol. 106, kpp. 699-706, 1984.

Giridharan, M.G., Lee, J.G., Krishnan, A., Yang, H.Q., Ibrahim, E., Chuech, S., and Przekwas, A.J., (1991) "A Computer Model for Liquid Jet Atomization in

Rocket Thrust Chambers," Final Report for NASA Marshall Space Flight Center, Report Number 4041/1.

Giridharan, M.G., Lee, J.G., Krishnan, A., Przekwas, A.J., and Gross, K., (1992), "A Numerical Model for Coupling Between ATomization and Spray Dynamics in Liquid Rocket Thrust Chambers," AIAA/SAE/ASME/ASEE 28th Joint Propulsion Conference, Nashville, TN, AIAA-92-3768.

Hanjalic, K., Launder, B.E., and Schiestel, R., (1980), "Multiple Time-Scale Concepts in Turbulent Shear Flows," Turbulent Shear Flows, Eds. L. J. S. Bradbury, Springer Verlag, New York

Harlow, F.H., and Welch, J.E., (1965), "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Physics Fluids*, vol. 8, pp. 2182-2189.

Jones, W.P., and Whitelaw, J.H., (1982), "Calculation Methods for Reacting Turbulent Flows: A Review," *Combustion and Flame*, vol. 48, pp. 1-26.

Kim, S. W. and Chen, C.P., (1988), "A Multiple Time-Scale Turbulence Model Based on Variable Partitioning of Turbulent Kinetic Energy Spectrum," Paper No. AIAA-88-1771.

Kuo, K.K. (1986), Principles of Combustion, Wiley & Sons.

Launder, B.E., and Spalding, D.B., (1974), *Comp. Methods Appl. Mech. Eng.*, vol.3, p. 269.

Liang, P.Y., (1986), "Liquid Rocket Computer Code Development," NASA CP-2372.

Meintjes, K. and Morgan, A.P., (1989), "Element Variables and Solution of Complex Chemical Equilibrium Problems," *Comb. Sci. Tech.*, vol. 68, pp. 35-48.

Patankar, S.V. and Spalding, D.B., (1972), "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows," *Int. J. Heat Mass Transfer*, vol. 15, pp. 1787-1806, 1972.

Patankar, S.V., (1980), Numerical Heat Transfer and Fluid Flow, Hemiphere Publishing Corp., New York.

Patel, V. C., Rodi, W. and Scheuerer, G. (1985), ``Turbulence Models for Near-Wall and Low Reynolds Number Flows: A Review," *AIAA Journal*, Vol. 23, No. 9, pp. 1308-1319.

Peric, M., (1985), "A Finite-Volume Method for the Prediction of Three-Dimensional Fluid Flow in Complex Ducts," PhD Thesis, University of London.

Przekwas, A.J., Singhal, A.K., and Tam, L.T., (1984), "Rocket Injector Anomalies Study," NASA CR-174702.

Przekwas, A.J., Edwards, J., and Gross, K., (1986), "SSME Thrust Chamber Modeling with Navier-Stokes Equations," AIAA-86-1517.

Rhie, C.M., and Chow, W.L., (1983), "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation," *AIAA Journal*, vol. 21, pp. 1525-1532.

Rogers, R.C. and Chinitz, W., (1983), *AIAA Journal*, vol. 20, pp. 586-592.

Thompson, J.F., Warsi, Z.U.A., and Mastin, C.W., (1985), Numerical Grid Generation, North-Holland, New York.

Van Doormal, J.P., and Raithby, G.D., (1984), "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows," *Numerical Heat Transfer*, vol. 7, pp. 147-163.

Westbrook, C.K., and Dryer, F.L., *Comb. Sci. & Tech.*, vol. 27, pp. 31-43, 1981.

Yang, H.Q., Habchi, S.D., and Przekwas, A.J., (1992), "A General Strong Conservation Formulation of Navier-Stokes Equations in Non-Orthogonal Curvilinear Coordinates," AIAA-92-0187.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Combustion Chamber Analysis Code - Final Report

5. Report Date

May 1993

6. Performing Organization Code

7. Author(s)

A.J. Przekwas, Y.G. Lai, A. Krishnan, R.K.Avva
and M.G. Giridharan

8. Performing Organization Report No.

9. Performing Organization Name and Address

CFD Research Corporation
3325 Trianan Blvd.
Huntsville, AL 35805

10. Work Unit No.

11. Contract or Grant No.

NAS8-37824

12. Sponsoring Agency Name and Address

NASA Marshall Space Flight Center
Alabama 35812

13. Type of Report and Period Covered

Final Report
Dec. 1989 - June 1993

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A three-dimensional, time dependent, Favre averaged, finite volume Navier-Stokes code has been developed to model compressible and incompressible flows (with and without chemical reactions) in liquid rocket engines. The code has a non-staggered formulation with generalized body-fitted-coordinates (BFC) capability. Higher order differencing methodologies such as MUSCL and Osher-Chakravarthy schemes are available. Turbulent flows can be modeled using any of the five turbulent models present in the code. A two-phase, two-liquid, Lagrangian spray model has been incorporated into the code. Chemical equilibrium and finite rate reaction models are available to model chemically reacting flows. The discrete ordinate method is used to model effects of thermal radiation. The code has been validated extensively against benchmark experimental data and has been applied to model flows in several propulsion system components of the SSME and the STME.

17. Key Words (Suggested by Author(s))

Computational Fluid Dynamics,
Reacting Flows, Two-Phase Sprays,
Thermal Radiation

18. Distribution Statement

Unclassified

19. Security Classif. (of this report)

20. Security Classif. (of this page)

21. No. of pages

88

22. Price