AN ABSTRACT OF THE THESIS OF

John King for the degree of Master of Science in Nuclear Engineering presented on April 17, 1987.

Title: Application of the Conjugate Gradient Method to the COMMIX-1B Three Dimensional Momentum Equation.

Redacted for Privacy

Abstract approved: Samim Anghaie

ABSTRACT

The conjugate gradient method is an efficient means of solving large sparse symmetric positive definite systems of linear equations which arise from finite difference approximations to self-adjoint elliptic partial differential equations. In obtaining a solution, the conjugate gradient method successively minimizes a certain norm of the error in different orthogonal directions, causing an exact solution to be obtained in less than N steps for an NxN system of equations. Because the conjugate gradient method is not widely known, it is seldom used in engineering applications in comparison to the successive over-relaxation (S.O.R.) method.

Although comparisons between the conjugate gradient and S.O.R. methods have been made, these comparisons usually focus on the solution of a single system of equations often arising from one or two dimensional problems. For this reason the purpose of this research was to compare the performance of these methods in the context of the COMMIX-1B three dimensional thermal hydraulics code where these methods are required to solve many different systems of equations in a given problem to the same level of convergence.

To accomplish its purpose, this thesis has three main objectives. The first is to give the reader sufficient background to understand the conjugate gradient method used in COMMIX-1B. The second is to show how the conjugate gradient method fits into the overall solution strategy of COMMIX-1B. The last is to compare the running times of the conjugate gradient and S.O.R. methods for general problems run with COMMIX-1B, and to discuss several factors affecting this comparison.

It is concluded that under many circumstances, the conjugate gradient method is more efficient than S.O.R.

Application of the Conjugate Gradient Method to the COMMIX-1B Three Dimensional Momentum Equation

by

John King

A THESIS

SUBMITTED TO

Oregon State University

in partial fulfillment of
the requirements for the
degree of
Master of Science

Completed April 17, 1987

Commencement June 1987

APPROVED:

Redacted for Privacy

Professor of Nuclear Engineering in charge of major

Redacted for Privacy

Head of Department of Nuclear Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented: April 17, 1987

Typed by P. Cunningham for John Barry King

Acknowledgement

I wish to thank my grandparents for raising me and helping me to develop my intelligence and my confidence in it.

I also wish to acknowledge the expert guidance of several people - Samim Anghaie for proposing the research and for helping me organize the results, Henry Domanus for his insight into the COMMIX-1B computer code, Bob Schmidt for his help in running the computer at Argonne National laboratory, and Ely Gelbard for his expert mathematical insight.

Finally, I would like to acknowledge the financial support of the Department of Energy through the Nuclear Engineering Fellowship, and the financial support of William Sha at Argonne National Laboratory who appropriated the computer funds for my research.

Table of Contents

List of Figures

List of Figures (continued)

List of Tables

Application of the Conjugate Gradient Method to the COMMIX-1B Three Dimensional Momentum Equation

CHAPTER 1

INTRODUCTION

Finite difference approximations to the continuity, momentum, and energy equations in thermal hydraulics codes result in an NXN system of equations for a problem having N field points. In a three dimensional problem, N increases as the problem becomes larger or more complex, and more rapidly as the mesh size is reduced. As a consequence, the execution time required to solve the problem increases, placing limits on the problem complexity or resolution. A conventional method of solution for this system of equations is the Successive Over Relaxation (S.O.R.) technique. However, for a wide range of problems the execution time may be reduced by using a more efficient linear equation solver. One such method is the conjugate gradient method which I implemented in the momentum section of the COMMIX-1B thermal hydraulics code. It was found that the execution time required to solve the resulting system of equations to the same level of convergence was reduced by a factor of about 2 for some problems.

Since the conjugate gradient method is not yet a common solution technique, it will be described in Chapter 2. The material in Chapter 2 is a detailed discussion and mathematical proofs which are intended to establish the convergence properties of the preconditioned conjugate gradient method used in COMMIX-1B. For a further discussion of this material see the thesis on variational interative methods by Rati Chandra [2]. The remainder of the material is such a small fraction of the total that the contributing sources will be referenced in Chapter 2 as they occur.

To show how the conjugate gradient method was implemented, the fluid modeling involved in COMMIX-1B will be discussed in Chapter 3. This discussion will not only describe the equations that the conjugate gradient method is used to solve but also show how this method is involved in the overall solution strategy. In addition, a convergence acceleration technique used in COMMIX-1B which is known as mass rebalancing will also be described. For a further discussion of this material see the COMMIX-1B reference manual [3].

After discussing these preliminary concepts, comparisons between the conjugate gradient and S.O.R. methods will be made in Chapter 4 for the problems run. In this discussion each of the problems will be described in detail with the flow patterns shown. Next, differences between problems in the comparison of computer running times will be discussed in terms of the differences between methods and the differences between the problems.

CHAPTER 2

PROPERTIES OF VARIATIONAL ITERATIVE METHODS

2.1 Introduction

The solution scheme used in COMMIX-1B is the preconditioned conjugate gradient method with incomplete Cholesky factorization. Since this method is a special case of the variational method, most of the following material will be devoted to the properties of the variational method. After these properties have been discussed, the preconditioned conjugate gradient method will be derived as a special case of the variational method. Before developing the properties of the variational method, however, it is necessary to provide an overview of the method itself.

The variational method is a means of solving systems of equations of the form,

$$(2.1.1) \qquad A\bar{x} = \bar{f},$$

for which A is an NXN symmetric matrix. The solution strategy is to march in a set of directions, $S = \{ \bar{p}_0, \bar{p}_1, \ldots, \bar{p}_{n-1}\}$, which are orthogonal to each other with respect to the inner product,

$$(2.1.2) \qquad \langle\bar{p}_j, \bar{p}_i\rangle_{A^\mu} = (\bar{p}_j, A^\mu \bar{p}_i) = \bar{p}j \bullet A^\mu \bar{p}_i.$$

When marching in a direction, $\bar{p}_i$, $\bar{x}_{i+1}$ is set to $\bar{x}_i + a_i\bar{p}_i$ with a_i chosen to minimize the error functional,

$$(2.1.3) \qquad E\mu (\bar{x}_{i+1}) = ((\bar{x}-\bar{x}_{i+1}), A^\mu (\bar{x}-\bar{x}i+1)).$$

(Note that $\bar{x}$ is the exact solution to (2.1.1), and $\bar{x}_{i+1}$ is the (i+1)st calculated approximation to $\bar{x}$.) After traveling in all n orthogonal directions, an exact solution is obtained if exact arithmetic exists. The conjugate gradient method is simply the special case where the exponent μ in (2.1.2) and (2.1.3) is set to 1.

Since the preconditioned conjugate gradient method is a special case of the variational method, several topics must first be discussed. Some linear algebraic concepts of inner products will be reviewed in section 2, and the method by which orthogonal directions are generated will be described in section 3. In section 4 the variational method will be motivated by geometric intuition and in section 5 its convergence behavior will be discussed. Several relationships among the variables of the variational method will be derived in section 6 for later use. The error bounds of the variational method will be derived in section 7, and more efficient methods of generating orthogonal direction vectors will be derived in section 8. In section 9 the use of matrix preconditioning to increase the convergence rate of the variational method will be discussed. Finally, the matrix preconditioning scheme and the preconditioned conjugate gradient algorithm will be described in section 10.

2.2 Some Linear Algebraic Concepts of Inner Products

For a further discussion of the following material see the textbook entitled Elementary Linear Algebra by Howard Anton [1].

An inner product on a vector space V is a function that associates a real number $\langle \bar{u}, \bar{v} \rangle$ with each pair of vectors $\bar{u}$ and $\bar{v}$ in V in such a way that the following axions hold.

(2.2.1) $\langle \bar{u}, \bar{v} \rangle = \langle \bar{v}, \bar{u} \rangle$

(2.2.2) $\langle \bar{u} + \bar{v}, \bar{w} \rangle = \langle \bar{u}, \bar{w} \rangle + \langle \bar{v}, \bar{w} \rangle$

(2.2.3) $\langle K\bar{u}, \bar{v} \rangle = K \langle \bar{u}, \bar{v} \rangle$

(2.2.4) $\langle \bar{v}, \bar{v} \rangle \geq 0$ and $\langle \bar{v}, \bar{v} \rangle = 0$ if and only if $\bar{v} = 0$.

A vector space with an inner product is called an inner product space.

It is important to note that since the dot product (Euclidean inner product) satisfies these axioms, it is one special case of the inner product.

(2.2.5) $||\bar{u}|| = \langle \bar{u}, \bar{u} \rangle^{\frac{1}{2}}$.

Also the cosine of the angle between two vectors is defined as,

(2.2.6) $\cos \Theta = \langle \bar{u}, \bar{v} \rangle / ||\bar{u}|| \, ||\bar{v}||$

By using this definition of cos Θ, the projection of a vector $\bar{u}$ onto another vector $\bar{v}$ can be determined by

(2.2.7) $\text{proj}_{\bar{v}}\bar{u} = ||\bar{u}|| \cos \Theta = \langle \bar{u}, \bar{v} \rangle / ||\bar{v}||$.

Since a unit vector in the $\bar{v}$ direction can be constructed as,

(2.2.8) $\bar{v}^* = \bar{v}/||\bar{v}|| = \bar{v}/\langle \bar{v}, \bar{v} \rangle^{\frac{1}{2}}$

(2.2.7) and (2.2.8) can be combined to subtract the projection of $\bar{u}$ onto $\bar{v}$ from $\bar{u}$.

(2.2.9) $\bar{u}' = \bar{u} - (\text{proj}_{\bar{v}}\, \bar{u})(\bar{v}^*) = \bar{u} - \dfrac{\langle \bar{u}, \bar{v} \rangle \bar{v}}{||\bar{v}||^2} = \bar{u} - \dfrac{\langle \bar{u}, \bar{v} \rangle \bar{v}}{\langle \bar{v}, \bar{v} \rangle}$

Furthermore, the new vector $\bar{u}'$ is now orthogonal to $\bar{v}$ since

$$(2.2.10) \qquad \langle\bar{u},\bar{v}\rangle = \langle\bar{u}-\frac{\langle\bar{u},\bar{v}\rangle}{\langle\bar{v},\bar{v}\rangle}\bar{v},\ \bar{v}\rangle=\langle\bar{u},\bar{v}\rangle - \frac{\langle\bar{u},\bar{v}\rangle\langle\bar{v},\bar{v}\rangle}{\langle\bar{v},\bar{v}\rangle} = 0$$

Thus, the concepts and definitions motivated for the dot product work together to produce analogous results for arbitrary inner products.

Another similar result is that if $S = \{ \bar{v}_1, \bar{v}_2, \ldots \bar{v}_n \}$ is an orthogonal set of nonzero vectors in an inner product space, then S is linearly independent.

Proof: Assume that

$$(2.2.11) \qquad k_1\bar{v}_1 + k_2\bar{v}_2 + \ldots + k_n\bar{v}_n = \bar{0}.$$

To show that S is linearly independent it suffices to show that $k_1 = k_2 = \ldots = k_n = 0$. For each $\bar{v}_i$ in S, it follows from (2.2.11) that

$$\langle k_1\bar{v}_1 + k_2\bar{v}_2 + \ldots + k_n\bar{v}_n,\ \bar{v}_i\rangle = \langle 0,\ \bar{v}_i\rangle = 0$$

or equivalently

$$k_1 \langle\bar{v}_1,\ \bar{v}_i\rangle + k_2 \langle\bar{v}_2,\ \bar{v}_i\rangle + \ldots + k_n \langle\bar{v}_n,\ \bar{v}_i\rangle = 0$$

From the orthogonality of S, $\langle\bar{v}_j,\ \bar{v}_i\rangle = 0$ when $j \neq i$, so that this equation reduces to

$$k_i \langle\bar{v}_i,\ \bar{v}_i\rangle = 0.$$

Since the vectors in S are assumed to be nonzero, $\langle\bar{v}_i,\ \bar{v}_i\rangle \neq 0$, and hence, $k_i = 0$.

Since the subscript is arbitrary, $k_1 = k_2 = \ldots = k_n = 0$. As a consequence S is linearly independent and spans the space.

Q.E.D.

2.3 The Lanczos Method for Generation of Orthogonal Vectors

Since the variational method minimizes the error functional (2.1.3) by marching in directions, $S = \{ \bar{p}_o, \bar{p}_1, \ldots \bar{p}_{n-1}\}$, that are A^μ orthogonal to each other, it is necessary to have some means of generating orthogonal vectors. One way of doing this is to choose an initial estimate of the (i + 1)st direction vector and then successively subtract its projections onto each of the other directions from it using (2.2.9). After this process the new vector will be orthogonal to each of the preceding vectors.

There are two problems associated with this method. First, the need to save each of the previous direction vectors would result in a large storage requirement. Second, the computational effort spent in successively determining and subtracting i+1 projections from the initial estimate of $\bar{p}_{i+1}$ would make the variational method inefficient. Through a wise choice of the initial estimate, the Lanczos algorithm eliminates these difficulties.

Algorithm (2.3.1) <u>The Lanczos Method for Generation of Orthogonal Vectors</u>

<u>Step 1</u>: Define $\bar{p}_{-1} = \bar{0}$

Choose $\bar{p}_o$

<u>Step 2</u>: Compute

$\delta_i = (A\bar{p}_i, A^\mu \bar{p}_{i-1})/(\bar{p}_{i-1}, A^\mu \bar{p}_{i-1})$,

$\gamma_i = (A\bar{p}_i, A^\mu \bar{p}_i)/(\bar{p}_i, A^\mu \bar{p}_i)$,

and

$$\bar{p}_{i+1} = A\bar{p}_i - \gamma_i\bar{p}_i - \delta_i \bar{p}_{i-1}. \qquad (2.3.1)$$

<u>Step 3</u>: Set i = i + 1 and go to Step 2.

The Lanczos method is equivalent to choosing $A\bar{p}_i$ to be the initial estimate of $\bar{p}_{i+1}$ and then subtracting the projections of $A\bar{p}_i$ onto $\bar{p}_i$ and $\bar{p}_{i-1}$ from $A\bar{p}_i$. What makes this method unique is that these two subtractions suffice to make $\bar{p}_{i+1}$ orthogonal to all the previous direction vectors.

Proof: By the symmetry of A it is sufficient to show that

(2.3.2) $(\bar{p}_i, A^{\mu}\bar{p}_j) = 0$ for $i > j$.

The proof is by induction on i.

Since $\bar{p}_{-1}$ is defined to be $\bar{0}$.

$(\bar{p}_1, A^{\mu}\bar{p}_0) = (A\bar{p}_0, A^{\mu}\bar{p}_0) - \gamma_0\,(\bar{p}_0, A^{\mu}\bar{p}_0)$,

which is 0 by the definition of γ_0. Therefore (2.3.2) holds for i=1. Assume it holds for $i \leq k$. Then

(2.3.3) $(\bar{p}_{k+1}, A^{\mu}\bar{p}_j) = (A\bar{p}_k, A^{\mu}\bar{p}_j) - \gamma_k\,(\bar{p}_k, A^{\mu}\bar{p}_j)$
$- \delta_k\,(\bar{p}_{k-1}, A^{\mu}\bar{p}_j)$.

If $j = k$, the last term is 0 by the induction hypothesis, and the remaining terms cancel by the definition of γ_k. If $j = k - 1$, the middle term in the right hand side is 0 by the induction hypothesis, and the remaining terms cancel by the definition of δ_k. Finally, if $j < k - 1$, the last two terms in the right hand side are 0 by the induction hypothesis and

(2.3.4) $(\bar{p}_{k+1}, A^{\mu}\bar{p}_j) = (A\bar{p}_k, A^{\mu}\bar{p}_j)$.

Since A is symmetric, A^{μ} is symmetric, and (2.3.4) may be rearranged.

$$
\begin{aligned}
(2.3.5)\quad (\bar{p}_{k+1}, A^{\mu}\bar{p}_j) &= (A\bar{p}_k, A^{\mu-1}(A\bar{p}_j)) \\
&= (A\bar{p}_k)^T(A^{\mu-1}(A\bar{p}_j)) \\
&= \bar{p}_k{}^T A^T A^{\mu-1}(A\bar{p}_j)
\end{aligned}
$$

$$= \bar{p}_k{}^T A^\mu(A\bar{p}_j)$$

$$= (\bar{p}_k, A^\mu(A\bar{p}_j)).$$

By (2.3.1) however

$$A\bar{p}_j = \bar{p}_{j+1} + \gamma_j\bar{p}_j + \delta_j\bar{p}_{j-1}$$

By substituting this expression into (2.3.5), the following equation results.

$$(\bar{p}_{k+1}, A^\mu\bar{p}_j) = (\bar{p}_k, A^\mu(\bar{p}_{j+1} + \gamma_j\bar{p}_j + \delta_j\bar{p}_{j-1}))$$

$$= (\bar{p}_k, A^\mu\bar{p}_{j+1}) + \gamma_j(\bar{p}_k, A^\mu\bar{p}_j) + \delta_j(\bar{p}_k, A^\mu\bar{p}_{j-1})$$

By the induction hypothesis all the terms on the right hand side are 0 for $j < k - 1$. Thus, $(\bar{p}_{k+1}, A^\mu\bar{p}_j) = 0$ for $j \leq k$, and the induction hypothesis holds for $i = k + 1$.

Q.E.D.

2.4 The Variational Method

As mentioned previously the variational method minimizes the error functional, $((\bar{x} - \bar{x}_{i+1}), A^\mu(\bar{x}-\bar{x}_{i+1}))$, by marching in the $\bar{p}_i$ direction. Furthermore, the directions in the set, $\{\bar{p}_o, \bar{p}_i, \ldots \bar{p}_n\}$ are all orthogonal to each other with respect to the inner product, $(\bar{p}_i, A^\mu\bar{p}_j)$. If μ is set to 0, the error functional becomes

$$(2.4.1) \quad Eu(\bar{x}_i) = ((\bar{x}-\bar{x}_i), (\bar{x}-\bar{x}_i)) = ||\bar{x}-\bar{x}_i||^2$$

so that the variational method is essentially trying to minimize the square of the error magnitude. Furthermore, the set of directions, $\{\bar{p}_o, \bar{p}_1, \ldots \bar{p}_{n-1}\}$, are now orthogonal to each other with respect to the Euclidean inner product, $(\bar{p}_j, \bar{p}_i)$. To further simplify matters assume that the space is 2 dimensional. Under these circumstances the minimization process can be shown geometrically as done in figure 1.

The strategy of the variational method is to march in the direction $\bar{p}_0$ by an amount $\bar{a}_0$ that

$$(2.4.2) \quad \bar{x}_1 = \bar{x}_0 + \bar{a}_0\bar{p}_0$$

minimizes

$$(2.4.3) \quad E\mu\ (\bar{x}_1) = ||\bar{x}-\bar{x}_1||^2 = ||\ (\bar{x}-\bar{x}_0-a_0\bar{p}_0)||^2$$

The way to do this is clearly to determine the projection of $\bar{x}-\bar{x}_0$ on the $\bar{p}_0$ direction, and then march in the $\bar{p}_0$ direction until the length of the march is equal to the projection. At this point the vector $\bar{x}-\bar{x}_1$ is orthogonal to $\bar{p}_0$. The projection of $\bar{x}-\bar{x}_0$ onto $\bar{p}_0$ is clearly $||\bar{x}-\bar{x}_0||\cos\Theta$. Since

$$((\bar{x}-\bar{x}_0),\ \bar{p}_0) = ||\bar{x}-\bar{x}_0||\ ||\bar{p}_0||\cos\Theta,$$

the projection is

$$||\bar{x}-\bar{x}_0||\cos\Theta = \frac{1}{||p_0||}((\bar{x}-\bar{x}_0),\ \bar{p}_0).$$

Having obtained this projection it should be multiplied by the unit vector $\bar{p}_0/||\bar{p}_0||$ and added to $\bar{x}_0$. Thus

$$(2.4.4) \quad \bar{x}_1 = \bar{x}_0 + \frac{((\bar{x}-\bar{x}_0),\bar{p}_0)}{||\bar{p}_0||^2}\ \bar{p}_0$$

By comparison to (2.4.2) it is seen that

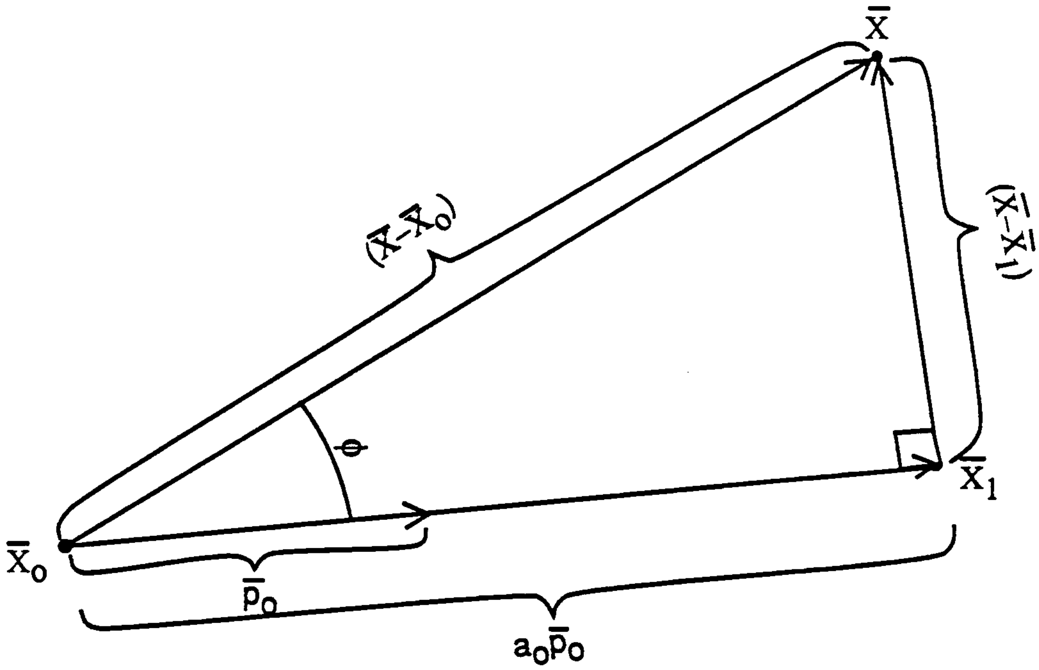

Figure 1. Showing the Minimization of the Error Magnitude in 2 Dimensional Space

$$(2.4.5)\quad a_0 = \frac{((\bar{x}-\bar{x}_0),\bar{p}_0)}{||\bar{p}_0||^2} = \frac{((\bar{x}-\bar{x}_0),\bar{p}_0)}{(\bar{p}_0,\bar{p}_0)}$$

It should next be observed that since the space is 2 dimensional, and $\bar{p}_1$ and $\bar{x}-\bar{x}_1$ are both orthogonal to $\bar{p}_0$, the process will converge on the next iteration.

The major flaw with this process is that a prior knowledge of the solution is necessary to evaluate a_0 and similarily for a_1. This problem arises, however, because μ was set to 0 in the variational method. Forcing $\mu \geq 1$ will prevent this difficulty and is the reason for involving more complicated orthogonality relationships in the method.

If $u \geq 1$, the same reasoning applies except that inner products of the form $(\bar{w}, A^{\mu}\bar{v})$ will replace inner products of the form $(\bar{w}, \bar{v})$. If these changes are made in (2.4.5), then

$$(2.4.6)\quad a_i = ((\bar{x}-\bar{x}_i),\ A^{\mu}\bar{p}_i)/(\bar{p}_i,\ A^{\mu}\bar{p}_i).$$

for a general subscript i. As it is written, a_i still depends upon a prior knowledge of the solution $\bar{x}$ for its evaluation. However, since A is symmetric, and $\mu \geq 1$, a_i may be rewritten as

$$(2.4.7)\quad a_i = (A(\bar{x}-\bar{x}_i),\ A^{\mu-1}\bar{p}_i)/(\bar{p}_i,\ A^{\mu}\bar{p}_i).$$

Since $A\bar{x} = \bar{f}$ which is known, and $A\bar{x}_i$ may be computed, a_i may be determined from known quantities. Furthermore, if the residual, $\bar{r}_i$, is defined as

$$(2.4.8)\quad \bar{r}_i = \bar{f} - A\bar{x}_i,$$

then

(2.4.9) $a_i = (\bar{r}_i, A^{\mu-1}\bar{p}_i)/(\bar{p}_i, A^{\mu}\bar{p}_i)$.

In addition, since $\bar{r}_{i+1} = \bar{f} - A\bar{x}_{i+1}$, and $\bar{x}_{i+1} = \bar{x}_i + a_i\bar{p}_i$,

(2.4.10) $\bar{r}_{i+1} = \bar{f}-A\bar{x}_i - a_iA\bar{p}_i = \bar{r}_i - a_iA\bar{p}_i$.

and may be easily updated. The variational method combines these concepts with the Lanczos algorithm to yield an effective solution scheme.

Algorithm (2.4.1) The Variational Method

```
Step 1: Choose an initial approximation $\bar{x}_0$ to $\bar{x}$
        Compute $\bar{r}_0 = \bar{f} - A\bar{x}_0$
        Set $\bar{p}_0 = \bar{r}_0$
            and $i = 0$
Step 2: Compute
        $a_i = (\bar{r}_i, A^{\mu-1}\bar{p}_i)/(\bar{p}_i, A^{\mu}\bar{p}_i)$
        $\bar{x}_{i+1} = \bar{x}_i + a_i\bar{p}_i$
        $\bar{r}_{i+1} = \bar{r}_i - a_i A\bar{p}_i$
        $\delta_i = (A\bar{p}_i, A^{\mu}\bar{p}_{i-1})/(\bar{p}_{i-1}, A^{\mu}\bar{p}_{i-1})$
        $\gamma_i = (A\bar{p}_i, A^{\mu}\bar{p}_i)/(\bar{p}_i, A^{\mu}\bar{p}_i)$
        and $\bar{p}_{i+1} = A\bar{p}_i - \gamma_i\bar{p}_i - \delta_i\bar{p}_{i-1}$
Step 3: If $\bar{x}_{i+1}$ is sufficiently close to $\bar{x}$, terminate the iteration
        process; else set $i = i+1$ and go to step 2.
```

2.5 Convergence Properties of the Variational Method

Since the variational method updates $\bar{x}_j$ on each iteration, it should be obvious that by the (i+1)st iteration

(2.5.1) $\bar{x}_{i+1} = \bar{x}_0 + \sum_{k=o}^{i} a_k\bar{p}_k$

Furthermore, the value of $\bar{x}_{i+1}$ so obtained makes the new error, $\bar{x}-\bar{x}_{i+1}$, A^μ orthogonal to each of the previous directions, $\{ \bar{p}_0, \bar{p}_1,\ldots\bar{p}_i\}$, used to obtain $\bar{x}_{i+1}$. Because of this orthogonality, the variational method converges in at most N steps.

Proof: To show that the method converges in at most N steps, it is first necessary to show that

$$(2.5.2)\quad ((\bar{x}-\bar{x}_{i+1}, A^\mu\bar{p}_j) = 0 \text{ for } j \leq i.$$

By (2.5.1) $\bar{x}_{i+1}$ may be expanded so that

$$(2.5.3)\quad ((\bar{x}-\bar{x}_{i+1}), A^\mu\bar{p}_j) = ((\bar{x}-\bar{x}_0 - \sum_{k=0}^{i} a_k\bar{p}_k), A^\mu\bar{p}_j)$$

$$= ((\bar{x}- \bar{x}_0), A^\mu\bar{p}_j) - a_j(\bar{p}_j, A^\mu\bar{p}_j),$$

where the final term results from the A^μ orthogonality of the $\bar{p}_k$'s.

Since

$$a_j = (\bar{x}-\bar{x}_j, A^\mu\bar{p}_j)/(\bar{p}_j, A^\mu\bar{p}_j)$$

by (2.4.6), (2.5.1) may again be used to expand $\bar{x}_j$ so that

$$a_j = ((\bar{x} - \bar{x}_0 - \sum_{k=o}^{j-1} a_k\bar{p}_k), A^\mu\bar{p}_j)/(\bar{p}_j, A^\mu\bar{p}_j)$$

Since all of the $\bar{p}_k$'s are orthogonal to $\bar{p}_j$ for $0 \leq k \leq j-1$,

$$(2.5.4)\quad a_j = ((\bar{x}- \bar{x}_0), A^\mu\bar{p}_j)/(\bar{p}_j, A^\mu\bar{p}_j).$$

Substituting this result into (2.5.3) yields

$$((\bar{x}-\bar{x}_{i+1}), A^\mu\bar{p}_j)) = ((\bar{x}-\bar{x}_0), A^\mu\bar{p}_j)- \frac{((\bar{x}-\bar{x}_0),A^\mu\bar{p}_j)(\bar{p}_j,A^\mu\bar{p}_j)}{(\bar{p}_j,A^\mu\bar{p}_j)}$$

which is 0. Thus $(\bar{x}-\bar{x}_{i+1})$ is orthogonal to the set $\{\bar{p}_0, \bar{p}_1, \dots, \bar{p}_i\}$ used to obtain $\bar{x}_{i+1}$. Furthermore, since the subscript is arbitrary it must be true that after $\bar{x}_n$ is calculated $(\bar{x}-\bar{x}_n)$ is orthogonal to the set $\{\bar{p}_0, \bar{p}_1 \dots \bar{p}_{n-1}\}$. Since this set has n orthogonal vectors which completely span the space, $\bar{x}-\bar{x}_n$ is orthogonal to all vectors in the space. As a consequence, $\bar{x}-\bar{x}_n=\bar{0}$, implying that $\bar{x}=\bar{x}_n$. Convergence in N steps is therefore guaranteed.

<u>Q.E.D.</u>

Another important property of the variational method is that for each i, $\bar{x}_{i+1}$ minimizes the error functional $E\mu(\bar{x})$ over the subspace spanned by $\bar{x}_0 + \{\bar{p}_0, \bar{p}_1, \dots, \bar{p}_i\}$

<u>Proof</u>: Let $\tilde{x} = \bar{x}_0 + \sum_{j=0}^{i} s_j\bar{p}_j$ where $\{s_j\}_{j=0}^{i}$ are some scalers.

Then, (2.5.5) $E\mu(\tilde{x}) = ((\bar{x}-\tilde{x}), A^{\mu}(\bar{x}-\tilde{x}))$

$$= ((\bar{x}-\bar{x}_0-\sum_{j=0}^{i} s_j\bar{p}_j), A^{\mu}(\bar{x}-\bar{x}_0-\sum_{j=0}^{i} s_j\bar{p}_j))$$

$$= ((\bar{x}-\bar{x}_0), A^{\mu}(\bar{x}-\bar{x}_0)) - \sum_{j=0}^{i} s_j (\bar{p}_j, A^{\mu}(\bar{x}-\bar{x}_0))$$

$$- \sum_{j=0}^{i} s_j((\bar{x}-\bar{x}_0), A^{\mu}\bar{p}_j) + \sum_{j=0}^{i} s_j^2 (\bar{p}_j, A^{\mu}\bar{p}_j)$$

where the last term arises from the orthogonality of the $\bar{p}_j$'s. Since the first term is simply $E\mu\,(\bar{x}_0)$ and the second term may be rearranged by the symmetry of A,

$$E\mu(\tilde{x}) = E\mu(\bar{x}_0) - \sum_{j=0}^{i} 2\, s_j\, ((\bar{x}-\bar{x}_0), A^{\mu}\bar{p}_j)$$

$$+ \sum_{j=0}^{i} s_j^2(\bar{p}_j, A^{\mu}\bar{p}_j).$$

By differentiating with respect to s_j, the necessary and sufficient condition for $E\mu(\tilde{x})$ to be a minimum is,

$$(2.5.6)\quad s_j = ((\bar{x}-\bar{x}_0),\ A^\mu \bar{p}_j)/(\bar{p}_j, A^\mu \bar{p}_j)$$

for $0 \leq j \leq i$. However, s_j corresponds to a_j from (2.4.6), and consequently,

$$(2.5.7)\ \tilde{x} = \bar{x}_0 + \sum_{j=o}^{i} s_j \bar{p}_j = \bar{x}_0 + \sum_{j=o}^{i} a_j \bar{p}_j = \bar{x}_{i+1}.$$

Thus $\bar{x}_{i+1}$ corresponds to $\tilde{x}$ for which $E\mu\ (\tilde{x})$ is a minimum.

Q.E.D.

2.6 Relationships of the Variational Method

Since much of the material presented in later sections depends heavily upon several relationships of the variational method, it is necessary to digress from discussions of its convergence properties and computational aspects to develop these relations. The following relations hold for the variational method. In these relations set notation is used to refer to the spaces spanned by the vectors in the brackets.

$$(2.6.1a)\quad A\bar{p}_i \ \varepsilon\ \{\bar{p}_0,\ \bar{p}_1, \ldots \bar{p}_{i+1}\};$$

$$(2.6.1b)\quad \bar{r}_i \ \varepsilon\ \{\bar{p}_0,\ \bar{p}_1, \ldots, \bar{p}_i\ \};$$

$$(2.6.1c)\quad \{\bar{p}_0, \bar{p}_1, \ldots, \bar{p}_i\} = \{\bar{p}_0, A\bar{p}_0, \ldots A^i\bar{p}_0\} = \{\bar{r}_0, A\bar{r}_0, \ldots A^i\bar{r}_0\};$$

$$(2.6.1d)\quad (\bar{r}_i,\ A^{\mu-1}\bar{p}_j) = 0,\ j < i\ ;$$

$$(2.6.1e)\quad (\bar{r}_i,\ A^{\mu-1}\bar{r}_j) = 0 \text{ if } i \neq j;$$

$$(2.6.1f)\quad (\bar{r}_i,\ A^{\mu-1}\bar{p}_j) = (\bar{r}_0,\ A^{\mu-1}\bar{p}_j),\ i \leq j.$$

Proof: (2.6.1a) follows directly from (2.3.1). Since $\bar{p}_0 = \bar{r}_0$, (2.6.1b)-(2.6.1d) hold for i = 0. To prove (2.6.1b)-(2.6.1d) by induction, assume they hold for $i \leq k$.

By this induction hypothesis $\bar{r}_k \varepsilon \{\bar{p}_0, \bar{p}_1, \ldots, \bar{p}_k\}$. Since $\bar{r}_{k+1} = \bar{r}_k - a_k A\bar{p}_k$, and $A\bar{p}_k \varepsilon \{\bar{p}_0, \bar{p}_1, \ldots \bar{p}_{k+1}\}$ by (2.6.1a), $\bar{r}_{k+1} \ \varepsilon \ \{\bar{p}_0, \bar{p}_1, \ldots \bar{p}_{k+1}\}$ and (2.6.1b) holds for i = k+1.

By the Lanczos algorithm,

$$\bar{p}_{k+1} = A\bar{p}_k - \gamma_k \bar{p}_k - \delta_k \bar{p}_{k-1},$$

and by the induction hypotheses

$$\{\bar{p}_0, \ \bar{p}_1, \ldots \bar{p}_k\} = \{\bar{p}_0, A\bar{p}_0, \ldots, A^k\bar{p}_0\}.$$

This implies that both $\bar{p}_k$ and $\bar{p}_{k-1}$ are elements of $\{\bar{p}_0, \ A\bar{p}_0, \ldots, A^k\bar{p}_0\}$, and hence

$$\bar{p}_{k+1} = A \ (\varepsilon \ \{\bar{p}_0, \ A\bar{p}_0, \ldots A^k\bar{p}_0\}) - \gamma_k(\varepsilon\{\bar{p}_0, A\bar{p}_0, \ldots A^k\bar{p}_0\})$$
$$- \ \delta_k(\varepsilon\{\bar{p}_0, \ A\bar{p}_0, \ldots A^k\bar{p}_0\})$$

As a consequence,

$$\bar{p}_{k+1} \ \varepsilon\{\bar{p}_0, A\bar{p}_0, \ldots A^{k+1}\bar{p}_0\}, \text{ and}$$

$$\{\bar{p}_0, \bar{p}_1 \ldots, \bar{p}_{k+1}\} \subseteq \{\bar{p}_0, \ A\bar{p}_0 \ldots A^{k+1}\bar{p}_0\}.$$

Since the $\bar{p}$'s are linearly independent,

$$\{\bar{p}_0, \ \bar{p}_1, \ldots \bar{p}_{k+1}\} = \{\bar{p}_0, \ A\bar{p}_0 \ldots A^{k+1}\bar{p}_0\}.$$

Furthermore,

$$\{\bar{p}_o, A\bar{p}_o, \ldots, A^{k+1}\bar{p}_o\} = \{\bar{r}_o, A\bar{r}_o, \ldots, A^{k+1}\bar{r}_o\}$$

since $\bar{p}_o = \bar{r}_o$, and (2.6.1c) holds for $i = k+1$.

By the definition of $\bar{r}_{k+1}$ in algorithm (2.4.1),

$$(\bar{r}_{k+1}, A^{\mu-1}\bar{p}_j) = (\bar{r}_k, A^{\mu-1}\bar{p}_j) - a_k (A\bar{p}_k, A^{\mu-1}\bar{p}_j)$$

If $j = k$, the right hand side is 0 by the definition of a_k in (2.4.9). If $j<k$, then the two terms on the right hand side vanish by the induction hypothesis and the orthogonality of the $\bar{p}$'s respectively. Thus (2.6.1d) holds for $i = k+1$, and (2.6.1b)-(2.6.1d) follow by induction.

To prove (2.6.1e) note that by (2.6.1b)

$$A^{\mu-1}\bar{r}_j \in \{A^{\mu-1}\bar{p}_o, A^{\mu-1}\bar{p}_1, \ldots, A^{\mu-1}\bar{p}_j\}$$

If $j<i$,

$$(\bar{r}_i, A^{\mu-1}\bar{r}_j) = (\bar{r}_i, \in \{A^{\mu-1}\bar{p}_o, A^{\mu-1}\bar{p}_1, \ldots A^{\mu-1}\bar{p}_j\})$$

By (2.6.1d) this is 0 for $j<i$ and $(\bar{r}_i, A^{\mu-1}\bar{r}_j) = 0$ is for $j<i$. However, by the symmetry of $A^{\mu-1}$

$$(\bar{r}_i, A^{\mu-1}\bar{r}_j) = (\bar{r}_j, A^{\mu-1}\bar{r}_i)$$

so that $(\bar{r}_i, A^{\mu-1}\bar{r}_j)=0$ for $i\neq j$.

To prove (2.6.1f) note that

$$\bar{r}_i = \bar{r}_o - \sum_{k=o}^{i-1} a_k A\bar{p}_k.$$

Therefore,

$$(\bar{r}_i, A^{\mu-1}\bar{p}_j) = (\bar{r}_o, A^{\mu-1}\bar{p}_j) - \sum_{k=o}^{i-1} a_k (A\bar{p}_k, A^{\mu-1}\bar{p}_j)$$

If $i \leq j$, the sum vanishes by the orthogonality of the $\bar{p}$'s and

$$(\bar{r}_i, A^{\mu-1}\bar{p}_j) = (\bar{r}_0, A^{\mu-1}\bar{p}_j) \text{ for } i \leq j$$

Q.E.D.

2.7 Error Bounds

As a first step in establishing error bounds for the variational method it must be shown that the variational method is optimal among all linear iterative methods with respect to the error functional $E\mu(x)$

Proof: Since $\{\bar{p}_0, \bar{p}_1 \ldots \bar{p}_{i-1}\} = \{\bar{r}_0, A\bar{r}_0, \ldots A^{i-1}\bar{r}_0\}$ by (2.6.1c)

$\bar{x}_i = \bar{x}_0 + \sum_{j=o}^{i-1} a_j \bar{p}_j = \bar{x}_0 + \sum_{j=o}^{i-1} s_j A^j \bar{r}_0$ for some scalers $\{s_j\}_{j=0}^{i-1}$ This may be written as

(2.7.1) $\bar{x}_i = \bar{x}_0 + P^*_{i-1}(A)\bar{r}_0$

where P^*_{i-1} (A) is a polynomial of degree at most i-1 in A. In addition, any consistent linear iterative method may be written as

(2.7.2) $\bar{x}_i = \bar{x}_0 + P_{i-1}(A)\ \bar{r}_0$

where $P_{i-1}(A)$ is also a polynomial of degree at most i-1 in A. However, since the variational method minimizes the error functional over the subspace $\bar{x}_0 + \{\bar{r}_0, A\bar{r}_0, \ldots A^{i-1}\bar{r}_0\}$, it must choose the particular polynomial $P^*_{i-1}(A)$ which is optimal with respect to $E\mu(\bar{x}_i)$ in the set of all polynomials, $\bar{P}_{i-1}(A)$. Since the polynomials $P^*_{i-1}(A)$ in (2.7.1) and $P_{i-1}(A)$ in

(2.7.2) are in the same set of polynomials, and $P^*_{i-1}(A)$ is the optimal polynomial, the variational method must be optimal with respect to the error functional among all linear iterative methods.

To establish error bounds on the solution, (2.7.1) must be manipulated somewhat

Since $\bar{x}_i = \bar{x}_0 + P^*_{i-1}(A)\bar{r}_0$,

$$\bar{x}-\bar{x}_i = \bar{x}-\bar{x}_0 - P^*_{i-1}(A)\bar{r}_0, \text{ and}$$

$$A(\bar{x}-\bar{x}_i) = A(\bar{x}-\bar{x}_0) - AP^*_{i-1}(A)\bar{r}_0 =$$

$$\bar{r}_i = \bar{r}_0 - AP^*_{i-1}(A)\bar{r}_0, \text{ or}$$

$$\bar{r}_i = (I - AP^*_{i-1}(A))\bar{r}_0.$$

If $\bar{R}_i(A)$ is defined as the set of polynomials of degree at most i in A, then $(I-AP^*_{i-1}(A))$ is clearly a polynomial in this set. Furthermore it must be the polynomial $R^*_i(A)$ that minimizes the error functional, and consequently

$$(2.7.3) \quad \bar{r}_i = R^*_i(A)\,\bar{r}_0$$

In addition, the error functional may be expressed as

$$(2.7.4) \quad E\mu(\bar{x}_i) = ((\bar{x}-\bar{x}_i), A^{\mu}(\bar{x}-\bar{x}_i)) = (\bar{r}_i, A^{\mu-2}\bar{r}_i)$$
$$= (R^*_i(A)\bar{r}_0, A^{\mu-2}R^*_i(A)\bar{r}_0)$$

Since A is symmetric, it has N orthonormal eigenvectors $\{\bar{v}_j\}_{j=1}^{N}$ satisfying,

$$(2.7.5) \quad A\bar{v}_j = \lambda_j\bar{v}_j,$$

where $\{\lambda_j\}_{j=1}^{N}$ are eigenvalues.

Since $\bar{r}_0 = \sum_{j=1}^{N} t_j\bar{v}_j$ for some scalers $\{t_j\}_{j=1}^{N}$,

$$(2.7.6)\quad R_i^*(A)\bar{r}_0 = \sum_{j=1}^{N} t_j R_i^*(A)\bar{v}_j = \sum_{j=1}^{N} t_j R_i^*(\lambda_j)\bar{v}_j.$$

In this expression $R_i^*(\lambda_j)$ is a polynomial of at most degree i in λ_j having the same form as $R_i^*(A)$ and arises from the relationship in (2.7.5) when $R_i^*(A)$ is multiplied by $\bar{v}_j$. Using (2.7.6) the error functional in (2.7.4) may be expanded as

$$\begin{aligned}(2.7.7)\quad E\mu(\bar{x}_i) &= (\sum_{j=1}^{N} t_j R_i^*(\lambda_j)\bar{v}_j, A^{\mu-2}\sum_{j=1}^{N} t_j R_i^*(\lambda_j)\bar{v}_j)\\ &= (\sum_{j=1}^{N} t_j R_i^*(\lambda_j)\bar{v}_j, \sum_{j=1}^{N} t_j R_i^*(\lambda_j)\lambda_j^{\mu-2}\bar{v}_j)\\ &= \sum_{j=1}^{N} t_j^2\ R_i^{*2}(\lambda_j)\lambda_j^{\mu-2}\\ &\leq (\max_{1<j<N} |\ R_i^*(\lambda_j)|)^2\ (\sum_{j=1}^{N} t_j^2\lambda_j^{\mu-2})\\ &= (\max_{1<j<N} |\ R_i^*(\lambda_j)|)^2\ (\sum_{j=1}^{N} t_j\bar{v}_j, \sum_{j=1}^{N} t_j\lambda_j^{\mu-2}\bar{v}_j)\\ &= (\max_{1<j<N} |\ R_i^*(\lambda_j)|)^2\ (\sum_{j=1}^{N} t_j\bar{v}_j,\ A^{\mu-2}\sum_{j=1}^{N} t_j\bar{v}_j)\\ &= (\max_{1<j<N} |\ R_i^*(\lambda_j)|)^2\ (\bar{r}_0,\ A^{\mu-2}r_0)\\ &= Q_i^2\ E\mu(\bar{x}_0)\end{aligned}$$

where

$$(2.7.8)\quad Q_i = \max_{1<j<N} |R_i^*(\lambda_j)|$$

For general eigenvalue distributions one cannot find the minimum polynomial and evaluate Q_i. However upper bounds on Q_i can be obtained. For positive definite A, Engeli, Ginzburg, Rutishauser and Strefel [5] used the following approach. They let [a,b], a,b>0 be an interval known to contain all the eigenvalues of A. Then to obtain a bound on Q_i, they chose $R_i(\lambda)$ to be the unique polynomial in $\bar{R}_i(\lambda)$ that minimizes the deviation from 0 on [a,b] viz., the normalized Chebyshev polynomial on [a,b], i.e.

$$R_i(\lambda) = T_i\left(\frac{-2\lambda+a+b}{b-a}\right)/T_i\left(\frac{b+a}{b-a}\right),$$

where $T_k(z) = \cos(k \arccos(z))$ is the kth order Chebyshev polynomial in z. This choice gives [4] and [5])

$$Q_i \leq 2\left(\frac{1-\sqrt{\alpha}}{1+\sqrt{\alpha}}\right)^i\text{: for } i > 0,$$

where $\alpha = a/b = \lambda\min/\lambda\max$. As a consequence, the iterates $\bar{x}_i$, $i \geq 0$ satisfy

$$(2.7.9)\quad ||\bar{x}-\bar{x}_i||A^{\mu} \leq 2\left(\frac{1-\sqrt{\alpha}}{1+\sqrt{\alpha}}\right)^i ||\bar{x}-\bar{x}_0||A^{\mu},$$

where $||\bar{x}-\bar{x}_i||A^{\mu} = ((\bar{x}-\bar{x}_i), A^{\mu}(\bar{x}-\bar{x}_i))^{1/2} = E_{\mu}(\bar{x}_i)^{1/2}$

2.8 Computational Aspects

As mentioned previously, the Lanczos algorithm forces the new direction vector $\bar{p}_{i+1}$, to depend only on $\bar{p}_i$ and $\bar{p}_{i-1}$. It was also mentioned that this strategy was far more efficient than "brute force" strategies that systematically force the new direction vector, $\bar{p}_{i+1}$, to be orthogonal to each of the previous directions $\{\bar{p}_0,\bar{p}_1\ldots,\bar{p}_i\}$. Although the Lanczos algorithm results in substantial computational savings, further reductions in computational effort are possible by using alternate formulae to calculate the direction vectors.

If

$$(2.8.1)\quad \tilde{p}_{i+1} = \bar{r}_{i+1} + b_i\bar{p}_i$$

where

$$(2.8.2)\quad b_i = (-\bar{r}_{i+1},\, A^{\mu}\bar{p}_i)/(\bar{p}_i,\, A^{\mu}\bar{p}_i),$$

and $\bar{p}_{i+1}$ is given by the Lanczos algorithm, then

$$(2.8.3)\quad \tilde{p}_{i+1} = -\, a_i\bar{p}_{i+1}$$

Proof: Since $\bar{r}_{i+1} = \bar{r}_i - a_iA\bar{p}_i$, (2.8.1) may be expanded as,

$$(2.8.4)\quad \tilde{p}_{i+1} = \bar{r}_i - a_iA\bar{p}_i + b_i\bar{p}_i.$$

Since $\bar{r}_i \varepsilon \{\bar{p}_0,\, \bar{p}_1,\ldots\bar{p}_i\}$ by (2.6.1b), (2.8.4) may be rewritten as,

$$(2.8.5)\quad \tilde{p}_{i+1} = \bar{u} - a_iA\bar{p}_i$$

where $\bar{u} \varepsilon \{\bar{p}_0,\bar{p}_1,\ldots\bar{p}_i\}$.

Moreover taking the inner product with $A^{\mu}\bar{p}_j$ yields.

$$(2.8.6)\quad (\tilde{p}_{i+1},\, A^{\mu}\bar{p}_j) = (\bar{r}_{i+1},\, A^{\mu}\bar{p}_j)+b_i\ (\bar{p}_i,\, A^{\mu}\bar{p}_j)$$

If $j=i$, the two terms on the right hand side cancel by the definition of b_i. If $j<i$, the last term is 0 by the orthogonality of the $\bar{p}$'s. Furthermore, the other term may be regrouped as

$(\bar{r}_{i+1}, A^{\mu}\bar{p}_j) = (\bar{r}_{i+1}, A^{\mu-1}(A\bar{p}_j))$

By (2.6.1a) $Ap_j \in \{\bar{p}_0, \bar{p}_1, \ldots, \bar{p}_{j+1}\}$ so that

$(\bar{r}_{i+1}, A^{\mu-1}(A\bar{p}_j)) = (\bar{r}_{i+1}, A^{\mu-1}(\in \{\bar{p}_0, \bar{p}_1, \ldots \bar{p}_{j+1}\}))$

Since (2.6.d) states that $(\bar{r}_i, A^{\mu-1}\bar{p}_j) = 0$ for $j<i$, this last term is 0, and

(2.8.7) $(\tilde{p}_{i+1}, A^{\mu}\bar{p}_j) = 0$ for $j<i+1$

Also, by the Lanczos algorithm

(2.8.8) $\bar{p}_{i+1} = \bar{v} + A\bar{p}_i$

where $\bar{v}\in \{\bar{p}_0, \bar{p}_1, \ldots \bar{p}_i\}$, and

(2.8.9) $(\bar{p}_{i+1}, A^{\mu}\bar{p}_j) = 0$, $j<i+1$

In addition multiplying (2.8.8) by a_i and adding this to (2.8.5) gives

$\tilde{p}_{i+1} + a_i\bar{p}_{i+1} = \bar{u} - a_iA\bar{p}_i + a_i\bar{v}_i + a_iA\bar{p}_i = \bar{u} + a_i\bar{v}$

Since $\bar{u}$ and $\bar{v}$ are each $\in \{\bar{p}_0, \bar{p}_1, \ldots \bar{p}_i\}$,

(2.8.10) $\tilde{p}_{i+1} + a_i\bar{p}_{i+1} \in \{\bar{p}_0, \bar{p}_1 \ldots \bar{p}_i\}$.

Since

$((\tilde{p}_{i+1} + a_i\bar{p}_{i+1}), A^{\mu} (\tilde{p}_{i+1} + a_i\bar{p}_{i+1})) =$

$(\tilde{p}_{i+1}, A^{\mu}(\tilde{p}_{i+1}+a_i\bar{p}_{i+1})) + a_i(\bar{p}_{i-1}, A^{\mu}(\tilde{p}_{i+1} + a_i\bar{p}_{i+1}))$

and since (2.8.10) implies that the last two terms are 0 by (2.8.7) and (2.8.9) respectively,

$$((\tilde{p}_{i+1} + a_i\bar{p}_{i+1}), A^\mu(\tilde{p}_{i+1} + a_i\bar{p}_{i+1})) = 0,$$

and consequently $\tilde{p}_{i+1} = - a_i\bar{p}_{i+1}$.

Q.E.D.

The new direction vectors are therefore the same as the old ones except that they have a different normalization. Since this does not effect the orthogonality relationships however, all of the results of previous sections hold for the new set of direction vectors.

Two problems arise when equations (2.8.1) and (2.8.2) are incorporated into the variational algorithm. First, since the new set of $\tilde{p}$'s are calculated from the old set of $\bar{p}$'s, it seems that a method must be incorporated to calculate and store the old set. As a consequence, some of the gains in computational efficiency that result from the new method may be lost. Second, since $\tilde{p}_{i+1}$ has a different length than $\bar{p}_{i+1}$, a new a'_{i+1} must be defined so that each new march, $a'_{j+1}\tilde{p}_{j+1}$, is equal to the corresponding old march, $a_{j+1}\bar{p}_{j+1}$. This condition is necessary so that $\bar{x}_{j+2}$.

$$(\bar{x}_{j+2} = \bar{x}_{j+1} + a_{j+1}\bar{p}_{j+1} = \bar{x}_{j+1} + a'_{j+1}\tilde{p}_{j+1})$$

has the same value under the new set of direction vectors. (Since the subscript j+1 is arbitrary, this condition insures that all of the $\bar{x}$'s and $\bar{r}$'s remain unchanged when these new vectors are incorporated into the variational algorithm.) Fortunately, both of these problems share one simple solution.

If b'_{i+1} and a'_{i+1} are the quantities obtained by replacing $\bar{p}_i$ with $\tilde{p}_i$ in the equations for b_{i+1} and a_{i+1}, then

(2.8.11) $\tilde{p}_{i+1} = \bar{r}_{i+1} + b_i\bar{p}_i = \bar{r}_{i+1} + b'_i\tilde{p}_i$,

and

(2.8.12) $\bar{x}_{i+2} = \bar{x}_{i+1} + a_{i+1}\bar{p}_{i+1} = \bar{x}_{i+1} + a'_{i+1}\tilde{p}_{i+1}$

As a consequence of (2.8.11) and (2.8.12), the new direction vectors can be calculated in the variational algorithm without reference to the old set and do not affect the convergence.

Proof: Since $\bar{r}_0$ and $\bar{p}_0$ would not be effected by incorporating the new vectors into the variational algorithm, a_0, $\bar{x}_1$, and $\bar{r}_1$, would remain the same. It is therefore sufficient to show that for given values of $\bar{x}_{i+1}$ and $\bar{r}_{i+1}$, the same values of $\bar{x}_{i+2}$ and $\bar{r}_{i+2}$ result with or without these modifications.

Since

$$b'_i = (-\bar{r}_{i+1}, A^\mu\tilde{p}_i)/(\tilde{p}_i, A^\mu\tilde{p}_i), \text{ and } \tilde{p}_i = -a_{i-1}\bar{p}_i,$$

$$b'_i = (-\bar{r}_{i+1}, -a_{i-1}A^\mu\bar{p}_i)/(-a_{i-1}\bar{p}_i, -a_{i-1}A^\mu\bar{p}_i) = -b_i/a_{i-1}$$

Furthermore,

$$\tilde{p}_{i+1} = \bar{r}_{i+1} + b'_i\tilde{p}_i = \bar{r}_{i+1} + \left(\frac{-b_i}{a_{i-1}}\right)((-a_{i-1})\bar{p}_i)$$

$$= \bar{r}_{i+1} + b_i\bar{p}_i$$

Since this last expression is equivalent to (2.8.1) using $\tilde{p}_i$ in place of $\bar{p}_i$ in (2.8.2) is a valid means of generating $\tilde{p}_{i+1}$.

In addition, since

$$a'_{i+1} = (\bar{r}_{i+1}, A^{\mu-1}\tilde{p}_{i+1})/(\tilde{p}_{i+1}, A^{\mu}\tilde{p}_{i+1})$$

$$= (\bar{r}_{i+1}, -a_i A^{\mu-1}\bar{p}_{i+1})/(-a_i\bar{p}_{i+1}, -a_i A^{\mu}\bar{p}_{i+1})$$

$$= -(\frac{a_{i+1}}{a_i}),$$

$$a'_{i+1}\tilde{p}_{i+1} = -(\frac{a_{i+1}}{a_i})(-a_i\bar{p}_{i+1}) = a_{i+1}\bar{p}_{i+1}.$$

This implies that $\bar{x}_{i+2}$

$$(\bar{x}_{i+2} = \bar{x}_{i+1} + a_{i+1}\bar{p}_{i+1} = \bar{x}_{i+1} + a'_{i+1}\tilde{p}_{i+1})$$

remains unchanged, and consequently so does $\bar{r}_{i+2}$.

Q.E.D.

The quantities a'_i and b'_i will henceforth be referred to as a_i and b_i respectively. If A is positive definite, then for each $i \geq 0$,

$$(2.8.13) \quad a_i = (\bar{r}_i, A^{\mu-1}\bar{r}_i)/(\bar{p}_i, A^{\mu}\bar{p}_i) \neq 0$$

and

$$(2.8.14) \quad b_i = (\bar{r}_{i+1}, A^{\mu-1}\bar{r}_{i+1})/(\bar{r}_i, A^{\mu-1}\bar{r}_i).$$

Proof: The proof of (2.8.13) is by induction on i. Since $\bar{p}_0=\bar{r}_0$, and $(\bar{r}_0, A^{\mu-1}\bar{r}_0)\neq 0$, (2.8.13) holds for i=0 by substituting $\bar{r}_0$ for $\bar{p}_0$ in the definition of a_0. Assume that it also holds for $i \leq k$. Then by the definition of a_{k+1},

$$(\bar{p}_{k+1}, A^{\mu}\bar{p}_{k+1})\ a_{k+1} = (\bar{r}_{k+1}, A^{\mu-1}\bar{p}_{k+1}).$$

The right hand side may be expanded by using (2.8.1) so that

$$(\bar{r}_{k+1}, A^{\mu-1}\bar{p}_{k+1}) = (\bar{r}_{k+1}, A^{\mu-1}(\bar{r}_{k+1}+b_k\bar{p}_k))$$

$$= (\bar{r}_{k+1}, A^{\mu-1}\bar{r}_{k+1}) + b_k(\bar{r}_{k+1}, A^{\mu-1}\bar{p}_k)$$

Since (2.6.1d) states that $(\bar{r}_i, A^{\mu-1}\bar{p}_j) = 0$ for $j < i$, the last term is 0, and

$$(2.8.15) \quad (\bar{p}_{k+1}, A^{\mu}\bar{p}_{k+1})\, a_{k+1} = (\bar{r}_{k+1}, A^{\mu-1}\bar{r}_{k+1}).$$

Therefore,

$$a_{k+1} = \frac{(\bar{r}_{k+1}, A^{\mu-1}\bar{r}_{k+1})}{(\bar{p}_{k+1}, A^{\mu}\bar{p}_{k+1})}, \text{ and}$$

(2.8.13) has been proved by induction.

To prove (2.8.14) first note that since $\bar{r}_{k+1} = \bar{r}_k - a_k A\bar{p}_k$,

$$(\bar{r}_{k+1}, A^{\mu-1}\bar{r}_{k+1}) = (\bar{r}_{k+1}, A^{\mu-1}(\bar{r}_k - a_k A\bar{p}_k))$$

$$= (\bar{r}_{k+1}, A^{\mu-1}\bar{r}_k) - a_k(\bar{r}_{k+1}, A^{\mu}\bar{p}_k).$$

Since (3.6.1e) states that $(\bar{r}_i, A^{\mu-1}\bar{r}_j) = 0$ for $i \neq j$, the term, $(\bar{r}_{k+1}, A^{\mu-1}\bar{r}_k)$, is 0, and

$$(2.8.16) \quad (\bar{r}_{k+1}, A^{\mu-1}\bar{r}_{k+1}) = - a_k(\bar{r}_{k+1}, A^{\mu}\bar{p}_k)$$

By (2.8.15) it is true that

$$(2.8.17) \quad (\bar{r}_k, A^{\mu-1}\bar{r}_k) = a_k(\bar{p}_k, A^{\mu}\bar{p}_k)$$

Dividing (2.8.16) by (2.8.17) implies that

$$\frac{(\bar{r}_{k+1}, A^{\mu-1}\bar{r}_{k+1})}{(\bar{r}_k, A^{\mu-1}\bar{r}_k)} = - \frac{a_k \, (\bar{r}_{k+1}, A^{\mu}\bar{p}_k)}{a_k \, (\bar{p}_k, A^{\mu}\bar{p}_k)}$$

$$= \frac{(-\bar{r}_{k+1}, A^{\mu-1}\bar{p}_K)}{(\bar{p}_k, A^{\mu}\bar{p}_k)} = b_k$$

Thus, (2.8.14) has been proved.

Incorporating the new set of direction vectors in algorithm (2.4.1) and using these last definitions for a_i and b_i yields algorithm (2.8.1).

Algorithm (2.8.1): The Variational Method for Positive Definite Systems:

Step 1: Choose an initial approximation $\bar{x}_0$ to $\bar{x}$.

Compute $\bar{r}_0 = \bar{f} - A\bar{x}_0$

Set $\bar{p}_0 = \bar{r}_0$

and $i = 0$

Step 2: Compute

$a_i = (\bar{r}_i, A^{\mu-1}\bar{r}_i)/(\bar{p}_i, A^{\mu}\bar{p}_i)$

$\bar{x}_{i+1} = \bar{x}_i + a_i\bar{p}_i$

$\bar{r}_{i+1} = \bar{r}_i - a_i A\bar{p}_i$

$b_i = (\bar{r}_{i+1}, A^{\mu-1}\bar{r}_{i+1})/(\bar{r}_i, A^{\mu-1}\bar{r}_i)$,

and $\bar{p}_{i+1} = \bar{r}_{i+1} + b_i\bar{p}_i$

Step 3: If $\bar{x}_{i+1}$ is sufficiently close to $\bar{x}$, terminate the iteration process; else set $i = i+1$ and go to step 2.

It can be seen that this algorithm involves less work and storage than algorithm (2.4.1).

2.9 The Preconditioned Variational Method

Another means of reducing the computational effort associated with the variational method is to precondition the matrix A. In particuular, if Q is a nonsingular matrix, the system,

(2.9.1) $A\bar{x} = \bar{f}$,

can be converted to the system

(2.9.2) $A'\bar{x}' = \bar{f}'$

where

(2.9.3) $A' = Q^{-1}AQ^{-T}$, $\bar{f}' = Q^{-1}\bar{f}$

and

(2.9.4) $\bar{x}' = Q^T\bar{x}$.

(The primes used in this section bear no relationship to those used in the previous section). Since A' is symmetric and positive definite, the variational methods introduced previously can be used on (2.9.2) to find approximations to $\bar{x}'$ from which approximations to $\bar{x}$ may be obtained from (2.9.4). Since this preconditioning increases the work per iteration, it must significantly reduce the number of iterations needed to also reduce the total execution time.

One means of reducing the number of iterations is to choose the matrix Q so that $M \equiv QQ^T$ is a symmetric positive definite matrix that is close to A. Since A may always be written as

(2.9.5) $A = M - R = QQ^T - R$,

for some matrix R, this choice of Q forces the elements of R to be small with respect to those of A and therefore with respect to the elements of M. Since the elements of R are small with respect to those of M, and since

$$(2.9.6) \quad A' = Q^{-1}AQ^{-T} = Q^{-1}MQ^{-T} - Q^{-1}RQ^{-T} = I - Q^{-1}RQ^{-T},$$

it is hoped that the elements of $Q^{-1}RQ^{-T}$ will be small with respect to those of I. If this condition holds, the eigenvalues of A' will be clustered about 1 by the Gerschgorin theorem of eigenvalue location. As a consequence the quantity $\alpha = \lambda_{min}/\lambda_{max}$ will be close to 1, and convergence will be more rapid by (2.7.9).

If the variational method were modified to solve (2.9.2) it would have the following form.

Algorithm (2.9.1): <u>A Possible Form of the Preconditioned Variational Method</u>

```
Step 1:  Choose $\bar{x}_0$
         Compute $\bar{x}_0' = Q^T\bar{x}_0$
         Compute $A' = Q^{-1}AQ^{-T}$
         Compute $\bar{f}' = Q^{-1}\bar{f}$
         Compute $\bar{r}_0' = \bar{f}' - A'\bar{x}_0'$
         Set $\bar{p}_0' = \bar{r}_0'$ and $i = 0$
Step 2:  Compute
         $a_i' = (\bar{r}_i', A'\mu^{-1}\bar{r}_i')/(\bar{p}_i', A'\mu\bar{p}_i')$,
         $\bar{x}'_{i+1} = \bar{x}'_i + a_i'\bar{p}_i'$
         $\bar{r}_{i+1}' = \bar{r}_i' - a_i'A'\bar{p}_i'$
         $b_i' = (\bar{r}_{i+1}', A'\mu^{-1}\bar{r}'_{i+1})/(\bar{r}_i', A'\mu^{-1}\bar{r}'_i)$
```

and $\bar{p}'_{i+1} = \bar{r}'_{i+1} + b_i'\bar{p}'_i$

Solve $\bar{x}_i = Q^{-T}\bar{x}'_i$

Step 3: If $\bar{x}_i$ is sufficiently close to $\bar{x}$, terminate the iteration process; else set $i = i+1$ and go to step 2.

As mentioned previously, the additional work involved in applying the preconditioning must be small so that the decrease in the number of iterations reduces the total computational effort. As a consquence, there are several reasons not to apply the variational method to (2.9.2) as straightforwardly as in algorithm (2.9.1). First, computing $\bar{x}'_i$, A', and f' in step 1 and solving $\bar{x}_i = Q^{-T}\bar{x}'_i$ in step 2 may be expensive. Second, since A' may not have as nice of a sparity structure as A, the work involved in computing matrix-vector products may significantly increase the work per iteration. Last, the quantities A', $\bar{x}'$, and $\bar{f}'$ require storage in addition to that required for A, $\bar{x}$, and $\bar{f}$. What is needed is an algorithm that gives the convergence properties of algorithm (2.9.1) while working directly with the system, $A\bar{x}=\bar{f}$.

Before developing this idea any further it is necessary to establish two relationships between the primed and the unprimed systems. As mentioned previously, $Q^{-T}\bar{x}'_k = \bar{x}_k$ for all k. Since $\bar{x}'_{k+1} = a'_k\bar{p}'_k + \bar{x}'_k$, it is necessary to define $Q^{-T}\bar{p}'_k = \bar{p}_k$, so that

$$(2.9.7) \quad Q^{-T}\bar{x}'_{k+1} = Q^{-T}(\bar{x}'_k+a'_k\bar{p}'_k) = \bar{x}_k + a'_k\bar{p}_k = \bar{x}_{k+1}.$$

(Notice that the coefficient a'_k is still primed. It will be seen shortly that this is a necessary condition to maintain the same convergence rate in the unprimed system.) In addition, by (2.9.3) and (2.9.4),

$$(2.9.8)\quad \bar{r}'_k = \bar{f}'-A'\bar{x}'_k = Q^{-1}\bar{f}-\ Q^{-1}AQ^{-T}Q^T\bar{x}_k$$
$$= Q^{-1}\ (\bar{f}-A\bar{x}) = Q^{-1}\bar{r}.$$

Having established these relationships it remains to show how to obtain the convergence properties of algorithm (2.9.1) while working directly with the system $A\bar{x}=\bar{f}$. Since algorithm (2.9.1) converges in $m \leq n$ steps, it must true that

$$(2.9.9)\quad \bar{x}' - \bar{x}'_m = \bar{x}'-\bar{x}'_0 - \sum_{j=o}^{m-1} a'_j\bar{p}'_j = \bar{0}.$$

However, since $\bar{x}_k = Q^{-T}\bar{x}'_k$ and $\bar{p}_k = Q^{-T}\bar{p}'_k$ for all k, (2.9.9) may be multiplied by Q^{-T} to yield

$$(2.9.10)\quad \bar{x}-\bar{x}_m = \bar{x}-\bar{x}_0 - \sum_{j=o}^{m-1} a'_j\bar{p}_j = \bar{0}.$$

Thus, the way to obtain rapid convergence in the unprimed system, is to calculate approximations to $\bar{x}$ by moving in directions $\{\bar{p}_k=Q^{-T}\bar{p}'_k\}_{k=o}^{m-1}$ and using the same coefficients $\{a_k = a'_k\}_{k=o}^{m-1}$ that are used in the primed system.

A difficulty that arises in implementing this idea is the calculation of the direction vectors. In algorithm (2.9.1) the direction vectors are calculated as $\bar{p}'_0 = \bar{r}'_0$ when i = 0 and $\bar{p}'_{i+1} = \bar{r}'_{i+1} + b'_i\bar{p}'_i$ otherwise. Keeping with the definition, $\bar{p}_k = Q^{-T}\bar{p}'_k$, implies that $\bar{p}_0=Q^{-T}\bar{r}'_0$ when i = 0, and $\bar{p}_{k+1} = Q^{-T}\bar{r}'_{k+1}+b'_k\bar{p}_k$ otherwise. The problem is that an algorithm which works directly with unprimed quantities would calculate values of $\bar{r}_k$ which are equal to $Q\bar{r}'_k$. Since the calculation of the direction vectors requires the quantity, $Q^{-T}\bar{r}'_k$, it is necessary to define the quantity

(2.9.11) $\tilde{r}_k = Q^{-T}\bar{r}'_k = Q^{-T}Q^{-1}\bar{r}_k = M^{-1}\bar{r}_k.$

As a consequence, values of $\tilde{r}_k$ can be obtained by solving the equation $M\tilde{r}_k = \bar{r}_k$, after each calculation of $\bar{r}_k$. Values of $\bar{p}_k$ can then be calculated as

(2.9.11a) $\bar{p}_0 = \tilde{r}_0$ for $i = 0$, and

(2.9.11b) $\bar{p}_{k+1} = \tilde{r}_{k+1} + b'_k\bar{p}_k$ for $k > 0$.

Another requirement is to be able to calculate the coefficients a'_k and b'_k in terms of unprimed quantities for use in updating values of $\bar{x}_{k+1}$ and $\bar{p}_{k+1}$. To develop a formula for a'_k, first note that the numerator of a'_k in algorithm (2.9.1) is equivalent to

(2.9.12) $(\bar{r}_k{}' \ A'\mu^{-1}\bar{r}_k{}') = \bar{r}'_k{}^T A'\mu^{-1}\bar{r}'_k$

$= \bar{r}'_k T(Q^{-1}AQ^{-T})\mu^{-1}\bar{r}'_k$

Since $\bar{r}'_k = Q^{-1}\bar{r}_k$, $\bar{r}'^T_k = \bar{r}_k^T Q^{-T}$. When this expression for $\bar{r}'_k{}^T$ is used, (2.9.12) becomes

(2.9.13) $(\bar{r}'_k, A'\mu^{-1}\bar{r}'_k) = \bar{r}_k T\ Q^{-T}(Q^{-1}AQ^{-T})\mu^{-1}\bar{r}'_k$

$= \bar{r}_k^T(Q^{-T}Q^{-1}A)\mu^{-1}(Q^{-T}\bar{r}'_k)$

$= \bar{r}_k^T(M^{-1}A)\mu^{-1}\tilde{r}_k = (\bar{r}_k, (M^{-1}A)\mu^{-1}\tilde{r}_k)$

Next note that the denominator for a'_k is equivalent to

(2.9.14) $(\bar{p}_k{}', A'\mu\bar{p}'_k) = \bar{p}'^T_k A'\mu\bar{p}'_k =$

$\bar{p}'_k{}^T(Q^{-1}AQ^{-T})\mu\bar{p}'_k =$

$$(\bar{p}_k^{'T}Q^{-1})A(Q^{-T}Q^{-1}A)^{\mu-1}(Q^{-T}\bar{p}_k^{'})$$

$$= (\bar{p}_k^{T})\ A\ (M^{-1}A)^{\mu-1}(\bar{p}_k) = (\bar{p}_k,\ A(M^{-1}A)^{\mu-1}\bar{p}_k)$$

By virtue of (2.9.13) and (2.9.14)

$$(2.9.15)\quad a'_k = \frac{(\bar{r}_k^{'}A'^{\mu-1}\bar{r}_k^{'})}{(\bar{p}_k^{'}A'^{\mu}\bar{p}_k^{'})} = \frac{(\bar{r}_k,\ (M^{-1}A)^{\mu-1}\tilde{r}_k)}{(\bar{p}_k,\ A(M^{-1}A)^{\mu-1}\bar{p}_k)}$$

Furthermore, by (2.9.13)

$$(2.9.16)\quad b'_k = \frac{(\bar{r}_{k+1}^{'},\ A'^{\mu-1}\bar{r}_{k+1}^{'})}{(\bar{r}_k^{'},\ A'^{\mu-1}\bar{r}_k^{'})} = \frac{(\bar{r}_{k+1},(M^{-1}A)^{\mu-1}\tilde{r}_{k+1})}{(\bar{r}_k,\ (M^{-1}A)^{\mu-1}\tilde{r}_k)}$$

From the previous development it is clear that an algorithm which extends the convergence properties of algorithm (2.9.1) to the unprimed system, must use the calculational scheme of (2.9.10) while using equations (2.9.11), (2.9.15) and (2.9.16) to calculate $\bar{p}_k$, $a_k^{'}$, and $b_k^{'}$. It can be seen by comparison to algorithm (2.9.1) that the following algorithm effectively incorporates these changes.

Algorithm (2.9.2) The Preconditioned Variational Method

Step 1: Choose $\bar{x}_0$

Compute $\bar{r}_0 = \bar{f} - A\bar{x}_0$

Solve $M\ \tilde{r}_0 = \bar{r}_0$

Set $\bar{p}_0 = \tilde{r}_0$, and $i = 0$

Step 2: Compute

$a_i = (\bar{r}_i,\ (M^{-1}A)^{\mu-1}\tilde{r}_i)/(\bar{p}_i, A(M^{-1}A)^{\mu-1}\bar{p}_i)$

$\bar{x}_{i+1} = \bar{x}_i + a_i\bar{p}_i$

$\bar{r}_{i+1} + \bar{r}_i - a_iA\bar{p}_i$

Solve $(M\tilde{r}_{i+1} = \bar{r}_{i+1})$

$b_i = (\bar{r}_{i+1}, (M^{-1}A)^{\mu-1}\tilde{r}_{i+1})/(\bar{r}_i,(M^{-1}A)^{\mu-1}\tilde{r}_i)$

and $\bar{p}_{i+1} = \tilde{r}_{i+1} + b_i\bar{p}_i$

Step 3: If $\bar{x}_{i+1}$ is sufficiently close to $\bar{x}$, terminate the iteration process; else set $i = i+1$ and go to Step 2.

It should be observed that although the variables a_k' and b_k' are referred to as a_k and b_k in this algorithm, they still have the values that would be obtained in algorithm (2.9.1).

Algorithm (2.9.2) minimizes the $A(M^{-1}A)^{\mu-1}$ norm of the error over subspaces of increasing dimension so that the iterates satisfy

$$(2.9.17) \quad ||\bar{x}-\bar{x}_i||_{A(M^{-1}A)^{\mu-1}} \leq 2\left(\frac{1-\sqrt{\alpha}}{1+\sqrt{\alpha}}\right)^i ||\bar{x}-\bar{x}_0||_{A(M^{-1}A)^{\mu-1}},$$

where $\alpha = \lambda_{min}(A')/\lambda_{max}(A')$.

Proof: Since algorithm (2.9.1) is simply an extension of the variational method to the system, $A'\bar{x}' = \bar{f}'$, equation (2.7.9) must apply to the iterates of this method so that

$$(2.9.18) \quad ||\bar{x}' - \bar{x}_i'||_{A'^{\mu}} \leq 2\left(\frac{1-\sqrt{\alpha}}{1+\sqrt{\alpha}}\right)^i ||\bar{x}' - \bar{x}_0'||_{A'^{\mu}}$$

where $\alpha = \lambda_{min}(A')/\lambda_{max}(A')$. Since algorithm (2.9.2) is derived from algorithm (2.9.1) by multiplying the $\bar{x}'$'s and $\bar{p}'$'s by Q^{-T}, the convergence of the $\bar{x}$'s in algorithm (2.9.2) is constrained by the convergence of the $\bar{x}'$'s in (2.9.18). By comparison of (2.9.18) to (2.9.17), it is sufficient to show that

$$(2.9.19) \quad ||\bar{x}' - \bar{x}_i'||_{A'^{\mu}} = ||\bar{x} - \bar{x}_i||_{A(M^{-1}A)^{\mu-1}}$$

$$||\bar{x}' - \bar{x}_i'||_{A'^{\mu}} = ((\bar{x}' - \bar{x}_i'), A'^{\mu}(\bar{x}' - \bar{x}_i'))^{\frac{1}{2}}$$

$$= ((\bar{x}'-\bar{x}_i')^T(Q^{-1}AQ^{-T})^{\mu}(\bar{x}'-\bar{x}_i'))^{\frac{1}{2}}$$

$$= \{(\bar{x}'-\bar{x}_i')^T Q^{-1})A(Q^{-T}Q^{-1}A)^{\mu-1}(Q^{-T}(\bar{x}'-\bar{x}_i'))\}$$

$$= \{(\bar{x}-\bar{x}_i)^T A(M^{-1}A)^{\mu-1}(\bar{x}-\bar{x}_i)\}^{\frac{1}{2}}$$

$$= ((\bar{x}-\bar{x}_i),\ A(M^{-1}A)^{\mu-1}(\bar{x}-\bar{x}_i))^{\frac{1}{2}}$$

$$= ||\bar{x}-\bar{x}_i||_{A(M^{-1}A)^{\mu-1}}$$

Using this result to replace the A'^{μ} norms of the current and initial errors in (2.9.18) gives (2.9.17).

Q.E.D.

2.10 Preconditioning and the Preconditioned Conjugate Gradient Method

In order to apply the preconditioned variational method, it is necessary to have a matrix Q such that $QQ^T=M$. Since the matrix M must be inverted on every iteration of algorithm (2.9.2), it is wise to choose Q to be a lower triangular matrix L' so that $M = L'L'^T$. One means of obtaining L' is to force its nonzero structure to agree with the lower triangular part of A, and then to force the product $L'L'^T$ to be identical to A in the nonzero locations of A. Unfortunately, however, the product $L'L'^T$ will also produce nonzero products in some of the zero locations of A. As a consequence, the product $L'L'^T$ = M is only approximately equal to A. For this reason the factorization is said to be incomplete, and this preconditioning method is refered to as incomplete Cholesky factorization.

To make the inversion of $M = L'L'^T$ more efficient, it is wise to rewrite this product. If D is a diagonal matrix containing the diagonals of L', and D^{-1} is its inverse, then

(2.10.1) $L'L'^T = (L'D^{-1})(DL'^T) = LU$

When L is a lower triangular matrix having 1's on its diagonal, and U is an upper triangular matrix. Because L has 1's on its diagonal, less storage is needed for it and less computational effort is required in the forward solution of

$$L(U\tilde{r}_i) = \bar{r}_i$$

If this factorization scheme is implemented in algorithm (2.9.2) with μ set to 1, the preconditioned conjugate gradient algorithm used in COMMIX-1B is obtained.

Algorithm (2.10.1) The Preconditioned Conjugate Gradient Method

```
Step 1:  Choose $\bar{x}_0$
         Compute $\bar{r}_0 = \bar{f} - A\bar{x}_0$
         Solve $M\tilde{r}_0 = \bar{r}_0$ $(LU\tilde{r}_0 = \bar{r}_0)$
         Set $\bar{p}_0 = \tilde{r}_0$, and $i = 0$
Step 2:  Compute
         $a_i = (\bar{r}_i, \tilde{r}_i)/(\bar{p}_i, A\bar{p}_i)$
         $\bar{x}_{i+1} = \bar{x}_i + a_i\bar{p}_i$
         $\bar{r}_{i+1} = \bar{r}_i - a_iA\bar{p}_i$
         Solve $M\tilde{r}_{i+1} = \bar{r}_{i+1}$ $(LU\tilde{r}_{i+1} = \bar{r}_{i+1})$
         $b_i = (\bar{r}_{i+1}, \tilde{r}_{i+1})/(\bar{r}_i,\tilde{r}_i)$
         and $\bar{p}_{i+1} = \tilde{r}_{i+1} + b_i\bar{p}_i$
Step 3:  If $\bar{x}_{i+1}$ is sufficiently close to $\bar{x}$, terminate the iteration
         process; else set $i = i+1$ and go to step 2.
```

It should be observed that setting $\mu = 1$ in the preconditioned variational method makes the preconditioned conjugate gradient method very efficient since all $(M^{-1}A)^{\mu-1}$ terms in the inner products are eliminated.

CHAPTER 3

COMMIX-1B

3.1 Introduction

COMMIX-1B is a single phase, 3 dimensional, thermal hydraulics code that can be used to analyze both steady state and transient problems. The solution strategy used in COMMIX-1B is to successively solve the momentum, continuity, and energy equations in each outer iteration to obtain the most recent update of fluid field parameters. Convergence is obtained when the relative differences between the values obtained on successive outer iterations is small for every calculated field parameter. The thermal interaction between solid structures and the fluid involves both heat conduction within the solid and convective heat transfer at the interface which is characterized by an empirically based heat transfer coefficient. Momentum interaction between solid structures and the fluid is based on the use of friction factor correlations. Although COMMIX-1B is primarily intended for the analysis of the thermal hydraulics of nuclear reactor systems, its versatile modeling allows for the analysis of any thermal hydraulic processes in single or multicomponent systems.

Since it was mentioned previously that the conjugate gradient method was implemented in the momentum section of COMMIX-1B, it is necessary to understand the overall behavior of this section to see exactly how the conjugate gradient method is applied. Within an outer iteration, COMMIX-1B first uses the previous iterate values of u, v, and w, the x, y, and z velocity fields, to calculate the convection and diffusion coefficients to be used in the x, y, and z momentum equations. Next, the pressure differences are removed from the source terms of the x, y, and z momentum equations by expanding the central cell velocities as the sum of pressure dependent and pressure independent contributions. The momentum equations are then solved explicitly to obtain the pressure independent contributions to the velocities which depend only on the convection and diffusion of momentum, gravity, and frictional resistance. Each velocity in the continuity equation is next replaced by the sum of its pressure dependent and pressure independent contributions. After this replacement, the previously calculated pressure independent contributions are taken to the right hand side to yield a system of pressure equations that is constrained to satisfy continuity. The conjugate gradient method is then used to solve for the field pressures. Next, the pressure dependent contributions are calculated and added to the pressure independent contributions to yield the total velocities. From this point the code moves on to solve the energy equation.

Since the system of pressure equations is derived from the momentum and continuity equations, several concepts must be discussed to understand the previous solution strategy. In section 2 the general form of the conservation equations will be derived. Section 3 will discuss the types of control volumes used in COMMIX-1B, and section 4 will show the integration of the general conservation equation over these control volumes. In section 5 the pressure equation will be derived from the momentum and continuity equations, and in section 6 a convergence acceleration technique called mass rebalancing will be described. Finally, the solution strategies used in COMMIX-1B will be given in section 7.

3.2 General Form of Conservation Equations

As mentioned previously, COMMIX-1B solves finite difference approximations to the conservation equations of mass, momentum, and energy to obtain successive updates to the fluid properties. Fortunately, the similarities in the transport of these quantities allow the conservation equations to be derived as one general form. For each conservation equation the rate of change of the quantity of interest may be written in general terms by the following balance equation for a control volume.

$$(3.2.1)\quad \begin{bmatrix}\text{Rate of Change}\\ \text{in C.V.}\end{bmatrix} + \begin{bmatrix}\text{Rate of Convection}\\ \text{out of C.V.}\end{bmatrix} = \begin{bmatrix}\text{Rate of Diffusion}\\ \text{into C.V.}\end{bmatrix} + \begin{bmatrix}\text{Rate of Production}\\ \text{from Sources in C.V.}\end{bmatrix}$$

To expand this balance equation let ϕ be the dependent variable defined to be 1 for mass conservation, u, v, or w for x, y, or z momentum conservation, and h for energy conservation. Next, use this definition of ϕ to apply the balance equation to a differential cell having dimensions dx, dy, and dz in Cartesian coordinates. The results of this process will be shown for each term in (3.2.1). The unsteady term becomes

$$(3.2.2)\quad \begin{bmatrix}\text{Rate of Change}\\ \text{in C.V.}\end{bmatrix} = dxdydz\frac{\partial\rho\phi}{\partial t}$$

The convection term is given by

$$(3.2.3)\quad \begin{bmatrix}\text{Rate of Convection}\\ \text{out of C.V.}\end{bmatrix} = dydz \left[(\rho u\phi)_{i+\frac{1}{2}}-(\rho u\phi)_{i-\frac{1}{2}}\right]$$

$$+ \quad dxdz \left[(\rho v\phi)_{j+\frac{1}{2}}-(\rho v\phi)_{j-\frac{1}{2}}\right]$$

$$+ \quad dxdy \left[(\rho w\phi)_{k+\frac{1}{2}}-(\rho w\phi)_{k-\frac{1}{2}}\right]$$

The diffusion term is

$$(3.2.4)\quad \begin{bmatrix}\text{Rate of Diffusion}\\ \text{into C.V.}\end{bmatrix} =$$

$$dydz \left[(\Gamma_\phi (-\frac{\partial\phi}{\partial x}))_{i-\frac{1}{2}} - (\Gamma\phi (-\frac{\partial\phi}{\partial x}))_{i+\frac{1}{2}}\right] +$$

$$dxdz \left[(\Gamma_\phi (-\frac{\partial\phi}{\partial y}))_{j-\frac{1}{2}} - (\Gamma\phi (-\frac{\partial\phi}{\partial y}))_{j+\frac{1}{2}}\right] +$$

$$dxdy \left[(\Gamma_\phi (-\frac{\partial\phi}{\partial z}))_{k-\frac{1}{2}} - (\Gamma\phi (-\frac{\partial\phi}{\partial z}))_{k+\frac{1}{2}}\right] ,$$

where Γ_ϕ is the fluid diffusivity for variable ϕ. Finally, the source term is

$$(3.2.5)\quad \begin{bmatrix}\text{Rate of Production}\\ \text{from Sources in C.V.}\end{bmatrix} = S_{v\phi}dxdydz,$$

where $S_{v\phi}$ is the per volume source strength for variable ϕ. By substituting (3.2.2), (3.2.3), (3.2.4), and (3.2.5) into (3.2.1), and dividing by dxdydz the following differential equation is obtained.

$$(3.2.6)\quad \frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x}(\rho u\phi + \frac{\partial}{\partial y}(\rho v\phi) + \frac{\partial}{\partial z}(pw\phi) =$$

$$\frac{\partial}{\partial x}(\Gamma\phi \frac{\partial\phi}{\partial x}) + \frac{\partial}{\partial y}(\Gamma\phi \frac{\partial\phi}{dy}) + \frac{\partial}{\partial_z}(\Gamma\phi \frac{\partial\phi}{\partial_z}) + Sv\phi$$

Since the fluid volume may not occupy the entire control volume, it is necessary to define the volume porosity, γ_v, as the ratio of the fluid volume to the total volume of a given cell. Furthermore, since the entire surface area of a cell may not be open to fluid flow, it is also necessary to define the directional surface porosities γ_x, γ_y, and γ_z as the fractions of surfaces areas perpendicular to the x, y, and z directions that are unobstructed to fluid flow. If these definitions are used in the previous development to replace cell volumes and surface areas with fluid volumes and flow areas, (3.2.6) becomes

$$(3.2.7)\quad \frac{\partial}{\partial t}(\gamma_v \rho \phi) = \frac{\Delta}{\Delta x}(\gamma_x \rho u \phi) + \frac{\Delta}{\Delta y}(\gamma_y \rho v \phi) + \frac{\Delta}{\Delta z}(\gamma_z \rho w \phi)$$

$$= \frac{\Delta}{\Delta x}(\gamma_x \Gamma\phi \frac{\partial \phi}{\partial x}) + \frac{\Delta}{\Delta y}(\gamma_y \Gamma\phi \frac{\partial \phi}{\partial y}) + \frac{\Delta}{\Delta z}(\gamma_z \Gamma\phi \frac{\partial \phi}{\partial z}) + \gamma_v S_v \phi$$

Obviously this result applies to a control cell having finite dimensions Δx, Δy, and Δz. The values of ϕ, s_ϕ, and Γ_ϕ are listed for each conservation equation in Table 1 for Cartesian coordinates and in Table 2 for cylindrical coordinates.

3.3 Control Volumes

In COMMIX-1B, the control volumes are referred to as computational cells and are defined by the locations of cell volume faces with grid points placed in the geometrical centers of the cells. The cells defined in COMMIX-1B may be nonuniform as shown in Figure 2. In addition, the neighboring cells and cell faces are defined according to the convention given in Table 3 and shown in Figure 3. The thermal

Table 1. Variable Values for the Conservation Equations in Cartesian Coordinates

Equation	Variable (ϕ)	Direction	Diffusion Coefficient (Γ_ϕ)	Source Term (S_ϕ)
Continuity	1	Scalar	0	0
Momentum (i)	u	x direction	μ	$\rho g_x + V_x - R_x - \left(\frac{\partial p}{\partial x}\right)$
(ii)	v	y direction	μ	$\rho g_y + V_y - R_y - \left(\frac{\partial p}{\partial y}\right)$
(iii)	w	z direction	μ	$\rho g_z + V_z - R_z - \left(\frac{\partial p}{\partial z}\right)$
Energy	h	Scalar	k	$\frac{dp}{dt} + \dot{Q}_{rb} + \dot{Q} + \Phi$

V_x, V_y, V_z : Balance of the viscous diffusion terms

R_x, R_y, R_z : Distributed resistances due to solid structures in a momentum control volume

$\dot{Q}_{rb}$: Rate of heat liberated from solid structures per unit fluid volume

$\dot{Q}$: Rate of internal heat generation per unit fluid volume

Φ : Dissipation function

Table 2. Variable Values for the Conservation Equations in Cylindrical Coordinates

Equation	Variable (ϕ)	Direction	Diffusion Coefficient (Γ_ϕ)	Source Term (S_ϕ)
Continuity	1	Scalar	0	0
Momentum (i)	v_r	r direction	μ	$\rho \frac{v_\theta^2}{r}^{*} + \rho g_r + V_r - R_r - \frac{1}{r} \frac{\partial}{\partial r} (rp)$
(ii)	v_θ	θ direction	μ	$- \frac{\rho v_r v_\theta}{r}^{**} + \rho g_\theta + V_\theta - R_\theta - \frac{1}{r} \frac{\partial}{\partial \theta} (p)$
(iii)	v_z	z direction	μ	$\rho g_z + V_z - R_z - \frac{\partial}{\partial z} (p)$
Energy	h	Scalar	k	$\frac{dp}{dt} + \dot{Q}_{rb} + \dot{Q} + \Phi$

*	:	Centrifugal force term
**	:	Coriolis force term
V_r, V_θ, V_z	:	Balance of the viscous diffusion terms
R_r, R_θ, R_z	:	Distributed resistance due to solid structures in a momentum control volume
$\dot{Q}_{rb}$	:	Rate of heat liberated from solid structures per unit fluid volume
$\dot{Q}$	:	Rate of internal heat generation per unit fluid volume
Φ	:	Dissipation function

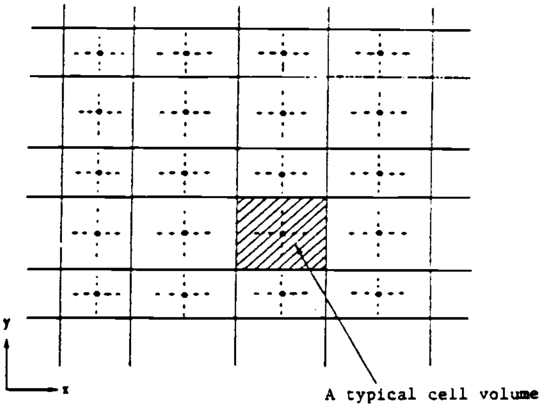

Fig. 2. Construction of Cell Volumes

Table 3. Convention Used in COMMIX-1B to Define Neighboring-Cell Control Volumes

Subscript	Cell Centers			Cell-Face Centers		
0	i,	j,	k			
1	i-1,	j,	k	i-1/2,	j,	k
2	i+1,	j,	k	i+1/2,	j,	k
3	i,	j-1,	k	i,	j-1/2,	k
4	i,	j+1,	k	i,	j+1/2,	k
5	i,	j,	k-1	i,	j,	k-1/2
6	i,	j,	k+1	i,	j,	k+1/2

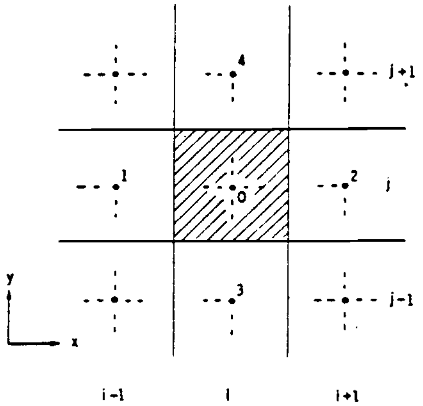

Fig. 3. Cell Volume around Point 0 in i,j,k Notation

hydraulic properties of the fluid are calculated at the cell grid points and are assumed to remain constant over the entire cell.

Since the values of fluid and flow properties are calculated at the grid points, the following dilema arises. Because the convective terms in the conservation equations require velocities on the cell faces, these surface velocities must be obtained as averages of grid point velocities which are less accurate. Furthermore, since each of the momentum equations has a pressure derivative in the source term, a central difference approximation to this pressure term must span 3 grid points. As a result, the derivative is an average over two cell lengths and is less accurate. The way in which to reduce the need for velocity averaging as well as to tighten the pressure derivative is to stagger the velocity grids so that the velocities are actually calculated on the cell faces. Since this staggering places velocity grids between adjacent pressure grids, the required pressure derivative can be approximated using the values at adjacent grid points. As a consequence, the central difference pressure approximation spans only one cell length and is more accurate.

Figure 4 shows the locations of u and v by short arrows on a two dimensional grid; the three dimensional counterpart can be easily imagined with respect to a grid point, the u location is displaced only in the x direction, the v location only in the y direction and so on. Since the velocity grids are staggered, the control volumes used for the conservation of momentum are also staggered. From now on these staggered control volumes will be referred to as momentum control volumes while the remainder will be referred to as main control

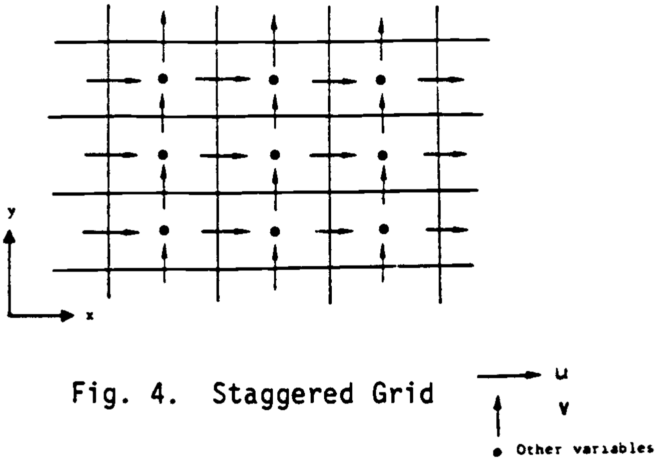

Fig. 4. Staggered Grid

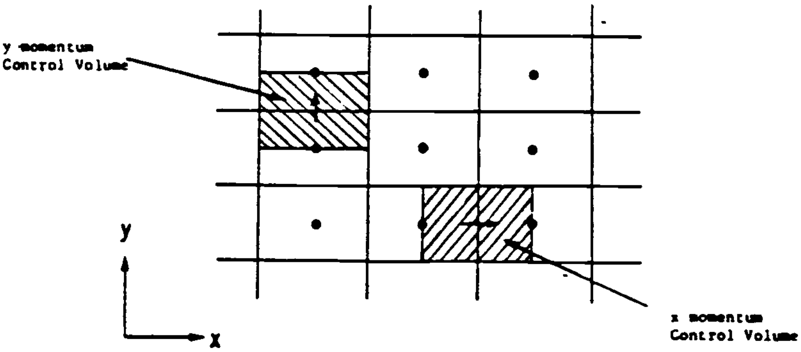

Fig. 5. Momentum Control Volumes

Table 4. Convention Used in COMMIX-1B to Define Neighboring Control Volumes for the i Direction Momentum Equation

Subscript	Momentum Control Volume Centers	Momentum Control Volume Face Centers
0	i+1/2, j, k	
1	i-1/2, j, k	i, j, k
2	i+3/2, j, k	i+1, j, k
3	i+1/2, j-1, k	i+1/2, j-1/2, k
4	i+1/2, j+1, k	i+1/2, j+1/2, k
5	i+1/2, j, k-1	i+1/2, j, k-1/2
6	i+1/2, j, k+1	i+1/2, j, k+1/2

volumes. As shown in Figure 5, the control volume for momentum will be staggered in the direction of the momentum such that the faces normal to that direction pass through the grid points. The convention used in COMMIX-1B to define neighboring momentum control volumes is shown for the i direction momentum equation in Table 4.

3.4 Finite Difference Approximations

Finite difference approximations to (3.2.7) are derived by integrating (3.2.7) over a control volume. The integration of (3.2.7) will be shown term by term in Cartesian coordinates.

Representation of the term $\frac{\partial}{\partial t}(\gamma_V \rho \phi)$ is obtained by assuming that the values ρ_O and ϕ_O prevail over the entire control volume. Integration of the unsteady term over the control volume then gives

$$(3.4.1) \qquad \int \frac{\partial}{\partial t}(\gamma_V \rho \phi) d_x d_y d_z = \frac{((\rho\phi_O)-(\rho\phi_O)^n)V_O,}{\Delta t}$$

where $V_O = \gamma_V \Delta_x \Delta_y \Delta_z$ is the volume of the fluid; the superscript n refers to old time-step values, and the superscript n+1 for the new time-step values is omitted for simplicity. For the x-momentum control volume the formulation is the same except that the volumes and densities used are averages of the values existing in the two main control volumes overlapped by the x momentum control volume. The volume of the fluid in the x-momentum control volume is given by

$$(3.4.2) \quad \bar{V}_O = 1/2\ (\gamma_{V_i} + \gamma_{V_{i+1}}) 1/2\ (\Delta x_i + \Delta x_{i+1})\ \Delta_y \Delta_z,$$

and the density is given by

(3.4.3) $$\bar{\rho} = \rho_{i+1/2} = \frac{\Delta x_i \rho_i + \Delta x_{i+1}\rho_{i+1}}{\Delta x_i + \Delta x_{i+1}} \quad .$$

In the material that follows a bar over a variable will be used to denote variables that pertain to momentum control volumes.

The integration of the convection terms over a main control volume gives

(3.4.4) $$\iint \left[\frac{\Delta(\gamma_x \rho u \phi)}{\Delta_x} + \frac{\Delta(\gamma_y \rho v \phi)}{\Delta_y} + \frac{\Delta(\gamma_z \rho w \phi)}{\Delta_z}\right] d_x d_y d_z$$

$$= F_2 \langle\phi\rangle_2^0 - F_1 \langle\phi\rangle_0^1 + F_4 \langle\phi\rangle_4^0 - F_3 \langle\phi\rangle_0^3 + F_6 \langle\phi\rangle_6^0 - F_5 \langle\phi\rangle_0^5$$

Here, F_k (=density x velocity x flow area) is the mass flux across surface k. For example,

(3.4.5) $$F_2 = F_{i+\frac{1}{2}} = \langle\rho\rangle_2^0 \, (\gamma_x u \Delta_y \Delta_z)_2 = \langle\rho\rangle_2^0 \, [uA_x]_2$$

$$= \langle\rho\rangle_{i+1}^{i} \, (uA_x)_{i+\frac{1}{2}}$$

is the mass flux across the east surface. In this expression

(3.4.6a) $\langle\rho\rangle_2^0 = \rho_0$ (if $u_2 \geq 0$), and

(3.4.6b) $\langle\rho\rangle_2^0 = \rho_2$ (if $u_2 < 0$).

In general the superscript location is to be used for positive velocity and the subscript location for negative velocity. In this way the value of ρ on the surface is identical to its value at the grid from which convection is assumed to occur. This means of assigning a property value is referred to as upwinding since the property value on the surface is assigned the value that exists at the grid directly upwind from it. Since the values of ϕ in (3.4.4) are similarily upwinded, the term, $F_2\langle\phi\rangle_2^0$, in (3.4.4) may be rewritten as

$$(3.4.7)\quad F_2\langle\phi\rangle_2^0 = |0,F_2|\phi_o - |0,-F_2|\phi_2,$$

where the quantity $|A,B|$ = max (A,B). Using this formulation (3.4.4) may be rewritten as

$$(3.4.8)\quad \int\left[\frac{\Delta(\gamma_x\rho u\phi)}{\Delta_x} + \frac{\Delta(\gamma_y\rho v\phi)}{\Delta y} + \frac{\Delta(\gamma_z\rho w\phi)}{\Delta_z}\right]d_x d_y d_z$$

$$= [|0, F_2| + |0, F_4| + |0, F_6| + |0, -F_1| + |0, -F_3| + |0,-F_5|]\phi_o$$

$$- [|0,-F_2|\phi_2 + |0,-F_4|\phi_4 + |0,-F_6|\phi_6 + |0,F_1|\phi_1 + |0, F_3|\phi_3$$

$$+ |0, F_5|\phi_5].$$

The convective fluxes for the main control volume are listed in Table 5. For the x-momentum control volume the formulation is the same except that the pecularities of the control volume force the expressions for the fluxes to be somewhat different. The convective fluxes for the x-momentum control volume are listed in Table 6.

The integration of the diffusion terms over a main control volume gives

$$(3.4.9)\quad \int\left[\frac{\Delta(\gamma_x\Gamma\phi\partial\phi/\partial_x)}{\Delta_x} + \frac{\Delta(\gamma_y\Gamma\phi\partial\phi/\partial_y)}{\Delta y} + \frac{\Delta(\gamma_z\Gamma\phi\partial\phi/\partial_z)}{\Delta_z}\right]d_x d_y d_z$$

$$= D_2(\phi_2-\phi_o) - D_1(\phi_o-\phi_1) + D_4(\phi_4-\phi_o) -$$

$$D_3(\phi_o-\phi_3) + D_6(\phi_6-\phi_o) - D_5(\phi_o-\phi_5)$$

$$= D_1\phi_1 + D_2\phi_2 + D_3\phi_3 + D_4\phi_4 + D_5\phi_5 + D_6\phi_6$$

$$- (D_1+D_2+D_3+D_4+D_5+D_6)\phi_o.$$

Table 5. Convective Fluxes for Main Control Volume

F_1	:	$(A_x u)_{i-1/2}$	$\langle\rho\rangle_0^1$
F_2	:	$(A_x u)_{i+1/2}$	$\langle\rho\rangle_2^0$
F_3	:	$(A_y v)_{j-1/2}$	$\langle\rho\rangle_0^3$
F_4	:	$(A_y v)_{j+1/2}$	$\langle\rho\rangle_4^0$
F_5	:	$(A_z w)_{k-1/2}$	$\langle\rho\rangle_0^5$
F_6	:	$(A_z w)_{k+1/2}$	$\langle\rho\rangle_6^0$

Table 6. Convective Fluxes for x Momentum Control Volume

$\bar{F}_1$	:	$\rho_0 \frac{1}{2}\left[(uA_x)_{i-1/2} + (uA_x)_{i+1/2}\right]$
$\bar{F}_2$	:	$\rho_2 \frac{1}{2}\left[(uA_x)_{i+1/2} + (uA_x)_{i+3/2}\right]$
$\bar{F}_3$	:	$\frac{1}{2}\left[\langle\rho\rangle_0^3 (vA_y)_{i,j-1/2} + \langle\rho\rangle_2^{23} (vA_y)_{i+1,j-1/2}\right]$
$\bar{F}_4$	:	$\frac{1}{2}\left[\langle\rho\rangle_4^0 (vA_y)_{i,j+1/2} + \langle\rho\rangle_{24}^2 (vA_y)_{i+1,j+1/2}\right]$
$\bar{F}_5$	:	$\frac{1}{2}\left[\langle\rho\rangle_0^5 (wA_z)_{i,k-1/2} + \langle\rho\rangle_2^{25} (wA_z)_{i+1,k-1/2}\right]$
$\bar{F}_6$	:	$\frac{1}{2}\left[\langle\rho\rangle_6^0 (wA_z)_{i,k+1/2} + \langle\rho\rangle_{26}^2 (wA_z)_{i+1,k+1/2}\right]$

In this equation D is the average diffusion strength across a surface of the control volume. Its value is obtained by assuming that a uniform value of diffusivity $\Gamma\phi$ prevails over each main control volume and using harmonic interpolation to obtain the average at the surface. For example,

$$(3.4.10)\quad D_2 = (A_x)_{i+\frac{1}{2}}\left[(\frac{\Delta x_0}{2\Gamma_0})+(\frac{\Delta x_2}{2\Gamma_2})\right]^{-1}$$

the expressions for the diffusion strengths are listed in Table 7 for the main control volume. For the x-momentum control volume the formulation is the same except that the pecularities of the control volume force the diffusion strengths to be averaged differently. The diffusion strengths for the x-momentum control volume are listed in Table 8. (In these expressions a quantity listed as Γ_{2k} means the value of Γ existing in the (i+1)st cell with respect to cell k about cell 0.)

The finite difference representation of the source term S is expressed as

$$(3.4.11)\quad S = S_{c\phi} + S_{p\phi}\phi_0,$$

where $Sc\phi$, $Sp\phi$, and ϕ_0 are assumed to prevail over the main control volume surrounding point 0. This linearization of the source is very practical in a finite difference formulation. For example, the gravity term in the x-momentum equation represents a constant source term while the frictional drag term depends upon velocity. In (3.4.11) the coefficient $Sp\phi$ must always be less than or equal to zero; otherwise instability, divergence or physically unrealistic solutions will result. The integration of the source term over the control volume gives.

Table 7. Diffusion Strengths for Main Control Volume

D_1	: $(A_x)_{i-1/2} \left[\left(\frac{\Delta x}{2\Gamma}\right)_0 + \left(\frac{\Delta x}{2\Gamma}\right)_1 \right]^{-1}$
D_2	: $(A_x)_{i+1/2} \left[\left(\frac{\Delta x}{2\Gamma}\right)_0 + \left(\frac{\Delta x}{2\Gamma}\right)_2 \right]^{-1}$
D_3	: $(A_y)_{j-1/2} \left[\left(\frac{\Delta y}{2\Gamma}\right)_0 + \left(\frac{\Delta y}{2\Gamma}\right)_3 \right]^{-1}$
D_4	: $(A_y)_{j+1/2} \left[\left(\frac{\Delta y}{2\Gamma}\right)_0 + \left(\frac{\Delta y}{2\Gamma}\right)_4 \right]^{-1}$
D_5	: $(A_z)_{k-1/2} \left[\left(\frac{\Delta z}{2\Gamma}\right)_0 + \left(\frac{\Delta z}{2\Gamma}\right)_5 \right]^{-1}$
D_6	: $(A_z)_{k+1/2} \left[\left(\frac{\Delta z}{2\Gamma}\right)_0 + \left(\frac{\Delta z}{2\Gamma}\right)_6 \right]^{-1}$

Table 8. Diffusion Strengths for x Momentum Control Volume

$\bar{D}_1$	: $\frac{1}{2}\left[(A_x)_{i-1/2} + (A_x)_{i+1/2}\right]\left(\frac{\Gamma}{\Delta x}\right)_0$
$\bar{D}_2$	: $\frac{1}{2}\left[(A_x)_{i+1/2} + (A_x)_{i+3/2}\right]\left(\frac{\Gamma}{\Delta x}\right)_2$
$\bar{D}_3$	: $\frac{1}{2}\left[(A_y)_{i,j-1/2} + (A_y)_{i+1,j-1/2}\right]\left[\frac{\Delta y_{j-1}}{(\Gamma_3 + \Gamma_{23})} + \frac{\Delta y_j}{(\Gamma_0 + \Gamma_2)}\right]^{-1}$
$\bar{D}_4$	: $\frac{1}{2}\left[(A_y)_{i,j+1/2} + (A_y)_{i+1,j+1/2}\right]\left[\frac{\Delta y_{j+1}}{(\Gamma_4 + \Gamma_{24})} + \frac{\Delta y_j}{(\Gamma_0 + \Gamma_2)}\right]^{-1}$
$\bar{D}_5$	: $\frac{1}{2}\left[(A_z)_{i,k-1/2} + (A_z)_{i+1,k-1/2}\right]\left[\frac{\Delta z_{k-1}}{(\Gamma_5 + \Gamma_{25})} + \frac{\Delta z_k}{(\Gamma_0 + \Gamma_2)}\right]^{-1}$
$\bar{D}_6$	: $\frac{1}{2}\left[(A_z)_{i,k+1/2} + (A_z)_{i+1,k+1/2}\right]\left[\frac{\Delta z_{k+1}}{(\Gamma_6 + \Gamma_{26})} + \frac{\Delta z_k}{(\Gamma_0 + \Gamma_2)}\right]^{-1}$

$$(3.4.12) \qquad \int_{\text{c.v.}} S\phi d_x d_y d_z = Sc\phi V_o + Sp\phi V_o \phi_o$$

for the main control volume. For the x-momentum control volume V_o is simply replaced by $\bar{V}_o$.

Having looked at each term of the general equation separately, it is now possible to assemble all of the terms of (3.4.1), (3.4.8), (3.4.9), and (3.4.12) for the main control volume to obtain the general finite difference equation.

$$(3.4.13) \int (\text{unsteady})+(\text{Convection})-(\text{Diffusion})-(\text{Source})\ d_x d_y d_z$$

$$= \underset{\text{(unsteady)}}{\frac{((\rho\phi)_o-(\rho\phi\ )_o^n)V_o}{\Delta t}} + \underset{\text{(convection)}}{\{|0,-F_1,|+|0,\ F_2|+\ldots\}\ \phi_o}$$

$$- \underset{\text{(Convection)}}{\{|0,F_1|\phi_1 + |0,-F_2|\phi_2 + \ \ldots\}} \ + \ \underset{\text{(Diffusion)}}{(D_1+D_2+\ldots)\phi_o}$$

$$- \underset{\text{(Diffusion)}}{[D_1\phi_1+D_2\phi_2 + \ \ldots]} \ - \ \underset{\text{(Source)}}{S_{c\phi}V_o - S_{p\phi}\phi_o V_o} = 0$$

Since the values of $\phi_o \ldots \phi_6$ may change over a time step, it makes sense to use a weighted average of new and old time values of ϕ in the convection, diffusion, and source terms. To accomplish this averaging the values of ϕ_i used in these terms is replaced by

$$(3.4.14) \quad \tilde{\phi}_i = \alpha\phi_i + (1-\alpha)\phi_i{}^n,$$

where α is an implicitness parameter that ranges between 0 and 1. By making this replacement in (3.4.13) and rearranging the terms so that only terms containing ϕ_o are on the left hand side, the following equation is obtained.

$$(3.4.15)\quad \phi_o \{\frac{\rho_o V_o}{\Delta t} + \alpha[(|0,-F_1|+\ldots+|0,F_6|)$$

$$+ (D_1 +\ldots+ D_6) - S_{p\phi}V_o]\}$$

$$= \alpha[(|0,F_1|+ D_1)\phi_1 +\ldots+(|0,-F_6|+D_6)\phi_6]$$

$$+ (1-\alpha)[(|0,F_1|+D_1)\phi_1{}^n+\ldots+(|0,-F_6|+D_6)\phi_6{}^n]$$

$$- \phi_o{}^n (1-\alpha)[(|0,-F_1|+\ldots+|0,F_6|)+(D_1+\ldots+D_6)-S_{p\phi}V_o]$$

$$+ (\frac{\rho_o{}^n\phi_o{}^nV_o}{\Delta t} + S_{c\phi}V_o)$$

(3.4.15) may be re-expressed as

$$(3.4.16)\quad a_o^\phi\phi_o = \alpha\cdot(a_1^\phi\phi_1+a_2^\phi\phi_2+a_3^\phi\phi_3+a_4^\phi\phi_4+a_5^\phi\phi_5+a_6^\phi\phi_6$$

$$+ b_{1o}^\phi + b_{2o}^\phi + b_{3o}^\phi$$

For ease in reading, the coefficients of (3.4.16) are given in Table 9. The equation for the x-momentum control volume is similar except that the quantities ρ, V_o, F, and D are replaced by $\bar{\rho}_o$, $\bar{V}_o$, $\bar{F}$, and $\bar{D}$. (In table 9 the quantity listed as a_o^ϕ (2) arises from an alternate derivation and is irrelevant to the present discussion.)

Table 9. General Finite Difference Equation for the Main Control Volume and Its Coefficients

$$a_0^\phi \phi_0 = \alpha \left(a_1^\phi \phi_1 + \cdots\cdot + a_6^\phi \phi_6\right) + \left(b1_0^\phi + b2_0^\phi + b3_0^\phi\right)$$

a_1^ϕ	: $(\lvert 0,F_1\rvert + D_1)$	a_2^ϕ	: $(\lvert 0,-F_2\rvert + D_2)$
a_3^ϕ	: $(\lvert 0,F_3\rvert + D_3)$	a_4^ϕ	: $(\lvert 0,-F_4\rvert + D_4)$
a_5^ϕ	: $(\lvert 0,F_5\rvert + D_5)$	a_6^ϕ	: $(\lvert 0,-F_6\rvert + D_6)$

$b1_0^\phi$	: $(1-\alpha)\left(a_1^\phi\phi_1^n + a_2^\phi\phi_2^n + a_3^\phi\phi_3^n + a_4^\phi\phi_4^n + a_5^\phi\phi_5^n + a_6^\phi\phi_6^n\right)$
$b2_0^\phi$	: $-(1-\alpha)\left[\left(\lvert 0,-F_1\rvert + \cdots\cdot + \lvert 0,F_6\rvert\right) + \left(D_1 + \cdots\cdot + D_6\right) - S_{p\phi}V_0\right]\phi_0^n$
$b3_0^\phi$	: $\left(\frac{\rho^n\phi^n}{\Delta t} + S_{c\phi}\right)_0 V_o$
$a_0^\phi(1)$ (1st form)	: $\frac{\rho_0 V_0}{\Delta t} + \alpha\left[\left(\lvert 0,-F_1\rvert + \cdots\cdot + \lvert 0,F_6\rvert + \left(D_1 + D_2 \cdots\cdot + D_6\right) - S_{p\phi} V_0\right]\right.$
$a_0^\phi(2)$ (2nd form)	: $\alpha\left(a_1^\phi + a_2^\phi + \cdots\cdot\ a_6^\phi\right) + \left(\frac{\rho^n}{\Delta t} - S_{p\phi}\right) V_0$ $+ (1-\alpha)\left(F_1 - F_2 + F_3 - F_4 + F_5 - F_6\right)$

3.5 Pressure Equation

As seen previously a pressure derivative term appears in the source term of the momentum equation. However, since the pressure is unknown, it must be determined from another equation. What is done in COMMIX-1B is to relate the velocity to pressure in such a way as to remove the pressure term from the momentum equation. Next, the pressure velocity relationships are used to replace the velocities in the continuity equation to yield a pressure equation that is constrained to satisfy continuity.

As mentioned previously the x-momentum equation may be written in the form

$$(3.5.1) \quad a_0^\phi \phi_0 = (a_1^\phi \phi_1 + \ldots . a_6^\phi \phi_6) + b_1^\phi o + b_2^\phi o + b_3^\phi o,$$

and from table 9 the term $b_3^\phi o$ contains the term $S_{c\phi} V_0$. Since a pressure derivative term appears in $S_{c\phi}$, (3.5.1) may be rewritten as

$$(3.5.2) \quad a_0^\phi \phi_0 = (a_1^\phi \phi_1 + \ldots + a_6 \phi_6^\phi) + b_1^\phi o + b_2^\phi o + b_3^\phi o' - \frac{\Delta P \bar{V} o}{\Delta x}$$

where $b_3^\phi o'$ is used to indicate that the pressure term has been removed from $b_3^\phi o$. To remove the pressure term from the equation, ϕ_0 is first written as the sum of pressure independent and pressure dependent terms

$$(3.5.3) \quad \phi_0 = \hat{\phi} - d^\phi \Delta P$$

Next (3.5.3) is substituted into (3.5.2) to yield

$$(3.5.4) \quad a_0^\phi \, (\hat{\phi} - d^\phi \Delta P) = (a_1^\phi \phi_1 + \ldots + a_6^\phi \phi_6) + b_1^\phi o + b_2^\phi o + b_3^\phi o' - \frac{\Delta P \bar{V}_0}{\Delta x}$$

In order for the pressure to vanish from the equation, it is essential that the coefficient of proportionality be given by

$$(3.5.5)\quad d^{\phi} = \frac{\bar{V}_o}{a_0^{\phi}\Delta x} = \frac{(1/2)(\gamma_{vi}+\gamma_{vi+1})(1/2)(\Delta_{xi}+\Delta_{xi+1})\Delta_y\Delta_z}{a_0^{\phi}(1/2)(\Delta_{xi}+\Delta_{xi+1})}$$

$$= (1/2)(\gamma_{vi}+\gamma_{vi+1})\Delta y\Delta z/a_0^{\phi}$$

Having removed the influence of pressure from the momentum equation, the momentum equation is used to solve for the pressure independent contributions to the new time velocities as

$$(3.5.6)\quad \hat{\phi} = ((a_1^{\phi}\phi_1+\ldots+a_6^{\phi}\phi_6) + b_1^{\phi}o + b_2^{\phi}o + b_3^{\phi}o')/a_0^{\phi}$$

Continuity is next used to constrain the pressure field. The continuity equation for a cell about point 0 is derived from (3.4.15) by substituting $\phi = 1$ diffusion strength $D = 0$, and source strength $S = 0$.

$$(3.5.7)\quad V_0(\partial\rho/\partial t) - (A_xu)_{i-\frac{1}{2}}\langle\rho\rangle_0^1 + (A_xu)_{i+\frac{1}{2}}\langle\rho\rangle_2^0$$

$$- (A_yv)_{j-\frac{1}{2}}\langle\rho\rangle_0^3 + (A_yv)_{j+\frac{1}{2}}\langle\rho\rangle_4^0$$

$$- (A_zw)_{k-\frac{1}{2}}\langle\rho\rangle_0^5 + (A_zw)_{k+\frac{1}{2}}\langle\rho\rangle_6^0 = 0$$

To obtain a pressure equation, the following relations are substituted into (3.5.7).

$$(3.5.8)\quad \begin{aligned} u_2 &= \hat{u}_2 - d_2^u\,(P_2-P_o); \\ u_1 &= \hat{u}_1 - d_1^u\,(P_o-P_1); \\ v_4 &= \hat{v}_4 - d_4^v\,(P_4-P_o); \\ v_3 &= \hat{v}_3 - d_3^v\,(P_o-P_3); \\ w_6 &= \hat{w}_6 - d_6^w\,(P_6-P_o), \end{aligned}$$

and

$$w_5 = \hat{w}_5 - d_5^w\,(P_o-P_5).$$

By making these substitutions in (3.5.7) the following equation is obtained

$$(3.5.9)\quad a_o^p P_o \;-\; \sum_{l=1}^{6} a_l^p\, P_l - b_o^p = 0$$

This is the pressure equation that the conjugate gradient method is used to solve. The coefficients of (3.5.9) are listed in Table 10. After solving for the new time pressure field, the total velocity fields are calculated from (3.5.3). From this point the code moves on to solve the energy equation.

3.6 Mass Rebalancing

The mass rebalancing scheme is a means of accellerating the convergence of the pressure equations by first making large scale corrections to the pressure field so that the linear equation solvers are required only for fine pressure adjustments. To make these large scale adjustments many computational cells are first grouped into N regions as shown in figure 6. To obtain a pressure correction for each region, the pressure equations of all cells within the region are added under two constraints. First, the pressure corrections

Table 10. Coefficients of Pressure Equation

a_1^P	:	$\left(\dfrac{A_x}{a_0^u}\right)_{i-1/2} \langle\rho\rangle_0^1 \frac{1}{2}\left(\gamma_{v0} + \gamma_{v1}\right) \Delta y \; \Delta z$
a_2^P	:	$\left(\dfrac{A_x}{a_0^u}\right)_{i+1/2} \langle\rho\rangle_2^0 \frac{1}{2}\left(\gamma_{v0} + \gamma_{v2}\right) \Delta y \; \Delta z$
a_3^P	:	$\left(\dfrac{A_y}{a_0^v}\right)_{j-1/2} \langle\rho\rangle_0^3 \frac{1}{2}\left(\gamma_{v0} + \gamma_{v3}\right) \Delta x \; \Delta z$
a_4^P	:	$\left(\dfrac{A_y}{a_0^v}\right)_{j+1/2} \langle\rho\rangle_4^0 \frac{1}{2}\left(\gamma_{v0} + \gamma_{v4}\right) \Delta x \; \Delta z$
a_5^P	:	$\left(\dfrac{A_z}{a_0^w}\right)_{k-1/2} \langle\rho\rangle_0^5 \frac{1}{2}\left(\gamma_{v0} + \gamma_{v5}\right) \Delta x \; \Delta y$
a_6^P	:	$\left(\dfrac{A_z}{a_0^w}\right)_{k+1/2} \langle\rho\rangle_6^0 \frac{1}{2}\left(\gamma_{v0} + \gamma_{v6}\right) \Delta x \; \Delta y$
a_0^P	:	$a_1^P + a_2^P + a_3^P + a_4^P + a_5^P + a_6^P$
b_0^P	:	$- V_0 \left.\dfrac{\partial\rho}{\partial t}\right\|_0 + \left(A_x\hat{u}\right)_{i-1/2} \langle\rho\rangle_0^1 - \left(A_x\hat{u}\right)_{i+1/2} \langle\rho\rangle_2^0$ $+ \left(A_y\hat{v}\right)_{j-1/2} \langle\rho\rangle_0^3 - \left(A_y\hat{v}\right)_{j+1/2} \langle\rho\rangle_4^0$ $+ \left(A_z\hat{w}\right)_{k-1/2} \langle\rho\rangle_0^5 - \left(A_z\hat{w}\right)_{k+1/2} \langle\rho\rangle_6^0$

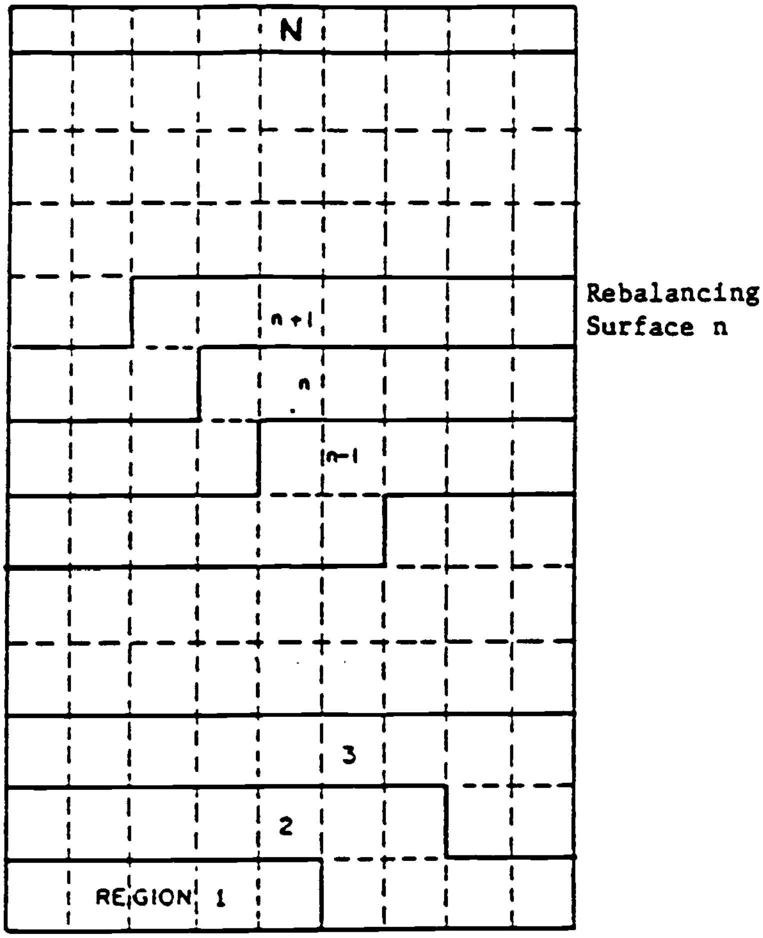

Fig. 6. Coarse Mesh Showing Rebalancing Regions

for all cells within a region are the same, and, second, the pressure correction for each region is determined such that the sum of mass residuals over all cells in that region equals zero.

The regions shown in figure 6 are chosen so that any region n has neighboring cells contained only in the neighboring regions (n-1) and (n+1). Mass leaving the region n and entering the region (n+1) does so through rebalancing surface n. Mass leaving the last region N goes into the remaining cells where no rebalancing performed. (It should be noted that in this discussion N refers to the number of regions not the number of cells. Consequently, N should not be confused with the N used in Chapter 2). Furthermore, region 1 has neighboring cells only in region 2.

To derive the pressure correction let P* be the pressure distribution which does not satisfy the continuity equation. The pressure equation for a cell m is

$$(3.6.1)\quad a_{mo}P^*_m - \sum_{l=1}^{6} a_{ml}P^*_l - b_{mo} = \delta^*_m,$$

where δ^*_m is the mass residual, and a_m and b_m are the pressure equation coefficients of (3.5.9). Next, the pressure equations for all cells m in the region n are added to yield:

$$(3.6.2)\quad \sum_{m\varepsilon n} [a_{mo}P^*_m - \sum_{l=1}^{6} (a_{ml}P^*_l) - b_{mo}] = \sum_{m\varepsilon n} \delta^*_m$$

If ΔP_1, ΔP_2, ...ΔP_N are the pressure corrections for the rebalancing regions 1 to N, the new pressure distribution can be written as

$$(3.6.3)\quad P_m = P^*_m + \Delta P_n;\ m\varepsilon n$$

If this new pressure field replaces the old one in (3.6.2), the following equation results for each region.

$$(3.6.4)\quad \sum_{m\varepsilon n} (a_{mo}P_m - \sum_{l=1}^{6} a_{ml}\, P_l - b_{mo}) = \sum_{m\varepsilon n} \delta_m$$

Since the second constraint requires that

$$(3.6.5)\quad \sum_{m\varepsilon n} \delta_m = 0,$$

expanding (3.6.4) by (3.6.3) yields,

$$(3.6.6)\quad \sum_{m\varepsilon n}(a_{mo}P_m^* - \sum_{l=1}^{6} a_{ml}P_l^* - b_{mo}) +$$

$$\Delta P_n\, [\sum_{m\varepsilon n} (a_{mo} - \sum_{l\varepsilon n}^{6} a_{ml})] - \Delta P_{n-1}[\sum_{m\varepsilon n} (\sum_{l\varepsilon n-1} a_{ml})$$

$$- \Delta P_{n+1}\, [\sum_{m\varepsilon n}(\sum_{l\varepsilon n+1} a_{ml})] = 0,\ (n = 1,....N).$$

The first term is simply $\sum_{m\varepsilon n} \delta_m^*$ from (3.6.2).

Furthermore, since $a_{mo} = \sum_{l=1}^{6} a_{ml}$, and since $a_{ml} = a_{lm}$, many cancellations will occur when the second term is evaluated with the result that

$$(3.6.7)\quad \sum_{m\varepsilon n}(a_{mo} - \sum_{l\varepsilon n}^{6} a_{ml}) = \sum_{m\varepsilon n} (\sum_{l\varepsilon n-1} a_{ml}) + \sum_{m\varepsilon n} (\sum_{l\varepsilon n+1} a_{ml}).$$

As a consequence, (3.6.6) may be rewritten as

$$(3.6.8)\quad A_o{}^n\Delta P_n - A_1{}^n\Delta P_{n-1} - A_2{}^n\Delta P_{n+1} - B^n = 0,$$

where

$$(3.6.9a)\quad A_1^n = \sum_{m\varepsilon n} (\sum_{l\varepsilon(n-1)} a_{ml});$$

(3.6.9b) $A_2^n = \sum_{m\varepsilon n} (\sum_{l\varepsilon(n+1)} a_{ml});$

(3.6.9c) $A_0^n = A_1^n + A_2^n$, and

(3.6.9d) $B^n = \sum_{m\varepsilon n} \delta^*_m.$

For the first rebalancing region, the coefficients $A_1^1 = 0$, and $A_0^1 = A_2^1$. Since no pressure is desired in the neighboring cells of the last rebalancing region, $\Delta P_{N+1} = 0$, and A_2^N can be evaluated from (3.9.6b).

(3.6.8) yields N equations for the N rebalancing regions. Since these equations may be solved using a tridiagonal matrix algorithm, the large scale pressure corrections are easily obtained. After obtaining these corrections, the linear equation solvers are used to make fine pressure adjustments so that mass is conserved on both a cell by cell and a regional basis.

3.7 Solution Procedures

As mentioned previously COMMIX-1B solves the conservation equations of momentum, mass, and energy in each outer iteration. Furthermore, it was shown in section 3.4 that the conservation equations allow values of the dependent variables to be represented as a weighted average of the old and new timestep values using an implicitness parameter α that ranges between 0 and 1. Although any value of α between 0 and 1 should work, the solution scheme has only been fully tested for α values of 0 and 1 at the present time. It is therefore recommended that only α values of 0 and 1 be used in the solution scheme. The solution schemes, resulting from these values of α are shown in tables 11 and 12 respectively.

Table 11. Algorithm of the Semi-Implicit Solution Scheme (α=0)

1. Calculate momentum coefficients using old-time step values of u, v, and w:

   $$\hat{\phi}, d^{\phi} \; ; (\phi = u, v, w) .$$

2. Calculate pressure equation coefficients using $\hat{\phi}$, d^{ϕ}:

   $$a_0^P, a_\ell^P, b_0^P .$$

3. Solve pressure equation for new-time pressure P^{n+1}:

   $$a_0^P P_0 - \sum a_\ell^P P_\ell - b_0^P = 0 .$$

4. Calculate new-time velocities using

   $$\phi = \hat{\phi} - d^{\phi} \Delta P \; ; (\phi = u, v, w)$$

   and new-time values of pressure.

5. Calculate energy equation coefficient using new-time values of velocities:

   $$a_0^h, a_\ell^h, b_0^h$$

6. Calculate new-time enthalpy h^{n+1}:

   $$h_0 = \sum a_\ell^h h_\ell^n + b_0^h / a_0^h .$$

Table 12. Fully Implicit Solution Sequence (α=1)

1. Calculate velocity-pressure relation coefficients from the previous iterate values of u, v, and w:

   $\hat{\phi}, d^{\phi}$; (ϕ = u, v, w).

2. Calculate pressure equation coefficients using $\hat{\phi}$, d^{ϕ}:

   a_0^P, a_ℓ^P, b_0^P .

3. Solve pressure equation for new-time, new-iterate pressure P:

   $a_0^P P_0 = \sum a_\ell^P P_\ell + b_0^P$

4. Calculate new-time, new iterate velocities u, v, w from velocity-pressure relations:

   $\phi = \hat{\phi} - d^{\phi} \Delta P$; (ϕ = u, v, w)

5. Calculate energy equation coefficients using new-time, new-iterate velocities:

   a_0^h, a_ℓ^h, b_0^h .

6. Solve energy equation for new-time, new-iterate enthalpy h:

   $a_0^h h_0 = \sum a_\ell^h h_\ell + b_0^h$.

7. Check for convergence of u, v, w, h; if not converged, return to Step 1.

CHAPTER 4

COMPARISONS BETWEEN THE S.O.R. AND CONJUGATE GRADIENT METHODS

4.1 Introduction

The comparisons between methods were made for six-steady state problems run with COMMIX-1B. In all of these cases the mass rebalancing technique described in Section 3.6 was used to accelerate the convergence of the pressure equation. In addition, for each outer iteration the maximum number of internal pressure iterations was limited only by the default value of 99. This choice was made for two reasons. First, an unknowledgable user would most probably rely on default values in the absence of any better information. Second, this choice allowed the pressure field to converge on most outer iterations with the result that the velocity fields on equivalent outer iterations were fairly similar. Since the velocity fields obtained affected the coefficients of the pressure matrix on the next outer iteration, allowing the pressure field to converge forced both methods to solve fairly similar systems of equations on each outer iteration. The over-relaxation factor was set to 1 for the first problem and 1.3 for the second. For the remaining problems this parameter was set to its default value of 1.5.

In comparing the performance of these methods three topics need discussion. In section 2 each of the problems run will be sufficiently described, and the total running times for each solution scheme will be given. In section 3 the significant differences between solution schemes will be discussed. These differences will be used in section

4 to explain differences in the ratios of running times between the problems described in section 2. In addition, several questions will be raised about other factors affecting the problem.

4.2 Problems

The first problem modeled was the flow of liquid sodium through a canned hexagonal 7 pin fuel assembly. Vertical and horizontal cross sections of this assembly are shown in figures 7 and 8, respectively. Fluid entered the assembly at the bottom of figure 7 where it was forced to have constant temperature and normal velocity values of 553°C and 2.15m/s. Fluid left the assembly at the top of figure 7. Here the temperature and normal velocity values were initialized to 553°C and 0m/s but allowed to vary to satisfy conservation requirements. In addition the pressure at this surface was assigned a constant value of $1.32X10^5$ Pa.

Because the fuel assembly is symmetric, only one quarter of the assembly was modeled as shown in figure 8. As a consequence, the uppermost and rightmost surfaces in this figure correspond to symmetric surfaces while the remaining surfaces correspond to the assembly canning. To model these surfaces free slip boundary conditions were used with the normal velocities on all four surfaces initialized to 0 m/s. No heat transfer was allowed across the surfaces of symmetry while heat transfer to the canning was allowed. In addition, the temperatures of these surfaces were initialized to 553°C but allowed to vary to satisfy conservation requirements.

Although the fuel rods and spacer grids are not shown in these figures, their influence on the fluid temperature and velocity was

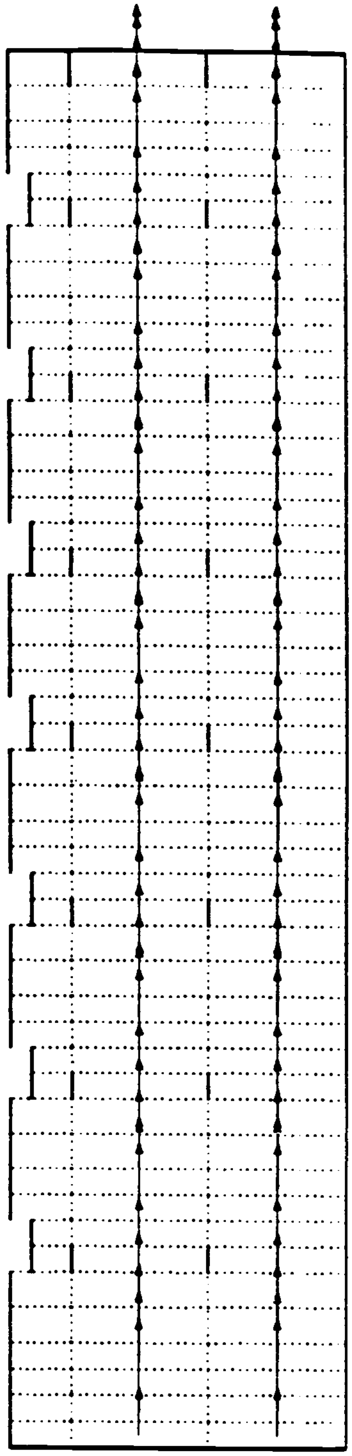

Figure 7. Vertical Cross Section of 7-Pin Fuel Assembly

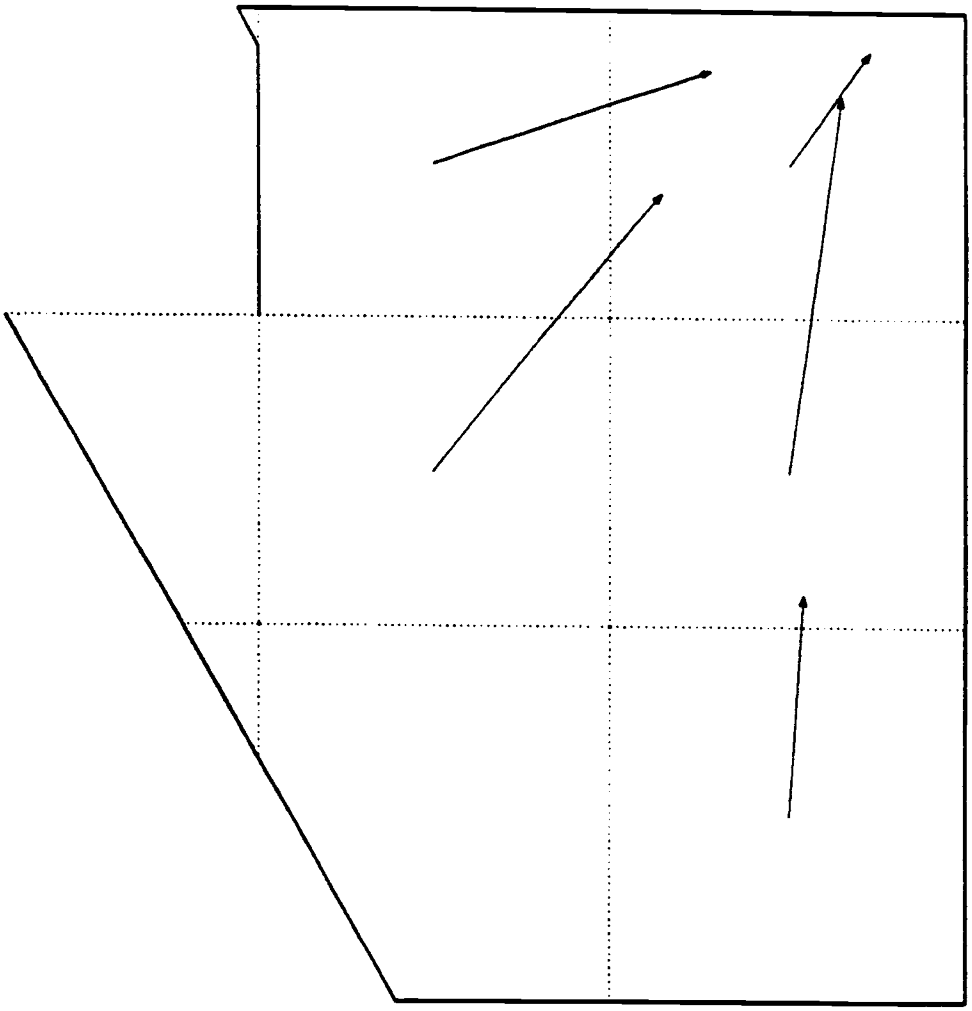

Figure 8. Horizontal Cross Section of 7-Pin Fuel Assembly

accounted for. The energy transport between these strucutres and the fluid was determined from heat conduction equations and convective heat transfer correlations. The influence on the fluid velocity was determined by taking both the direct resistance and frictional resistance of the structures into account. The direct resistance was determined by adjusting the volume porosities and directional surface porosities of each cell to make the flow pattern account for the shape of these structures. The frictional resistance was determined by the use of friction factor correlations. Because the rods affected fluid temperatures and velocities, this problem required the solution of all three conservation equations.

This problem converged in 52 outer iterations for both the conjugate gradient method and the S.O.R. method. The total time spent solving the pressure equation was 6.14s for the conjugate gradient method and 11.60s for S.O.R.

The second problem involved the flow of water in a scaled model of the Clinch River Breeder Reactor outlet plenum. A cross-section of the resulting flow pattern is shown in figure 9. In this figure fluid enters at two places in the bottom and leaves at the knob on the left hand surface. The first inlet occurs at the two large cells in the bottom which are shown with dotted lines. This inlet corresponds to the reactor coolant coming from the fuel and was assigned constant temperature and normal velocity values of 81.67°C and .904 m/s. The second inlet occurs in the cells adjacent to these and corresponds to the reactor coolant which was diverted to cool the breeding blanket. This inlet was assigned constant temperature and normal velocity boun-

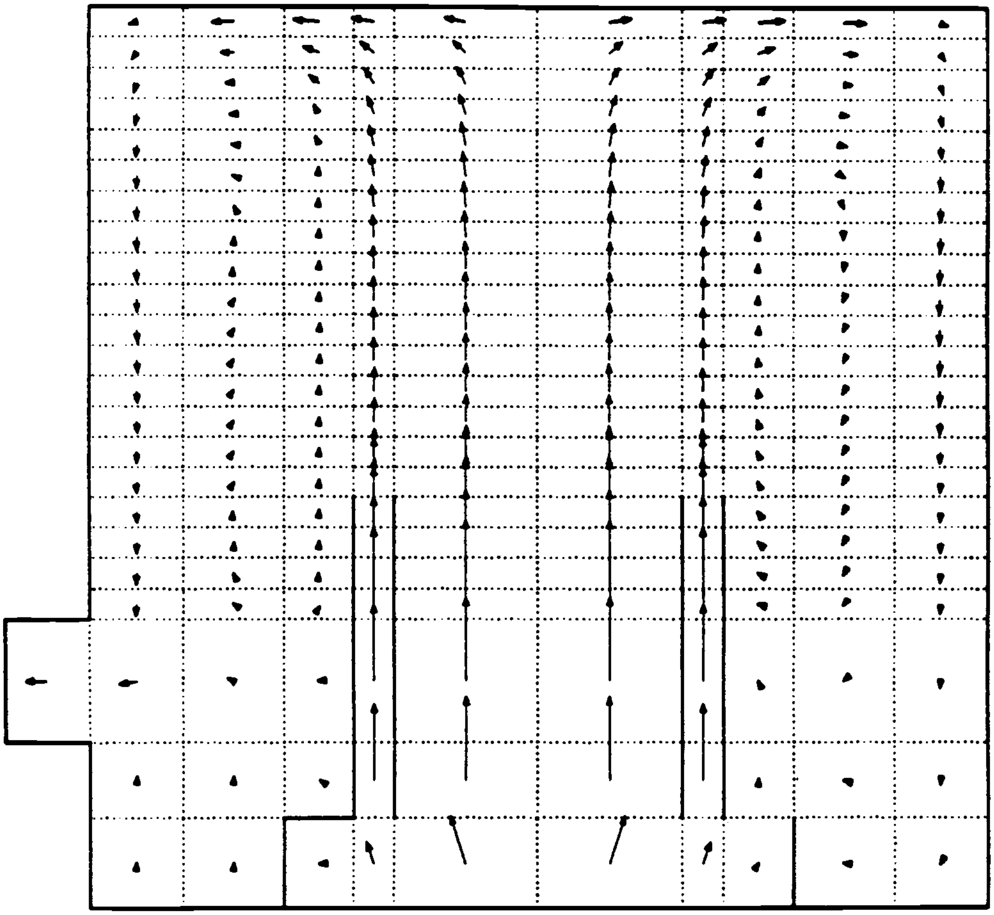

Figure 9. Vertical Cross Section of C.R.B.R. Outlet Plenum

dary conditions of 49.5°C and .179 m/s. The temperature difference between these two streams required the energy equation to be involved in the problem solution. The temperature and normal velocity at the exit were initialized to 81.67°C and 0 m/s but allowed to vary to satisfy conservation requirements. The pressure at this surface was assigned a constant value of 98,569.1 Pa.

The upper surface was assigned an initial temperature boundary condition of 81.67°C but allowed to vary. The centerline was assigned a free slip velocity boundary condition while the remaining surfaces were assigned a constant normal velocity value of 0 m/s.

For this problem convergence was obtained in 236 outer iterations using either solution scheme. The total time spent solving the pressure equation was 32.96s for the conjugate gradient method and 74.06s for S.O.R. Out of curiosity this problem was tried without mass rebalancng causing convergence to be obtained in 236 outer iterations with the conjugate gradient method and 211 iterations with S.O.R. The total time spent solving the pressure equation was 35.26s for the conjugate gradient method and 105.26s for S.O.R.

The third problem involved the flow of air through an atmospheric fluidized bed combustor. Vertical and horizontal cross-sections of the resulting flow pattern are shown in figure 10 and figure 11 respectively. In this problem air entered the combustor from 3 places. The first inlet was at the bottom surface which was assigned constant temperature and normal velocity boundary conditions of 871°C and 2.58 m/s. The second and third inlets were located in the rear most plane

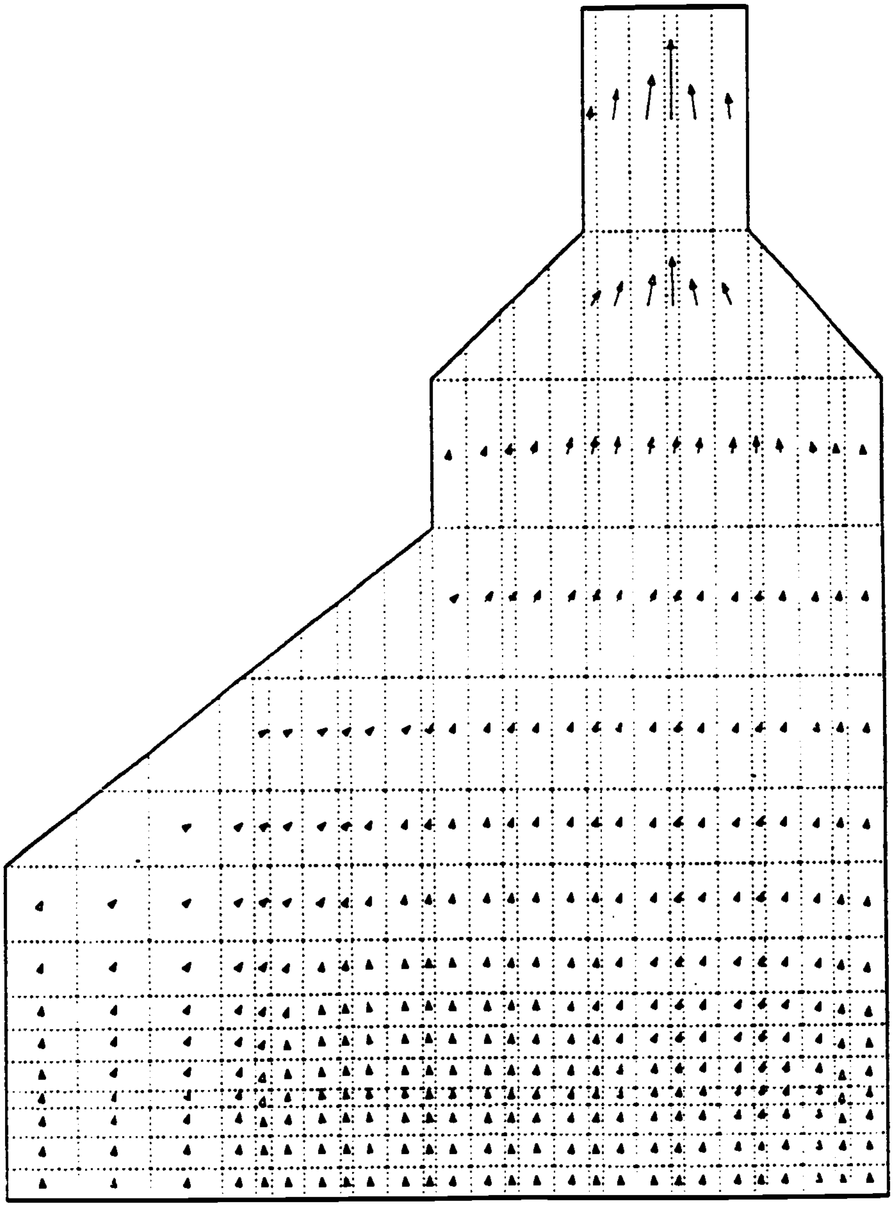

Figure 10. Vertical Cross Section of Atmospheric Fluidized Bed Combustor

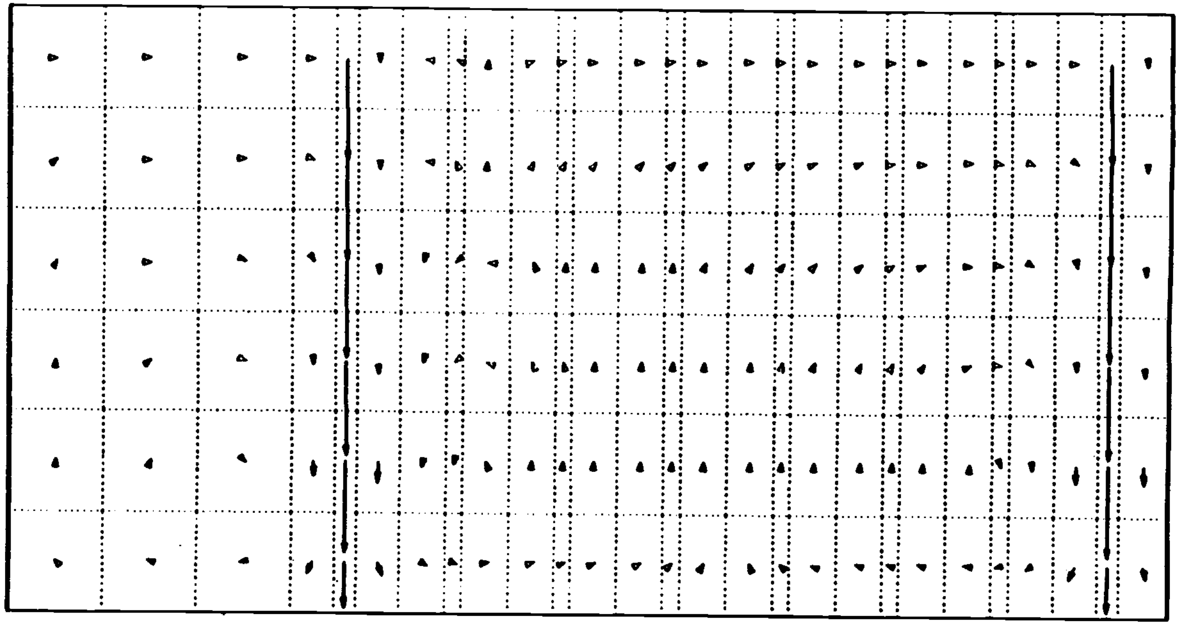

Figure 11. Horizontal Cross Section of Atmospheric Fluidized Bed Combustor

of figure 10 so that the flow from these inlets came straight out of the figure and perpendicular to the upward flow from the bottom. These two inlets represented ports which were used to inject low temperature air at high velocity into the combustor. The action of these ports is shown in the horizontal plane of figure 11. Both of these ports were assigned constant temperature and normal velocity boundary conditions of 288°C and 91.44 m/s. The temperature difference between these ports and the bottom inlet required the energy equation to be involved in the problem solution.

The air left the combustor at the uppermost surface in figure 10 which was assigned a constant pressure boundary condition of 1.0135x 10^5 Pa. Although the temperature and normal velocity of this surface were initialized to 871°C and 0 m/s, they were allowed to vary to satisfy conservation requirements. The normal velocities of all remaining surfaces were held constant at 0 m/s while their temperatures were initialized to 871°C but allowed to vary.

For this problem convergence was obtained in 148 outer iterations using either solution scheme. The total time spent solving the pressure equation was 253.88s for the conjugate gradient method and 458.35s for S.O.R.

The fourth problem involved the flow of water in the cold leg and downcomer of a PWR. The resulting flow pattern is shown in figure 12. In this figure the horizontal portion is the cold leg having the high pressure injector on the top right and the downcomer on the left. The two flow inlets for this problem were at the far right of the cold leg and at the top of the high pressure injector. The

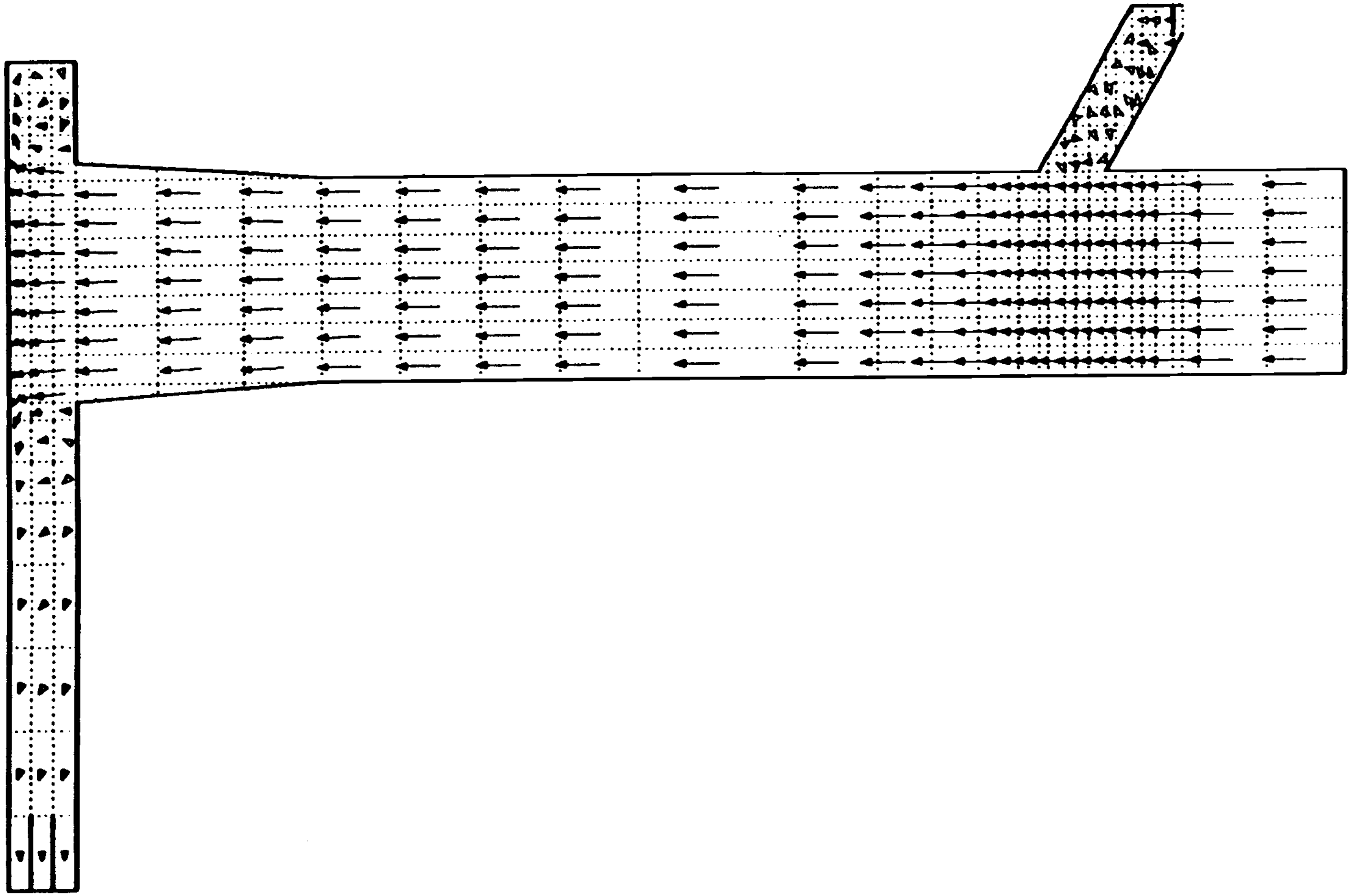

Figure 12. Cross Section of Cold Leg, Downcomer and High Pressure Injector

first inlet was assigned constant temperature and normal velocity values of 70°C and .0913 m/s. Since the second inlet was assigned constant values of 70°C and 0 m/s, it was effectively shut off, and the fluid in the injector was consequently stagnant. (The arrows shown in the injector mean nothing because the norms of these velocities were 4 orders of magnitude below the main velocity in the cold leg, resulting from the viscous diffusion of momentum.)

The fluid exit was located at the bottom of the downcomer. At this exit the temperature and normal velocity values were initialized to 70°C and 0 m/s but allowed to vary after that. The normal velocities of all other surfaces were held constant at 0 m/s. The temperatures of these surfaces as well as that of the internal fluid were initialized to 70°C. Since all temperatures were set at 70°C, no significant energy calculation needed to be performed.

For this problem convergence was obtained in 206 outer iterations with the conjugate gradient method and 208 iterations with S.O.R. The total time spent solving the pressure equation was 51.35s for the conjugate gradient method and 103.70s for S.O.R.

The fifth problem involved isothermal air flow through a pipe having half of its flow area blocked a fifth of the way to the top. The resulting flow pattern is shown in figure 13. Air entered at the bottom surface which was assigned constant temperature and normal velocity values of 25°C and 1 m/s. Air left at the top where these values were initialized to 25°C and 0 m/s and then allowed to vary. In addition the pressure at the top was held constant at 7000 Pa.

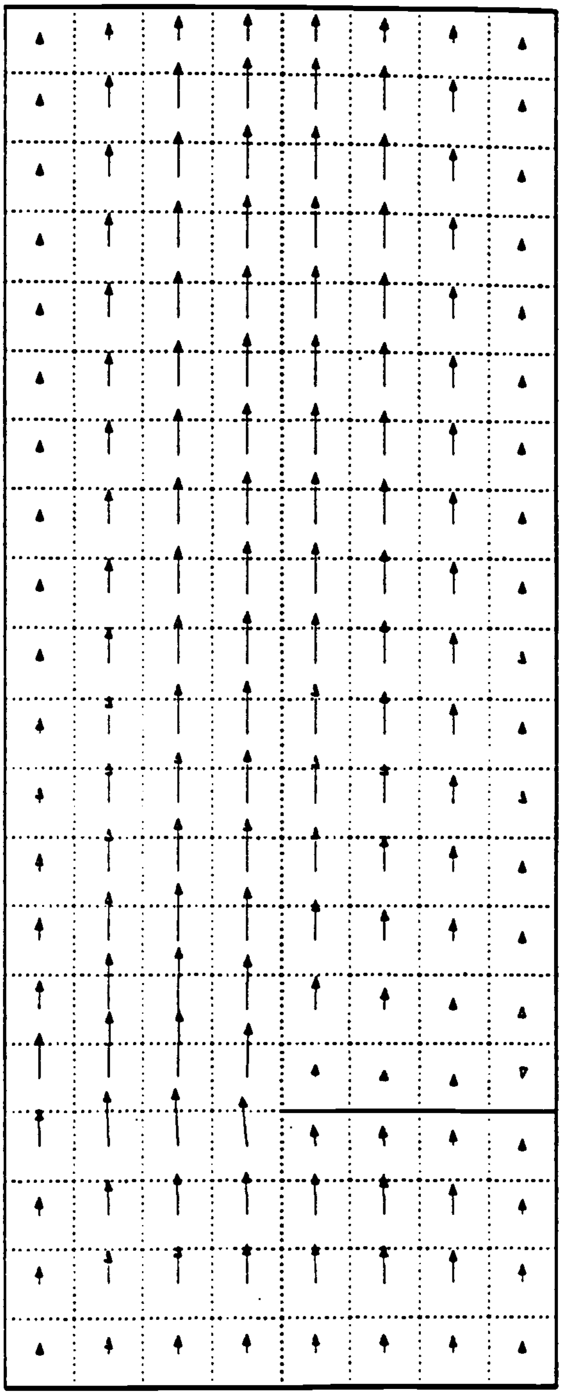

Figure 13. Cross Section of Isothermal Air Flow through a Pipe

The normal velocity at the outside surface of the pipe was held constant at 0 m/s and the temperature was initialized to 25°C. A free slip boundary condition was used at the center of the pipe, and the internal air temperature was initialized to 25°C. Because of the isothermal conditions at 25°C, no energy calculation was required.

This problem converged in 110 outer iterations using either solution scheme. The total time spent solving the pressure equation was 21.35s for the conjugate gradient method and 15.60s for S.O.R. Since the simplicity of the problem geometry made this problem easy to scale, it was decided to see how decreasing the mesh size in the r and z directions would influence the running times. The flow patterns for these cases are shown in figures 14 and 15. For the case shown in figure 14 the mesh size in these directions was reduced to 2/3 of that shown in figure 13. This problem converged in 156 outer iterations using the conjugate gradient method and 175 iterations using S.O.R. The total time spent solving the pressure equation was 71.67s for the conjugate method and 48.32s for the S.O.R. For the case shown in figure 15 the mesh size was reduced to half of that shown in figure 13 in the r and z directions. For this problem convergence was obtained in 194 outer iterations using the conjugate gradient method and 282 iterations using S.O.R. The total time spent solving the pressure equation was 155.45s for the conjugate gradient method and 117.19s for S.O.R.

The last problem modeled the flow of water in the cold leg and downcomer of a PWR in which blockage occurred in the downcomer. The resulting flow pattern is shown in figure 16. In this figure

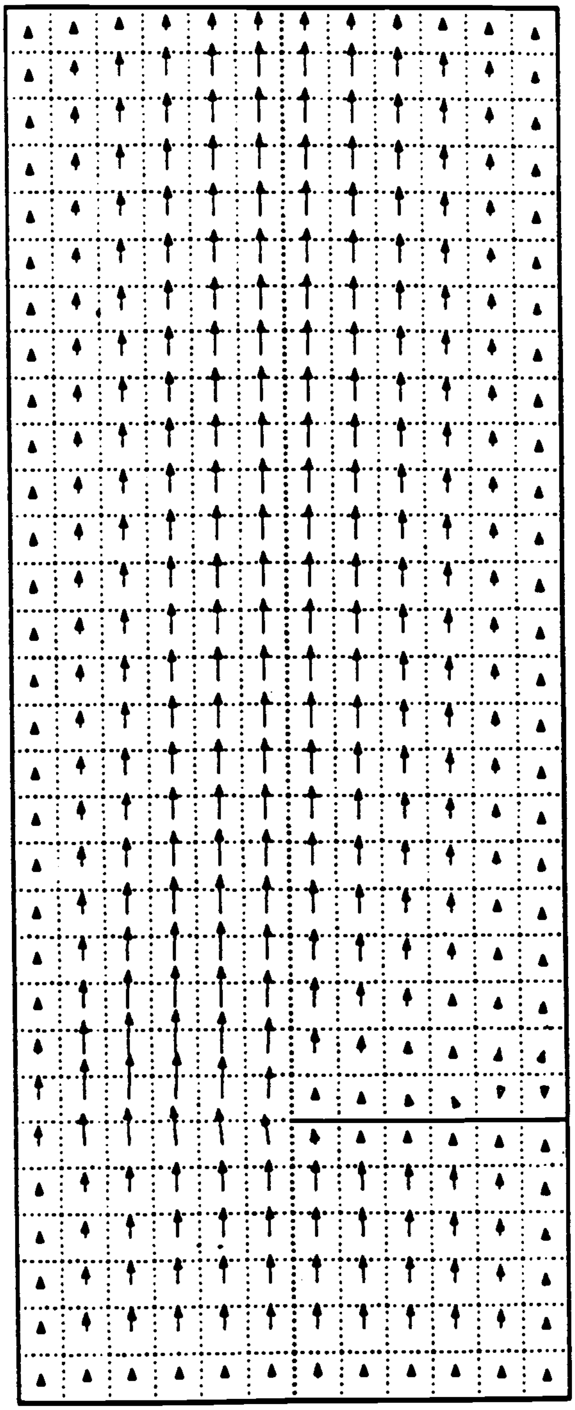

Figure 14. Cross Section of Isothermal Air Flow through a Pipe

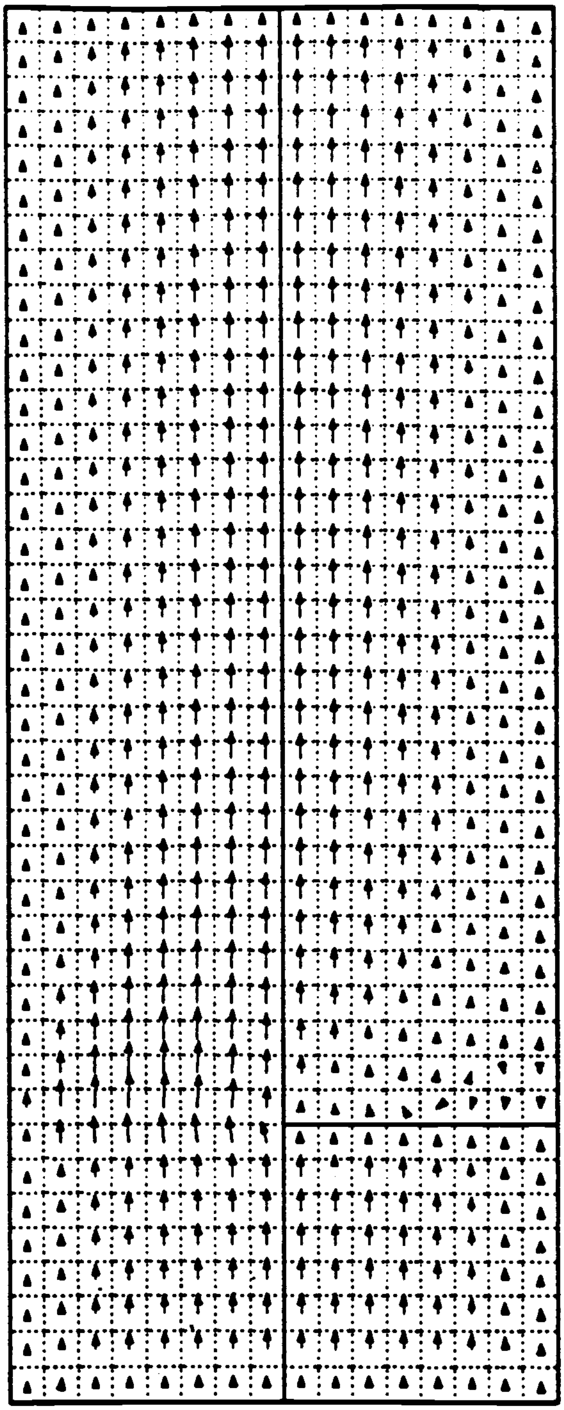

Figure 15. Cross Section of Isothermal Air Flow through a Pipe

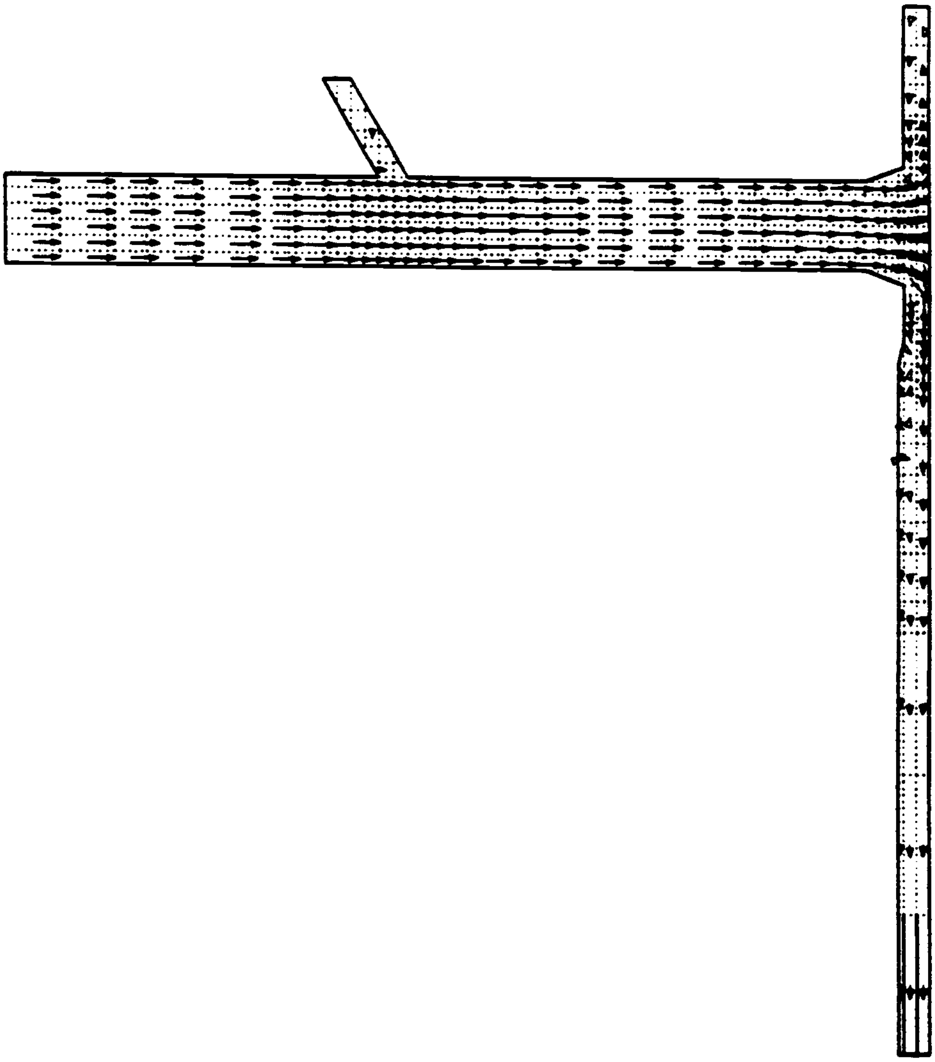

Figure 16. Cross Section of Cold Leg, Downcomer, and High Pressure Injector

the horizontal portion is the cold leg having the high pressure injector on the top left and the downcomer to the right. In figure 17 a view of the downcomer in a different plane shows the blockage which is the cutaway portion near the top of the right surface.

In this problem there were two fluid inlets. The first inlet was to the far left of the cold leg where the fluid was assigned constant temperature and normal velocity boundary conditons of 64.08°C and .0161 m/s . The second inlet was on top of the high pressure injector where the corresponding boundary conditions were 64.08°C and 0 m/s. Because the normal velocity was set to 0 m/s at this inlet, the high pressure injector was essentially shut off.

The fluid outlet was located at the bottom of the downcomer. Although the temperature and normal velocity values were initialized to 64.08°C and 0 m/s at this outlet, they were allowed to vary to satisfy conservation requirements. The normal velocities of all other surfaces were fixed at 0 m/s while their temperatures were initialized to 64.08°C. The internal fluid temperature was also initialized to 64.08°C. Because the temperatures of all the surfaces and the internal fluid were set to 64.08°C, the problem was isothermal, and the energy equation was therefore not required in the problem solution.

Convergence for this problem was obtained in 266 outer iterations using the conjugate gradient method and 262 iterations using S.O.R. The total time spent solving the pressure equation was 207.27s for the conjugate gradient method and 224.56s for S.O.R.

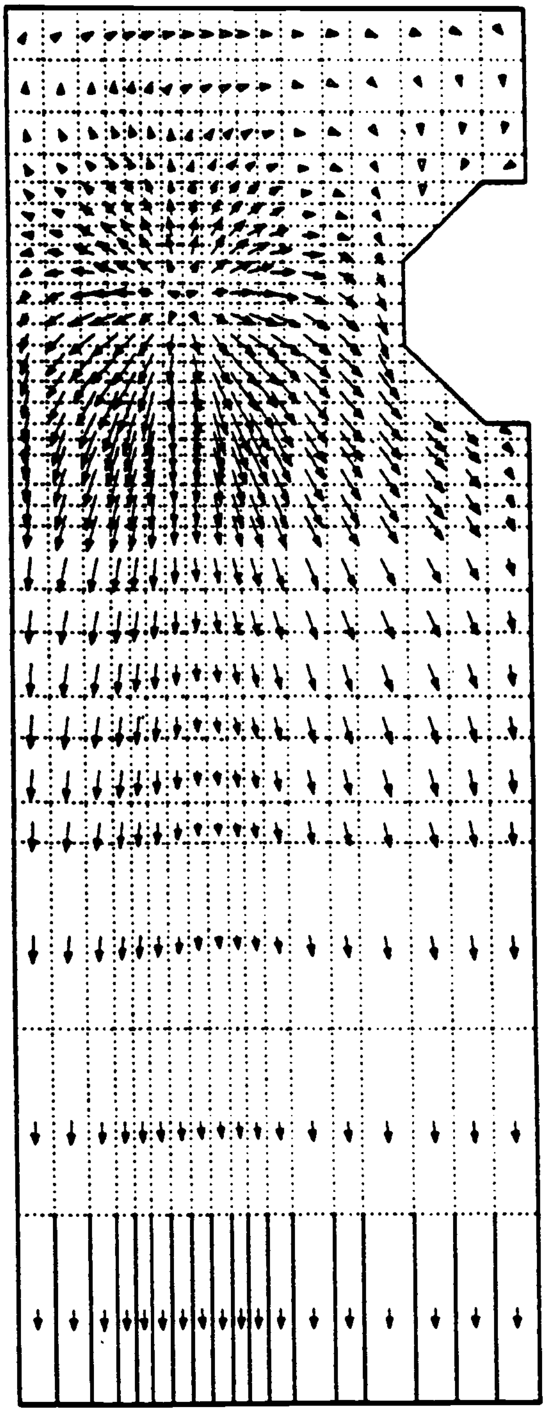

Figure 17. Cross Section of Downcomer Showing Blockage

The results of all six problems are summarized in table 13. It should be noticed from this table that the conjugate gradient method significantly out performed the S.O.R. method in the first four problems while losing in the fifth and nearly tieing in the sixth. To understand this behavior it is first necessary to discuss some differences between the two solution schemes.

4.3 Differences Between Solution Schemes

As can be seen from algorithm 2.9.2, the conjugate gradient method is executed in three steps. Since step 1 always occurs before the repetition of step 2, the time taken by this method to solve a system of equations may be divided into two categories. The first is the amount of time that is taken to perform step 1 and is fairly constant for a given problem on each outer iteration. The second category is the time taken in step 2 which is the product of the time required for one iteration of step 2 and the number of iterations of step 2. This result may be expressed mathematically to determine the time taken by the conjugate gradient method to obtain convergence in one outer iteration.

$$(4.3.1)\quad T_{Ctotal} = T_{C1} + T_{C2}N_C$$

For the S.O.R. method, however, there is no initial setup step and the time needed is therefore directly proportional to the number of iterations. This may be expressed mathematically as

$$(4.3.2)\quad T_{Stotal} = T_S N_S$$

Table 13. Summary of Results

	Problem	# of Cells	Total	Time(s)	Time Ratio
#	Type		C.G.	S.O.R.	(S.O.R./C.G.)
1	German 7 Pin Assembly	432	6.14	11.6	1.89
2a	C.R.B.R. Outlet Plenum	346	32.9	74.06	2.25
2b	C.R.B.R. Outlet Plenum (no rebalancing)	346	35.26	105.26	2.99
3	Atmospheric Fluidized Bed	2148	253.88	458.35	1.81
4	Cold Leg and Down Commer of a P.W.R.	978	51.35	103.76	2.02
5a	Isothermal Air Pipe Flow	800	21.35	15.60	.73
5b	Isothermal Air Pipe Flow (with more cells)	1800	71.67	48.32	.67
5c	Isothermal Air Pipe Flow (with more cells)	3200	155.45	117.19	.75
6	Cold Leg and Downcommer of a PWR with blockage	3404	207.27	224.56	1.08

For the problems run the setup time T_{C1} in (4.3.1) varied between 2.5 and 4.22 times the iterative S.O.R. time T_S. Furthermore, since the conjugate gradient method involves more bookkeeping work per iteration than the S.O.R. method, T_{C2} in (4.3.1) varied between 1.5 and 1.78 times the iterative S.O.R. time. If it is assumed that $T_{C1} = 3T_S$ and $T_{C2} = 1.6T_S$, the following crude equation may be derived from (4.3.1).

(4.3.3) $T_{Ctotal} = 3T_S + 1.6T_SN_C$

By equating (4.3.3) and (4.3.2) a relationship between the numbers of iterations of the different methods that result in the same total execution time is obtained.

(4.3.4) $3 + 1.6N_C = N_S$.

Equation (4.3.4) has important consequences for the competition between the methods. When the number of inner iterations is large for both methods, the setup time becomes negligible and the number of iterations of the conjugate gradient method must be smaller by a factor of 1.6 to tie the running time of the S.O.R. method. Obviously it must be lower by a factor of 3.2 to reduce the running time in half. When the number of iterations is small the setup time is no longer negligible, and the conjugate gradient method must beat the S.O.R. method by greater margins. For example, when N_S=10, $N_C \simeq 4$, and the conjugate gradient method must beat the number of S.O.R. iterations by a factor of 2.5 to tie and by a factor of 5 to reduce the running time in half. Finally, when $N_S = 5$, $N_C \simeq 1$ so that for $N_S \leq 4$ the S.O.R. method becomes unbeatable.

From the previous development it is obvious that the burden placed on the conjugate gradient method becomes greater as the number of S.O.R. iterations becomes smaller. Furthermore, when the number of S.O.R. iterations decreases below 5, the S.O.R. method becomes unbeatable. As a consequence, the number of inner iterations per outer iteration must remain high in order for the conjugate gradient method to remain competitive.

4.4 Summary of Results

The difference in the ratios of running times for different problems can be explained from the development of the previous section. In all problems run with the S.O.R. method the number of inner iterations per outer iteration was large in the beginning of a run and decreased to very low numbers towards the end of a run. As a consequence, the conjugate gradient method beat the S.O.R. method during the first part of a problem and lost toward the end. It is therefore obvious that the conjugate gradient method beat the S.O.R. method by greater than 2 to 1 margins in the beginnings of problems for which the total S.O.R. execution time was beaten by a factor of 2. On the other hand, in cases where the total S.O.R. execution time was lower than that of the conjugate gradient method, the conjugate gradient method did not beat the S.O.R. method by a large enough margin in the beginning to compensate for its loss in the end.

The iteration histories that typify these two extreme cases are shown in figures 18 and 19 which are sketches intended to represent

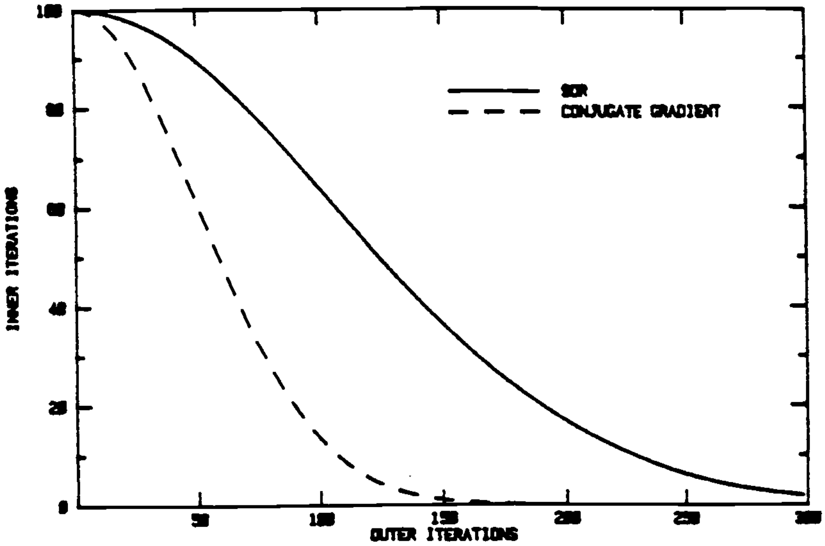

Figure 18. Showing a Typical Iteration History for which the Conjugate Gradient Method Beats the S.O.R. Method.

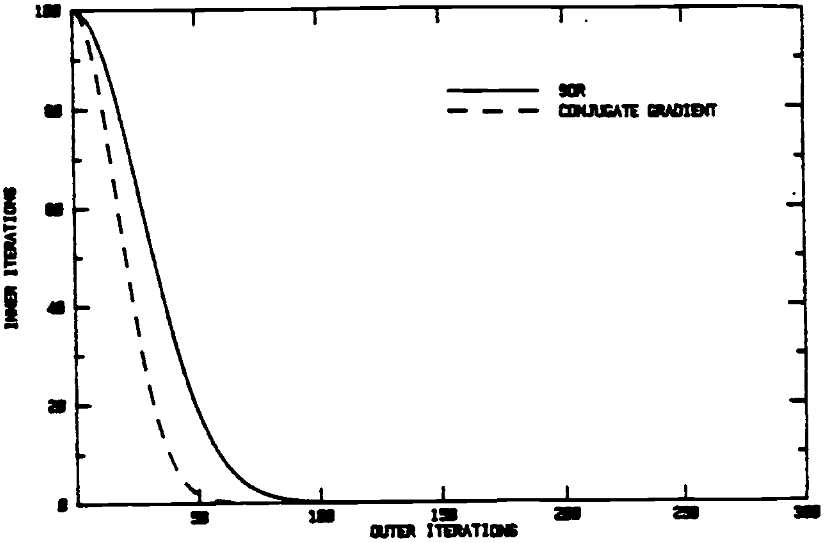

Figure 19. Showing a Typical Iteration History for which the S.O.R. Method Beats the Conjugate Gradient Method.

general trends and not actual cases. Figure 18 illustrates the case where the conjugate gradient method beats the S.O.R. method by a significant margin over a large range and consequently has a lower total execution time. The opposite case occurs in figure 19 where the number of S.O.R. iterations rapidly diminishes as a function of the outer iteration number. In this case the conjugate gradient method beats the S.O.R. method by only a small margin over a narrow range and consequently has a higher total execution time. The cases where the total running times are fairly equal are intermediate between these two.

The sketch in figure 18 is tyical of the iteration histories in the first four problems. Although the decrease in the number of iterations was not as smooth as shown in figure 18, the common feature is that the conjugate gradient method beat the S.O.R. method by significant margins over long ranges in the beginning of the problems.

The three cases run for problem five had curves similar to those shown in figure 19. The number of inner iterations per outer iteration quickly decreased for the S.O.R. method so that the conjugate gradient method never had a chance to beat it. Throughout most of these problems the conjugate gradient method converged in fewer iterations than the S.O.R. method but not sufficiently fewer to reduce the total execution time.

Problem 6 had iteration curves intermediate between those in figures 18 and 19. In this problem the conjugate gradient method beat the S.O.R. method by a significant margin over a moderate range in the beginning of the run. After this the methods competed rather equally for a long time, and finally the S.O.R. method beat the conjugate gradient method at the end of the problem. As a consequence, the total execution times were fairly equal.

The curves representing the iteration histories of problems 5 and 6 motivate two questions. First, what gives the iteration curves of these problems their characteristic shapes? Second, since problems 4 and 6 model very similar problems, why should their iteration histories differ so much? One possible explanation for both of these occurrences is the effect of mass rebalancing. One major difference between problem 4 and problem 6 is that problem six is divided more finely into 37 rebalancing regions whereas problem 4 is divided into only 6 regions. This would imply that the mass rebalancing in problem 6 produces a greater degree of resolution in the pressure field than that in problem 4. As a consequence, it may be that the pressure field obtained after mass rebalancing is so close to the iterative solution that the S.O.R. method converges in few inner iterations, leaving little room for the conjugate gradient method to beat it.

This hypothesis would also support the rapid convergence obtained in problems 5a-5c. Since these problems involve linear flow patterns in the z direction, there is essentially no pressure gradient in the other directions. Furthermore, since plane by plane rebalancing is used normal to the direction of flow, the pressure field obtained after rebalancing is so close to the final solution that the S.O.R. method converges rapidly, giving the conjugate gradient method no opportunity to beat it.

This explanation, however, motivates another question. Since problems 1 and 5 both involve linear flow patterns with plane by plane rebalancing normal to the flow direction, why can the conjugate gradient method compete in the former problem but not in the latter? Some minor differences between these problems are that problem 1 involves a more complicated flow domain with more thermal and momentum interaction between the structures and the fluid. A more significant difference, however, is that the over-relaxation factor is 1 for problem 1 and 1.5 for problem 5.

Although mass rebalancing accelerates the convergence of the pressure equation in some problems, its effects are sometimes peculiar. For instance, in problem 2, mass rebalancing greatly reduced the execution time of the S.O.R. method while hardly affecting the execution time of the conjugate gradient method. There are, however, other problems in which mass rebalancing greatly reduces the execution time of the conjugate gradient method. (These problems, however, were not mentioned previously since no data was taken for them.) This behavior motivates the following question. What factors contribute to the difference between solution methods in their sensitivity toward mass rebalancing?

One of the potential factors affecting the sensitivity to rebalancing is how the rebalancing regions match the problem geometry and flow field. For example, plane by plane rebalancing in the z direction would be inapprrpriate for a problem in which the flow was normal to the x direction. In addition, a rebalancing scheme may be damaging when it attempts to subdivide regions in which the flow is circular. For example, if plane by plane rebalancing in the z direction were used in problem 2, the flow would go up through the center of a rebalancing region and down through the sides of it. This situation would imply that the pressure gradient is negative in the center of the regions and positive on the sides. Under such circumstances two possible situations may occur. First, it might happen that the pressures in the center of the region need to be adjusted downward while those on the side need to be adjusted upward. Since the mass rebalancing scheme would apply the same pressure correction throughout each region, the pressures of all cells within a region would be adjusted in one direction so that the pressure field may be farther from convergence. On the other hand, it may be that the pressures in the center need a slight downward adjustment while those on the sides need a drastic downward adjustment. The effect of the uniform pressure change in the region may be a compromised solution that significantly overshoots the required central adjustments and undershoots the required peripheral adjustments. As a consequence, mass rebalancing may be ineffective under some circumstances. Furthermore, the interplay of the rebalancing regions and the problem geometry may favor one solution scheme more than the other.

Another potential factor affecting the sensitivity to rebalancing is the number of rebalancing regions into which the flow field is divided. As seen previously, finer divisions give a greater resolution of the adjusted pressure field after rebalancing. Depending upon whether or not the geometric effects of rebalancing are favorable to the given problem, the greater resolution may favor or damage one method more than the other.

Another possibility is that the differences in the methods themselves may make one method more sensitive to rebalancing. One difference between the two methods is that for the conjugate gradient method the pressures are updated in terms of quantities calculated from the values on the previous iteration. For S.O.R., however, the pressure is adjusted cell by cell so that the adjustments are made in terms of both the new and old iterate fields. Whatever the reason is, it remains clear that for some problems there is a difference in the sensitivity of different solution schemes to mass rebalancing. This effect should be investigated.

In the light of the previous discussion, it seems that for problems like 5a-5c, mass rebalancing is sufficient to obtain rapid convergence, and that combining the conjugate gradient method with mass rebalancing is essentially "overkill." For other cases like problems 1-4, convergence is still fairly time consuming after rebalancing so that the conjugate gradient method may be effectively applied. Furthermore, it was discussed that under certain circumstances the problem geometry may make the mass rebalancing scheme ineffective or even detrimental. Since the trend in fluid mechanics codes is to move toward larger problems for which convergence is slower and toward more complicated geometries for which mass rebalancing may be ineffective, the conjugate gradient method has merit as a linear equation solver.

Bibliography

1. Anton, Howard. Elementary Linear Algebra. John Wiley and Sons, New York, 1981.

2. Chandra, R. Conjugate Gradient Methods for Partial Differential Equations, Ph.D. Thesis, Yale University, May 1978.

3. COMMIX-1B: A Three-dimensional Transient Single-phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems. W.T. Sha, H.M. Domanus, et al., ANL-85-42, NUREG/CR-4348, September 1985.

4. Daniel, J.W. The Approximate Minimization of Functionals. Prenctice-Hall, 1971.

5. Engeli, H., T. Ginssburg, H. Ruthishauser, and E. Stiefel. Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems. Mitteilungen aus dem Institute fur Angewandte Mathematik: 8, Birkhauser Verlag, Basel, Stuttgart, 1959.