Journal of Computational Physics 228 (2009) 4836–4857

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

# Solution-limited time stepping to enhance reliability in CFD applications

Chenzhou Lian *, Guoping Xia, Charles L. Merkle

*School of Mechanical Engineering, Purdue University, ARMS 3142, 701 W. Stadium Ave., West Lafayette, IN 47907-2045, United States*

ARTICLE INFO

*Article history:*
Received 14 August 2008
Received in revised form 25 March 2009
Accepted 26 March 2009
Available online 7 April 2009

*MSC:*
65B99
76M12

*Keywords:*
Computational fluid dynamics
Numerical methods
Convergence and reliability

ABSTRACT

A method for enhancing the reliability of implicit computational algorithms and decreasing their sensitivity to initial conditions without adversely impacting their efficiency is investigated. Efficient convergence is maintained by specifying a large global Courant (*CFL*) number while reliability is improved by limiting the local *CFL* number such that the solution change in any cell is less than a specified tolerance. The method requires control over two key issues: obtaining a reliable estimate of the magnitude of the solution change and defining a realistic limit for its allowable variation. The magnitude of the solution change is estimated from the calculated residual in a manner that requires negligible computational time. An upper limit on the local solution change is attained by a proper non-dimensionalization of variables in different flow regimes within a single problem or across different problems. The method precludes unphysical excursions in Newton-like iterations in highly non-linear regions where Jacobians are changing rapidly as well as non-physical results such as negative densities, temperatures or species mass fractions during the computation. The method is tested against a series of problems all starting from quiescent initial conditions to identify its characteristics and to verify the approach. The results reveal a substantial improvement in convergence reliability of implicit CFD applications that enables computations starting from simple initial conditions without user intervention.



## 1. Introduction

Computational algorithms for systems of partial differential equations traditionally use iterative methods to obtain numerical solutions. The efficiency and reliability of such procedures represent crucial issues for practical applications. An implicit goal of any algorithm is to maximize computational efficiency while minimizing user intervention and the specification of user defined control variables at input. An ideal algorithm should converge reliably from any physically meaningful initial condition while using a standard set of predefined control variables. For example, an ideal time-marching procedure would use a fixed Courant (*CFL*) number for every problem and converge reliably from any easily specified initial condition to the correct solution without user intervention. Such an ideal is far from conventional experience. Iterative procedures are sensitive to initial conditions, are adversely affected by grid stretching and cell aspect ratios and by the nonlinearities that are characteristic of fluid mechanics. Issues such as these combine to require significantly different *CFL* values for various problems and often require changes in *CFL* during the iteration to preclude divergence and ensure efficiency. Without watchful user intervention, solutions often diverge or 'blow up' as nonlinearities cause local Jacobians to drive the solution in improper directions or produce unrealistic physical variables such as negative temperatures, densities or mass fractions during the iteration.

While convergence acceleration has been the subject of many studies [1–7], methods for ensuring code reliability generally rely upon *ad hoc* procedures. One common method for enhancing reliability is to add increased artificial dissipation to

* Corresponding author. Tel.: +1 765 426 8695; fax: +1 765 494 0307.
*E-mail address:* clian@purdue.edu (C. Lian).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.03.040

**Nomenclature**

| | |
|---|---|
| $\mathcal{F}$ | inviscid flux vector |
| $\mathcal{F}_v$ | viscous flux vector |
| $H$ | source term |
| $\tau_{ij}$ | Reynolds stress tensor |
| $\delta_{ij}$ | Kronecker delta |
| $\mu$ | molecular viscosity |
| $h^0$ | total enthalpy |
| $\Gamma$, $\Gamma_p$ | matrix coefficient |
| $Q$ | conservative transport variables |
| $Q_p$ | primitive transport variable |
| $\widehat{Q}$ | characteristic transport variable |
| $\tau$ | pseudo time |
| $u$, $v$, $w$ | Cartesian velocity component |
| $x$, $y$, $z$ | Cartesian coordinate |
| $i$ | grid point index |
| $CFL$ | Courant–Friedrich–Levy number |
| $A$, $A_{p,i}$, $A_{p,v}$ | Jacobian matrices |
| $\rho$ | density |
| $p$ | static pressure |
| $p^0$ | stagnation pressure |
| $T$ | temperature |
| $T^0$ | stagnation temperature |
| $u'$, $v'$, $w'$ | fluctuation of velocity in $x$, y and $z$ direction |
| $I$ | identity matrix |
| $U$ | initial velocity magnitude |
| $Re$ | Reynolds number |
| $\beta$ | flow angle |
| $Ma$ | Mach number |
| $\Delta Q_p^{est}$ | estimated solution change |
| $\Delta Q_p$ | solution change |
| $Q_{pref}$ | reference variable |
| $\alpha$ | max allowable fractional change in solution |

the algorithm in the early stages of the computation as, for example, starting with a first-order procedure and then switching to a higher-order scheme as the solution nears convergence. The first-order algorithm, however, may not be sufficiently dissipative to ensure that the initial condition leads to convergence and the choice of when to transition to the desired order of accuracy is unknown. Further, even a converged first-order solution may be too far from the final solution to guarantee convergence of the higher-order scheme.

A second common procedure for ensuring reliability is to ramp the *CFL* during the solution process. The iteration is started from a small *CFL* and increased to an upper limit based on past experience. Again, the starting value, the final value and the allowable rate of increase are all unknown and can vary dramatically from problem to problem. The alternative of using small *CFL* numbers throughout the iteration, in principle, enhances reliability, but results in inordinately long computation time. The choice of an *a priori CFL* schedule, however, remains an open issue and values are generally set arbitrarily as the literature on optimal strategies for *CFL* specification is limited [4–7].

A third common procedure is to reset positive-definite variables (such as density or temperature) to a small positive value when computational excursions cause them to change sign. While this action precludes divide checks and *NaN's*, such arbitrary changes often interact adversely with the convergence process requiring increased numbers of iterations. Further, resetting variables does not remove the underlying cause and negative quantities generally reappear in succeeding time steps.

Often, all three of the above methods are combined with other similar techniques to try to improve code reliability. A major limitation is that all are performed in an 'open-loop' manner—the correction is devised before the problem appears, and is implemented without regard for why or when the difficulty appears.

The goal of the present article is to investigate a 'closed-loop' procedure for improving algorithm reliability that controls the local *CFL* during the iteration on the basis of the convergence process itself rather than by means of an *a priori* procedure. The method chosen is based upon a concept suggested by Luke et al. [8], but differs from their approach in that it requires negligible computational overhead and has been extended to improve its effectiveness. The objectives are to control the change of all variables at every grid point and iteration in such a manner that a single, pre-specified value of *CFL* can be used

for all computations while simultaneously reducing sensitivity to initial conditions and grid distribution and ensuring that non-physical values are not encountered. The closed-loop control is achieved by specifying a maximum solution change that allows the code to determine a local *CFL* in such a manner that this tolerance is not exceeded. Appropriate limits on the fractional change of the solution should ensure that the linearization procedure for converting the differential equations to algebraic form will remain valid so that Newton-based convergence processes do not diverge because of large changes in Jacobians and that positive-definite quantities cannot change sign. The approach allows a large *CFL* in non-sensitive regions of the convergence process while restricting the *CFL* in rapidly changing regions. We call this control method the solution-limited time stepping method. Our goal is to demonstrate the characteristics of this automatic method for determining *CFL* and its impact on code reliability.

In the following sections we first summarize the conservation equations of fluid mechanics along with the representative discretization procedure that is used for the examples. We then describe the solution-limited time stepping procedure and apply it to a scalar problem and the one-dimensional fluid equations followed by results for a series of problems of increasing difficulty to assess the effectiveness and generality of the method. We close with a summary of the findings.

## 2. Governing equations and discretization method

Although the solution-limited time stepping procedure presented here is applicable to any implicit computational algorithm, for definiteness we specify the equations of motion and the solution method that are used in the examples. The Navier–Stokes equations can be written as:

$$\frac{\partial Q}{\partial \tau} + \nabla \cdot (\mathcal{F} - \mathcal{F}_v) = 0 \tag{1}$$

where $Q = (\rho, \rho u, \rho v, \rho h^0 - p)^T$ is the conservative variables vector, $\tau$ is a pseudo time used for iterative solution and $\mathcal{F}$ and $\mathcal{F}_v$ are the inviscid and viscous flux vectors that each include three components:

$$\mathcal{F} = E\vec{i} + F\vec{j} + G\vec{k} \quad \text{and} \quad \mathcal{F}_v = E_v\vec{i} + F_v\vec{j} + G_v\vec{k} \tag{2}$$

The inviscid flux vectors are:

$$E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uh^0 \end{pmatrix}, \quad F = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho vh^0 \end{pmatrix}, \quad G = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho wh^0 \end{pmatrix} \tag{3}$$

while the viscous flux vectors are:

$$E_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx} + v\tau_{yx} + w\tau_{zx} + kT_x \end{pmatrix}, \quad F_v = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{zy} + kT_y \end{pmatrix}, \quad G_v = \begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + kT_z \end{pmatrix} \tag{4}$$

The quantities, $\tau_{ij}$, represent the nine components of the stress tensor:

$$\tau_{ij} = -\frac{2}{3}\mu(u_x + v_y + w_z)\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{5}$$

and, $x$, $y$, and $z$, are the Cartesian coordinates.

The conservation equations must be augmented by appropriate constitutive relations for the fluids of interest to obtain a closed set. A general Gibb's function, $g(p, T)$ provides a convenient constitutive relation for a broad range of fluids problems from which the density and enthalpy can be found [9]. In addition, the fluid viscosity, $\mu(p, T)$, and thermal conductivity, $k(p, T)$, are likewise specified as arbitrary functions of pressure and temperature.

Numerical solutions to Eq. (1) require an appropriate discretization procedure and a suitable iterative scheme. We use finite volume methods to discretize the spatial derivatives and solve the resulting algebraic system by treating the pseudo-time derivative in Eq. (1) as an iterative variable that can be marched from a selected initial condition to a converged final result. Because $\partial Q/\partial\tau$ vanishes as pseudo time goes to infinity and so has no impact on the final solution it can be expressed in non-conservative form without impacting the final solution. Accordingly we use the chain rule to transform from conservative variables, $Q$, to the primitive variables, $Q_p = (p, u, v, w, T)^T$ so that Eq. (1) becomes:

$$\Gamma_p \frac{\partial Q_p}{\partial \tau} + \nabla \cdot (\mathcal{F} - \mathcal{F}_v) = 0 \tag{6}$$

The pressure and temperature in the primitive variables complement the variables in the constitutive relations while the Jacobian matrix, $\Gamma_p$, provides a transformation between $Q$ and $Q_p$. Although the physical Jacobian can be used, both solution accuracy and convergence efficiency indicate that artificially defined matrices, $\Gamma_p$, are sometimes preferable. Although such artificial matrices provide improvements in solution accuracy and convergence in general computations, they can also increase sensitivity to initial conditions. The methods described herein decrease sensitivity to initial conditions considerably for applications with either physical or artificial pseudo-time coefficients, $\Gamma_p$.

The finite volume discretization of the inviscid and viscous divergence operators is considered separately. The divergence operator for the inviscid flux, $\nabla \cdot \mathcal{F}$, can be discretized by central differences [10–12], upwind flux-vector splitting [13–16] or flux-difference splitting [17–19]. Our examples use an upwind, flux-difference splitting method based upon Gauss' theorem with the fluxes on each face determined by an approximate Riemann solver. After integrating the continuous divergence operator, $\nabla \cdot \mathcal{F}$, over a volume, $\Omega$, a discrete divergence operator, $\nabla_D \cdot \mathcal{F}$, for the inviscid fluxes can be defined as:

$$\nabla_D \cdot \mathcal{F} = \frac{1}{\Omega} \sum_{k=1}^{K} \left[ \frac{1}{2} (\mathcal{F}_L + \mathcal{F}_R) - \frac{1}{2} \Gamma_p \left| \Gamma_p^{-1} A_{p,i} \right| (Q_{pR} - Q_{pL}) \right]_k S_k \tag{7}$$

In this expression, the flux Jacobian is defined as $A_{p,i} = \partial \mathcal{F} / \partial Q_p$, the vectors, $\mathcal{F}_L$ and $\mathcal{F}_R$, represent the normal components of the flux crossing the $k$th face, and $S_k$ is the face area. The second term in Eq. (7) is an artificial dissipation term introduced in the approximate Riemann procedure that depends upon the matrices $\Gamma_p$ and $A_{p,i}$. This matrix $\Gamma_p \left| \Gamma_p^{-1} A_{p,i} \right|$ must be defined carefully to ensure a proper amount artificial dissipation. The preconditioning $\Gamma_p$ developed by Turkel [20], Van Leer [21], Viviand [22], Briley [23], or Merkle [24] can ensure an appropriate amount of artificial dissipation. The physical matrix $\Gamma_p$ is given by:

$$\Gamma_p = \begin{pmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ u\rho_p & \rho & 0 & 0 & u\rho_T \\ v\rho_p & 0 & \rho & 0 & v\rho_T \\ w\rho_p & 0 & 0 & \rho & w\rho_T \\ h^0\rho_p - (1 - \rho h_p) & \rho u & \rho v & \rho w & h^0\rho_T + \rho h_T \end{pmatrix}$$

For the examples shown below, the physical quantity, $\rho_p$, in this matrix was replaced by an artificial property term, $\rho'_p = \frac{1}{V_p^2} - \frac{\rho_T (1 - \rho h_p)}{\rho h_T}$, and the preconditioned velocity scale $V_p$ is defined as, $V_p = \min\left[ \max\left( \sqrt{u^2 + v^2 + w^2}, \frac{\mu}{\rho \Delta x}, \frac{\mu}{\rho \Delta y}, \frac{\mu}{\rho \Delta z} \right), c \right]$, where $c$ is the sound speed. Note that the flux Jacobian couples the discretization with the pseudo-time parameter of the iterative procedure.

The divergence operator for the viscous fluxes is discretized by a Galerkin procedure that results in a discrete viscous divergence operator given as:

$$\nabla_D \cdot \mathcal{F}_v = \frac{1}{\Omega} \sum_{k=1}^{K} (\mathcal{F}_v)_k \cdot S_k \tag{8}$$

To complete the setup of the iterative procedure we replace the pseudo-time operator in Eq. (6) by an Euler implicit difference and replace the remaining terms by their discrete forms from Eq. (7) to yield the final algebraic system:

$$\Gamma_p \frac{Q_p^{k+1} - Q_p^k}{\Delta \tau} = -\{ \nabla_D \cdot (\mathcal{F} - \mathcal{F}_v) \}^{k+1} \tag{9}$$

Here, the superscript $k$ represents the running variable in pseudo time.

To solve Eq. (9), we linearize the terms on the right side at the new pseudo-time level by Taylor's series expansions:

$$(\nabla_D \cdot \mathcal{F})^{k+1} \approx (\nabla_D \cdot \mathcal{F})^k + \nabla A_{p,i} \cdot \Delta Q_p \tag{10a}$$

and

$$(\nabla_D \cdot \mathcal{F}_v)^{k+1} \approx (\nabla_D \cdot \mathcal{F}_v)^k + \nabla A_{p,v} \cdot \Delta Q_p \tag{10b}$$

where the Jacobians are defined as: $A_{p,i} = \partial \mathcal{F} / \partial Q_p, A_{p,v} = \partial \mathcal{F}_v / \partial Q_p$ and $\Delta Q_p \equiv Q_p^{k+1} - Q_p^k$ where the flux Jacobian is defined as $A_p = A_{p,i} - A_{p,v}$. Upon rearranging, we obtain the final algebraic system,

$$\left\{ I + \Delta \tau \Gamma_p^{-1} (\nabla \cdot A_p) \right\}^k \Delta Q_p = -\Delta \tau \left[ \Gamma_p^{-1} \{ \nabla_D \cdot (\mathcal{F} - \mathcal{F}_v) \} \right]^k \tag{11}$$

The implicit operator in Eq. (11) is linear but requires a matrix inversion at each pseudo-time step. Because of its wide-banded nature in multi-dimensional problems, it is generally solved by approximate methods such as alternating direction implicit (ADI) [25,26], point Gauss–Seidel [27–29], line Gauss–Seidel (LGS) [30,31] or GMRES [32]. Here, we use a line Gauss–Seidel method. Because our implementation is based on an unstructured format, a least squares method is used to calculate the gradients of the primitive variable which are used for the reconstruction on the interface between adjacent cells.

## 3. The solution-limited time stepping method

As intimated above, large *CFL* values are needed to ensure *efficient* convergence, but small *CFL*'s are needed to ensure *reliable* convergence. In general, larger *CFL*'s can be used as the iteration nears convergence, while smaller values are imperative when it is far from the final solution. An important aspect of controlling solution reliability without adversely impacting efficiency is to recognize that at a given time in the iteration, some portions of the flowfield will be changing rapidly (and, hence, will need careful protection against divergence and unphysical results) while other portions are changing slowly (and accordingly can and should be advanced aggressively). The goal of the present work is to devise a means for monitoring the solution change at every grid point and every step in an effort to determine local *CFL* values that maximize the progress toward convergence while simultaneously minimizing the likelihood of divergence. The method involves two important concepts: first, an efficient means must be identified for estimating how large the solution change will be, and second, an acceptable magnitude for the allowable solution change must be established. These two items are discussed separately below.

### 3.1. The estimated solution change

An efficient estimate of the manner in which the solution change, $\Delta Q_p$, varies with the local time step size is imperative. Defining acceptable values of $\Delta\tau$ at each grid point by computing a series of implicit time steps is computationally intractable since the preferred $\Delta\tau$ values will be different at every grid point and the change at any grid point will be dependent on the change at neighboring cells as well as local values. Following the general spirit of the method suggested by Luke et al. [8], we set the estimated solution change (which we designate as $\Delta Q_p^{est}$) equal to the residual of the implicit update relation in Eq. (11):

$$\Delta Q_p^{est} = -\Delta\tau\left[\Gamma_p^{-1}\{\nabla_D\cdot(\mathcal{F}-\mathcal{F}_v)\}\right]^k \tag{12}$$

Thus, $\Delta Q_p^{est}$ is obtained by omitting the divergence terms on the left side of Eq. (11) and is identical to the update relation that would be obtained from an Euler explicit discretization of Eq. (6). Because the residual must be computed as part of the implicit algorithm, the computation of $\Delta Q_p^{est}$ requires no additional manipulations or iterations. Thus, in contrast to the procedure used by Luke et al. [8] whose procedure required two iterations and was relatively expensive to employ, our estimate has a negligible impact on the overall CPU time.

The estimated solution change in Eq. (12) is proportional to the time step, $\Delta\tau$. Consequently, for any cell in which $\Delta Q_p^{est}$ is deemed large enough to cause convergence difficulties, the local time step is reduced to bring $\Delta Q_p^{est}$ to an acceptable magnitude. This time-step-limiting procedure will be useful only to the extent that $\Delta Q_p^{est}$ provides a reasonable bound on $\Delta Q_p$. The adequacy of this estimate must be verified, but as a first step in this direction, we compare relative magnitudes of $\Delta Q_p^{est}$ and $\Delta Q_p$ by combining Eqs. (11) and (12) to obtain:

$$\Delta Q_p^{est} = \left\{I+\Delta\tau\Gamma_p^{-1}(\nabla_D\cdot A_p)\right\}^k\Delta Q_p \tag{13}$$

For small pseudo-time steps, $\Delta\tau$, the second term in the brackets are negligible and the actual calculated and estimated solution changes are approximately equal, $\Delta Q_p^{est}\approx\Delta Q_p$. For large $\Delta\tau$, the estimated change, $\Delta Q_p^{est}$, increases linearly without bound (see Eq. (12)) whereas the actual change, $\Delta Q_p$, approaches a constant value. The behavior at these two limits suggests that the estimated solution change may serve as a nominal upper bound on the calculated change for all $\Delta\tau$ and that the norm of $\Delta Q_p$ is bounded by the norm of $\Delta Q_p^{est}$. This supposition must however be verified. In Section 4, we use both analytical and numerical methods to demonstrate the effectiveness of $\Delta Q_p^{est}$ as a useful estimator of the calculated solution change, $\Delta Q_p$.

### 3.2. Setting the allowable solution change

As a formal procedure for defining the allowable magnitude of the solution change we introduce a reference vector, $Q_p^{ref}$ and a scalar multiplier, $\alpha$. We then limit the local pseudo-time step in such a manner that the estimated solution change, $\Delta Q_p^{est}$, will be less than the product of $\alpha$ and $Q_p^{ref}$:

$$\left|\Delta Q_p^{est}\right| \leqslant \alpha Q_p^{ref} \tag{14}$$

This limit is to be applied to each variable at each grid point with the local time step being chosen to satisfy the most restrictive variable. The effectiveness of Eq. (14) depends upon defining appropriate values for both the reference vector and the scalar multiplier. If the method is to be useful, it is imperative that the definitions for both these quantities be applicable over widely varying flow regimes.

Picking a value for $\alpha$ is quite straightforward, whereas the reference quantities require more consideration. For positive-definite quantities such as pressure, density or temperature that are not allowed to change sign, positivity is ensured by choosing $\alpha$ less than unity. For example, restricting the solution to a 10% change ($\alpha = 0.1$) will ensure positivity, while also

eliminating strong non-linear effects in Newton-based iteration procedures by precluding large changes in Jacobians from step to step while still enabling efficient convergence. Because $\Delta p$ and $\Delta T$ are only estimates of the change, sign changes are still possible but small values of $\alpha$ provide a substantial margin of safety and we have not observed sign changes for a wide variety of problems.

Indefinite variables such as velocity that must be allowed to change sign require a scaling factor greater than unity, or a reference value larger than that in the local cell. As discussed below, our results have shown that pressure is the most sensitive variable, whereas velocity is relatively insensitive.

Having defined representative values for the scaling constant, $\alpha$, we next consider methods for defining the reference variables with special emphasis on the pressure.

### 3.3. Scaling the reference variables

The reference vector, $Q_p^{ref}$, should be based upon local conditions in the computational domain as defined by the current iteration. A first choice is to set the reference vector for any cell equal to the current solution vector in that cell, $Q_p^{ref} = Q_{p,k}^n$, where $k$ refers to the local cell, and $n$ refers to the present time step. Of the pressure, velocity and temperature, the pressure change is more sensitive and requires more care because the pressure gradient, rather than the pressure, appears in the equations of motion. For example, in incompressible flows, the thermodynamic pressure clearly would not serve as an appropriate reference variable. Thus, while the reference value for determining the maximum allowable pressure change should depend upon the local pressure in the given cell, this local solution must be scaled appropriately to incorporate the local flow physics in a proper manner. Improper normalization can allow solution changes that are too large (thus allowing divergence) or too small (thereby slowing convergence).

The reference pressure must be determined from the local pressure gradient which can be estimated from the momentum equation with a pseudo-time term added to represent the iterative procedure. The pressure gradient depends upon flow characteristics such as the local Mach number, the local Reynolds number, the local acceleration or the local body forces and is essentially independent of the magnitude of the thermodynamic pressure. A properly defined reference pressure should reflect these several variations in flow characteristics in a manner that is representative of the local physics. Using the one-dimensional momentum equation as an example,

$$\rho\frac{\partial u}{\partial \tau}+\frac{\partial \rho u^2}{\partial x}+\frac{\partial p}{\partial x}=\mu\frac{\partial^2 u}{\partial x^2}+\rho g \tag{15}$$

it is immediately seen that multiple terms can balance the pressure gradient. Pertinent scaling variables can be obtained from formal perturbation expansion methods [33]. For brevity, we here simply tabulate pertinent scalings in the representative limits.

For high Reynolds number flows, the convective term balances the pressure gradient, and the pressure change scales as the local dynamic pressure resulting in the reference pressure $p_{ref} = \rho u^2$. At low Reynolds numbers, the pressure gradient is balanced by viscous forces and the pertinent reference pressure is the local shear stress, $p_{ref} = \tau_{ij}$. For problems dominated by buoyancy or body force terms, the reference pressure depends upon the gravitational term, $p_{ref} = \rho g h$. Finally, the pressure change can be determined by local gradients in the solution giving rise to a reference pressure, $p_{ref} = \Delta p$, where $\Delta p$ represents the pressure gradient across the local cell. In a general computation, using the local reference pressure as the largest of these several values along with a choice of $\alpha$ less than unity, ensures small changes in the Jacobians and precludes sign changes in the pressure.

An appropriate reference temperature is generally somewhat easier to identify than the pressure. Like pressure, the thermodynamic temperature cannot change sign. In solutions attempted to date, using the local thermodynamic temperature as the reference condition has proven sufficient to ensure both positivity and insensitivity to initial conditions. Nevertheless, problems for which the local temperature gradient must also be included might be encountered. As a special check on temperature scaling, we have included an example in which the flowfield contains substantial amounts of heat transfer (the temperature changes by an order of magnitude) as noted later. Ensuring that pressure and temperature are positive and that their changes are restricted to a small fraction of their previous values also ensures that the density remains well-behaved.

The reference velocity is somewhat different in that the velocity must be allowed to change sign. The reference velocity can be based upon the local velocity in a given cell or upon a velocity change imposed by a pressure gradient across the cell as determined by the Method of Characteristics (MOC). For the former limit, we set $u_{ref} = \sqrt{u_i^2}$ for each local cell, while for the latter, we have, $u_{ref} = c\Delta p/\gamma p$. The MOC procedure clearly can lead to velocity changes. Accordingly, we take $\alpha = 0.1$ for the MOC limit, and $\alpha = 2$ for the local velocity limit to allow it change sign in non-zero velocity regions. However, regions in which the velocity is large, the local velocity limit may have little effect, but a carefully defined pressure limit appears to be sufficient to protect the solution change in such regions. To circumvent difficulties in cells in which the reference velocity is identically zero by either of these conditions, we have imposed a third reference velocity as $10^{-9}$ times some easily identifiable characteristic velocity in the global flowfield. Thus, for the velocity components, $u_i$, in cell $k$, we limit $|\Delta u_{i,k}| \leqslant 2\sqrt{u_{1,k}^2+u_{2,k}^2+u_{3,k}^2}$. Representative solutions given later critically test this limit by starting with initial conditions in which the entire velocity field is set to zero and the pressure is uniform. This proves to be a rather conservative limit when the velocity components are small, but as the solution approaches convergence (or for initial conditions where the velocity is

large) it can be overly lenient and allow large fluctuations in the Jacobians. The special attention to the pressure limit, however, offsets this wider limit on the velocity components and we have found this method to be successful in all solutions to date. The present paper also includes examples containing recirculation regions behind a backward facing step and heat addition in a pipe flow starting from rest to test for difficulties when the local velocity passes through zero during the iteration. Other computations (not included) based upon using a global characteristic velocity as the reference speed with the value of $\alpha$ set equal to 0.1 as for a positive-definite variable have also proven effective for applications in which such a velocity is easily discerned.

### 3.4. Summary of solution-limited time stepping procedure

The purpose of the limiting procedure is to define a local time step, $\Delta\tau$, at every grid point such that $\left|\Delta Q_p^{est}\right| \leqslant \alpha Q_p^{ref}$. As shown next, this limit effectively ensures that the L2-norm of the calculated solution change will be bounded by a similar condition, $|\Delta Q_p| \leqslant \alpha Q_p^{ref}$. Since $\Delta Q_p^{est}$ is equal to the residual of the implicit algorithm, it is already computed as a part of the calculated solution, and therefore requires negligible CPU resources. We implement the solution-limited time stepping method in the following manner:

1. Specify a global Courant number, $CFL_{specified}$, that is large enough to provide efficient convergence in the limit as the solution approaches convergence. (The value used in the examples is $CFL_{specified} = 1000$.) The $CFL$ is defined as:
$$CFL = \frac{\lambda \cdot \Delta t}{\Delta l}$$
where $\lambda$ is the maximum eigenvalue of the convective Jacobian, $\Delta l$ is the longest dimension of the cell and $\Delta t$ is the time step.
2. Choose the reference quantities, $Q_p^{ref}$, based upon local conditions:
$$p_{ref} = \max\left(\frac{\rho u^2}{2}, \mu\frac{u_i \Delta x_i}{\Delta x_j^2}, \rho g h, \Delta p, p_{global} \times 10^{-9}\right)$$
$$u_{ref} = \max\left(\sqrt{u_i^2}, \frac{c}{\gamma}\frac{\Delta p}{p}, 10^{-9} \times V_{global}\right)$$
$$T_{ref} = T$$
where $V_{global}$ is some global velocity chosen to ensure $u_{ref}$ is always greater than zero and $p_{global}$ is some global reference pressure chosen to ensure $p_{ref}$ is always greater than zero.
3. Set the maximum allowable fractional change, $\alpha$ (we have used $\alpha = 0.1$ for all variables except the velocity change based on the local velocity where we have used $\alpha = 2$).
4. Compute the residual of the implicit algorithm in Eq. (11) which gives the estimated solution change, $\Delta Q_p^{est}$ as noted in Eq. (12).
5. Compare $\Delta Q_p^{est}$ at every grid point with the allowable solution change, $\alpha Q_p^{ref}$, and for those cells that exceed the limit, decrease the local Courant number according to: $CFL_{allowable} = CFL_{specified}\alpha Q_p^{ref} \Big/ \left|\Delta Q_p^{est}\right|$. This limiting step takes advantage of the linearity of $\Delta Q_p^{est}$ with $\Delta\tau$ and is to be done for each component of $Q_p$ with the minimum $CFL$ being selected.
6. Use the resulting sequence of Courant numbers to compute the calculated solution change, $\Delta Q_p$, and complete the time step.

It is important to use the time step limiting procedure for every variable at every iteration and every cell in the computation, starting from the initial condition. The local pseudo-time step at every cell and grid point is based upon the smallest $\Delta\tau$ from all variables. As shown later, the limiting procedure is typically operative over small portions of the flowfield during limited parts of the convergence process. As the computation progresses, the entire field can typically be advanced at the global Courant number.

## 4. Bounding assessment of calculated and estimated solution changes

To understand the effectiveness of the estimated solution change as a bound for the implicit calculated change, we first look at solution changes for a scalar wave equation, and then for a one-dimensional vector system to compare the estimated solution change with the actual solution change.

### 4.1. Scalar wave equation

As a first step, we consider the calculated and estimated solution changes for the scalar wave equation,

$$\frac{\partial u}{\partial \tau} + a\frac{\partial u}{\partial x} = 0 \tag{16}$$

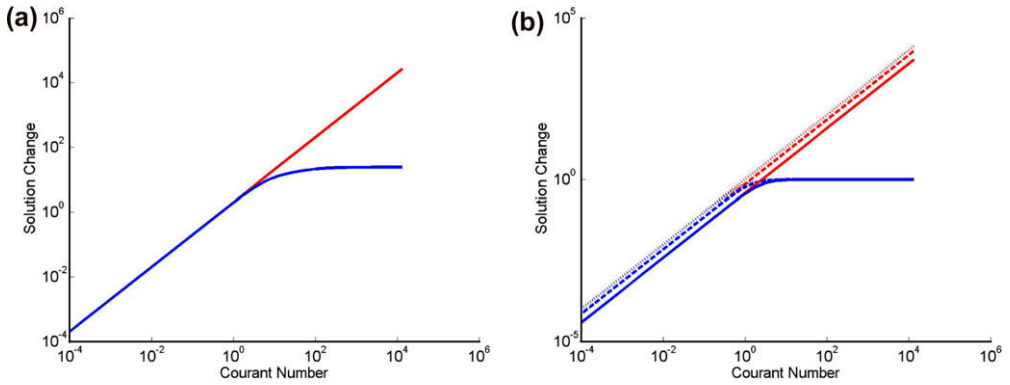


**Fig. 1.** Solution change obtained from numerical analysis (a, left) and from Fourier analysis (b, right). Both plots are for central differencing in time. Fourier modes on right are for $\phi = \pi/2$, $\pi/4$ and $\pi/8$. Estimated solution change: red; Solution change: blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 1**
Fourier analysis of discretized scalar equation for central differences

| Estimated solution change | Solution change |
|---|---|
| $u_i^{n+1} - u_i^n + \frac{a\Delta\tau}{2\Delta x}(u_{i+1} - u_{i-1})^n = 0$ | $u_i^{n+1} - u_i^n + \frac{a\Delta\tau}{2\Delta x}(u_{i+1} - u_{i-1})^{n+1} = 0$ |
| $(\Delta\hat{u})^{est} = -\hat{u}\frac{a\Delta\tau}{\Delta x} iS$ | $(\Delta\hat{u}) = -\hat{u}\frac{\frac{a\Delta\tau}{\Delta x} iS}{1+\frac{a\Delta\tau}{\Delta x} iS}$ |

as a function of the time step size for Courant numbers ranging from $10^{-4}$ to $10^4$. We consider a domain $0 \leqslant x \leqslant \pi$ with 50 evenly space grid points and an initial condition given by $u(x, 0) = \sin x$. The L2 norm of the solution change obtained from numerical solutions of Eq. (16) with central differencing in space are shown in the plot on the left of Fig. 1 as a function of the Courant number for the indicated range. The plot clearly shows that the L2 norm of the calculated solution is bounded by the L2 norm of the estimated solution for all Courant numbers. The calculated solution change is essentially equal to the estimated change for *CFL*'s up to approximately unity, while for values greater than unity the calculated solution change approaches a constant while the estimated solution change continues to increase linearly. Analogous findings have also been obtained for first-order upwind differencing.

As a second assessment of this simple equation, we consider a Fourier analysis that compares the ratio of the solution changes for specific modes in the solution. Table 1 gives the discretization and the stability analyses for both the explicit and implicit systems. In Table 1, the quantities, $u_i^n$, represents the numerical solution at grid point, $i$, in space at time level, $n$, and the change in the solution is defined as $\Delta u_i = u_i^{n+1} - u_i^n$. The amplitude of any Fourier mode is given by $\hat{u}$ with the superscripts *est* for estimated changes. Here we use the shorthand notation $S = \sin(2\pi/N)$ following the imaginary unit $i$, where $N$ is the Fourier mode. The results in the table clearly show that the estimated solution change, $\Delta u^{est}$, is a linear function of $\Delta\tau$, while the calculated solution change, $\Delta u$, is not. Plots of this Fourier function in Fig. 1(b) for mode numbers, $\pi/2$, $\pi/4$ and $\pi/8$, are analogous to the numerical results given in Fig. 1(a). Both the L2 norm comparisons and the Fourier transform comparisons indicate that for the scalar case the estimated solution change serves as an effective bound for the calculated solution change.

### 4.2. Comparison of scalar and vector system

The scalar results given above are also applicable to a vector system analogous to the Euler equations of fluid dynamics as is easily shown by a transformation to characteristic variables. The vector system:

$$\frac{\partial Q}{\partial \tau} + A\frac{\partial Q}{\partial x} = 0 \tag{17}$$

can be transformed to characteristics form by pre-multiplying by the modal matrix, $M$, whose columns correspond to the right eigenvectors of the Jacobian matrix, $A$, giving:

$$\frac{\partial \widehat{Q}}{\partial \tau} + \Lambda\frac{\partial \widehat{Q}}{\partial x} = 0 \tag{18}$$

where $\widehat{Q}$ represents the Riemann variable $d\widehat{Q} \equiv M^{-1}\,dQ$ and $\Lambda$ is the diagonal matrix whose elements are the eigenvalues of $A$. As in the scalar example, the Fourier transform of the estimated and calculated solution changes are:

$$\Delta\widehat{Q}^{est} = -iS\Lambda\frac{\Delta\tau}{\Delta x}\widehat{Q} \tag{19}$$

$$\Delta\widehat{Q} = -\left(I + iS\Lambda\frac{\Delta\tau}{\Delta x}\right)^{-1} iS\Lambda\frac{\Delta\tau}{\Delta x}\widehat{Q} \tag{20}$$

Since the matrix $\Lambda$ in Eq. (18) is a diagonal matrix, the scalar analyses of Section 4.1 are also valid for each component of the characteristic variable, $\widehat{Q}$, so the conclusion that the L2 norm of the calculated solution is bounded by the L2 norm of the estimated solution continues to hold for a system of equations.

Because the magnitude of the calculated solution change is not absolutely bounded by the estimated solution change but only by its L2 norm, there could (and will) be points in the domain for which the estimated change is identically zero while those for the calculated change are finite. This specific point is addressed in the following examples by using problems with a uniform initial condition throughout the domain except for a step change at the boundary. The estimated solution change is identically zero for all grid points except those adjacent to the boundary (the effect of the boundary condition propagates only one cell per time step), while the calculated solution change extends over the entire domain (boundary condition effects propagate over multiple cells in implicit scheme). Thus, in every example, we deal with multiple points for which the calculated solution change exceeds the estimated solution change, but the estimator still provides an effective limit on the iteration.

## 5. Results

To demonstrate the solution-limited time stepping procedure, we start with a simple, one-dimensional problem to test the general concept and assess its behavior. Following this, we present a series of multi-dimensional examples with an additional physical concept added in each problem to assess its effectiveness in various flow regimes. The initial condition for all cases sets the velocity at all cells in the field to zero and the pressure and temperature to a constant value. The initial density at every grid point is then obtained from the equation of state. Upstream boundary conditions typically correspond to a specified stagnation pressure, $p^0$, and temperature, $T^0$, and the flow angle. The downstream boundary condition is a specified back pressure, $p_{back}$. This initial condition is easy to define, simple to implement and broadly useful for a wide variety of problems.

### 5.1. One-dimensional inviscid flow

As the first step in developing the method, we consider the one-dimensional flow of a perfect gas in a constant area passage. The converged solution for this problem is a trivial uniform flow but it provides a useful first test of the method. The upstream boundary conditions were chosen as: $p^0$ = 1.4e5 Pa and $T^0$ = 400 K and a series of back pressures were imposed such that the Mach number of the converged flow ranged from low subsonic to supersonic speeds. The global *CFL* was set to 1000 for all cases and the limiting procedure was used to control the local *CFL* to see if it would respond appropriately to the initial condition and provide reliable convergence. There were 100 uniformly distributed cells in the domain. The geometry and flow conditions are shown in Fig. 2.

Two sets of constant values were used for the initial condition on pressure. In one case, the initial pressure was set equal to the upstream stagnation pressure while in the second it was set to the back pressure. With the first initial condition, the outflow boundary is exposed to the back pressure at the first time step and the iterative procedure is initiated by an expansion fan propagating into the computational domain from the downstream boundary in pseudo time. We refer to this as an expansion-initiated case. With the second initial condition, the upstream cell is exposed to the stagnation pressure at the first time step and the convergence process starts from a shock wave propagating through the flowfield. We describe this as the shock-initiated case. In all cases the initial velocity was set to zero and the temperature was set to the upstream stagnation value.

As a first step, we look in detail at the initial condition and the first several iterative steps using the expansion-initiated case as an example. From the initial condition, the velocity in all cells is zero and the pressure is equal to the upstream stagnation value at every cell except the last one which is equal to the back pressure. The estimated solution change returns a machine zero for $\Delta p^{est}$ for all cells except the last which returns a finite value. In the companion two-dimensional version of this problem (discussed below), the value of $\Delta p^{est}$ for the entire downstream column of cells was finite, while the $\Delta p^{est}$ for all remaining cells were round-off zero rather than a machine zero. For this reason, the final limiting terms was added to $p_{ref}$ definition and the reference pressure for all but the last cell was chosen as the nominal value, $10^{-9}p_{global}$ where $p_{global}$ was chosen as the stagnation pressure. The reference pressure in the last cell was limited by the pressure change across the cell, $\Delta p$ (which in this case was $p^0 - p_{back}$). The corresponding reference velocity was $10^{-9}V_{global}$ for all but the last cell where the reference velocity was set by the MOC limit. In performing the limiting procedure, the Courant number was


**Fig. 2.** Geometry and boundary conditions for constant area duct.

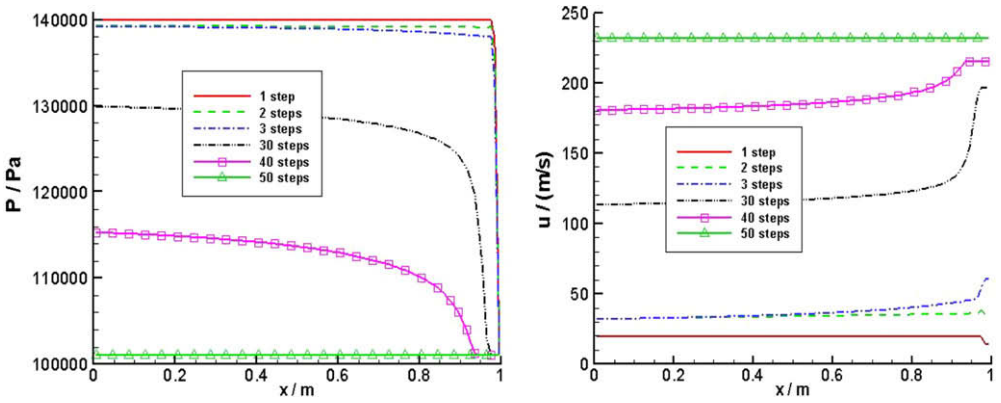


**Fig. 3.** Pressure (left) and velocity (right) profiles at various steps.

unlimited in all cells except the last where the velocity limit resulted in $CFL \approx 0.5$ for the first step. The resulting velocity and pressure after this first step are shown in Fig. 3. Note that the velocity change propagates through the entire domain (because of the $CFL = 1000$ condition at all cells except the last), while the pressure propagates more slowly.

The second step proceeds in similar fashion, except that the non-zero velocity field results in non-zero explicit solution changes for most cells. The reference velocity for step 2 was now based on the local velocity in nearly all cells whereas the pressure was still based on the nominal global change in all but the last several cells where it was generally based on the dynamic pressure. The solution for this step is also shown on Fig. 3, along with the companion solution at steps 3, 30, 40 and 50. It can be seen that the highly non-uniform initial condition at step 1 is quickly absorbed into the solution and that by limiting only the last few cells, physically meaningful implicit solution changes are obtained at all cells.

The overall convergence for this sequence of one-dimensional cases is given in Fig. 4 which shows the $L_\infty$ norm of the global convergence for a series of downstream pressures corresponding to converged Mach numbers of 2.2, 0.7, 0.3, 0.03, and 0.003. All results on Fig. 4 except for the $Ma = 2.2$ case are for the expansion-initiated initial condition. The shock-initiated initial condition was used for the supersonic case. For clarity only the pressure convergence is given. Other variables converge at similar rates. In general the convergence process is composed of an initial region where the $CFL$ values in a small fraction of the cells are limited so that $\Delta p/p_{ref}$ is approximately constant followed by a region in which the solution converges very rapidly and ending with a constant region where the solution has reached machine accuracy. The initial constant region corresponds to the iterations during which the limiter controls the time step at some points in the domain. In all cases, the velocity limit provided the dominant time step limit for the first one or two iterations, but the pressure limit quickly became the limiting criterion after that. Throughout the computation, $CFL$ limitations were restricted to the downstream end of the domain. The region of rapid convergence corresponds to the unlimited regime where the large Courant number provides effective convergence. This character of controlled convergence while the global characteristics of the solution are established, followed by a region of rapid convergence to machine accuracy is the expected goal of the solution-limited method. That goal is met quite effectively in these one-dimensional examples, providing confidence to move to multi-dimensions.

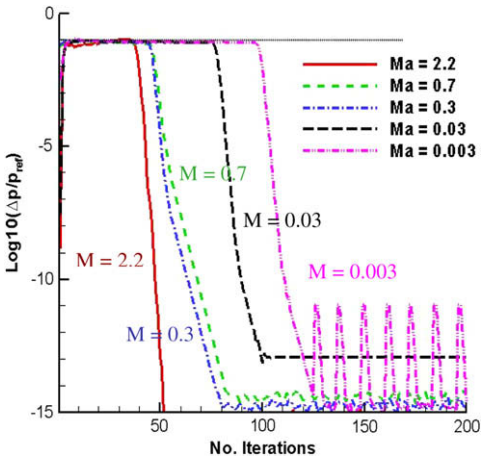


**Fig. 4.** $L_\infty$ norm of solution-limited convergence rate for one-dimensional flow in constant area duct for converged Mach numbers of 2.2, 0.7, 0.3, 0.03, and 0.003.

### 5.2. Two-dimensional inviscid flow in constant area channel

As a first multi-dimensional problem, we consider a two-dimensional version of the above constant area channel. The purpose is to identify any new phenomena introduced by the added dimension and to ensure that the procedure continues to provide robust convergence. The ratio of channel length to the passage height is $L/H = 20$. The grid has uniform spacing in $x$ and is stretched near the wall in preparation for the viscous examples in Sections 5.3 and 5.4. Three cases with the maximum cell aspect ratio (AR) at the wall equal to 100, 500 and 5000 are presented.

The boundary conditions are: $p^0 = 1.4e5$ Pa, $T^0 = 300$ K at the inlet and $p_{back} = 1.01e5$ Pa at the outlet which correspond to a converged Mach number of 0.7. The initial condition was expansion initiated. Because of the symmetry of the problem, only one half of the channel is computed. As in the one-dimensional problem, the global *CFL* is set to 1000 and the solution-limited procedure is used to provide robustness.

The global character of the convergence process is shown in the left plot of Fig. 5 in terms of the $L_\infty$ norm of $\Delta p/p_{ref}$ for the three different grid aspect ratio cases. The other parameters converged in similar fashion but for clarity, only the pressure is shown. In general, the convergence is similar to the 1-D case except the number of iterations is somewhat larger and increases with cell aspect ratio.

The plot on the right of Fig. 5 compares the estimated change (after adjusting the time step) with the actual solution change for the AR = 500 case. Iterations in which the estimated change is 0.1 imply that the time step is limited in at least one cell in the domain while steps in which the estimated solution change is less than 0.1 indicate the solution is not limited and *CFL* = 1000 is applied throughout the domain. As can be seen, the time-step limit is enforced during approximately the first 200 time steps but, following this the *CFL* = 1000 time step is applied to all cells causing the solution to converge rapidly. Note that the implicit solution change is nearly always less than the estimated change indicating that the estimate does, indeed, provide a reasonable bound on the solution change. The 10% fractional change ($\alpha = 0.1$), however, is more than sufficient to ensure positivity throughout the complete convergence process.

Detailed investigation of the convergence process shows that at the initial time step, the selected reference conditions in the last column of cells adjacent to the exit boundary were based on the pressure gradient across the cell and the corresponding the MOC expression for the velocity. The reference conditions in the remainder of the cells were set by global limits for the velocity and pressure (zero velocity and zero pressure gradients). Accordingly, in this first step, the resulting *CFL* limit was applied only to the last column and was based on the pressure change. After the first step, the *CFL*'s in a small fraction of the cells near the exit plane (extending more than one column away) were limited by the dynamic pressure.

The remaining line on the right side of Fig. 5 gives the fraction of cells in which the solution is limited in any given iteration. Note that a large fraction of cells are limited at the beginning, but that this number decreases as the velocity increases from zero throughout the domain, then reaches zero as the flowfield nears the correct solution. Except for the first step, the limiting reference condition was the dynamic pressure.

These results demonstrate the role of the limiting procedure. Limiting allows the flowfield to safely undergo large changes from the initial condition until it reaches a reasonable approximation to the exact solution without encountering difficulty or diverging. As the flowfield approaches the exact solution, limiting ceases and the large *CFL* drives the solution rapidly to the correct answer. The limiting method continues to ensure reliable convergence when the first simple example is extended to two dimensions. Again, the global *CFL* was set to 1000, and the computation progressed from a quiescent initial condition to the converged solution without user intervention, but with large variations in the local *CFL*'s during the computation.

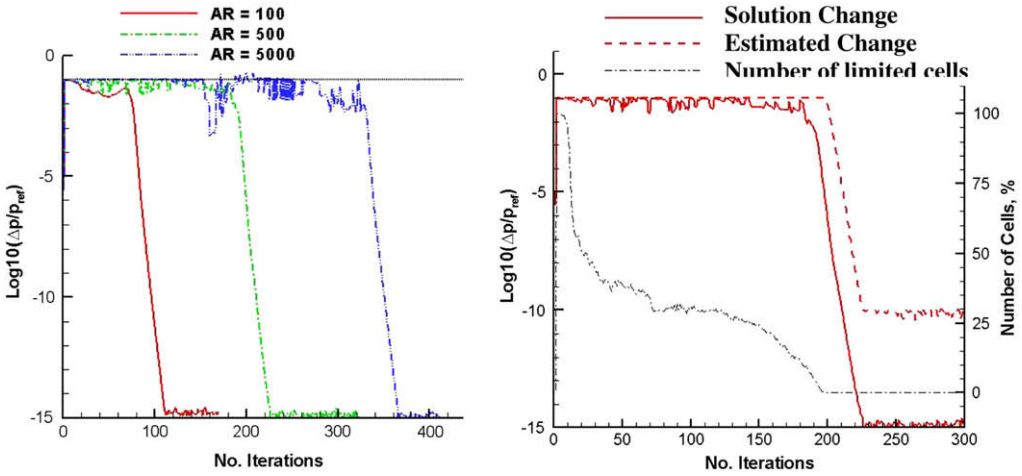


**Fig. 5.** $L_\infty$ norm of solution-limited convergence rate for two-dimensional flow in constant area duct. $L/H = 20$, $Ma = 0.7$. Expansion wave initiated case. Left: comparison for different maximum cell aspect ratio (AR). Right: comparison for calculated and estimated solution changes and number of cells limited during the convergence. AR = 500.

As the next step, we impose a no-slip condition on the duct walls and look at the effectiveness of the limiting method in the presence of viscosity in the entry region of a channel.

### 5.3. Viscous flow in constant area channel

To investigate the effect of viscous terms on the limiting procedure, we consider developing flow in a channel using the same geometry as for the above inviscid example. To resolve the boundary layers in the entry region the grid is stretched in both the axial and radial directions (the radial stretching is the same as that used in the inviscid solution) with a total grid of 32,000 cells and a maximum aspect ratio of 500. The channel length to passage height ratio is again, $L/H = 20$ and the Reynolds number is set at 1000. The estimated entry length [34] for this problem is $L/H = 44.2$, indicating that the boundary layers should grow to a reasonable fraction of the channel in the domain but should not reach the fully developed state. The boundary conditions chosen for this problem are identical to the conditions that resulted in $Ma = 0.7$ flow in the above case, specifically: $p^0 = 1.4e5$ Pa, $T^0 = 300$ K at the inlet and $p_{back} = 1.01e5$ Pa at the outlet. The problem was again started from rest with an expansion wave from the downstream end. Again, the single user-specified quantity for this calculation was $CFL = 1000$.

A graphical description of the convergence and limiting process for this viscous problem is given in Fig. 6 showing quantities analogous to those for the inviscid problem in Fig. 5. For the viscous problem, the solution change was limited for approximately the first 150 steps after which it converged rapidly toward the solution. The solution change was generally less than the estimated change, and never exceeded it by more than a few percent. During the computation, the number of limited cells increased from the last row of cells in the first step to a maximum and then decreased to zero as a meaningful flowfield was established. The primary difference was that the reference pressure was sometimes based upon the viscous term instead of the dynamic pressure, reflecting the different physics in the near-wall regions of the domain. All in all, this computation also converged reliably from the initial quiescent condition to the final converged solution which is shown in Fig. 7 for reference.

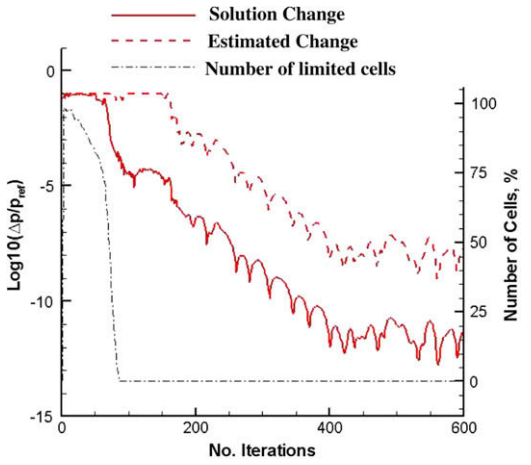


**Fig. 6.** $L_\infty$ norm of solution-limited convergence rate for viscous flow in entry region of channel. $L/H = 20$, $Re = 1000$, Expansion wave initiated.

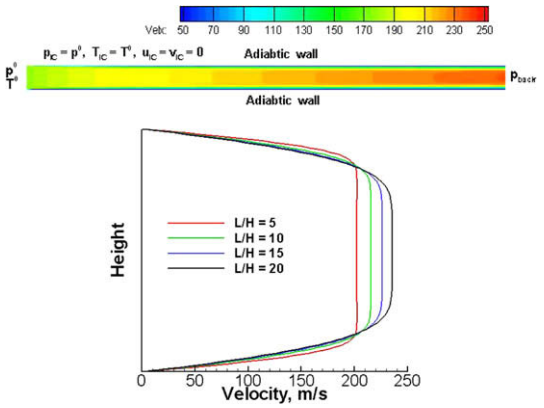


**Fig. 7.** Converged velocity contours in channel (top) and velocity profiles at various locations (bottom).

### 5.4. Viscous flow with strong heat addition

As the next demonstration, we consider the developing flow region of the same two-dimensional channel with strong heat addition through the walls. The heat addition tests the manner in which the limiting procedure responds to strong temperature gradients. In addition, the wall heating also has a dominant effect on the start-up transient leading to a very different convergence path. The grid and boundary conditions are the same as those used in Section 5.3 except that the wall temperature is set to 3000 K introducing a ten-to-one temperature change in the solution. These conditions again give a converged solution with a Mach number of nominally 0.7. We again set the global *CFL* = 1000 and use the solution-limited procedure to provide convergence without user intervention. The geometry and the converged temperature contours are shown in Fig. 8 in conjunction with a summary of the boundary conditions. Note that the boundary layers are considerably thickened by the heat addition and the consequent density change at the walls.

The initial condition is again chosen as quiescent, $u = v = 0$, with the pressure and temperature, $p = p^0$ and $T = T^0$, but the hot wall causes the initial transient to be more complicated than the previous examples, presenting a more severe test to the limiting procedure. The beginning transient in the computation is composed of the superposition of a longitudinal expansion wave analogous to those in the earlier problems plus a transverse compression wave created by the expansion of the heated fluid adjacent to the hot wall. The strong wall heating rapidly generates a transverse $v$-velocity along the entire channel which, in turn, induces an axial velocity causing the fluid to flow from the center of the pipe toward both ends, producing outflow at the 'inlet' as well as the exit. Thus, at early times in the solution, the fluid in the left half of the pipe moves in the upstream direction but changes sign and flows downstream when the expansion wave reaches it. Thus, in addition to introducing a substantial $v$-component that was not present in the unheated viscous calculation, the present calculation also requires a sign change in the axial velocity. As the solution nears convergence, the perturbation propagating from the downstream end overcomes these heat-transfer-induced motions and produces the final, familiar, entry flow field. The heat addition therefore introduces substantial complications into the initial transient as well as into the final converged solution and tests not only the effects of strong temperature changes, but also the effects of velocity sign changes during the transient convergence process.

The calculated and estimated solution changes of the pressure for the heated, viscous flow case are shown in Fig. 9. With solution limiting enforced and the *CFL* set to 1000, convergence proceeded in a manner similar to that for the earlier cases. Again, the calculated solution change during the early portion of the convergence process occasionally exceeds the estimated value, but remains of the same order of magnitude and provides convergence with no user interaction. Because of the wall heating the temperature is limited during the first 700 iterations (Fig. 9), but once the field is set up, convergence is rapid with the total iterations required for convergence being approximately 900. The strong temperature gradients that occur during the convergence process and in the final solution are adequately handled by the local limit on temperature

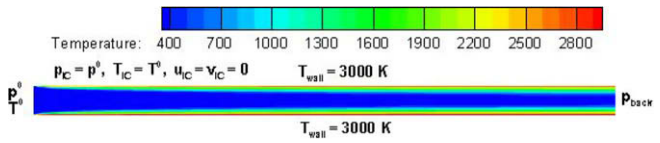


**Fig. 8.** Converged temperature contours for channel flow with heating. Fluid inlet temperature, 300 K; wall temperature, 3000 K.

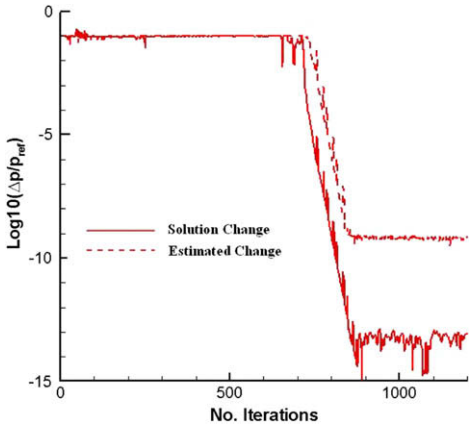


**Fig. 9.** $L_\infty$ norm of solution-limited convergence rate for strongly heated viscous flow in entry region of channel. $L/H = 20$, $Re = 1000$, Expansion wave initiated.

($\alpha = 0.1$) and did not require redefinition of the reference temperature. In addition, the initial backward flow in the upstream half of the tube and the subsequent reversal were handled by the velocity change limit.

Some details of the convergence process are given in Figs. 10 and 11 to support the convergence plots in Fig. 9. Fig. 10 shows the *CFL* distribution for the first step of the iteration. The expansion wave at the downstream end causes the time step at the outlet plane of cells to be limited. The wall thermal condition imposes a limit at the wall boundary cells from the very beginning, which is not so clearly shown in the contours plot because of the refined grid near the wall but is shown by the more detailed plots of the *CFL* profiles on the top left of Fig. 10.

Fig. 11 shows a sequence of axial velocity contour plots throughout the convergence process. At time steps one and two, the velocity profile is seen developing near both ends with larger values in the middle of the channel than along the walls

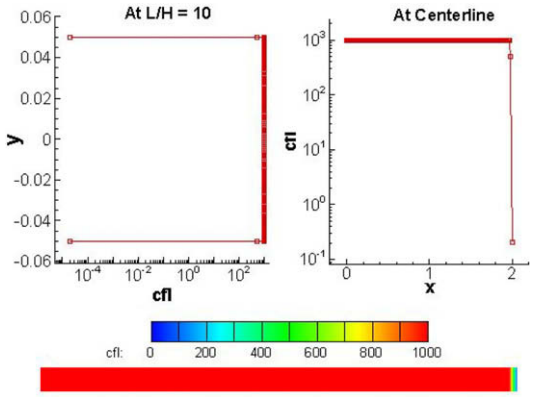


**Fig. 10.** *CFL* distributions for the first step of the iteration. Top: *CFL* profiles at two different locations. Bottom: *CFL* contours in the channel.

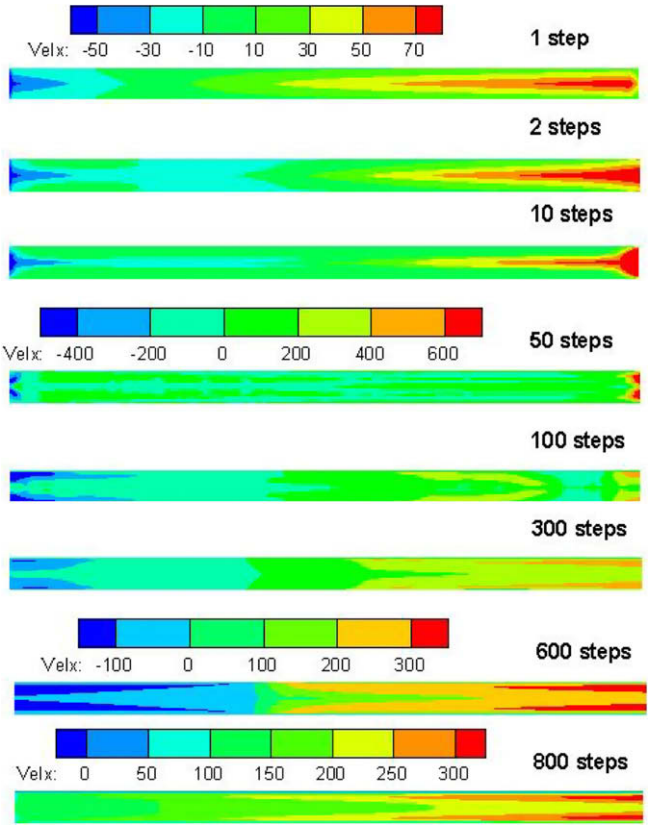


**Fig. 11.** Axial velocity contour plots showing solution during convergence. For clarity, the color scheme for the contour plots has been changed for different iteration steps. Plots not having a color legend follow the legend above them.

because of the smaller time steps on the walls. Note that from 50 to 600 time steps that there are large regions with substantial negative axial velocity components. As the solution approaches convergence (800 steps), the velocity field becomes positive throughout with the solution eventually reaching the familiar pattern for an entry duct.

### 5.5. Viscous flow in a converging–diverging nozzle

The above test problems all involved nearly parallel flow with small streamline curvature. To incorporate more complex flow features, we next consider two test cases with significant transverse velocity components. The first of these concerns a converging–diverging nozzle with an inlet-to-throat area ratio of 100. The Reynolds number based on the throat diameter and flow conditions at the throat is 200 and the walls are adiabatic. The total grid size is 14,000 with a maximum aspect ratio of 200. We present two solutions for this geometry, one with an unchoked throat, and the other for choked conditions. The boundary conditions for the unchoked cased are: $p^0 = 1.4e5$ Pa, $T^0 = 300$ K and zero flow angle at the inlet and $p_{back} = 1.01e5$ Pa at the outlet. The initial condition is again an expansion-initiated condition with a converged throat Mach number of approximately $Ma = 0.7$.

The geometry and converged Mach number contours for the subsonic case are given in Fig. 12 while the convergence process is given in Fig. 13. Again, the same input control parameters ($CFL = 1000$, $\alpha = 0.1$) were used without user intervention and provided reliable convergence. The presence of the reasonably strong transverse velocity components had little or no influence on reliability. Again, the limiting procedure holds the solution change to a constant plateau for the first 200 iterations after which it converged reasonably rapidly.

Solutions for this geometry were also computed for the choked condition by lowering the downstream pressure to a value small enough to ensure all supersonic flow in the short divergent section. Mach number contours for this case are given in Fig. 14, with convergence results in Fig. 15. Solution limiting was present for approximately the first 200 steps followed by convergence to machine accuracy in an additional 600 steps.

These results also produce robust convergence as can be seen. The presence of transonic velocities in a significant portion of the flowfield had little effect on convergence rates, and the solution-limited time stepping approach continues to enhance reliability. Additional computations for converging–diverging nozzles with various inlet-to-throat area ratios were also

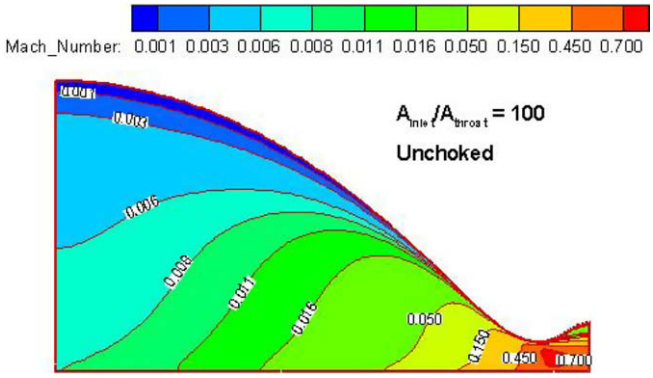


**Fig. 12.** Converged Mach number contours for unchoked viscous flow in converging–diverging nozzle.

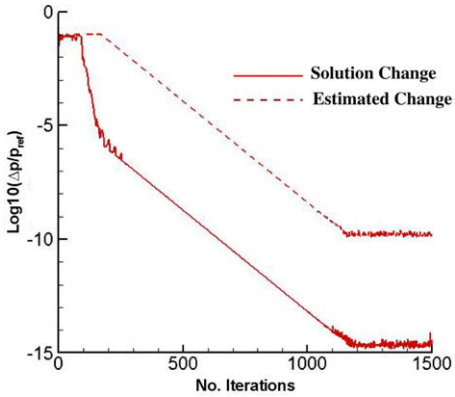


**Fig. 13.** $L_\infty$ norm of solution-limited convergence rate for viscous flow in converging–diverging nozzle. Unchoked throat condition.

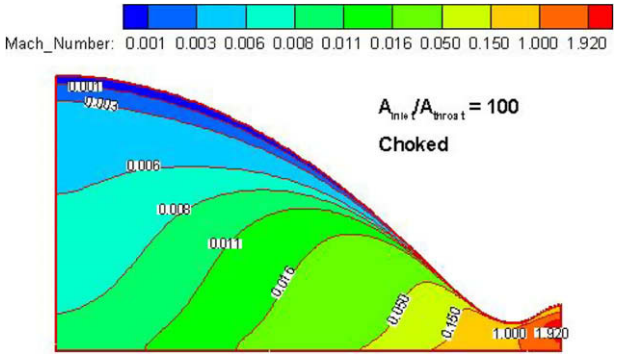


**Fig. 14.** Converged Mach number contours for choked viscous flow in converging–diverging nozzle.

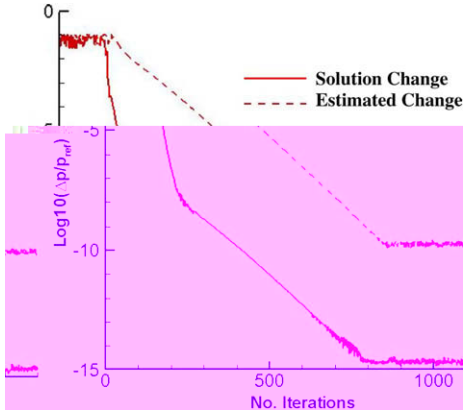


**Fig. 15.** $L_\infty$ norm of solution-limited convergence rate for viscous flow in converging–diverging nozzle. Choked throat condition.

conducted for both choked and unchoked conditions. In all cases, the solution-limited time stepping method provided effective convergence with no change in user-specified conditions.

### 5.6. Flow past a backward-facing step

The next example concerns the flow over a backward-facing step. The backward facing step problem is significant for the solution-limited time stepping procedure in that the flow must change sign during the convergence process and both velocity components have regions of positive and negative signs in the converged solution. The results verify that the velocity limiting procedure does not adversely affect convergence when velocity sign changes are a characteristic of the converged flowfield. Two different step heights were considered as shown in Fig. 16. The first [35] has a step height to inlet height ratio of 1:9 while the second has a ratio of 1:2 corresponding to the experimental configuration of Armaly et al. [36]. Both cases are compression wave initiated with specified upstream Mach numbers (0.128 and 0.004) and the same upstream temperature (300 K) and the same back pressure ($1.01e^5$ Pa). The Reynolds numbers based on step height are 370 and 50. The grid sizes are 17,000 and 23,000 with maximum cell aspect ratio of 300. The convergences for the two cases are similar and only that for the second case is shown in Fig. 17. The convergence rate decreases after around 500 iterations because of the recirculation zone at the backward facing step. The time stepping method is operational only during approximately the first 200 iterations. The results clearly show that the limiting procedure remains effective in flow fields characterized by sign changes in both velocity components.

### 5.7. Transonic flow over an RAE 2822 airfoil

The above problems consider relatively simple problems that test various characteristics of the solution-limited time stepping method. To illustrate the method in more practical applications, we include two additional cases in the next two sections.

The first of these concerns transonic flow over an RAE 2822 airfoil. A grid of 24,000 cells is used. The free stream Mach number is 0.729 while the static pressure and temperature are $1.0e^5$ Pa and 273 K, respectively. The solution shown is for an

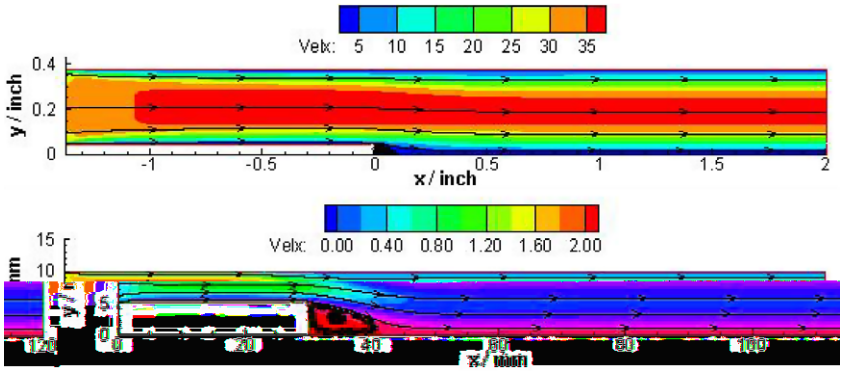


**Fig. 16.** Schematic of backward-facing step and converged velocity contours. Top: 1:9 step height ratio; Bottom: 1:2 exit height ratio.

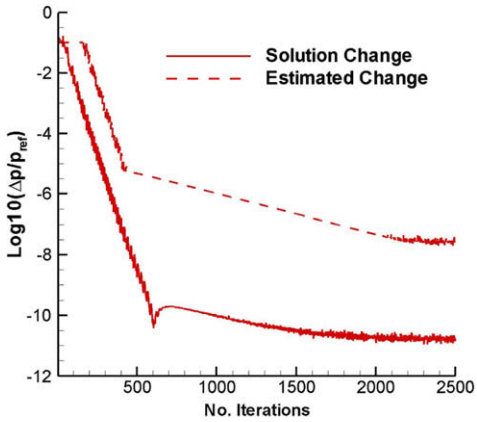


**Fig. 17.** $L_\infty$ norm of solution-limited convergence rate for backward-facing step flow. *CFL* = 1000, $\alpha = 0.1$.

angle of attack of 2.31°. These conditions correspond to a Reynolds number of 6.5 million based on a chord length of 1.0 ft. A contour plot of the Mach number near the airfoil is shown in Fig. 18. The pressure coefficient on the airfoil ($C_p$) is compared experimental measurements in Fig. 19.

The turbulence characteristics were simulated by the $k$–$\omega$ turbulence model [37]. To apply the limiting procedure, we choose normalization values for $k$ and $\omega$ according to the procedure described in Section 3.3. For the turbulent kinetic energy, we choose a reference value $k_{ref} = \frac{3}{2}(I \cdot U_{ref})^2$ where $I$ is the turbulence intensity. The turbulence dissipation rate $\omega$ has a

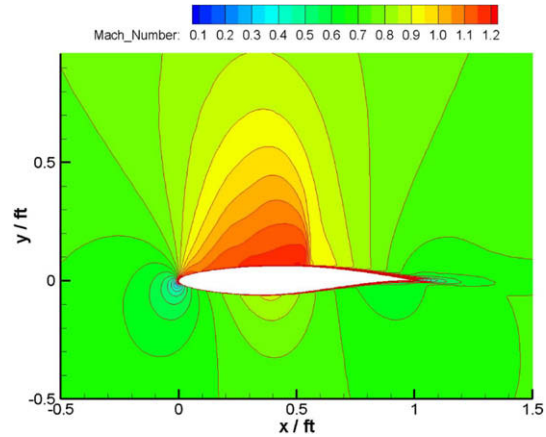


**Fig. 18.** Mach number contours near airfoil region.

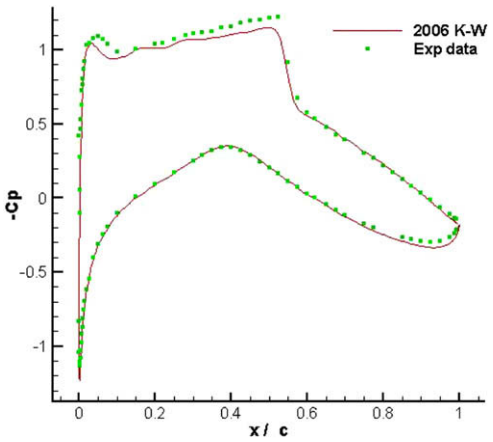


**Fig. 19.** $C_p$ distribution around RAE 2822 airfoil. Experimental data: green dots; Computational result: red line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

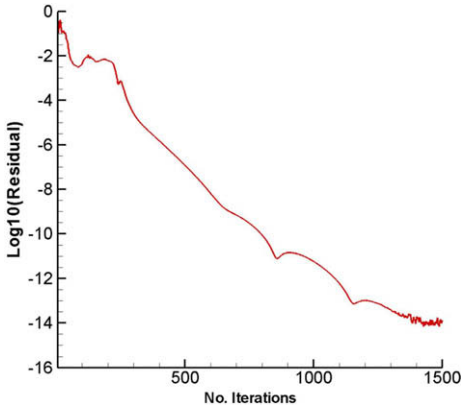


**Fig. 20.** Convergence rate for transonic flow past a RAE2822 airfoil.

maximum value near the wall and decreases quickly away from the wall with magnitude variations that are typically more than five orders of magnitude. For the reference condition here, we set $\omega_{ref} = \omega_{wall}$.

The convergence history for the RAE 2822 case is recorded in Fig. 20. The $L_\infty$ norm of $\Delta p/p_{ref}$ for the solution-limited convergence rates is given in Fig. 21. The results show that the calculated solution change is well estimated and limited by the method during the initial 'non-linear' convergence region. The total number iterations required for convergence is approximately 1500.

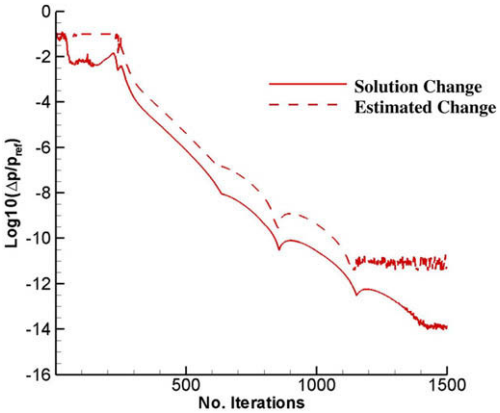


**Fig. 21.** $L_\infty$ norm of solution-limited convergence rate for transonic flow past a RAE2822 airfoil. $CFL = 1000$, $\alpha = 0.1$.

### 5.8. Mixing and combustion of hydrogen ($H_2$) and oxygen ($O_2$) in rocket engine combustor

As a final example of convergence control by means of the limiting procedure, we present results for a reacting, turbulent flow in a model rocket engine combustor [38]. The combustor is fed by a coaxial injector with oxygen in the central passage and hydrogen in the surrounding annulus. The computational domain and boundary conditions are shown in Fig. 22. The grid contains 46,000 cells. The combustion chamber length is 93 mm and its diameter is 38.1 mm. The inlet diameter of the central oxidizer jet is 5.26 mm, and the inner and outer diameters of the annular fuel passage are 6.30 and 7.49 mm, respectively. The Reynolds numbers of the oxidizer and fuel inlet are 604,000 and 169,000, respectively. The mass flow rates and total temperatures are specified as upstream boundary conditions and a back pressure of 5.2e6 Pa is applied at the unchoked nozzle exit. All wall surfaces are set as no-slip, adiabatic conditions.

A 9 species, 17 reaction step chemical kinetics model was used to represent the combustion of hydrogen ($H_2$) and oxygen ($O_2$) [39]. Thermodynamic and transport properties of all species are expressed as arbitrary functions of pressure and temperature with appropriate mixing relations used to obtain mixture properties. Enthalpies, viscosity and thermal conductivity for each species are taken from polynomial expressions. Species diffusivities are obtained from a Chapman–Enskog expression.

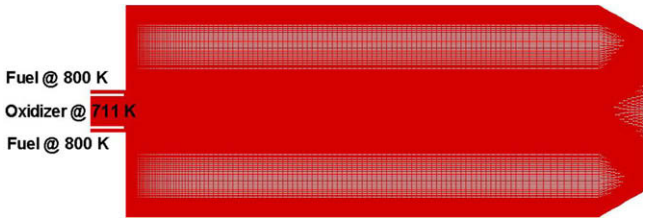


**Fig. 22.** Generic geometry and computational domain. Hydrogen and oxygen enter through injector passages on left.

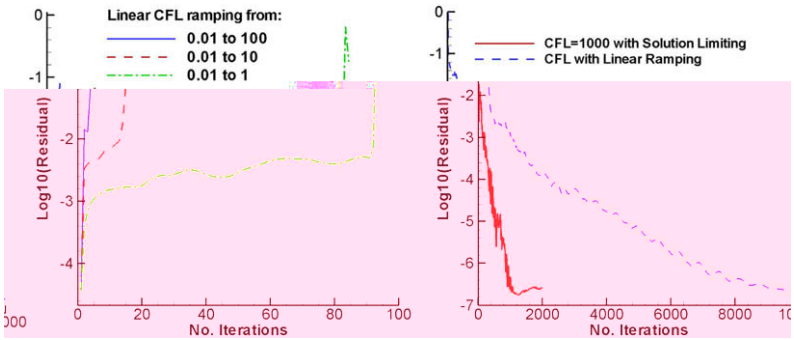


**Fig. 23.** Convergence rate comparisons for the time stepping method and traditional *CFL* ramping method.

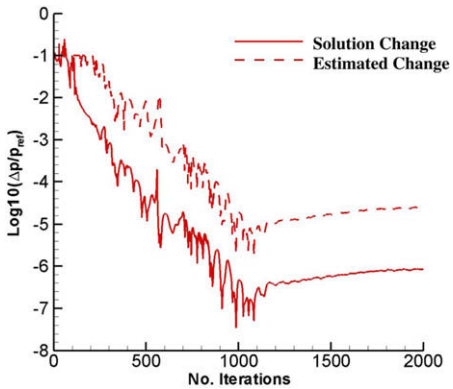


**Fig. 24.** $L_\infty$ norm of solution-limited convergence rate for combustion in rocket engine combustor. *CFL* = 1000, $\alpha = 0.1$.

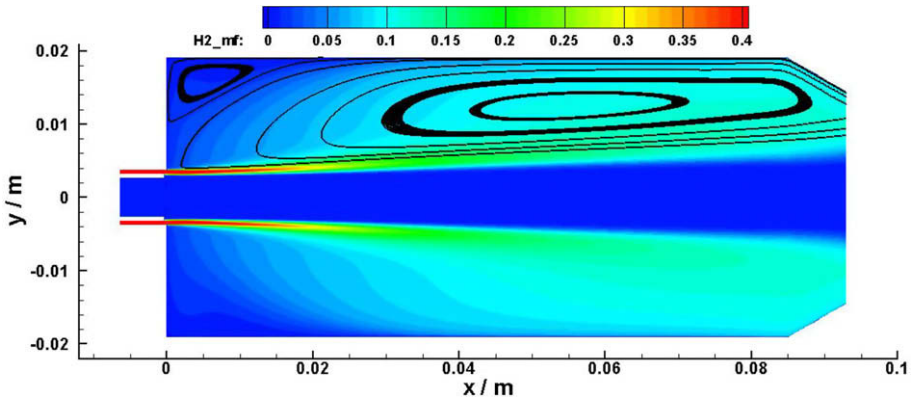


**Fig. 25.** Streamlines imposed on the contours of $H_2$ mass fraction.

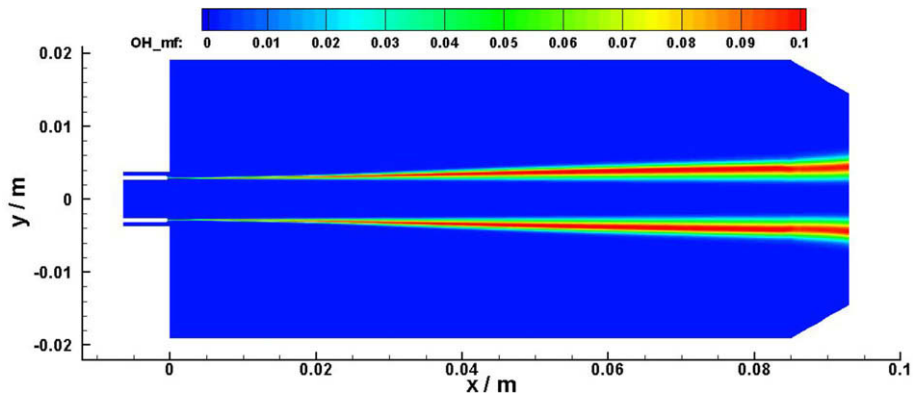


**Fig. 26.** Contours of mass fraction of OH radical.

To demonstrate the difficulty of converging this problem by conventional methods, we first attempt to solve the problem by ramping the *CFL* number from a small value to a larger one. The results of a series of three unsuccessful attempts are given in the left plot of Fig. 23 while a successful ramping procedure is given in the right side of Fig. 23 along with results for the solution-limiting procedure. The three attempts shown in the plot on the left corresponds to ramping the *CFL* from 0.01 to 100, 0.01 to 10 and 0.01 to 1 during the first 100 iterations. All three cases diverge before the *CFL* has reached its terminal value, although the ramping to *CFL* = 1, nearly reaches 100 iterations. As a fourth attempt, we ramp *CFL* from 0.01 to 1 in 500 iterations rather than in 100 iteration (right side of Fig. 23). This ramping schedule results in a converged solution although the convergence is considerably slower than that obtained with the solution-limited time stepping method. The results indicate that the solution-limiting procedure not only enhances code robustness without user intervention, but also improve convergence by a factor of ten. Clearly, the manually scheduled *CFL* ramping could be tuned to improve convergence but the uncertainties in determining the schedule and considerable user intervention that has already been done strongly contrasts with the effectiveness of the solution-limiting procedure. The calculated and estimated solution changes given in Fig. 24, again shows the calculated change is well bounded by the estimated solution change. Hydrogen mass fraction contours of the converged solution along with streamlines showing the recirculation zones are given in Fig. 25 along with contours of the OH radical which marks the flame location in Fig. 26 to demonstrate the character of this flowfield.

## 6. Summary and conclusions

A method for diminishing user intervention in implicit CFD computations is presented. The method is based upon controlling the local *CFL* in such a manner as to ensure that the local solution change is never more than a small fraction of the magnitude of the solution variable. The magnitude of the solution change is estimated from the residual and the computational resources required for implementing the procedure are a very minor fraction of the overall per-step computational requirements. As a nominal limit on the allowable solution change, positive-definite quantities are limited to a maximum change of 10%, while changes in indefinite quantities are allow to exceed the local value by a factor of 2. These magnitudes are sufficient to ensure that linearization procedures based upon solution Jacobians remain meaningful, thereby

precluding solution migrations to unphysical regions, yet large enough to allow rapid solution convergence. In addition the solution-change limit ensures that quantities such as pressure, density and temperature cannot become negative, again preventing a common difficulty in CFD solutions. Other limiting magnitudes have been tried, but the 10% limit appears to be a reasonable compromise between overly conservative changes that slow convergence unacceptably and more aggressive changes that may fail to keep the solution within reasonable bounds over the complete computation.

Perhaps the most important element for success of the method is determining acceptable reference conditions upon which to base the solution change. Our experience indicates that while all variables must be limited, the most critical parameter is the pressure. The characteristic pressure in a given problem or a series of different problems may be related to the local dynamic pressure, the local shear stress, the local flow acceleration or the local body force terms. Proper balancing of these with the pressure gradient is important for controlling the allowable change in pressure. Velocity, likewise, must be limited, but in such a manner that the velocity may change sign, whereas temperature limitation based simply upon the thermodynamic temperature appears sufficient in the conditions computed to date.

As a demonstration of the method, more than a half-dozen different problems, each with an added degree of difficulty, are presented. All solutions started from a quiescent initial condition with either an expansion fan from the exit boundary or a shock from the inlet boundary. For all cases, a single, user-specified *CFL* = 1000 was used and in all cases, the solution converged reliably and efficiently without user intervention or changes in the control parameters apart from those variations determined by the limiting procedure itself. The specific cases and variables tested along with their relevance include:

- One-dimensional flow: verifies ability to scale the solution change for conditions in which the velocity is zero in all cells and the pressure is constant everywhere except for one cell;
- Two-dimensional inviscid flow: verifies that findings from one-dimensional solutions are applicable in multi-dimensions;
- Two-dimensional viscous flow: incorporates effects of boundary layers and shear stress in solution change limits and reference conditions;
- Viscous flow with strong heating: adds considerable complexity to the starting transient, including directional changes in the velocity, while also incorporating the effects of strong temperature gradients and tests the effects of solution limits for temperature;
- Nozzle flows: tests the capability of the solution-limiting method in the presence of flows with large changes in the flow angle for both choked and unchoked conditions;
- Flow over a backward-facing step: incorporates the effect of recirculation, reverse flow and sign changes in the velocity;
- Transonic flow over RAE 2822 airfoil: incorporates turbulent flow at high Reynolds number flow along with shocks;
- Mixing and combustion of hydrogen and oxygen: tests the capability of the solution-limited time stepping method for multi-species and reacting flow in a complicated flow field.

In addition to the cases represented here, the method has also been tried for a large number of other problems. Our experience has been that the method greatly decreases the number of unsuccessful runs and nearly eliminates the need for user intervention in a wide variety of problems. Overall, the solution-limited time stepping procedure automatically adjusts the *CFL* number for each grid according to the estimated solution change and greatly reduces the sensitivity to initial conditions for various types of flows. The method requires negligible additional computing time and is easily implemented into any implicit CFD code.

## References

[1] R.H. Bush, G.D. Power, C.E. Towne, WIND: the production flow solver of the NPARC alliance, in: 36th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 12–January 15, 1998, AIAA Paper 1998-0935.
[2] N.T. Frink, S.Z. Pirzadeh, P.C. Parikh, M.J. Pandya, M.K. Bhat, The NASA tetrahedral unstructured software system (TetrUSS), in: 22nd International Congress of Aeronautical Sciences, Harrogate, United Kingdom, August 27–September 1, 2000, ICAS Paper No. 0241.
[3] V. Sankaran, C.L. Merkle, X. Zeng, D. Li, Influence of large-scale pressure changes on preconditioned solutions at low speeds, AIAA J. 42 (12) (2004) 2490–2498.
[4] W. Mulder, B. van Leer, Experiments with implicit upwind methods for the Euler equations, J. Comput. Phys. 59 (1985) 232–246.
[5] E. Issman, G. Degrez, H. Deconinck, Implicit upwind residual-distribution Euler and Navier–Stokes solver on unstructured meshes, AIAA J. 34 (10) (1996) 2021–2028.
[6] D. Vanderstraeten, A. Csk, D. Rose, An expert-system to control the CFL number of implicit upwind methods, Technical Report TM 304, Universiteit Leuven, March 2000.
[7] H.M. Bücker, B. Pollul, A. Rasch, On CFL evolution strategies for implicit upwind methods in linearized Euler equations, Technical Report, RWTH Aachen University, August 7, 2006.
[8] E.A. Luke, X.-L. Tong, J. Wu, L. Tang, P. Cinnella, A step towards 'Shape-Shifting' algorithms: reactive flow simulations using generalized grids, AIAA Paper No. 2001-0897, 2001.
[9] C. Merkle, D. Li, V. Sankaran, Multi-disciplinary computational analysis in propulsion, AIAA Paper No. 2006-4575, 2001.
[10] H. Nessyahu, E. Tadmor, Non-oscillatory central differencing for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 408–463.
[11] G. Jiang, E. Tadmor, Non-oscillatory central schemes for multidimensional hyperbolic conservation laws, SIAM J. Sci. Comput. 19 (1998) 1892–1917.
[12] P.D. Lax, Weak solutions of nonlinear hyperbolic equation and their numerical computation, Commun. Pure Appl. Math. 7 (1954) 159–193.
[13] B. van Leer, Flux-vector splitting for the Euler equation, NASA Langley Research Center Hampton, ICASE Report 82-30, 1982.
[14] T. Linde, A practical, general-purpose Riemann solver for hyperbolic conservation laws, in: Seventh International Conference on Numerical Methods in Fluid Dynamics, Clarendon, 2001.
[15] M.-S. Liou, C. Steffen, A new flux splitting scheme, J. Comput. Phys. 109 (1993) 23–93.
[16] M.-S. Liou, J. Edwards, Low-diffusion flux splitting methods for flows at all speed, AIAA J. 36 (9) (1998) 1610–1617.

[17] P. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[18] S. Godunov, A finite-difference method for the numerical computation and discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271–306.
[19] S. Osher, F. Solomon, Upwind schemes for hyperbolic systems of conservation laws, Math. Comput. 38 (1982) 339–374.
[20] E. Turkel, Preconditioning methods for solving the incompressible and low speed compressible equations, J. Comput. Phys. 72 (1987) 277–298.
[21] B. van Leer, W. Lee, P. Roe, Characteristic time-stepping or local preconditioning of the euler equations, AIAA 1991-1552-CP, Computational Fluid Dynamics Conference, Honolulu, 1991.
[22] H. Viviand, Pseudo-unsteady systems for steady inviscid flow calculation, in: F. Angrand et al. (Eds.), Numerical Methods for the Euler Equations of Fluid Dynamics, SIAM, Philadelphia, 1985.
[23] W. Briley, H. McDonald, S. Shamroth, A low Mach number Euler formulation and application to time iterative LBI schemes, AIAA J. 21 (10) (1983) 1467–1469.
[24] Y. Choi, C. Merkle, The application of preconditioning to viscous flows, J. Comput. Phys. 105 (1993) 207–223.
[25] W. Briley, H. McDonald, On the structure and use of linearized block implicit schemes, J. Comput. Phys. 34 (1980) 54–73.
[26] W. Briley, H. McDonald, An overview and generalization of implicit Navier–Stokes algorithms and approximate factorization, Comput. Fluids 30 (2001) 807–828.
[27] J. Thomas, R. Walters, Upwind relaxation algorithm for the Navier–Stokes equations, AIAA J. 25 (4) (1987) 527–534.
[28] K. Vanden, D. Whitfield, Direct and iterative algorithms for the three-dimensional euler equations, AIAA 1993-3378-CP, 1993.
[29] Joo Sung Kim, Oh Joon Kwon, An efficient and robust implicit operator for upwind point Gauss–Seidel method, J. Comput. Phys. 224 (2007) 1124–1144.
[30] C. Lombard, J. Bardina, E. Venkatapathy, J. Oliger, Multi-dimensional formulation of CSCM – an upwind flux difference eigenvector split method for the compressible Navier–Stokes equations, AIAA Paper No. 1983-1895, 1983.
[31] R. MacCormack, G. Candler, The solution of the Navier–Stokes equations using Gauss–Seidel line relaxation, Comput. Fluids 17 (1989) 135–150.
[32] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear system, SIAM J. Sci. Stat. Comput 7 (3) (1986) 856–869.
[33] H. Guillard, C. Viozat, On the behaviour of upwind schemes in the low Mach number limit, Comput. Fluids 28 (1) (1999) 63–86.
[34] Durst Franz, An Introduction to the Theory of Fluid Flows, 2008, ISBN:9783540713425.
[35] D. Driver, H. Seegmiller, Features of a reattaching turbulent shear layer in divergent channel flow, AIAA J. 23 (2) (1985) 163–171.
[36] B.F. Armaly, F. Durst, J.C.F. Pereira, B. Schonung, Experimental and theoretical investigation of backward-facing step flow, J. Fluid Mech. 127 (1983) 473–496.
[37] D. Wilcox, Turbulence Modeling for CFD, second ed., DCW Industries, La Canada, CA, 1998.
[38] P. Tucker, S. Menon, C. Merkle, J. Oefelein, V. Yang, An approach to improved credibility of CFD simulations for rocket injector design, in: 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, July 8–July 11, Cincinnati, OH, 2007.
[39] M. Ó Conaire, H.J. Curran, J.M. Simmie, W.J. Pitz, C.K. Westbrook, A comprehensive modeling study of hydrogen oxidation, Int. J. Chem. Kinet. 36 (2004) 603–622.