

## NUMERICAL SIMULATION METHODS OF INCOMPRESSIBLE FLOWS AND AN APPLICATION TO THE SPACE SHUTTLE MAIN ENGINE

J. L. C. CHANG,\* D. KWAK,† S. E. ROGERS‡ AND R.-J. YANG§

*NASA Ames Research Center, Moffett Field, CA 94035, U.S.A.*

### SUMMARY

This paper discusses incompressible Navier–Stokes solution methods with an emphasis on the pseudocompressibility method. A steady-state flow solver based on the pseudocompressibility approach is then described. This flow solver code has been used to analyse the internal flow in the Space Shuttle main engine hot-gas manifold. Salient features associated with this three-dimensional realistic flow simulation are discussed. Numerical solutions relevant to the current engine analysis and the redesign effort are discussed along with experimental results. This example demonstrates the potential of computational fluid dynamics as a design tool for aerospace applications.

KEY WORDS Incompressible flow, Navier–Stokes equations Finite differences

### INTRODUCTION

Incompressible flows are encountered in many realistic engineering problems related to hydrodynamics and also to low-speed aerodynamics. However, to date, computational flow simulation has not been a critical element in resolving many of these problems. For instance, impellers, automobiles, submarines, chemical reactors, etc. have been designed reasonably well by empirical means. As technologies advance, the design of modern flow devices tends to be more compact and highly efficient. Therefore the approach relying on empiricism and simplified analysis becomes inadequate for resolving problems associated with those devices requiring advanced analysis. For example, in analysing and redesigning the current Space Shuttle main engine (SSME) power head, computational simulations became an economical and time-saving supplement to experimental data. There are vast numbers of other real-world problems which demand accurate viscous incompressible flow solutions, such as rocket engine fuel flow, flow through an impeller (as in the turbo pump in the SSME) and biomedical flow (as in an artificial heart). There have been many computational simulations reported in the past on these subjects. In response to these demands, the present study attempts to develop an alternative method for simulating three-dimensional viscous incompressible flows. This paper is a summary of our recent effort (NASA Ames,

---

\* Senior Staff Scientist, Rocketdyne, Rockwell International, Canoga Park, CA, U.S.A.

† Research Scientist, NASA Ames Research Center, Moffett Field, CA, U.S.A.

‡ Research Scientist, Sterling Federal Systems, Palo Alto, CA, U.S.A.

§ Technical Staff, Rocketdyne, Rockwell International, Canoga Park, CA, U.S.A.

Rocketdyne and NASA Marshall Space Flight Center) to develop and apply an incompressible Navier–Stokes solver with a special emphasis on SSME applications.

A major problem involved in solving the incompressible Navier–Stokes equations comes from the lack of a pressure term in the continuity equation. In two-dimensional (2D) problems one can bypass this difficulty by using a stream function–vorticity formulation. In three dimensions (3D) a similar approach can be adopted using a vorticity–velocity formulation. However, the pressure term is removed at the expense of introducing three vorticity equations as well as requiring vorticity boundary conditions. In 2D, computational efficiency is not a major problem; however, in realistic 3D applications, satisfying continuity in a reasonable amount of computing time becomes a primary issue. Naturally, computational efficiency is of primary importance in addition to accuracy and robustness. For convenience and flexibility, a primitive variable formulation in generalized curvilinear co-ordinates is chosen, and our discussion is limited to this formulation using a finite difference approach. Derivation of equations and algorithmic details can be found in our earlier publications<sup>1–3</sup> and only equations relevant to the present discussion will be given in this paper.

### SOLUTION METHODS

Unsteady 3D incompressible flow with constant density is governed by the following Navier–Stokes equations:

$$\partial u_i / \partial x_i = 0, \quad (1)$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad (2)$$

where  $t$  is the time,  $x_i$  are the Cartesian co-ordinates,  $u_i$  are the corresponding velocity components,  $p$  is the pressure and  $\tau_{ij}$  is the viscous stress tensor. Here all variables are non-dimensionalized by a reference velocity and length scale. The viscous stress tensor can be written as

$$\tau_{ij} = 2\nu S_{ij} - R_{ij}, \quad (3)$$

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (4)$$

Here  $\nu$  is the kinematic viscosity,  $S_{ij}$  is the strain rate tensor and  $R_{ij}$  are the Reynolds stresses. Various levels of closure models for  $R_{ij}$  are possible. In the present study turbulence is simulated by an eddy viscosity model using a constitutive equation of the form

$$R_{ij} = \frac{1}{3} R_{kk} \delta_{ij} - 2\nu_t S_{ij}, \quad (5)$$

where  $\nu_t$  is the turbulent eddy viscosity. By including the normal stress  $R_{kk}$  in the pressure,  $\nu$  in equation (3) can be replaced by  $\nu + \nu_t$ , namely

$$\tau_{ij} = 2(\nu + \nu_t) S_{ij} = 2\nu_T S_{ij}. \quad (6)$$

In the remainder of this paper the total viscosity  $\nu_T$  will be represented simply by  $\nu$ . Therefore the incompressible Navier–Stokes equations are modified to allow variable viscosity in the present formulation.

In this section a few solution methods used in previous flow solver development work are briefly discussed.

*Method using Poisson's equation for pressure*

Perhaps the oldest and most commonly used method of satisfying the continuity equation is with the use of a derived equation, namely Poisson's equation for pressure.<sup>4</sup> The usual computational procedure is to solve the pressure field such that continuity is satisfied at the next time level. This procedure usually requires a relaxation scheme iterating on pressure until the divergence-free condition is satisfied within a specified numerical accuracy. To obtain a divergence-free velocity field numerically, spatial differencing of the Laplacian operator in Poisson's equation has to be consistent with the velocity differencing. Instead of a Laplacian operator, divergence of a gradient operator may be necessary. Even though this approach in general needs acceleration schemes to enhance computational efficiency further for three-dimensional problems, successful computations have been made.<sup>5,6</sup>

*Fractional step method*

The fractional step method can be used for time-dependent computations of the incompressible Navier–Stokes equations.<sup>7–14</sup> Here the time evolution is approximated by several steps. Various operator splittings can be adopted by treating the momentum equation as a combination of convection, pressure and viscous terms. The common application of this method is done by two steps. The first step is to solve for an auxiliary velocity field using the momentum equation in which the pressure gradient term can be computed from the pressure in the previous time step<sup>10</sup> or can be excluded entirely.<sup>11</sup> In the second step the pressure is computed which will map the auxiliary velocity onto a divergence-free velocity field.

A generalized flow solver based on this approach using a staggered grid has been developed in conjunction with the present work mainly for time-dependent computations. Details of this solver, which require lengthy description of various aspects such as the spatial-differencing scheme which maintains conservativeness, time integration and associated boundary conditions, can be found in Reference 15.

*Pseudocompressibility method*

Recent advances in the state of the art in computational fluid dynamics (CFD) have been made in conjunction with compressible flow computations. Therefore it is of significant interest to be able to use some of these compressible flow algorithms. To do this, the artificial compressibility method of Chorin<sup>16</sup> can be used. In this formulation the continuity equation is modified by adding a time derivative of the pressure term, resulting in

$$\frac{1}{\beta} \frac{\partial p}{\partial t} + \frac{\partial u_i}{\partial x_i} = 0. \quad (7)$$

Together with the unsteady momentum equations, this forms a hyperbolic–parabolic type of time-dependent system of equations. Thus fast implicit schemes developed for compressible flows, such as the approximate factorization scheme by Beam and Warming<sup>17</sup> or Briley and McDonald,<sup>18,19</sup> can be implemented. Various applications evolved from this concept have been reported for obtaining steady-state solutions.<sup>1–3,20,21</sup> Merkle and Athavale<sup>22</sup> have successfully computed time-dependent flows by subcycling each time step to satisfy the continuity equation.

Strigberger<sup>23</sup> and, independently, Briley *et al.*<sup>24</sup> interpreted the pseudocompressible formulation as a simplified form of low-Mach-number compressible equations. Then the low-Mach-

number approximation of the unsteady compressible flow equations becomes

$$\frac{\partial p}{\partial t} + \left(\frac{1}{M}\right)^2 \frac{\partial u_i}{\partial x_i} + u_i \frac{\partial p}{\partial x_i} = 0, \quad (8)$$

where  $M$  is the Mach number. Therefore the pseudocompressible formulation represented by equation (7) does not fully satisfy the time-dependent compressible flow formulation and is intended mainly for steady-state computations. As shown above, unsteady flow calculations may be performed by including the third term in the above equation. If approximate factorization is applied, where  $1/M^2$  corresponds to  $\beta$ , a large value of  $\beta$  will introduce factorization error. Therefore the unfactored form would be preferable for time-dependent calculations.

In an incompressible flow a disturbance in the pressure causes waves which travel with infinite speed. When pseudocompressibility is introduced, waves of finite speed result in which the magnitude of the speed depends upon the pseudocompressibility constant  $\beta$ . In a true incompressible flow the pressure field is affected instantaneously by a disturbance in the flow, but with pseudocompressibility there will be a time lag between the flow disturbance and its effect on the pressure field. In viscous flows the behaviour of the boundary layer is very sensitive to the streamwise pressure gradient, especially when the boundary layer is separated. If separation is present, a pressure wave travelling with finite speed will cause a change in the local pressure gradient which will affect the location of the flow separation. This change in separated flow will feed back to the pressure field, possibly preventing convergence to a steady state. An extensive mathematical account of the pseudocompressibility approach is given by Temam.<sup>25</sup>

Among the various solution techniques, the method of pseudocompressibility is chosen to gain efficiency and robustness in solving 3D real-world problems which require large numbers of grid points in curvilinear co-ordinates. In the present paper the algorithm and its physical interpretation are described. Also, other practical considerations such as grid-induced error, boundary conditions and a multiple-zone approach are presented. To improve the computational efficiency and accuracy, various acceleration schemes and modifications have been tried. The numerical scheme implemented in the present flow solver (INS3D code) will be summarized next.

## NUMERICAL SCHEME

### *Governing equations in generalized co-ordinates*

To perform calculations on 3D arbitrarily shaped geometries, the physical co-ordinates are transformed into general curvilinear co-ordinates by introducing the independent variables

$$\tau = t, \quad \xi_i = \xi_i(x, y, z, t).$$

Applying the transformation to the governing equations (2) and (7) yields

$$\frac{\partial}{\partial \tau} \hat{D} + \frac{\partial}{\partial \xi_i} (\hat{E}_i - \hat{E}_{vi}) = 0. \quad (9)$$

Here

$$\hat{D} = \frac{D}{J} = \frac{1}{J} \begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix}, \quad \hat{E}_i = \frac{1}{J} \begin{bmatrix} \beta U_i + (\xi_i)_t (p - \beta) \\ u U_i + (\xi_i)_x p \\ v U_i + (\xi_i)_y p \\ w U_i + (\xi_i)_z p \end{bmatrix},$$

where  $J$  is the Jacobian of the transformation. The contravariant velocities  $U_i$  (without metric normalization) are defined as

$$U_i = (\xi_i)_t + (\xi_i)_x u + (\xi_i)_y v + (\xi_i)_z w. \tag{10}$$

The viscous terms are given by

$$\hat{E}_{vi} = \frac{v}{J} \left\{ \nabla \xi_i \cdot \left( \nabla \xi_i \frac{\partial}{\partial \xi_i} \right) \begin{bmatrix} 0 \\ u \\ v \\ w \end{bmatrix} + \left( \xi_{ix} \frac{\partial u}{\partial \xi_j} + \xi_{iy} \frac{\partial v}{\partial \xi_j} + \xi_{iz} \frac{\partial w}{\partial \xi_j} \right) \begin{bmatrix} 0 \\ \frac{\partial}{\partial x} \xi_j \\ \frac{\partial}{\partial y} \xi_j \\ \frac{\partial}{\partial z} \xi_j \end{bmatrix} \right\}. \tag{11a}$$

When  $v$  is constant, the contribution of the second group of terms sums to zero when the velocity field is divergence-free. However, since in general  $v$  varies in space and time, these terms have to be kept. For the flow with constant  $v$  in orthogonal co-ordinates, the above full viscous terms can be simplified further as follows:

$$\hat{E}_{vi} = \frac{v}{J} \nabla \xi_i \cdot \left( \nabla \xi_i \frac{\partial}{\partial \xi_i} \right) \begin{bmatrix} 0 \\ u \\ v \\ w \end{bmatrix} = \gamma_i D. \tag{11b}$$

*Difference equations*

There are a number of different ways of defining variables in a grid system. Either standard cell node oriented or a staggered arrangement can be chosen. In Cartesian co-ordinates a staggered grid has several favourable properties.<sup>4,11</sup> In generalized co-ordinates these advantages become obscured because of the interpolation required. However, a fully conservative differencing scheme can be devised maintaining the convenience of a staggered arrangement in a Poisson solver.<sup>15</sup> Using any grid system, spatial differencing can be done either in finite difference or finite volume form. The finite volume scheme usually behaves better near geometric singularities. In spatial differencing, both central differencing and upwind differencing have been tried. In this report only the work using central differencing will be discussed.

The numerical algorithm used to advance equation (9) in time is an implicit, non-iterative, approximately factored, finite difference scheme.<sup>17-19</sup> The time differencing used by this scheme is generally known as the trapezoidal rule and is given by

$$\hat{D}^{n+1} = \hat{D}^n + \frac{\Delta\tau}{2} \left[ \left( \frac{\partial \hat{D}}{\partial \tau} \right)^n + \left( \frac{\partial \hat{D}}{\partial \tau} \right)^{n+1} \right] + O(\Delta\tau^3), \tag{12}$$

where the superscript  $n$  refers to the  $n$ th time step.

The problem is to solve for  $D^{n+1}$ , and this is non-linear in nature since  $\hat{E}^{n+1} = \hat{E}(D^{n+1})$  is a non-linear function of  $D^{n+1}$ . The following linearization procedure is used. A local Taylor expansion

about  $D^n$  yields

$$\hat{E}_i^{n+1} = \hat{E}_i^n + \hat{A}_i^n (D^{n+1} - D^n) + O(\Delta\tau^2), \quad (13)$$

where  $\hat{A}_i$  is the Jacobian matrix

$$\hat{A}_i = \frac{\partial \hat{E}_i}{\partial D}. \quad (14a)$$

The Jacobian matrices can be represented by

$$\hat{A}_i = \frac{1}{J} \begin{bmatrix} L_0 & L_1\beta & L_2\beta & L_3\beta \\ L_1 & Q + L_1u & L_2u & L_3u \\ L_2 & L_1v & Q + L_2v & L_3v \\ L_3 & L_1w & L_2w & Q + L_3w \end{bmatrix}, \quad (14b)$$

where

$$\begin{aligned} Q &= L_0 + L_1u + L_2v + L_3w, \\ L_0 &= (\xi_i)_t, \quad L_1 = (\xi_i)_x, \quad L_2 = (\xi_i)_y, \quad L_3 = (\xi_i)_z, \\ \xi_i &= (\xi, \eta \text{ or } \zeta). \end{aligned}$$

Substituting equation (13) into equation (9) results in the governing equation in delta form:

$$\begin{aligned} & \left( I + \frac{h}{2} J^{n+1} [\delta_\xi(\hat{A}_1^n - \Gamma_1) + \delta_\eta(\hat{A}_2^n - \Gamma_2) + \delta_\zeta(\hat{A}_3^n - \Gamma_3)] \right) (D^{n+1} - D^n) \\ &= -\Delta\tau J^{n+1} [\delta_\xi(\hat{E}_1 - \hat{E}_{v1})^n + \delta_\eta(\hat{E}_2 - \hat{E}_{v2})^n + \delta_\zeta(\hat{E}_3 - \hat{E}_{v3})^n] + \left( \frac{J^{n+1}}{J^n} - 1 \right) D^n, \end{aligned} \quad (15)$$

where

$$\begin{aligned} \Gamma_i D^{n+1} &= \hat{E}_{vi}^{n+1}, \\ h &= \begin{cases} \Delta\tau & \text{for trapezoidal,} \\ 2\delta\tau & \text{for Euler,} \end{cases} \\ \delta_\xi &= \text{finite difference operator for } \partial/\partial\xi. \end{aligned}$$

At this point it should be noted that the notation of the form  $[\delta_\xi(A - \Gamma)]D$  refers to

$$\left[ \frac{\partial}{\partial\xi}(AD) - \frac{\partial}{\partial\xi}(\Gamma D) \quad \text{and not} \quad \frac{\partial A}{\partial\xi}D - \frac{\partial\Gamma}{\partial\xi}D. \right]$$

#### Approximate factorization

The solution of equation (15) would involve a formidable matrix inversion problem. With the use of an alternating direction implicit (ADI) type scheme, the problem could be reduced to the inversion of three matrices of small bandwidth, for which there exist some efficient solution algorithms. The particular ADI form used here is known as approximate factorization (AF). It is difficult to implement the AF scheme to equation (15) in its full matrix form. Noting that at steady state the left-hand side of equation (15) approaches zero, a simplified expression for the viscous

term as shown in equation (11b) is used on the left-hand side. To maintain the accuracy of the solution, the entire viscous term is used on the right-hand side. Using this, the governing equation becomes

$$L_{\xi}L_{\eta}L_{\zeta}(D^{n+1} - D^n) = \text{RHS}, \tag{16}$$

where

$$\begin{aligned} L_{\xi} &= \left[ I + \frac{\Delta\tau}{2} J^{n+1} \delta_{\xi}(\hat{A}_1^n - \gamma_1) \right], \\ L_{\eta} &= \left[ I + \frac{\Delta\tau}{2} J^{n+1} \delta_{\eta}(\hat{A}_2^n - \gamma_2) \right], \\ L_{\zeta} &= \left[ I + \frac{\Delta\tau}{2} J^{n+1} \delta_{\zeta}(\hat{A}_3^n - \gamma_3) \right] \end{aligned} \tag{17}$$

and RHS is the right-hand side of equation (15). When second-order central differencing is used, the solution to this problem becomes the inversion of three block tridiagonal matrices. The inversion problem is reduced to the three inversions:

$$\begin{aligned} (L_{\eta})\Delta\bar{D} &= \text{RHS}, \\ (L_{\xi})\Delta\bar{D} &= \Delta\bar{D}, \\ (L_{\zeta})\Delta D^{n+1} &= \Delta\bar{D}. \end{aligned} \tag{18}$$

These inversions are carried out for all interior points, and the boundary conditions can be implemented explicitly. It is possible, however, to implement the boundary conditions implicitly.

The factorization has introduced the following second-order cross-product term into the equation:

$$h^2[\delta_{\xi}A_1\delta_{\eta}A_2 + \delta_{\eta}A_2\delta_{\zeta}A_3 + \delta_{\zeta}A_3\delta_{\xi}A_1]\Delta D + O(h^3), \tag{19}$$

where

$$A_1 = \hat{A}_1^n - \gamma_1, \quad A_2 = \hat{A}_2^n - \gamma_2, \quad A_3 = \hat{A}_3^n - \gamma_3, \quad h = \frac{\Delta\tau}{2} J^{n+1}.$$

To maintain the second-order accuracy of the scheme, the added terms must be kept smaller than the original terms in the equations everywhere in the computational domain. This puts a restriction on the size of the pseudocompressibility parameter  $\beta$ . The proper choice of  $\beta$  is discussed in our earlier reports.<sup>1,2</sup> In applying the approximate factorization scheme, it has been found that the stability of the scheme is dependent on the use of some higher-order smoothing terms. These are used to damp out higher-frequency oscillations which arise in the solution because of the use of second-order central differencing, and will be discussed in a following section.

*Numerical dissipation/smoothing*

Higher-order smoothing terms are required to make the present algorithm stable. These terms help to damp out the higher-order oscillations in the solution that are caused by the use of second-order central differencing. The smoothing term can be related to an upwind finite difference approximation. An extensive discussion on numerical dissipation can be found in Reference 26. In this report, specifics relevant to the pseudocompressible formulation are discussed.

Including these smoothing terms, equations (16) and (17) become

$$L_\xi L_\eta L_\zeta (D^{n+1} - D^n) = \text{RHS of (15)} - \varepsilon_c [(\nabla_\xi \Delta_\xi)^2 + (\nabla_\eta \Delta_\eta)^2 + (\nabla_\zeta \Delta_\zeta)^2] D^n, \tag{16'}$$

where

$$\begin{aligned} L_\xi &= \left[ I + \frac{\Delta\tau}{2} J^{n+1} \delta_\xi (\hat{A}_1^n - \gamma_1) + \varepsilon_i \nabla_\xi \Delta_\xi \right], \\ L_\eta &= \left[ I + \frac{\Delta\tau}{2} J^{n+1} \delta_\eta (\hat{A}_2^n - \gamma_2) + \varepsilon_i \nabla_\eta \Delta_\eta \right], \\ L_\zeta &= \left[ I + \frac{\Delta\tau}{2} J^{n+1} \delta_\zeta (\hat{A}_3^n - \gamma_3) + \varepsilon_i \nabla_\zeta \Delta_\zeta \right]. \end{aligned} \tag{17'}$$

Here  $\nabla$  and  $\Delta$  represent forward and backward spatial-differencing operators respectively. To preserve the tridiagonal nature of the system, only second-order smoothing can be used on the left-hand side of the equation, whereas fourth-order smoothing is used on the right-hand side. When the diagonal algorithm (described in a later section) is used, however, it is feasible to increase the bandwidth of the system to a pentadiagonal. This makes it possible to use fourth-order smoothing on the left-hand side of the equations also. When this combination of dissipation terms is used, the AF algorithm will be stable if  $\varepsilon_i$  and  $\varepsilon_c$  satisfy a certain relation.<sup>26</sup>

To study the nature of the numerical smoothing, one-dimensional forms of dissipation terms are represented as follows:

$$[1 - \varepsilon_i \nabla_\xi \Delta_\xi](p^{n+1} - p^n) = -\varepsilon_c (\nabla_\xi \Delta_\xi)^2 p^n. \tag{20}$$

Suppose  $p$  is represented by the discrete Fourier expansion

$$p = \sum_n \hat{p}(k) e^{ik\xi}, \tag{21}$$

where

$$\begin{aligned} k &= \frac{2\pi}{N\Delta\xi} n = \text{wave number}, \\ n &= -N/2, \dots, 0, 1, \dots, (N/2 - 1), \\ N &= \text{number of mesh points.} \end{aligned}$$

Substituting equation (21), equation (20) can be written as

$$[1 - \varepsilon_i k'](\hat{p}^{n+1} - \hat{p}^n) = -\varepsilon_c (k')^2 \hat{p}^n, \tag{22}$$

where

$$\begin{aligned} k' &= -2 + 2 \text{coocos}(k), \\ (k')^2 &= 6 - 8 \cos(k) + 2 \cos(2k). \end{aligned}$$

From this, the amplification factor can be defined as

$$\sigma = \frac{\hat{p}^{n+1}}{\hat{p}^n} = \frac{[1 - \varepsilon_i k' - \varepsilon_c (k')^2]}{[1 - \varepsilon_i k']}. \tag{23}$$

To damp out the numerical fluctuations as time advances, the absolute value of the amplification factor  $\sigma$  has to be less than one for all possible frequencies, i.e.

$$|\sigma| < 1.$$



Noting that  $k'$  is always negative, this requirement leads to the relation

$$\varepsilon_e(k')^2 \leq 2(1 - \varepsilon_i k'). \quad (24a)$$

It can be shown that the above inequality is always satisfied if

$$2\varepsilon_e \leq \varepsilon_i. \quad (24b)$$

The exact relation between these two coefficients can be determined only by a non-linear stability analysis. In the present code  $\varepsilon_i$  is taken to be three times larger than  $\varepsilon_e$ . From the expression given in equation (23), it is clear that if  $\varepsilon_i$  is too large, the rate of damping will be diminished. It may not be advantageous to take an excessively large value for  $\varepsilon_i$  over  $\varepsilon_e$ . The choice of  $\varepsilon_e$  depends on the Reynolds number and the grid spacings. However, as discussed later, large values of  $\varepsilon_e$  adversely affect the accuracy of the continuity equation. Therefore the magnitude of  $\varepsilon_e$  is usually taken to be small. If grid sizes are fine enough to resolve the changes in the flow field, it can be taken as small as  $10^{-3}$ .

There are two major sources of inaccuracy associated with the numerical dissipation terms, namely (1) the numerical dissipation terms effectively change the Reynolds number of the flow and (2) the explicit smoothing terms added to the continuity equation do not conserve mass. In particular the explicit smoothing on the pressure can affect whether or not the solution converges to an incompressible flow field. Chang and Kwak<sup>2</sup> showed that the pseudopressure waves decay exponentially with time and vanish as the solution converges. Thus the change in pressure with time approaches zero. When there is no explicit smoothing added to the equation, the divergence of the velocity field identically approaches zero. However, when explicit smoothing is included, as the change in pressure approaches zero, the divergence of velocity approaches

$$\frac{\partial u_i}{\partial x_i} \rightarrow -\frac{\varepsilon_e(1)}{\beta \Delta \tau} [(\nabla_{\zeta} \Delta_{\zeta})^2 + (\nabla_{\eta} \Delta_{\eta})^2 + (\nabla_{\zeta} \Delta_{\zeta})^2] p, \quad (25)$$

where  $\varepsilon_e(1)$  is the explicit smoothing parameter for the pressure. If  $\varepsilon_e$  is scaled by  $h$ , i.e.  $\varepsilon_e = \Delta \tau \varepsilon_e$ , equation (25) becomes independent of the time step. This term can deteriorate the conservation of mass depending on the magnitude of  $\beta$  and the local pressure gradient. When the pressure gradients become substantial, as in the case when a region of separation is present, the smoothing term does not approach zero, and this contaminates the divergence of the velocity field. In this situation mesh refinement usually does not help reduce the magnitude of this term. To alleviate this problem, it may become necessary to decrease the smoothing coefficient in the continuity equation as the solution converges.

### Diagonal algorithm

In a diagonal algorithm a similarity transform is used to diagonalize the Jacobian matrices and uncouple the set of equations. The equations can then be solved by solving scalar tridiagonal matrices instead of solving block tridiagonal matrices. A similarity transform which symmetrizes and diagonalizes the matrices of the compressible gas dynamic equations has been used by Warming *et al.*<sup>27</sup> and Turkel.<sup>28</sup> This method was used by Pulliam and Chaussee<sup>29</sup> to produce a diagonal algorithm for the Euler equations. This can be applied to the compressible Navier–Stokes equations to obtain considerable savings in computing time.<sup>30</sup> In this section similarity transforms for the matrices used in the method of pseudocompressibility are presented. They are used in a diagonal algorithm which results in a substantial reduction in computer time.

Similarity transformations exist which diagonalize the Jacobian matrices

$$\hat{A}_i = T_i \hat{\Lambda}_i T_i^{-1}, \tag{26}$$

where  $\hat{\Lambda}_i$  is a diagonal matrix, whose elements are the eigenvalues of the Jacobian matrices, which is given by

$$\hat{\Lambda}_i = \begin{bmatrix} Q & 0 & 0 & 0 \\ 0 & Q & 0 & 0 \\ 0 & 0 & Q+C & 0 \\ 0 & 0 & 0 & Q-C \end{bmatrix}. \tag{27}$$

Here  $C$  is the pseudospeed of sound, which is given by

$$C = \sqrt{[(Q - L_0)^2 + \beta(L_1^2 + L_2^2 + L_3^2)]}.$$

The  $T_i$  matrix is composed of the eigenvectors of the Jacobian matrix. For  $i = 1$  ( $\xi$ -sweep) the first two eigenvectors are given by

$$X_1 = \begin{bmatrix} 0 \\ \tilde{x}_\zeta \\ \tilde{y}_\zeta \\ \tilde{z}_\zeta \end{bmatrix}, \quad X_2 = \begin{bmatrix} 0 \\ \tilde{x}_\eta \\ \tilde{y}_\eta \\ \tilde{z}_\eta \end{bmatrix}. \tag{28}$$

For  $i = 2$  ( $\eta$ -sweep) they are

$$X_1 = \begin{bmatrix} 0 \\ \tilde{x}_\xi \\ \tilde{y}_\xi \\ \tilde{z}_\xi \end{bmatrix}, \quad X_2 = \begin{bmatrix} 0 \\ \tilde{x}_\zeta \\ \tilde{y}_\zeta \\ \tilde{z}_\zeta \end{bmatrix}. \tag{29}$$

For  $i = 3$  ( $\zeta$ -sweep) they are

$$X_1 = \begin{bmatrix} 0 \\ \tilde{x}_\eta \\ \tilde{y}_\eta \\ \tilde{z}_\eta \end{bmatrix}, \quad X_2 = \begin{bmatrix} 0 \\ \tilde{x}_\xi \\ \tilde{y}_\xi \\ \tilde{z}_\xi \end{bmatrix}. \tag{30}$$

The determinant of  $T_i$  is given by

$$\det(T_i) = 2\tilde{C}^3,$$

which remains bounded independent of the geometry. For more detail on the derivation of these matrices, see Reference 3.

The implementation of the diagonal scheme involves replacing the Jacobian matrices in the implicit operators with the product of the similarity-transform matrices and the diagonal matrix as given in equation (26). The identity matrix in the implicit operators is replaced by the product of the similarity-transform matrix and its inverse. A modification is made to the implicit viscous terms so that the transformation matrices may also be factored out of these terms. This implicitly adds an additional viscous dissipation term to the pressure. The transformation matrices are now

factored out of the implicit operators to give

$$\begin{aligned} \mathcal{L}_\xi &= T_\xi \left[ I + \frac{\Delta\tau}{2} J\delta_\xi(\Lambda_\xi - \hat{\gamma}_1) \right] T_\xi^{-1}, \\ \mathcal{L}_\eta &= T_\eta \left[ I + \frac{\Delta\tau}{2} J\delta_\eta(\Lambda_\eta - \hat{\gamma}_2) \right] T_\eta^{-1}, \\ \mathcal{L}_\zeta &= T_\zeta \left[ I + \frac{\Delta\tau}{2} J\delta_\zeta(\Lambda_\zeta - \hat{\gamma}_3) \right] T_\zeta^{-1}, \end{aligned} \tag{31}$$

where the implicit viscous terms are now given by

$$\hat{\gamma}_i = \frac{\nu}{J} \nabla \xi_i \cdot \nabla \xi_i I \delta_{\xi_i}.$$

Since the transformation matrices are dependent on the metric quantities, factoring them outside the difference operators introduces an error. No modification has been made to the right-hand side of the equation; therefore these linearization errors will not affect the steady-state solution, only the convergence path toward the solution.

The implementation of this algorithm over the block algorithm will result in a substantial reduction in computational time per iteration because of the decrease in the number of operations performed. Additionally, considerably less memory is required to store the elements on the left-hand side. This additional memory was used to further vectorize the existing code as follows. Since the solution of a tridiagonal block or scalar matrix is recursive, it is not vectorizable for loops which use the current sweep direction as the inner do-loop index. However, if a large number of these matrices are passed into the inversion routines at once, then vectorization can take place in the ‘non-sweep’ direction. This diagonalized version is given as an option in the INS3D code.

*Boundary conditions*

An important part of any computer code is the proper implementation of boundary conditions. The code must be capable of handling the several different types of boundaries encountered in numerical simulations, which include solid surface, inflow and outflow, and far-field boundaries.

*Solid surface.* At a solid surface boundary the usual no-slip condition is applied. In general the grid point adjacent to the surface will be sufficiently fine so that constant pressure normal to the surface in the viscous boundary layer can be assumed. For a  $\zeta = \text{constant}$  surface this can be expressed as

$$\left( \frac{\partial p}{\partial \zeta} \right)_{L=1} = 0. \tag{32}$$

*Inflow, outflow and far-field conditions.* The inflow and outflow boundary conditions for an internal flow problem and the far-field boundary conditions for an external flow problem can be handled in much the same way. The incoming flow for both problems can be set to some appropriate constant as dictated by the problem. For example, at the inlet to a pipe the pressure can be set to a constant and the velocity profile can be specified to be uniform. The conditions in the downstream region, however, are the most difficult to provide. Simple upwind extrapolation is not well posed. The best convergence rate is obtained if global mass is conserved. So to ensure the best results, the velocities and pressure are first updated using a second-order upwind extrapol-

ation. Then these extrapolated velocities are integrated over the exit boundary to obtain the outlet mass flux. The extrapolated velocity components are weighted by the mass flux ratio to conserve global mass, i.e.

$$\mathbf{V}^{n+1} = \frac{\dot{m}_{\text{in}}}{\dot{m}_{\text{out}}} \mathbf{V}^{\tilde{n}}. \quad (33)$$

If nothing further is done to update the boundary pressure, this can lead to discontinuities in the pressure because momentum is not being conserved. A method of weighting the pressure by a momentum correction is used to obtain a pressure which corresponds to the mass-weighted velocities:

$$p^{n+1} = p^{\tilde{n}} - \frac{1}{\zeta_z} [(wW)^{n+1} - (wW)^{\tilde{n}}] + \frac{v}{\zeta_z} (\nabla\zeta \cdot \nabla\zeta) \left[ \left( \frac{\partial w}{\partial \zeta} \right)^{n+1} - \left( \frac{\partial w}{\partial \zeta} \right)^{\tilde{n}} \right], \quad (34)$$

where  $W$  is the contravariant velocity. In obtaining this formula, it has been assumed that the streamlines near the exit plane are nearly straight. Any appreciable deviation will cause a discontinuity in the pressure and may lead to an instability. To avoid this, a momentum-weighted pressure was used. This was obtained by integrating the momentum-corrected pressure  $p^{n+1}$  and the extrapolated pressure  $p^{\tilde{n}}$  across the exit:

$$I_p^{n+1} = \int_{\text{exit}} p^{n+1} d\hat{a}, \quad I_p^{\tilde{n}} = \int_{\text{exit}} p^{\tilde{n}} d\hat{a}. \quad (35)$$

The final outlet pressure is then taken to be

$$p^{n+1} = \left( \frac{I_p^{n+1}}{I_p^{\tilde{n}}} \right) p^{\tilde{n}}. \quad (36)$$

Using these downstream boundary conditions, global conservation of mass and momentum are ensured, and the scheme will not introduce instabilities into the flow field.

## SSME POWER HEAD FLOW SIMULATION

### *Validation of flow solver*

The present flow solver (INS3D code) has been validated by computing fundamental fluid dynamics problems such as channel flow, the flow over a backward-facing step and the flow over a circular cylinder. Three-dimensional cases include flow over an ogive cylinder, flow through a rectangular duct, wind tunnel inlet flow, cylinder wall juncture flow and flow through multiple posts mounted between two plates. These results have been reported in our earlier publications and will not be repeated here.<sup>1-3, 31-33</sup> The most striking application of the present code is the flow simulation in the power head portion of the SSME.<sup>34, 35</sup> In the remaining parts of this section several important results obtained to date will be presented.

### *Background of SSME flow analysis*

For future scientific and commercial applications, an upgrade of the SSME power head has been under way which will substantially increase the operating margin and the engine durability. To achieve this goal without increasing the weight and size of the existing engine, it became essential to understand the dynamics of the hot-gas flow in the engine power head. Because of the complexity of the geometry, an experimental approach is extremely difficult as well as time-

consuming. Computational simulation therefore offers an economical alternative to complement the experimental work in analysing the current configuration and to suggest new, improved design possibilities. In the past few years major milestones have been established from this effort. In this report highlights of our initial task are presented.

In the SSME staged combustion cycle the fuel is partially burned at very high pressure and relatively high temperature in the preburners. The resulting hot gas is used to run the turbine and is then routed to the main injector where, along with additional oxidizer, it is injected into the main combustion chamber. The Reynolds number of the primary flow in the manifold is of order  $10^6$  per inch. Because of the high gas temperature, the Mach number is less than 0.12. The flow is turbulent and is practically incompressible.

Figure 1 illustrates the current arrangement for the SSME power head components. Hot gas discharged from the gas turbine enters the annular turnaround duct (TAD) and experiences a  $180^\circ$  turn before it diffuses into the fuel bowl. This assembly is called the hot-gas manifold (HGM). The gas flows into the main injector through three transfer ducts on the left-hand side of the power head (fuel preburner side) and enters into the region of the main injector posts. On the right-hand side of the power head (oxidizer preburner side) there are two transfer ducts connected to the right-hand side of the main injector assembly. Around the bottom portion of each liquid oxygen (LOX) post in the main injector assembly there are a number of small holes through which the hot gas flows into the main combustion chamber. There it mixes with the oxidizer, which comes through circular passages along the centreline of each LOX post. As a part of the current redesign effort, a CFD study has been conducted to simulate the dynamics of the hot-gas flow in the power head.

#### *The computer model and the grid*

A computational model of the power head is chosen to analyse critical areas where the dynamics of the hot-gas flow is expected to have a significant effect on the overall performance of the SSME. As shown in Figure 2, the model starts from the gas turbine exit on the fuel preburner side and extends to the main injector assembly. The main injector consists primarily of a bundle of LOX posts, which is physically modelled by a porous medium.

Figures 2(a) and 2(b) demonstrate the 3D grid for the SSME HGM. They show a horizontal and a vertical cross-section respectively of the HGM. Figure 2(c) illustrates the details of the H-grid for the cross-section of the three transfer ducts. This H-grid is generated for a unit circle. Near the boundary the grid lines are concentric circles except in the vicinity of the four singular points. Using the nearly orthogonal grid in this unit circle, one can obtain H-grids for tubes or ducts of any given shape and dimension by a simple linear transformation. Figure 2(d) shows an unwrapped surface of the annular fuel bowl with openings. The elliptical transfer duct shown in Figure 2(c) represents an advanced two-duct design which will be explained later in this section.

The grid for the entire HGM system is generated by using algebraic functions and is written with a high degree of flexibility for changing geometric configurations. By specifying the shape, the dimension and the desired number of transfer ducts, a grid for a variety of new HGM configurations can be obtained in a short time. The ducts described in this paper are connected directly to the fuel bowl without any fairings, while in the current engine the three transfer ducts are connected smoothly to the annular fuel bowl with fairings. This configuration with an abrupt change in geometry is more demanding computationally than smooth configurations.

#### *Multiple-zone computation*

A large number of mesh points is required to resolve the 3D turbulent flow in the SSME. To overcome the limitation in computer core memory, the domain of interest is divided into several

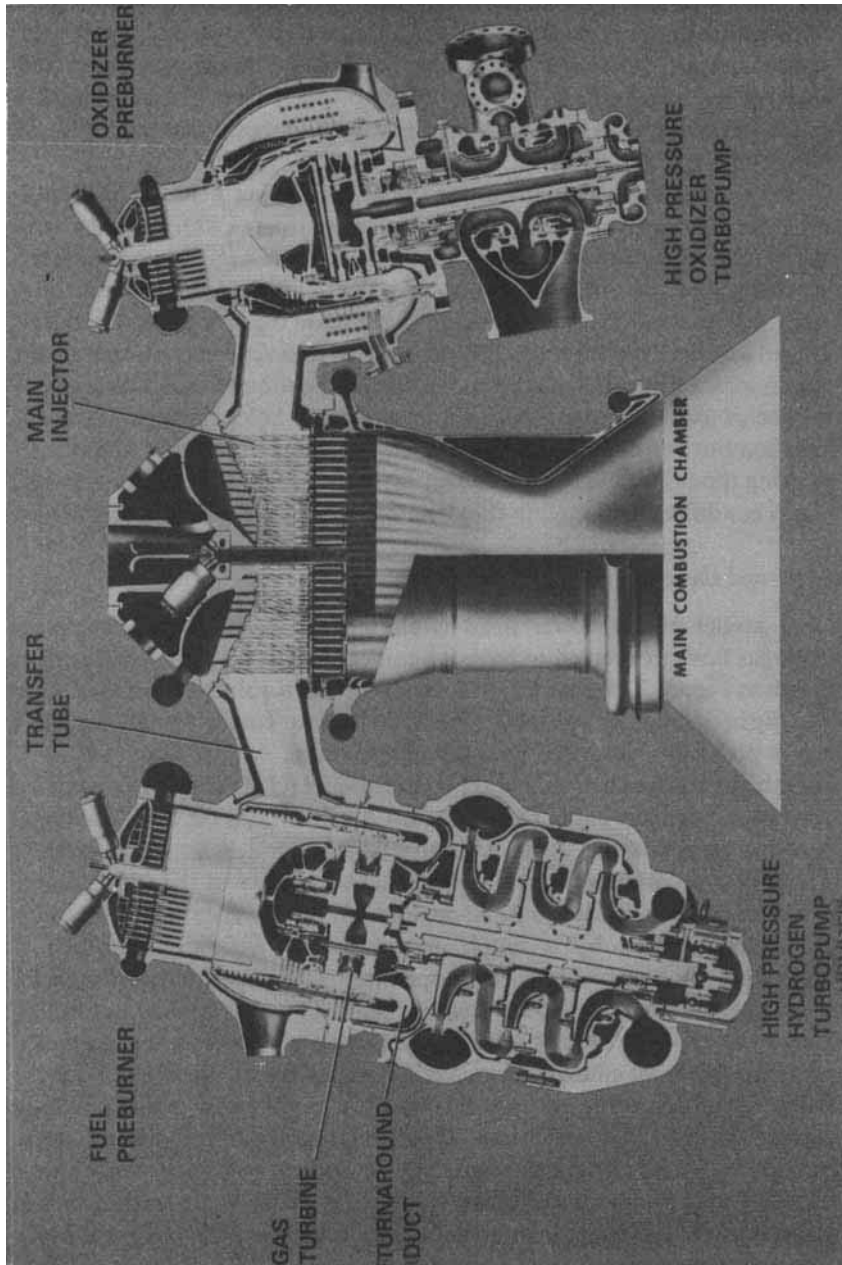


Figure 1

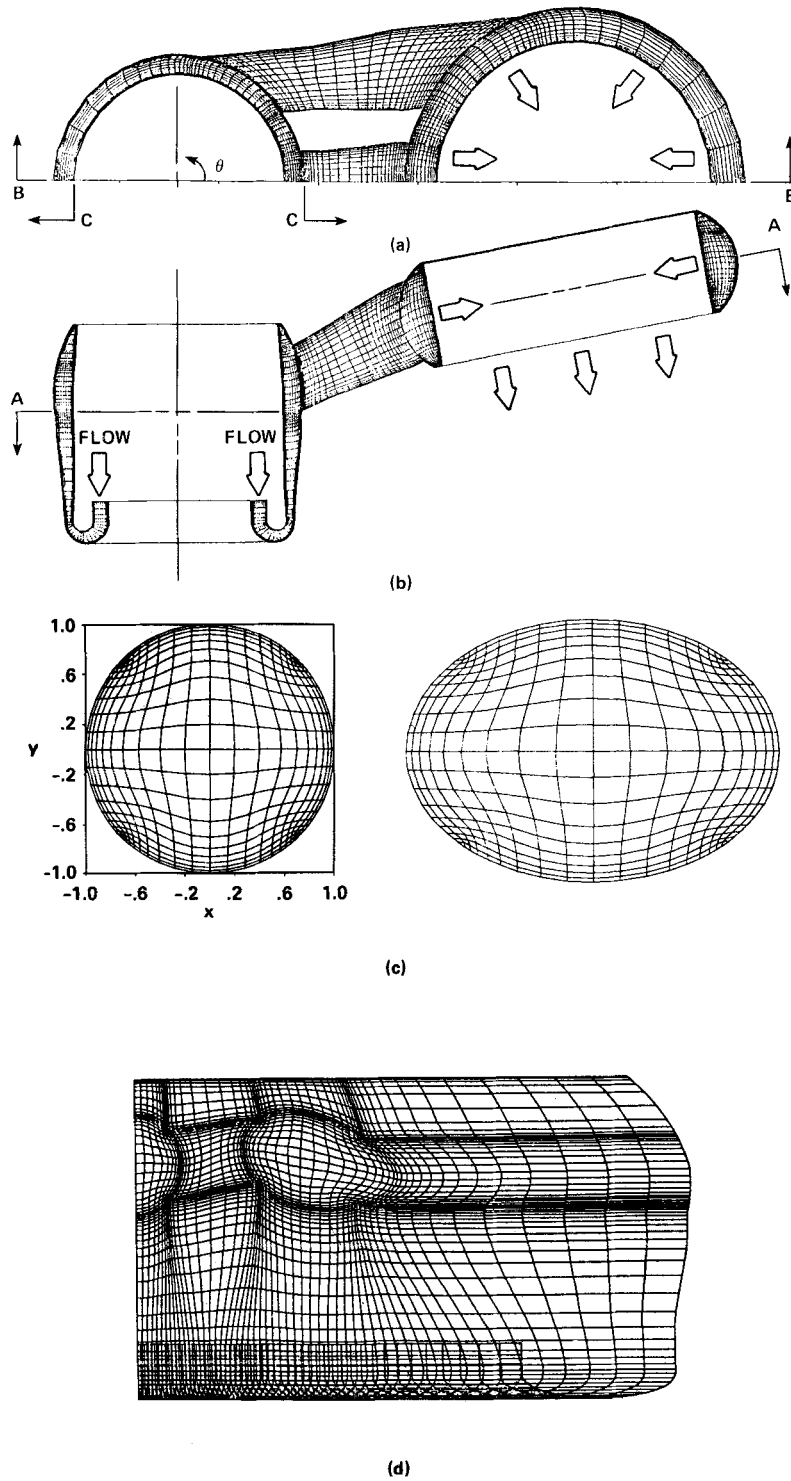
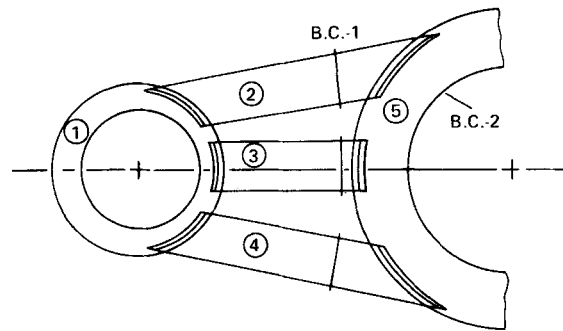


Figure 2



ZONE  
 ① TAD AND FUEL BOWL  
 ②, ③, AND ④ TRANSFER DUCTS  
 ⑤ RACETRACK (FUTURE WORK)  
 OVERLAPPING GRID IN ZONE INTERFACES

Figure 3

zones. This requires a special treatment at the interface for a smooth continuation of the solution between zones. Figure 3 illustrates a five-zone arrangement for the HGM flow field. Zone 1 is allocated for the TAD and fuel bowl. Zones 2, 3 and 4 are for the three transfer ducts. The racetrack of the main injector is represented by zone 5. Also shown in the figure are some overlapping grids in the various zonal interfaces. The grid is chosen to be continuous and smooth across the zonal boundaries. In this paper the racetrack (zone 5) is not included in the computation. Since the vertical plane through the centre of the fuel bowl and the main injector is taken to be a plane of symmetry, only half of the HGM is computed.

The equations of motion given in equations (2) and (7) are of hyperbolic type with parabolic-type viscous diffusion terms. Waves are propagating in both upstream and downstream directions while the solution approaches a steady state. The interfaces between zones of the present problem are locations where the geometry changes abruptly. Therefore in the neighbourhood of these interfaces flow is expected to experience a rapid change. To maintain a smooth continuation of the solutions across these zones, and hence to achieve a stable and fast-converging computation, a means of providing adequate communication for the travelling waves must be established. Overlapping regions and a proper zonal interpolation scheme are thus required for this purpose.

A forward or backward differencing, if applied to the interfaces of multiple zones, would distort the geometric representation. To maintain a smooth transition of the flow field across a zonal boundary, the Jacobian and the metrics at the interfaces are computed using grid points in neighbouring zones. Then the pressure and the velocities  $Q$  are updated explicitly at each iteration. Let values at  $n+1/2$  denote the state of conditions to be used to advance the computation to  $n+1$ . The values of  $Q^{n+1/2}$  for zone 1 at the exit plane are obtained from the values of the corresponding plane of zone 2 at  $n$ , i.e.

$$[Q_{BC}^{n+1/2}]_{\text{zone 1}} = [Q_{\text{interior}}^n]_{\text{zone 2}}; \quad (37)$$

and values of  $Q^{n+1/2}$  for zone 2 at the zonal interfaces are taken from the latest computed result of zone 1 as

$$[Q_{BC}^{n+1/2}]_{\text{zone 2}} = [Q_{\text{interior}}^{n+1}]_{\text{zone 1}}. \quad (38)$$

When more than two points are overlapped, the latest values in the interior of this overlapping



region must be properly transmitted to the next zone. There are a number of ways to treat this problem. The simplest one is to take an average of the two values computed in zones 1 and 2 as follows:

$$[Q^{n+1/2}] = \frac{1}{2}([Q^n]_{\text{zone2}} + [Q^{n+1}]_{\text{zone1}}). \quad (39)$$

A scheme using updated zonal boundary values, but with no interior updates, has also been tested. Either way, converged steady-state solutions have been obtained. However, the scheme with interior updates converges at a much faster rate.

#### Grid-induced error

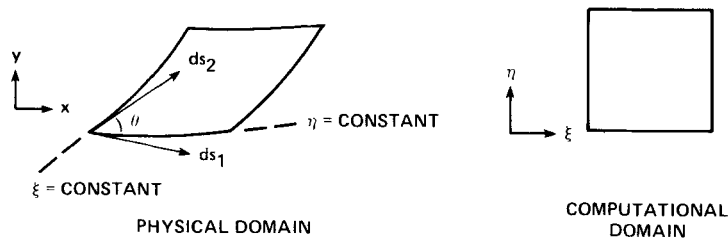
To generate grids for realistic 3D problems like the SSME, it is convenient to combine different types of grids depending on the problem's geometric characteristics. In the present application, for continuity and smoothness across zonal boundaries, an H-type grid is chosen for the circular transfer duct as shown in Figure 2(c). This is then connected to the side-wall grid of the HGM as illustrated in Figure 2(d). In generating this grid for the SSME, errors are introduced mainly due to grid singularities, skewness and stretching.

The present AF algorithm integrates the difference equations along the transformed coordinate ( $\xi$ ,  $\eta$  and  $\zeta$ ) directions. At the junction of the two H-grid directions, flow particles in the two co-ordinate directions could communicate only indirectly via the interior mesh points. This produces some corner-effect error. Even though this error is not as severe as the one caused by external flows, an *ad hoc* method of eliminating this corner effect is devised based on a finite element concept. Let  $j$  and  $k$  denote the indices for the grid points along the increasing  $\xi$ - and  $\eta$ -directions respectively; and let  $j = k = 1$  be the corner point, which is singular. First, the pressure at this point,  $p_{11}$ , is determined by an extrapolation along the diagonal direction  $j = k$ . Secondly,  $p_{13}$  and  $p_{31}$  are obtained in the usual manner. Then  $p_{12}$  and  $p_{21}$  are established by an interpolation along the circular surface.

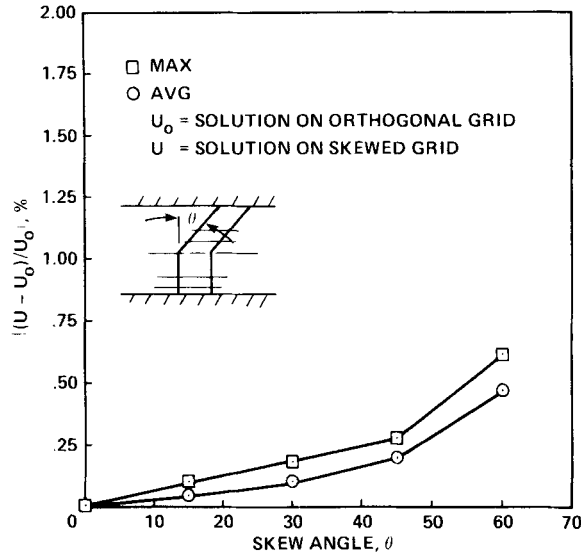
The full viscous term given by equation (11a) can be simplified to equation (11b) when the grid is orthogonal. Even though full viscous terms can be used, it is convenient and economical to keep only the orthogonal part. It is therefore of practical interest to estimate the overall error caused by orthogonal formulation when a non-orthogonal grid system is used. Using the notation defined in Figure 4(a), the skewness can be related to the metric terms as

$$\begin{aligned} ds_1 \cdot ds_1 &= (\eta_x^2 + \eta_y^2) \left( \frac{d\xi}{J} \right)^2, \\ ds_2 \cdot ds_2 &= (\xi_x^2 + \xi_y^2) \left( \frac{d\eta}{J} \right)^2, \\ ds_1 \cdot ds_2 &= -(\xi_x \eta_x + \xi_y \eta_y) \left( \frac{d\xi d\eta}{J^2} \right), \\ \frac{ds_1 \cdot ds_2}{(ds_1^2 ds_2^2)^{1/2}} &= \cos \theta. \end{aligned}$$

As a quick measure of an overall error caused by grid skewness, a 2D channel flow is computed using two grids, namely (1) a stretched Cartesian grid (orthogonal) and (2) a non-orthogonal grid where the skewness is controlled on the upper half of the channel as sketched in Figure 4(b). In the computation only orthogonal terms are kept. Converged solutions on the lower half, where the two grids are identical, are then compared as shown in Figure 4(b). The total error depends



(a)



(b)

Figure 4

additionally on the Reynolds number and the third-directional skewness. However, this quick experiment indicates that the orthogonal assumption can be used without significantly impacting on the overall solution if the grid is reasonably orthogonal.

#### Turbulence models

Several levels of turbulence models have been implemented into the code. These include the Baldwin and Lomax<sup>36</sup> algebraic model in which the length scale is determined by the location of the maximum moment of vorticity. This model has been widely applied in external flow problems. However, the maximum moment of vorticity is not as well defined for fully turbulent internal flows as for external flows. In particular, the moment of vorticity is almost constant for a fully developed pipe or channel flow except in the sublayer region. For the present problem it is proposed that the length scale is determined by the point of minimum vorticity. This length scale is incorporated into an extended Prandtl-Karman mixing-length theory. The combination of these automatically accounts for the curvature effect. Full details of this model are given in other reports.<sup>35,37</sup> For the present SSME flow computations this extended Prandtl-Karman mixing-length model is used. A review on various levels of turbulence models can be found in Reference 38 or 39.

### Computed results

Flow solutions in a variety of different HGM configurations at various Reynolds numbers are presented. Here the Reynolds number is based on the mean velocity and the duct width at the entrance of the TAD.

*Three-circular-duct HGM.* First, steady-state solutions are obtained for the current three-circular-duct HGM at  $Re = 1000$ . As illustrated in Figure 5, in the present analysis, the racetrack of the main injector is not connected to the ducts. The three transfer ducts are assumed to discharge the flow separately. There is no communication of the pressure between the centre and the outer ducts at their exit planes. For this reason, small residual waves have remained in the computed results. However, the root-mean-square value of the change in the flow variables,  $\Delta Q$ , has dropped below  $10^{-5}$ , and an essentially steady-state solution has been obtained.

In Figures 6(a) and 6(b) velocity vectors are shown in the horizontal and vertical cross-sections corresponding to Figures 2(a) and 2(b). The flow in the centre transfer duct, as illustrated in Figure 6(b), is highly non-uniform, and a large separation region is formed just downstream of the entrance to the transfer ducts. By comparison of the vector length in Figure 6(a), the flow in the centre duct is much slower than it is in the outer ones. The results shown in these figures agree qualitatively with the airflow test data conducted at an  $Re$  of about  $10^6$ . The predicted mass flow through the centre duct is 9.8% of the total mass flow, which agrees with the test data.

Figures 7(a) and 7(b) illustrate the 3D velocity vectors at the unwrapped centre plane and at a plane near the inner wall of the fuel bowl. A reverse flow pattern is clearly visible near the inner wall. Three-dimensional swirl patterns are predicted in the vicinity of the entrance to the transfer ducts. Figure 7(c) is a photograph that indicates, by means of surface streak (shear pattern) visualization, the similar swirls at the corresponding locations in the airflow test.

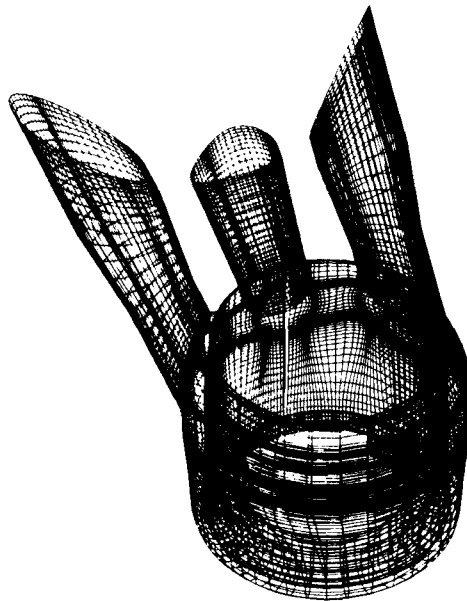
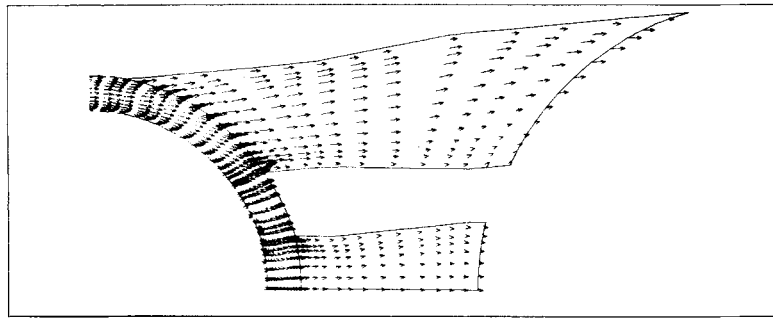
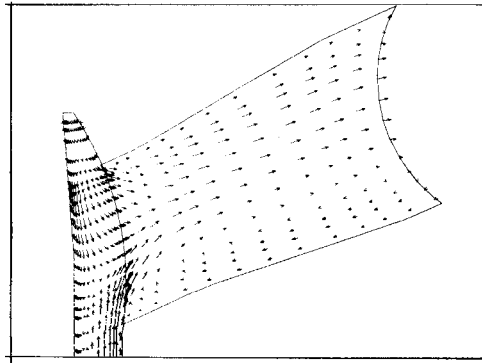


Figure 5



(a)



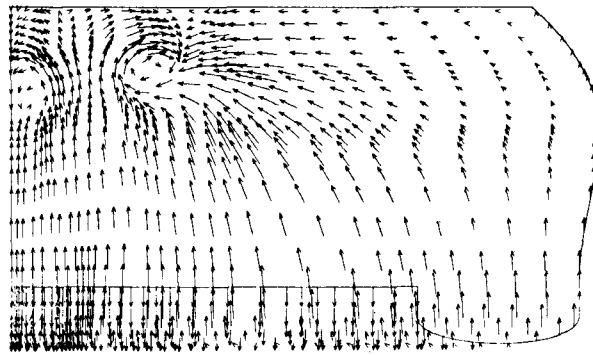
(b)

Figure 6

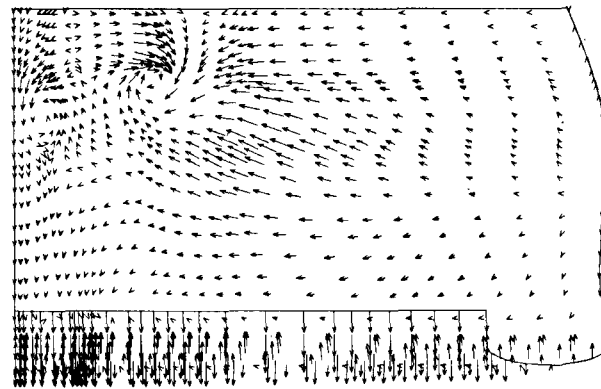
The existence of the swirls can be explained as follows. The flow coming from below has a large momentum due to the relatively small width of the annular duct. Among the streamlines of this flow in between the two ducts there exists a dividing streamline. This streamline has a stagnation point at the top of the fuel bowl as shown in Figures 7(a) and 7(b). On the left-hand side of this dividing streamline the flow is bent leftward to the centre duct. Because of symmetry, a rightward flow is also approaching the centre from the other side. When these opposite currents approach each other, another dividing streamline is formed with a stagnation point again at the top wall. The stagnation pressure forces the streams to bend downward, and at the same time the streams make a right-angle turn into the circular duct. Conservation of momentum thus requires the formation of swirls.

The pattern of the swirl and its centre depend on the relative strength of the approaching currents. Near the centre duct double swirls of equal strength are formed because of symmetry. In the vicinity of the entrance to the outer duct the current approaching leftward from the rear part of the bowl is more massive than the one approaching rightward. A stronger swirl is thus formed and is located sideways toward the weaker stream.

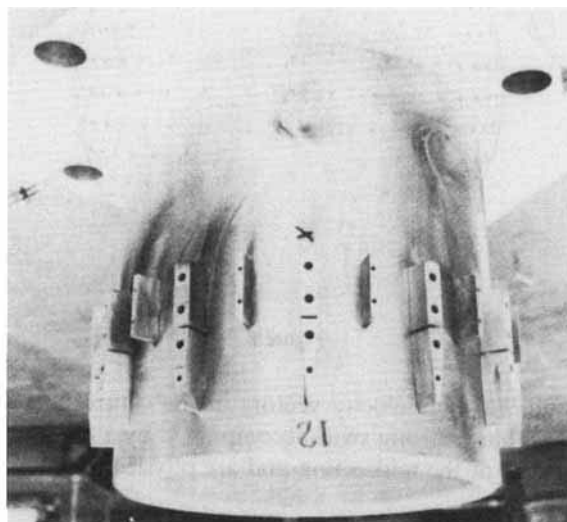
Figures 8 and 9 are the perpendicular cross-sectional views showing three different sections of the transfer ducts, namely near the entrance, at the midsection and near the exit plane. Near the



(a)



(b)



(c)

Figure 7

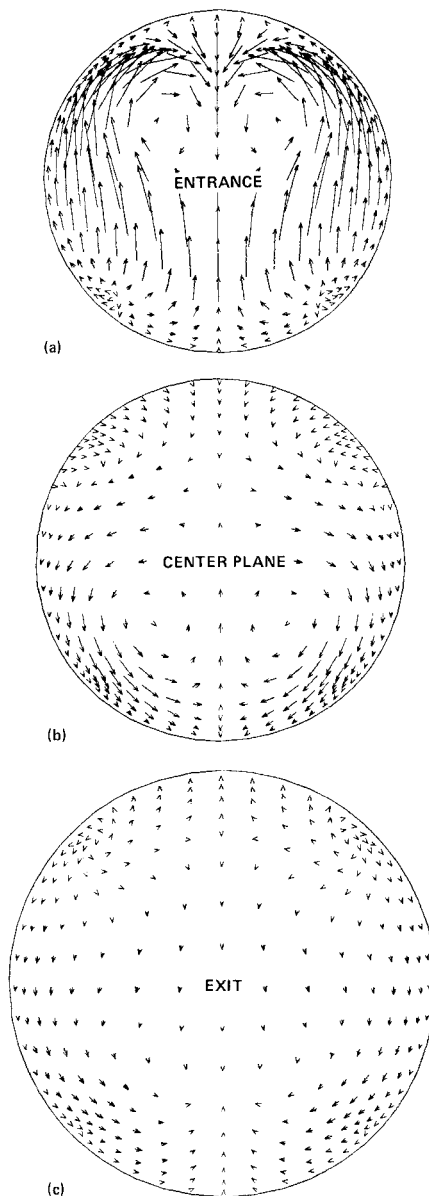


Figure 8

entrance (Figures 8(a) and 9(a)) the velocity vectors in the centre duct have symmetric double swirls, while the outer duct has a strong swirl accompanied by a much weaker one. The swirling velocities are largely reduced at the midsection and are physically dissipated before entering the main injector regions.

*New two-elliptical-duct HGM.* From this computational flow analysis and also from experiments, the centre duct of the current three-duct HGM is found to transfer a limited amount of mass flow (about 10% of the total flow). Also the transverse pressure gradient remains large and is

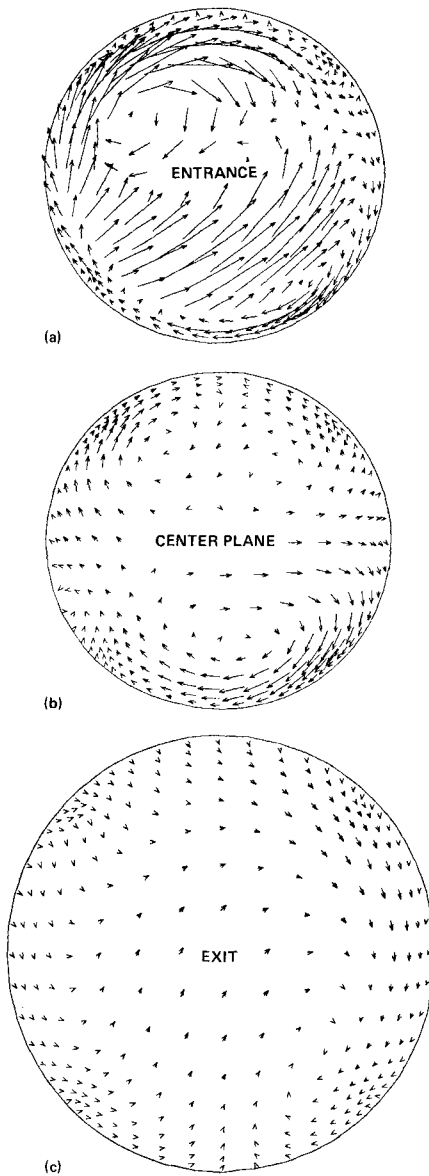


Figure 9

accompanied by a large bubble of separation after the  $180^\circ$  turn. To improve the quality of the flow, a large-area, two-duct design concept has been developed. In addition, the ducts are chosen to have an elliptical shape in order to distribute the mass flow evenly to the main injector region. An outer-surface grid for a two-duct model is illustrated in Figure 10.

First, to reduce the large separation bubble after the  $180^\circ$  turn, a parametric study is performed to find the best possible configuration. In Figures 11(a) and 11(b) comparison of the current three-duct configuration and the new two-duct design is illustrated. As shown in Figure 11(a), a large separation bubble existing in the present design is practically removed in the new configuration

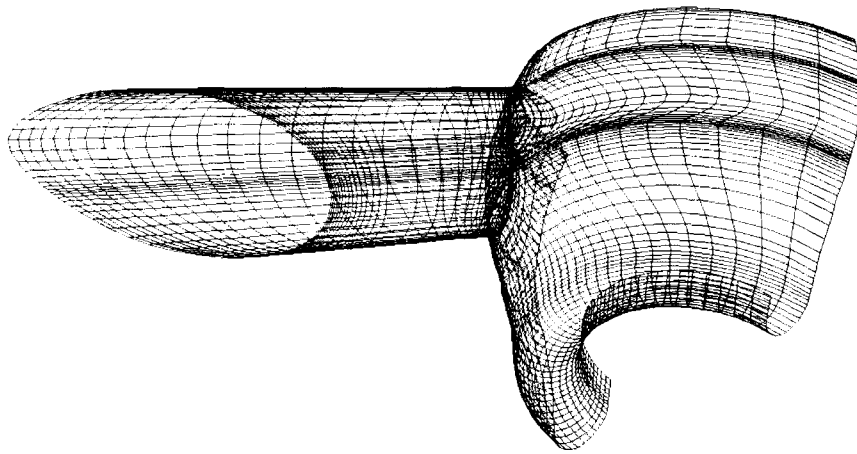


Figure 10

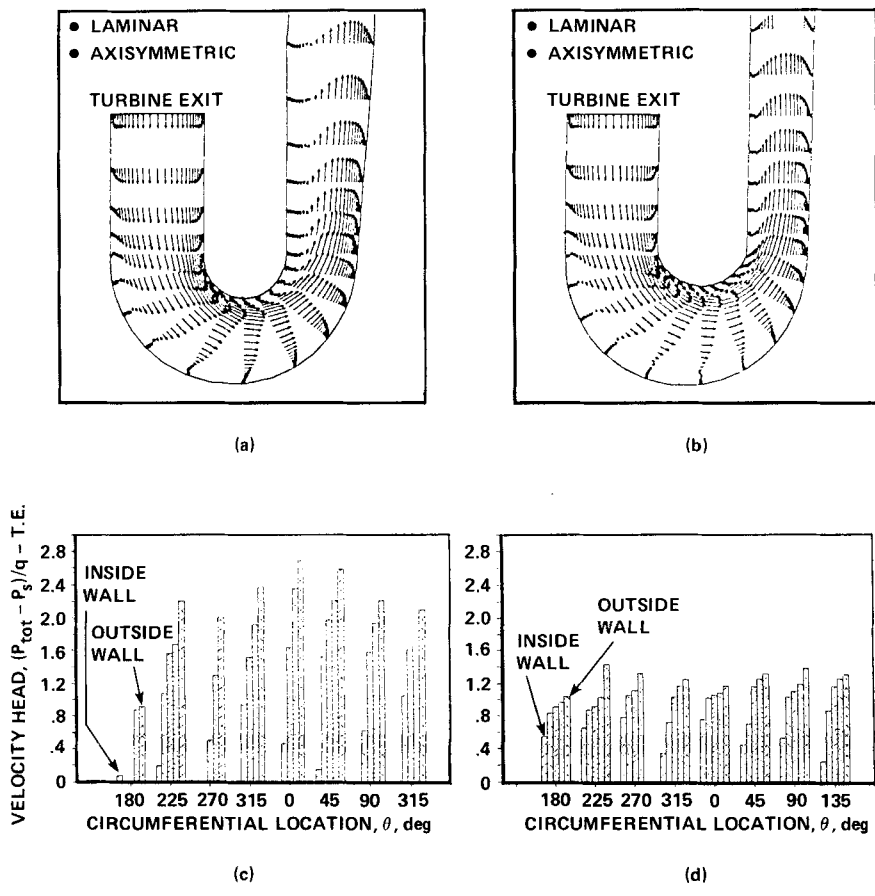


Figure 11

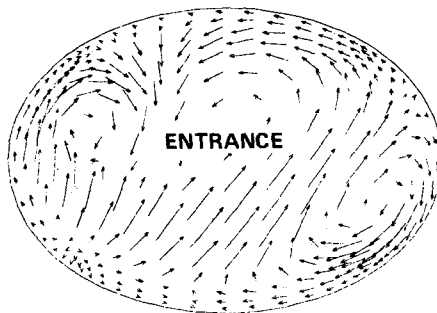


shown in Figure 11(b). This is confirmed by experiment as shown in Figures 11(c) and 11(d) for the current and new designs respectively, where velocity measurements at five different locations across the channel between the inner and the outer wall are shown.

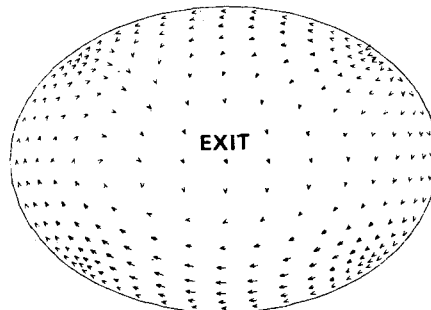
Figure 12(a) illustrates an example of the triple-swirl pattern in the elliptical duct near the entrance at  $Re = 10^3$ . Here, because of the absence of the centre duct, the stream approaching from the left-hand side is much stronger than in the previous case. Another point of interest is that the upcoming stream entering the elliptical duct directly from below is also more massive. The three currents are almost of the same strength, resulting in a triple-swirl flow. The swirling is greatly dissipated along the duct. Figure 12(b) shows the remaining small swirling vectors at the duct exit. A steady-state, turbulent flow solution for an HGM with two elliptical transfer ducts at  $Re = 10^5$  has been obtained.<sup>35</sup> In Figure 13 the swirling flow in the elliptical transfer duct is illustrated. In comparing these results with the laminar solution in Figure 12, it is seen that only a double-swirl pattern exists.

The most significant aspect of the present study is to pinpoint the locations where the flow experiences the most energy losses. An important measure of the energy losses is the mass-weighted average total pressure along the flow. Figure 14 illustrates the decreasing coefficient of the mass-weighted total pressure along the centreline of the TAD, the fuel bowl and the transfer duct. The total pressure coefficient  $C_{p0}$  is defined as

$$C_{p0} = \frac{\bar{p}_0 - \bar{p}_{01}}{\bar{p}_{01}},$$

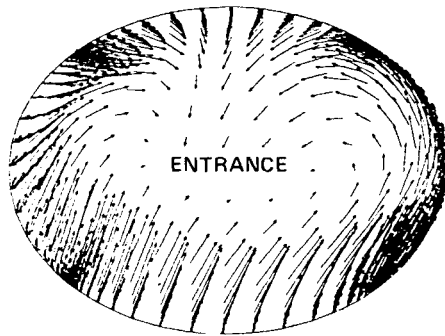


(a)

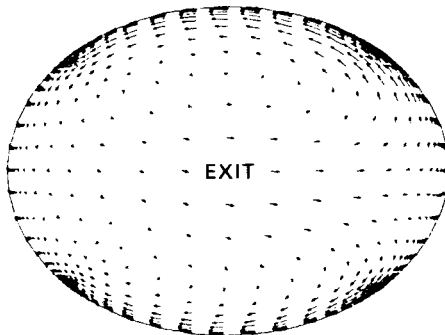


(b)

Figure 12



(a)



(b)

Figure 13

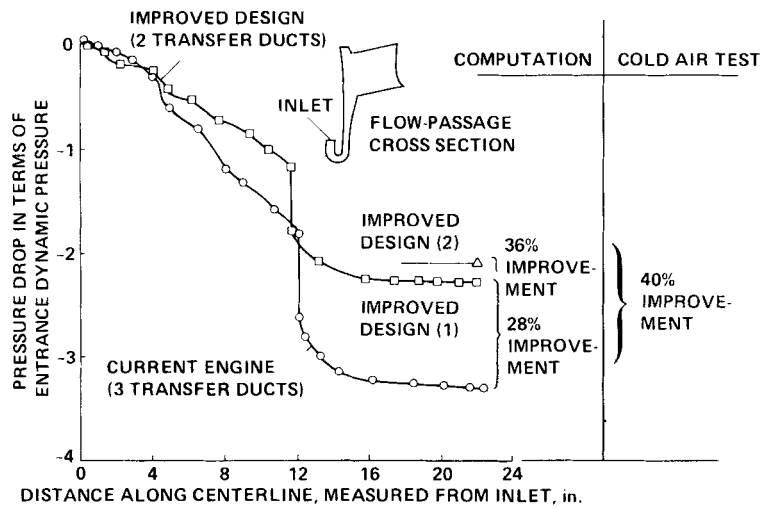


Figure 14

where

$$\bar{p}_0 = \frac{1}{M} \int [P + \frac{1}{2}(u^2 + v^2 + w^2)] dm.$$

The discontinuities shown in the figure correspond to the entrance of the duct where energy fluxes are computed over different planes. In the figure, three different HGM configurations are compared. The initial two-duct design shows 28% less total pressure drop compared to the current three-duct version. After fine-tuning the two-duct configuration computationally, the pressure drop decreased even further to 36% less than the original configuration. This final configuration is then tested using cold airflow, which shows 40% reduction in pressure loss.

### CONCLUDING REMARKS

This paper presents a summary of incompressible Navier–Stokes flow solver development work. The SSME power head has been simulated using this flow solver. Computational results compare favourably with test data and offer information not readily available from experiments. The results show that CFD can reduce the development time and cost by suggesting the best possible configurations for final verification by experiments. For example, in redesigning the TAD, over 20 different configurations were studied computationally, thus providing the best geometry for designers. Further study is in progress, and the total performance improvement will be compared in the future. At the time the present SSME analysis was performed, the computational model of the power head was designed to obtain solutions within a reasonable turnaround time. Therefore the total number of grid points was limited and the model could include the TAD and transfer ducts only. Despite its limitation, the present application provides an excellent example of how the present CFD and computer capabilities can be integrated into the aerospace design process.

### ACKNOWLEDGEMENT

This work is partially sponsored by the NASA Marshall Space Flight Center.

### REFERENCES

1. D. Kwak, J. L. C. Chang, S. P. Shanks and S. Chakravarthy, 'A three-dimensional incompressible Navier–Stokes flow solver using primitive variables', *AIAA J.*, **24**(3), 390–396 (1986).
2. J. L. C. Chang and D. Kwak, 'On the method of pseudo compressibility for numerically solving incompressible flows', *AIAA Paper 84-0252*, Reno, Nevada, 1984.
3. S. E. Rogers, J. L. C. Chang and D. Kwak, 'A diagonal algorithm for the method of pseudocompressibility', *AIAA Paper 86-1060*, Atlanta, Georgia, May 1986; also *J. Comput. Phys.*, **73**, 364–379 (1987).
4. F. H. Harlow and J. E. Welch, 'Numerical calculation of time-dependent viscous incompressible flow with free surface', *Phys. Fluids*, **8**(12), 2182–2189 (1965).
5. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Co., New York, 1980.
6. G. D. Raithby and G. E. Schneider, 'Numerical solution of problems in incompressible fluid flow: treatment of the velocity–pressure coupling', *Numer. Heat Transfer*, **2**, 417–440 (1979).
7. A. J. Chorin, 'Numerical solution of Navier–Stokes equations', *Math. Comput.* **22**(104), 745–762 (1968).
8. N. N. Yanenko, *The Method of Fractional Steps*, Springer-Verlag, Berlin, 1971.
9. G. M. Marchuk, *Methods of Numerical Mathematics*, Springer-Verlag, 1975.
10. H. S. Dwyer, M. Soliman and M. Hafez, 'Time accurate solutions of the Navier–Stokes equations for reacting flows', *Proc. 10th Int. Conf. on Numerical Methods in Fluid Dynamics*, Beijing, China, June 1986, Springer-Verlag, pp. 247–251.
11. J. Kim and P. Moin, 'Application of a fractional-step method to incompressible Navier–Stokes equations', *J. Comput. Phys.* **59**(2), 308–323 (1985).
12. S. V. Patankar and D. B. Spalding, 'A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows', *Int. J. Heat Mass Transfer*, **15**, 1787–1806 (1972).
13. S. A. Orszag, M. Israeli and M. O. Deville, 'Boundary conditions for incompressible flows', *J. Sci. Comput.* **1**, 75–111 (1986).

14. P. M. Gresho and R. L. Sani, 'On pressure boundary conditions for the incompressible Navier–Stokes equations', *Int. j. numer. methods Fluids*, **7**, 1111–1145 (1987).
15. M. Rosenfeld, D. Kwak and M. Vinokur, 'A solution method for unsteady, incompressible Navier–Stokes equations in generalized coordinate systems', *AIAA Paper 88-0718*, *AIAA 26th Aerospace Sciences Meeting*, Reno, Nevada, 11–14 January 1988.
16. A. J. Chorin, 'A numerical method for solving incompressible viscous flow problems', *J. Comput. Phys.*, **2**, 12–26 (1967).
17. R. M. Beam and R. F. Warming, 'An implicit finite-difference algorithm for hyperbolic systems in conservation-law form', *J. Comput. Phys.*, **22**, 87–110 (1976).
18. W. R. Briley and H. McDonald, 'Solution of the three-dimensional compressible Navier–Stokes equations by an implicit technique', *Proc. Fourth Int. Conf. on Numerical Methods in Fluid Dynamics; Lecture Notes in Physics*, Vol. 35, Springer-Verlag, New York, 1975, pp. 105–110.
19. W. R. Briley and H. McDonald, 'On the structure and use of block implicit schemes', *J. Comput. Phys.*, **334**, 54–73 (1980).
20. J. L. Steger and P. Kutler, 'Implicit finite-difference procedures for the computation of vortex wakes', *AIAA J.*, **15**(4), 581–590 (1977).
21. D. Choi and C. L. Merkle, 'Application of time-iterative schemes to incompressible flow', *AIAA J.*, **23**(10), 1518–1524 (1985).
22. C. L. Merkle and M. Athavale, 'Time-accurate unsteady incompressible flow algorithms based on artificial compressibility', *AIAA Paper 87-1137*, *AIAA 8th Computational Fluid Dynamics Conf.*, Honolulu, Hawaii, 9–11 June 1987.
23. J. Strigberger, 'Relationship between pseudocompressible and unsteady compressible flow at low Mach numbers', *AIAA J.*, **25** (11), 163–165 (1987).
24. W. R. Briley, H. McDonald and S. J. Shamroth, 'A low Mach number Euler formulation and application to time-iterative LBI schemes', *AIAA J.*, **21** (10), 1467–1469 (1983).
25. R. Temam, *Navier Stokes Equations*, Revised edn, North-Holland, 1979.
26. T. H. Pulliam, 'Artificial dissipation models for the Euler equations', *AIAA Paper 85-0438*, *AIAA 23rd Aerospace Sciences Meeting*, Reno, Nevada, 14–17 January 1985.
27. R. F. Warming, R. M. Beam and B. J. Hyett, 'Diagonalization and simultaneous symmetrization of the gas-dynamic matrices', *Math. Comput.*, **29**, 1037–1045 (1975).
28. E. Turkel, 'Symmetrization of the fluid dynamic matrices with applications', *Math. Comput.*, **27**, 729–736 (1973).
29. T. H. Pulliam and D. S. Chaussee, 'A diagonal form of an implicit approximate-factorization algorithm', *J. Comput. Phys.*, **39**, 347–363 (1981).
30. J. Flores, 'Convergence acceleration for a three-dimensional Euler/Navier–Stokes zonal approach', *AIAA Paper 85-1495*, Cincinnati, Ohio, July 1985.
31. S. E. Rogers, D. Kwak and U. Kaul, 'On the accuracy of the pseudocompressibility method in solving the incompressible Navier–Stokes equations', *AIAA Paper 85-1689*, *AIAA 18th Fluid Dynamics and Plasmadynamics and Laser Conf.* Cincinnati, Ohio, 16–18 July 1985.
32. U. Kaul, D. Kwak and C. Wagner, 'A computational study of saddle point separation and horseshoe vortex system', *AIAA Paper 85-0182*, Reno, Nevada, January 1985.
33. D. Kwak, S. E. Rogers, U. K. Kaul and J. L. C. Chang, 'A numerical study of incompressible juncture flows', *Tenth Int. Conf. on Numerical Methods in Fluid Dynamics*, Beijing, China, 23–27 June 1986.
34. J. L. C. Chang, D. Kwak and S. C. Dao, 'A three dimensional incompressible flow simulation method and its application to the Space Shuttle main engine. Part I. Laminar flow', *AIAA Paper 85-0175*, Reno, Nevada, January 1985.
35. J. L. C. Chang, D. Kwak, S. C. Dao and R. Rosen, 'A three-dimensional incompressible flow simulation method and its application to the Space Shuttle main engine. Part II. Turbulent flow', *AIAA Paper 85-1670*, *AIAA 18th Fluid Dynamics and Plasmadynamics and Laser Conf.*, Cincinnati, Ohio, 16–18 July 1985.
36. B. S. Baldwin and H. Lomax, 'Thin layer approximation and algebraic model for separated turbulent flows', *AIAA Paper 78-257*, Huntsville, Alabama, January 1978.
37. J. L. C. Chang, R.-J. Yang and D. Kwak, 'Numerical study of turbulent internal layer flow in an axisymmetric diffusion U-duct', *AIAA Paper 88-0718*, *AIAA 26th Aerospace Sciences Meeting*, Reno, Nevada, 11–14 January 1988.
38. J. H. Ferziger, 'Incompressible turbulent flows', *J. Comput. Phys.*, **69**, 1–48 (1987).
39. W. C. Reynolds, 'Computation of turbulent flows', *Ann. Rev. Fluid Mech.*, **8**, 183–208 (1976).