TIME ACCURACY AND THE USE OF IMPLICIT METHODS

Thomas H. Pulliam*

Senior Staff Scientist, Fluid Dynamics Division
NASA Ames Research Center

**Abstract**

Implicit time differencing methods are usually developed to be time accurate. The typical implementation and use, though, is either non-time accurate for steady state convergence acceleration or the schemes are modified (e.g. approximate factorization, explicit boundary conditions, linearization error, etc.) to enhance their efficiency, practical use, or tractability on conventional computers. This papers examines these departures from time accuracy and introduces methods to enhance the time accuracy of conventional schemes. Approximations are examined for their effect on time accuracy, subiteration techniques designed to improve time accuracy are presented, and analysis for time accuracy assessment are introduced.

## Introduction

Since 1976, when Steger [1] first introduced a practical implicit finite difference scheme for the Euler and Navier-Stokes equations, there have been numerous (too numerous to reference here) modifications and new methods developed which use implicit time approximations with various spatial discretion techniques. Until recently, most of the effort was directed toward steady state or slowly varying unsteady applications. Most of these use either large time steps, spatially variable scalings, or preconditionings techniques to accelerated convergence. The next generation of challenges in CFD, though, will be in the area of unsteady time accurate calculations. At face value one might assume that explicit techniques are the choice for such computations. The demands of adequate boundary layer and complicated geometric resolution make explicit methods too restrictive in terms of time steps and efficiency for most practical cases. The alternative is to develop more efficient and accurate implicit methods. One advantage of implicit methods over explicit is that larger time steps can be used than would be permitted by explicit stability bounds. For instance, a high Reynolds number viscous transonic airfoil computation using fine grid resolution at the surface typically requires CFL numbers in the wall normal direction on the order $10^3$. These arguments have been delineated many times in many references and applications with the result that implicit methods of one form or another are widely used today.

This paper will address some of the approximations used to make implicit methods more efficient and practical for the solution of the Euler and Navier-Stokes equations. In particular, approximate factorizations, diagonalizations and linearization approximations will be reviewed and categorized. A subiteration correction scheme commonly used today will be presented, improved, demonstrated and analyzed. This scheme is used to produce a second order accurate, more robust implicit method for unsteady flow computations.

## Nonlinear Equations

A generic form of a system of partial differential equations will be used for demonstration and analysis.

$$\partial_t \mathbf{Q} + F(\mathbf{Q}) = 0 \tag{1}$$

will represent our generic nonlinear scalar or system of equations. This generic system of equations may represent any number of conventional problem definitions, for example:

The one- or multi- dimensional Euler or Navier-Stokes equations. Where, for example,

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \quad F(\mathbf{Q}) = \partial_x \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{bmatrix} + \partial_y \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{bmatrix} \tag{2.1}$$

for the 2D Euler equations.

A more conventional way to write the 2D Euler equations would be

$$\partial_t \mathbf{Q} + \partial_x \mathbf{E}(\mathbf{Q}) + \partial_y \mathbf{F}(\mathbf{Q}) = 0 \tag{2.2}$$

where $\mathbf{Q}$ is a vector of $n$ variables and $\mathbf{E}, \mathbf{F}$ represent the nonlinear flux vectors.

There are many more choices for nonlinear equations encountered in the literature, e.g. the full Navier-Stokes equations, viscoelastic equations, equations involving chemical reactions, etc. In each case the system is castable into a form where $F(\mathbf{Q})$ is uniquely defined.

* Associate Fellow, AIAA

## Implicit Forms

An implicit approximation in time for the solution of Eq.(1) can be written as

$$\Delta\mathbf{Q}^n = \frac{\vartheta\Delta t}{1+\varphi}\frac{\partial}{\partial t}(\Delta\mathbf{Q}^n) + \frac{\Delta t}{1+\varphi}\frac{\partial}{\partial t}\mathbf{Q}^n + \frac{\varphi}{1+\varphi}\Delta\mathbf{Q}^{n-1} + O\left[(\vartheta - \frac{1}{2} - \varphi)\Delta t^2 + \Delta t^3\right] \quad (3)$$

with $\Delta\mathbf{Q}^n = \mathbf{Q}^{n+1} - \mathbf{Q}^n$ and $\mathbf{Q}^n = \mathbf{Q}(n\Delta t)$. The parameters $\vartheta$ and $\varphi$ can be chosen to produce different schemes of either first or second order accuracy in time.

The values $\vartheta = 1$ and $\varphi = 0$, results in the first order Euler implicit scheme, $\vartheta = 1/2$ and $\varphi = 0$ for a trapezoidal implicit or for $\vartheta = 1$ and $\varphi = 1/2$ gives the three point backward second order implicit scheme.

Replacing $\frac{\partial}{\partial t}\mathbf{Q}$ with $-F(\mathbf{Q})$ we have (neglecting the higher order terms)

$$\Delta\mathbf{Q}^n + \frac{\vartheta\Delta t}{1+\varphi}F(\mathbf{Q}^{n+1}) = \frac{(\vartheta-1)\Delta t}{1+\varphi}F(\mathbf{Q}^n) + \frac{\varphi}{1+\varphi}\Delta\mathbf{Q}^{n-1} \quad (4)$$

Equation (4) is nonlinear in terms of $\mathbf{Q}^{n+1}$ due to the second term on the left hand side. The nonlinear terms are linearized in time about $\mathbf{Q}^n$ by a Taylor series such that

$$F(\mathbf{Q}^{n+1}) = F(\mathbf{Q}^n) + \mathcal{A}(\mathbf{Q}^n)\Delta\mathbf{Q}^n + \mathcal{O}(\Delta t^2) \quad (5)$$

where $\mathcal{A} = \partial F(\mathbf{Q})/\partial\mathbf{Q}$ is typically called the Jacobian of $F$ and $\Delta\mathbf{Q}^n$ is $\mathcal{O}(\Delta t)$. Note that the linearizations are carried out to the $\mathcal{O}(\Delta t^2)$ terms. These terms will be multiplied (see below) by $\Delta t$ and so if a second order time scheme had been chosen the linearizations would not degrade the time accuracy.

Replacing $F(\mathbf{Q}^{n+1})$ using Eq.(5) in Eq.(4) one gets

$$\left[I + \frac{\vartheta\Delta t}{1+\varphi}\mathcal{A}(\mathbf{Q}^n)\right](\mathbf{Q}^{n+1} - \mathbf{Q}^n) = -\frac{\Delta t}{1+\varphi}F(\mathbf{Q}^n) + \frac{\varphi}{1+\varphi}\Delta\mathbf{Q}^{n-1} \quad (6)$$

which is second order accurate in time for $\varphi = 0, \vartheta = \frac{1}{2}$ or $\varphi = \frac{1}{2}, \vartheta = 1$ . Equation (6) is kept in what is called "Delta Form," referring to the $\Delta\mathbf{Q}^n = \mathbf{Q}^{n+1} - \mathbf{Q}^n$ term on the left hand side. The left hand side of Eq.(6) and similar equations will be referred to as the *implicit side* and the right hand side as the *explicit side* .

Equation (6) is the basis for most implicit time integration schemes. If the original problem of interest was a system of nonlinear partial differential equations, e.g. the Euler equations, then the *implicit side* operator of Eq.(6) would represent a matrix operator of the order of the system size. In addition, if finite difference operators were used for the spatial derivatives then the *implicit side* would represent an even larger matrix system of the order of the system size times the discrete grid dimensions. Typical examples of such schemes can be found in Refs. [2,3] among numerous others. We shall not go into the detail of various implicit solution algorithms, except where it pertains to the approximations used to solve such schemes.

## Implicit Approximations

In general, Eq.(6) is not too difficult, but prohibitively time consuming, to solve directly. Large sparse matrix systems usually result from conventional finite difference or finite volume schemes for the spatial derivatives. Approximations to the *implicit side* operator in Eq.(6) are employed to either improve the efficiency, reduce computer storage requirements, enhance the stability, or map the system onto nonconventional computers (e.g. parallel implementation, Jespersen and Levit [4]).

Equation(6) is rewritten as

$$\mathcal{L}(\mathbf{Q}^n)\Delta\mathbf{Q}^n = \mathcal{R}(\mathbf{Q}^n, \mathbf{Q}^{n-1}) \quad (7)$$

Approximations to $\mathcal{L}(\mathbf{Q}^n)$ are made for a variety of reasons. In steady state computations (where one is just interested in satisfying $\Delta\mathbf{Q}^n = 0$), any approximation to $\mathcal{L}(\mathbf{Q}^n)$ which produces a stable and convergent solution process is admissible. In fact, $\mathcal{L}(\mathbf{Q}^n)$ is typically taken as some $\mathcal{O}(\Delta t)$ approximation. Some example approximations are presented below.

Beam and Warming [5] developed an approximate factorization of multidimensional implicit schemes applied to the Euler and Navier-Stokes equations. The implicit scheme applied to Eq.(2.2) yields

$$[I + \alpha\partial_x\mathbf{A}^n + \alpha\partial_y\mathbf{B}^n]\Delta\mathbf{Q}^n = \mathbf{R}(\mathbf{Q}^n, \mathbf{Q}^{n-1}) = -\beta(\partial_x\mathbf{E}(\mathbf{Q}^n) + \partial_y\mathbf{F}(\mathbf{Q}^n)) + \frac{\varphi}{1+\varphi}\Delta\mathbf{Q}^{n-1} \quad (8)$$

with $\mathbf{A}^n = \frac{\partial\mathbf{E}}{\partial\mathbf{Q}}$, $\mathbf{B}^n = \frac{\partial\mathbf{F}}{\partial\mathbf{Q}}$, $\alpha = \frac{\vartheta\Delta t}{1+\varphi}$ and $\beta = \frac{\Delta t}{1+\varphi}$.

Second order central differences are typically used for the $\partial$ derivatives in Eq.(8), which would lead to a very large sparse matrix representing $\mathcal{L}(\mathbf{Q}^n)$. A template representation of the sparse system is

$$
\begin{bmatrix}
[] & [] & & & [] & & & & & & & \\
[] & [] & [] & & & [] & & & & & & \\
 & [] & [] & [] & & & [] & & & & & \\
 & & [] & [] & & & & [] & & & & \\
[] & & & & [] & [] & & & [] & & & \\
 & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & & \\
 & & -B & & & -A & I & A & & & B & \\
 & & & [] & & & [] & [] & & & & [] \\
 & & & & [] & & & & [] & [] & & \\
 & & & & & [] & & & [] & [] & [] & \\
 & & & & & & [] & & & [] & [] & [] \\
 & & & & & & & [] & & & [] & []
\end{bmatrix}
$$

where the band width of the above matrix is directly proportional to minimum grid dimension which can be quite large.

Factorization of Eq. (8) produces

$$
\begin{aligned}
&[I+\alpha\partial_x\mathbf{A}^n]\,[I+\alpha\partial_y\mathbf{B}^n]\,\Delta\mathbf{Q}^n \\
&\qquad -\alpha^2(\partial_x\mathbf{A}^n)(\partial_y\mathbf{B}^n)\Delta\mathbf{Q}^n=\mathbf{R}(\mathbf{Q}^n,\mathbf{Q}^{n-1})
\end{aligned}
\tag{9}
$$

The $\alpha^2$ term in Eq.(9) can be neglected since with $\Delta\mathbf{Q}^n=\mathcal{O}(\Delta t)$ that term is now $\mathcal{O}(\Delta t^3)$ which maintains the second order accuracy of the implicit scheme. In Eq.(9), each of the separate *implicit side* operators are now banded block tridiagonal matrices and are much easier to invert, see [2] for more details. A template representation of each of the block tridiagonal operators is

$$
\begin{bmatrix}
[\ ] & [\ ] & & & & & & \\
[\ ] & [\ ] & [\ ] & & & & & \\
 & [\ ] & [\ ] & [\ ] & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & -A & I & A & & \\
 & & & & [\ ] & [\ ] & [\ ] & \\
 & & & & & [\ ] & [\ ] & [\ ] \\
 & & & & & & [\ ] & [\ ]
\end{bmatrix}
$$

where the band width is now proportional to the system dimension which is four for the two-dimensional Euler equations. The approximate factorization at least maintains the accuracy of the implicit scheme, although it may affect iterative convergence and stability, see Pulliam [2].

Pulliam and Chaussee [6] introduce a further approximation where they diagonalized the implicit operators in Eq.(9) using the eigensystems of $\mathbf{A}^n=\mathbf{X}(\mathbf{Q})\Lambda_A\mathbf{X}^{-1}(\mathbf{Q})$ and $\mathbf{B}^n=\mathbf{Y}(\mathbf{Q})\Lambda_B\mathbf{Y}^{-1}(\mathbf{Q})$.

Briefly, the eigenvector matrices for $\mathbf{A}^n$ and $\mathbf{B}^n$ are factored out of the implicit operators producing

$$
\begin{aligned}
\mathbf{X}(\mathbf{Q})\,&[I+\alpha\partial_x\Lambda_A]\,\mathbf{X}^{-1}(\mathbf{Q})\mathbf{Y}(\mathbf{Q}) \\
&[I+\alpha\partial_y\Lambda_B]\,\mathbf{Y}^{-1}(\mathbf{Q})\Delta\mathbf{Q}^n \\
&+\mathcal{O}(\Delta t^2)=\mathbf{R}(\mathbf{Q}^n,\mathbf{Q}^{n-1})
\end{aligned}
\tag{10}
$$

The advantage of this scheme is reduced operation counts for the inversions. Instead of banded block matrices, one now deals with matrix multiples and banded scalar inversions, since the bracketed terms are diagonal. Other advantages of this scheme are described in Ref. [2]. One disadvantage is that the maximum time accuracy is now first order (the error term in Eq.(10) is $\mathcal{O}(\Delta t^2)$). Another disadvantage is that the unsteady nature of this scheme is nonconservative due to the solution dependence of eigenvector matrices and the errors induced by the diagonalization. On the other hand, the steady state solutions for this scheme would be identical to the nondiagonalized scheme (the approximation only affect the *implicit side* which at convergence goes to zero leaving $\mathbf{R}(\mathbf{Q}^n,\mathbf{Q}^{n-1})=0$ for both schemes), see Pulliam [6].

A number of other common approximations have been used with varying degrees of consequences in terms of accuracy, efficiency, and stability. Time and space would only allow one to address a small subset of the numerous approximations presented in the literature. One common approximation is to use low order accurate spatial difference operators on the *implicit side* of Eq.(6) and high order operators on the *explicit side* . This is especially advantageous when high order upwind differences are used in conjunction with flux splitting or flux difference methods, see Steger and Warming [7] or Rai [8]. Other approximations include, modified flux Jacobians, incomplete decompositions of the implicit side matrices, approximate artificial dissipation operators and many more. In general, we consider our system of equations to be of the form

$$
\mathcal{L}(\mathbf{Q}^n)\Delta\mathbf{Q}^n+\mathcal{O}(\Delta t^{r-1})\Delta\mathbf{Q}^n=\mathcal{R}(\mathbf{Q}^n,\mathbf{Q}^{n-1}) \tag{11}
$$

where the second term on the *implicit side* is $\mathcal{O}(\Delta t^r)$ since $\Delta\mathbf{Q}^n$ is $\mathcal{O}(\Delta t)$. Of concern is the effect of the error term on stability, iterative convergence to a steady state, and in particular for this paper, the resulting time accuracy when approximations are applied.

We shall now examine an example of the type of approximation discussed above. Consider the model linear problem $u_t + u_x = 0$ (the 1D scalar wave equation) and an implicit time differencing of the form

$$\left[1 + \frac{\vartheta \Delta t}{1+\varphi}\widehat{\delta_x}\right](u^{n+1} - u^n) = -\frac{\Delta t}{1+\varphi}\delta_x u^n \qquad (12)$$

where we allow $\widehat{\delta_x}$ to be distinct from $\delta_x$. The use of different differencing stencils on the *implicit side* and *explicit side* is one way to produce an efficient algorithm and still maintain a minimum level of temporal and spatial accuracy. For example, fourth order five point difference stencils can be used on the *explicit side* . Then either, the same five point differencing on the *implicit side* leading to pentadiagonal inversions or for efficiency three point central differences could be used on the *implicit side* to reduce the inversion requirements to the more efficient tridiagonal operator. This is particularity important for systems such as the Euler and Navier-Stokes equations where the inversion requirements for a fully implicit treatment of a fourth order scheme, even with approximate factorization, would lead to block pentadiagonals in contrast to block tridiagonals, see Ref. [9].

Although, the use of higher order differences on the *explicit side* and lower order on the *implicit side* enhances the efficiency of the scheme it may reduce the time accuracy to $\mathcal{O}(\Delta t)$. As shown below, for the implicit schemes examined here, the only stable time integration is the Euler implicit $\mathcal{O}(\Delta t)$ scheme. If the scheme converges, then the steady-state solution achieves the desired spatial accuracy used on the *explicit side* . To recover second order accuracy one might at first employ either the trapezoidal or the three point backward implicit schemes. Von Neumann analysis of the implicit time differencing schemes for the choices of $\delta_x u_j = (u_{j+1} - u_{j-1})/(2\Delta x)$ (second order central differences with a Fourier signature $\frac{i\sin(j\Delta x)}{\Delta x}$) and $\widehat{\delta_x} u_j = (-u_{j+1} + 8u_{j+1} - 8u_{j-1} + u_{j-2})/(12\Delta x)$ the five point fourth order central difference gives an amplification factor

$$\sigma(\vartheta) = \frac{1 - \frac{\Delta t\,\vartheta}{\Delta x} i \sin j\Delta x \left[\frac{1}{\vartheta} - 1 + \frac{2}{3\vartheta}\sin^2\frac{j\Delta x}{2}\right]}{1 + \frac{\Delta t\,\vartheta}{\Delta x} i\sin j\Delta x} \qquad (13)$$

for $\varphi = 0$ and a similar but more complicated expression for $\varphi \neq 0$. It is easy to show that the norm of the amplification factor $\sigma(\vartheta)$ for all $\Delta t$ and $\Delta x$ is less than one (unconditionally stable) for $\vartheta = 1$ and greater than one (unconditionally instability) for $\vartheta = 1/2$ . Results for the three point backward time implicit scheme also show unconditionally instability for the above combination of operators. This choice of operators produces a stable algorithm for first order Euler implicit and unstable for trapezoidal implicit and 3pt backward implicit. Thus the only stable method is at best only first order in time. As we shall see below, second order time accuracy can be recovered by employing the diagonal algorithm (using scalar pentadiagonal operators to obtain a stable integration) and subiteration to regain time accuracy.

## Subiteration to Second Order Accuracy

In the case of time accurate computations, the second order fully implicit scheme would seem to be the proper choice. We have seen above that approximate factorization can make such a scheme reasonably efficient and does not reduce the time accuracy below second order. In practice though, even the implicit factor algorithm may be computationally expensive and in the case of very sophisticated spatial difference operators (e.g., fourth order accuracy leads to block penta-diagonal inversions) the cost may be prohibitive.

One way to take advantage of simplifying approximations to the *implicit side* and still maintain a desired accuracy is to employ a subiteration time stepping procedure, for example see Rai [8]. We shall examine such a process and develop improvements, analysis and optimization procedures for its use.

The goal is to obtain a second order in time difference approximation in time to Eq.(1)

$$\begin{aligned}\Delta t\left[\frac{\partial \mathbf{Q}}{\partial t} + F(\mathbf{Q})\right] &\approx \Delta\mathbf{Q}^n + \frac{\vartheta\Delta t}{1+\varphi}F(\mathbf{Q}^{n+1}) = \\ &\frac{(\vartheta-1)\Delta t}{1+\varphi}F(\mathbf{Q}^n) + \frac{\varphi}{1+\varphi}\Delta\mathbf{Q}^{n-1} + \mathcal{O}(\Delta t^3)\end{aligned} \qquad (14)$$

for proper choices of $\varphi$ and $\vartheta$.

Neglecting for now the error term, rewrite Eq.(14) substituting a new iterative index $p+1$ for the $n+1$ terms and add $\mathbf{Q}^p$ to both sides of the equality, giving

$$\begin{aligned}\Delta\mathbf{Q}^p + \frac{\vartheta\Delta t}{1+\varphi}F(\mathbf{Q}^{p+1}) &= \\ -(\mathbf{Q}^p - \mathbf{Q}^n) &+ \frac{(\vartheta-1)\Delta t}{1+\varphi}F(\mathbf{Q}^n) \\ &+ \frac{\varphi}{1+\varphi}\Delta\mathbf{Q}^{n-1}\end{aligned} \qquad (15)$$

Linearizing about $\mathbf{Q}^p$ and writing in "Delta Form" we have

$$\left[I + \frac{\vartheta \Delta t}{1+\varphi} F'(\mathbf{Q}^p)\right] \left(\mathbf{Q}^{p+1} - \mathbf{Q}^p\right) = -\frac{\Delta t}{1+\varphi} F(\mathbf{Q}^p) - \left[\mathbf{Q}^p - \frac{1+2\varphi}{1+\varphi}\mathbf{Q}^n + \frac{\varphi}{1+\varphi}\mathbf{Q}^{n-1}\right] \tag{16}$$

with $F'(\mathbf{Q}^p) = \partial F / \partial \mathbf{Q}$.

Equation (16) is the basic subiteration time advance scheme which will yield second order time accuracy independent of the choice of $F'(\mathbf{Q}^p)$ if the subiteration process converges. The $n$and $n-1$ terms are evaluated from previous time levels and after iterating $p$ times the solution at time level $n+1$ will be taken from the most recent $\mathbf{Q}^{p+1}$. Choosing $\varphi = \frac{1}{2}$ and $\vartheta = 1$, in the limit (assuming the iterative process converges) $\mathbf{Q}^{p+1} = \mathbf{Q}^p$, setting $\mathbf{Q}^{n+1} = \mathbf{Q}^{p+1}$ we have

$$\frac{3\mathbf{Q}^{n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t} + F(\mathbf{Q}^{n+1}) = 0 \tag{17}$$

which is a second order in time fully implicit approximation to Eq. (1).

Equation (16) can be written in a form similar to Eq.(7)

$$\mathcal{L}(\mathbf{Q}^p)\Delta\mathbf{Q}^p = \mathcal{R}(\mathbf{Q}^p, \mathbf{Q}^n, \mathbf{Q}^{n-1}) \tag{18}$$

where $\mathcal{R}(\mathbf{Q}^p, \mathbf{Q}^n, \mathbf{Q}^{n-1})$ represents the fully implicit approximation to Eq. (1). We can now consider the *implicit side* as the error term of order $\mathcal{O}(\Delta t^r)$ with $r \geq 2$ .

Analysis of Eq.(18) will determine conditions on the approximate implicit operator $\mathcal{L}(\mathbf{Q})$. The first criteria on $\mathcal{L}(\mathbf{Q})$ is that the resulting iterative scheme does converge. The additional requirement is that the local subiteration process be performed until the error is second order in time. The convergence of the iterative process and the accuracy requirement are directly linked, if the iterative scheme fails to converge the error term will be large.

The solution procedure when employing Eq.(18) is as follows. Given a choice of the approximate operator $\mathcal{L}(\mathbf{Q})$ and an initial solution $\mathbf{Q}^n$ (probably from one iteration of a first order scheme) a given number of subiterations $p = 0, 1, \ldots, np$ are performed (typically 3, but some analysis below will provide guidelines for a proper choice). Comparing Eq.(16) and Eq.(6) we see a change of variable ($n$ to $p$) and the addition of a few terms to the *explicit side* . If one assumes that a given numerical code represents the basic operators $\mathcal{L}$ and $\mathcal{R}$, then the addition of the subiteration process is a trivial source term included in the numerical code. The amount of extra computational work is directly proportional to the number of subiterations required. Note that all operations of the original numerical code, including boundary operators are applied to $\mathbf{Q}^p$. After the subiteration process is adequately converged the solution $\mathbf{Q}^{n+1}$ is updated.

<u>Analysis of Subiteration Scheme</u>

The *implicit side* operator $\mathcal{L}$ in Eqs. (16) or (18) can be chosen as any scheme which will guarantee convergence of the iteration in $p$. Rewriting Eq.(16) in terms of a general implicit operator $\mathcal{L}(\mathbf{Q}^p)$ we have

$$\mathcal{L}(\mathbf{Q}^p)\left(\mathbf{Q}^{p+1} - \mathbf{Q}^p\right) = -\frac{\Delta t}{1+\varphi} F(\mathbf{Q}^p) - \left[\mathbf{Q}^p - \frac{1+2\varphi}{1+\varphi}\mathbf{Q}^n + \frac{\varphi}{1+\varphi}\mathbf{Q}^{n-1}\right] \tag{19}$$

a perturbation $\mathbf{e}^p = \mathbf{Q}^p - \overline{\mathbf{Q}}$ is introduced, where $\overline{\mathbf{Q}}$ satisfies the implicit scheme to our desired time accuracy. We can examine the stability and convergence of the subiteration scheme by analyzing

$$\mathcal{L}(\overline{\mathbf{Q}} + \mathbf{e}^p)(\mathbf{e}^{p+1} - \mathbf{e}^p) = -\frac{\Delta t}{1+\varphi} F(\overline{\mathbf{Q}} + \mathbf{e}^p) - \left[\mathbf{e}^p + \overline{\mathbf{Q}} - \frac{1+2\varphi}{1+\varphi}\mathbf{Q}^n + \frac{\varphi}{1+\varphi}\mathbf{Q}^{n-1}\right] \tag{20}$$

which for small $\mathbf{e}$ and linearized about $\overline{\mathbf{Q}}$ reduces to

$$\mathcal{L}(\overline{\mathbf{Q}})(\mathbf{e}^{p+1} - \mathbf{e}^p) = -\frac{\Delta t}{1+\varphi}\left[F(\overline{\mathbf{Q}}) + F'(\overline{\mathbf{Q}})\mathbf{e}^n + \mathcal{O}(\Delta t^2)\right] - \left[\mathbf{e}^p + \overline{\mathbf{Q}} - \frac{1+2\varphi}{1+\varphi}\mathbf{Q}^n + \frac{\varphi}{1+\varphi}\mathbf{Q}^{n-1}\right] \tag{21}$$

Ignoring the linearization error (which is $\mathcal{O}(\Delta t^3)$) and collecting terms, we have

$$\mathbf{e}^{p+1} = \mathcal{L}(\overline{\mathbf{Q}})^{-1}\left[\mathcal{L}(\overline{\mathbf{Q}}) - \left(I + \frac{\Delta t}{1+\varphi}F'(\overline{\mathbf{Q}})\right)\right]\mathbf{e}^p \tag{22}$$

The requirement for stability and convergence of the subiteration scheme is that

$$\parallel \mathcal{L}(\overline{\mathbf{Q}})^{-1}\left[\mathcal{L}(\overline{\mathbf{Q}}) - \left(I + \frac{\Delta t}{1+\varphi}F'(\overline{\mathbf{Q}})\right)\right] \parallel \leq 1 \tag{23}$$

implying that

$$\mathcal{L} \approx \left(I + \frac{\Delta t}{1+\varphi}F'(\overline{\mathbf{Q}})\right) \tag{24}$$

Note this requires that the appropriate values of $\varphi$ be used in the approximate *implicit side* operator. In fact, an improper implementation will suffer in terms of convergence of the subiteration process and in some cases stability. This then, gives some guidance in the choice of $\mathcal{L}$. Again, the criteria beyond convergence of the subiterative process, is a choice of $\mathcal{L}$ which is efficient in terms of its implementation while not requiring excessive subiterations for the desired accuracy.

The requirement for accuracy is easily defined in terms of Eq.(19) as having

$$\mathcal{L}(\mathbf{Q}^n)(\mathbf{Q}^{n+1} - \mathbf{Q}^n) \approx \mathcal{O}(\Delta t^3) \tag{25}$$

fall within the truncation error of a second order accurate scheme. This term is exactly the residual of the integration operator which most numerical codes use as a measure of convergence. Error estimates should be obtainable from measures of the subiteration residual and/or its rate of convergence. In future work, we shall be investigating some of the classical ODE techniques for error estimation in an attempt to estimate the number of subiterations required for a certain level of solution accuracy. In practice one finds that two to three subiterations gives adequate accuracy for most problems. Examples are given below in the results sections.

## Results

The diagonal form of the implicit Beam-Warming scheme, as define in [6], will be used as the base integration scheme $\mathcal{L}$ with fourth order spatial differencing and artificial dissipations on the *explicit side* as defined in Ref. [2]. This is an efficient and practical algorithm which is widely used in CFD. The subiteration process defined above in Eq.(16), with $\vartheta = 1.0$ and $\varphi = 0.5$, i.e. the second order backward implicit scheme, will be used unless noted below. In some of the cases a full second order block tridiagonal algorithm will be used for comparison purposes. The implicit code, ARC2D [2] will be used for the results presented below. ARC2D can be used in either the diagonal form, a block tridiagonal option, or a block pentadiagonal option. Typical CPU time in microseconds per iteration per grid point on a CRAY YMP for various options are shown in Table 1, where both periodic and nonperiodic cases are examined.

| $10^{-6}$ seconds/time step/grid point | | |
|---|---|---|
| Option | Nonperiodic | Periodic |
| Diagonal | 6.3 | 8.3 |
| Block Tri | 11.6 | 19.2 |
| Block Penta | 17.5 | $\approx 40.0$ |

**Table 1.** Timing Comparison From ARC2D.

Note that the block tridiagonal option is approximately a factor of 1.84 for nonperiodic case and 2.3 in the periodic case more CPU time per time step per grid point than the diagonal option and approximately 2.77 more expensive for the block pentadiaonal option. The result of a factor of 4.8 for the periodic block pentadiaonal is an approximation based on operation count estimates.

Pulliam and Chaussee [6] show that the use of the diagonal form of the implicit approximate factorization scheme leads to a nonconservative behavior for unsteady shock motions. They demonstrate such a problem for the 1D shock tube problem solved using the 1D Euler equations. Figure 1a shows a similar result where the second order fully implicit (Block tridiagonal) scheme, the exact solution, the first order diagonal algorithm, and a corrected second order subiteration results are shown for a typical case. An initial pressure jump of 10 is propagated to the same physical time in all four results. It is quite evident that the first order diagonal scheme results in an improper shock jump and shock speed. The two second order results compare quite well with the exact solution getting both the shock jump and the speed correct. Figure 1b shows a closer view of the solution in the vicinity of the shock. For this case three subiterations were used, although two would have been sufficient. The subiteration process not only corrected the time accuracy but also restored the time conservation properties of the original scheme.

The second example is 2D Navier-Stokes flow past a circular cylinder at a Mach number, $M_\infty = 0.1$ and a Reynolds number, $Re = 120$. Fourth order spatial differences were used along with low levels of artificial dissipation which did not interfere with the physical dissipation. At these conditions the flow is in a state of unsteady vortex shedding behind the cylinder with a shedding frequency (Strouhl number) $St \approx 0.186$ which is consistent with published results. Figure 2 contrasts results using the 2D Navier-Stokes code ARC2D [2] comparing $St$ versus $\Delta t$ for the standard algorithm (first order Diagonal implicit) and subiteration scheme results using both two and three subiterations. There is a large variation of the predicted $St$ with decreasing $\Delta t$ for the first order diagonal algorithm, where adequate results are only found for $\Delta t \leq 0.05$, see Fig. 2. In the case of just two subiterations, comparable accuracy is obtained at a $\Delta t = 0.4$ a factor of eight in time step balanced against only a factor of two in computational work. For the three subiteration case, second order time accurate results are obtained at $\Delta t \approx 0.5$, a factor of ten larger than the standard algorithm while only increasing the computational work by a factor of three. This translates into a factor of three to four improvement in efficiency over a small time step first order diagonal result, comparable savings have been realized for a variety of other cases. A second order in time fourth order in space block penta diagonal solution would be comparable in computational cost to a three subiteration diagonal algorithm result. But, as we noted above and will see below, there are other advantages to the subiteration process in terms of time conservation and robustness which make it the algorithm of choice.

Another example of the effectiveness of the subiteration process is its affect on boundary conditions. Boundary condition operators can be highly nonlinear and quite complicated, e.g. characteristic inflow/outflow conditions. In general, their implementation in implicit codes is handled explicitly, although implicit treatment is advised whenever practical. A common occurrence with the use of explicit boundary conditions is the appearance of pressure and density oscillations near the boundary which are a direct result of the time lagged nature of the boundary schemes in relation to the implicit update of the interior. Although numerical dissipation mitigates possible problems associated with this, especially in the transient development of a flowfield, it is not unusual for this phenomena to adversely affect both the stability and convergence of typical computations. The possible time step or convergence acceleration to a steady state may be restricted. For unsteady problems, initial condition dependent transients may take a long time to relax and time steps for tracking unsteady features may be restricted. Subiteration can be used to alleviate this problem and thereby enhance to robustness and convergence of a numerical code. Take for example the case of the artificial wake cut behind a 2D airfoil in a "C mesh" grid, see Fig. 3. The wake cut is an computational convenience along which boundary values are typically provided from averages of the values from upper and lower points on either side of the cut. (It should be pointed out that fully implicit treatments of the wake cut are not difficult to implement.) If explicit treatment is used in updating wake cut values it is not unusual for the transient solution in that region to exhibit wild oscillations of pressure and density which may cause computational stability and convergence problems. Figure 4 shows trailing edge pressure contours from the early time development of an inviscid "C Mesh" NACA 0012 airfoil computation at $M_\infty = 0.8$ and $\alpha = 1.25$ using the implicit finite difference code ARC2D [2] and explicit treatment of the wake cut boundary. In the case where the standard algorithm is used without subiteration, the wake region exhibits large oscillation of pressure, as seen in Fig. 4a. A larger time integration step would produce negative pressure and "blow up." When subiterations are used the smooth flow shown in Fig. 4b results and larger time steps can be used to accelerate the solution convergence. In this case six subiterations steps per time advance produces the smooth contours, but three subiterations are sufficient to produce a more robust transient development.

## Summary

Until recently, numerical algorithm development in CFD has concentrated on steady state or slowly varying unsteady applications. The next generation of algorithm development will concentrate on unsteady time accurate computation. Unsteady wave propagation for aeroacoustics, computational electromagnetics, unsteady shock boundary-layer interactions, complicated vortex surface interactions, large eddy simulation, etc, are just a short list of the important problem areas. The requirements of mesh and time resolution balanced against limited computer resources will always lead to simplifications. One technique to recover the time accuracy of implicit approximations is through the use of subiteration. This paper attempts to delineate some common approximations, examine their effect of stability, iterative convergence, and time accuracy. The subiteration approach can be used to recover time accuracy without increasing computational work (in most cases producing substantial savings). Criteria for implicit approximation choices, automatic time step procedures and subiteration limits are the next areas to be investigated.

References

1 Steger, J. L., "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils," AIAA Paper 77-665, 1977.

2 Pulliam, T.H., "Efficient Solution Methods for The Navier-Stokes Equations," Lecture Notes for the von Kármán Institute For Fluid Dynamics Lecture Series : Numerical Techniques for Viscous Flow Computation In Turbomachinery Bladings, von Kármán Institute, Rhode-St-Genese, Belgium, 1985.

3 Yoon, S. and Kwak, D., "An Implicit Three-Dimensional Navier-Stokes Solver for Compressible Flows," AIAA Paper 91-1555-CP, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 24-27, 1991.

4 Jespersen, D.C., and Levit, C., "A Computational Fluid Dynamics Algorithm On A Massively Parallel Computer," The Internatinal Journal of Supercomputer Applications, Vol. 3, No. 4, pp. 9-27, 1989.

5 Beam, R. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," J. Comp. Phys., Vol. 22 1976, pp. 87-110.

6 Pulliam, T. H. and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate Factorization Algorithm," J. Comp. Phys.,Vol. 39, p. 347, 1981.

7 Steger J. L. and Warming, R. F., "Flux Vector Splitting of the Inviscid Gas Dynamic Equations with Applications to Finite Difference Methods," J. Comp. Phys., Vol. 40, pp. 263-293, 1981.

8 Rai, M.M., "Navier-Stokes Simulations of Blade-Vortex Interaction Using High-Order Accurate Upwind Schemes," AIAA Paper No. 87-0543, 1987.

9 Pulliam T. H. and Steger J. L., "On Implicit Finite-Difference Simulations of Three- Dimensional Flow," AIAA Journal Vol. 18, 1980 p. 159.


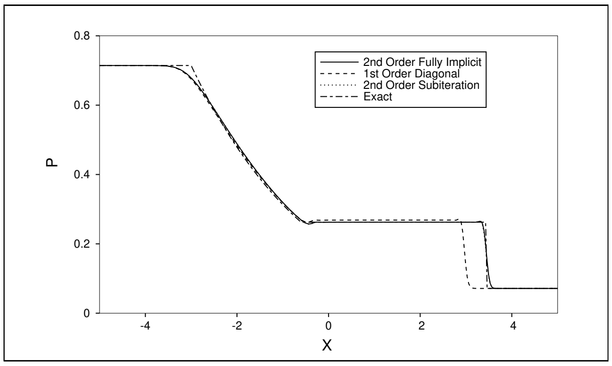


FIGURE 1A. 1D Euler Shock Tube Results.

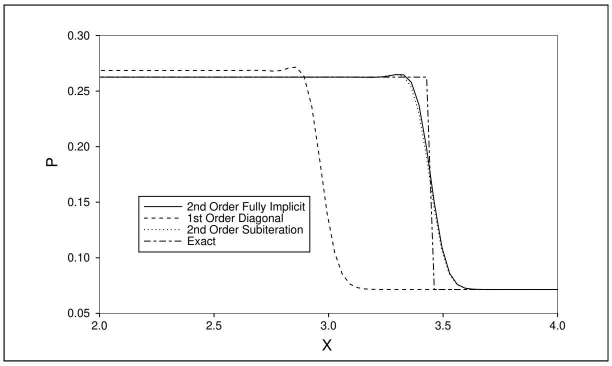


FIGURE 1B. Close Up Of 1D Euler Shock Tube Results.

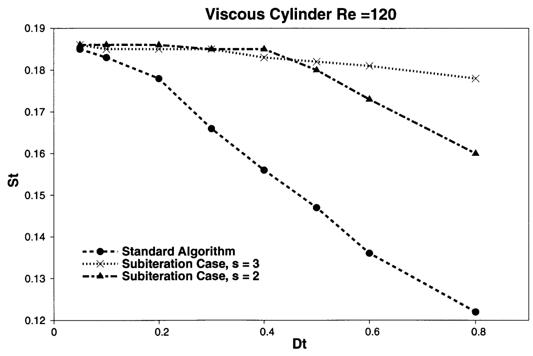


FIGURE 2. Strouhl Number vrs $\Delta t$ for 2D Cylinder at $M_\infty = 0.1$ and $Re = 120$.

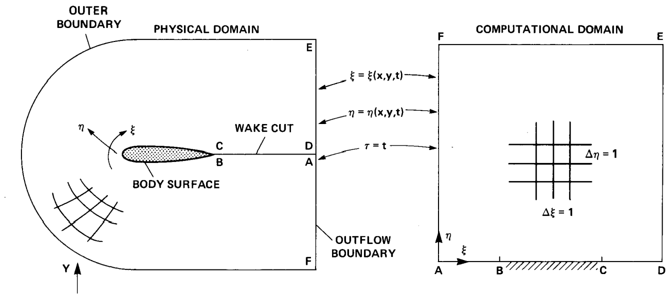


FIGURE 3. "C Mesh" Physical to Computational Space Mapping Showing Wake Cut.

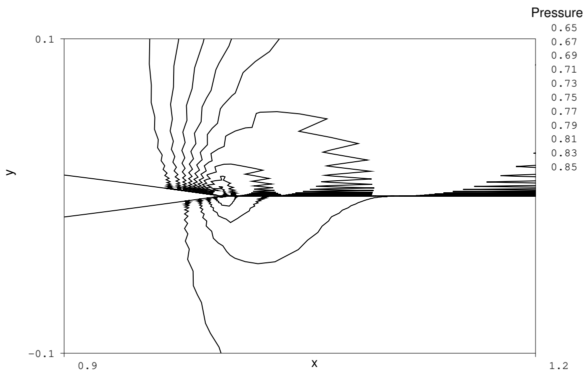


FIGURE 4A. Trailing Edge Pressure Contours Showing Explicit BC Wake Cut Oscillations.

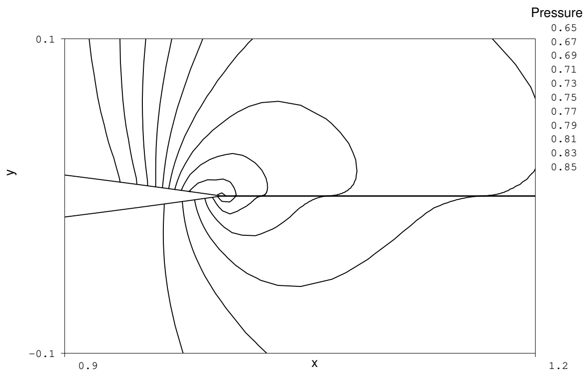


FIGURE 4B. Trailing Edge Pressure Contours Showing Effect of 6 Subiterations.