A98-31699

ICAS-98-7,4,1

IMPROVED APPROXIMATE FACTORISATION ALGORITHM FOR THE STEADY SUBSONIC AND TRANSONIC FLOW OVER AN AIRCRAFT WING

Eddie Ly

Department of Mathematics
RMIT University
GPO Box 2476V, Melbourne, Victoria 3001, Australia.

Abstract

This paper describes an approximate factorisation (AF) algorithm, for the solution of the two-dimensional steady Subsonic Small Disturbance (SSD) Equation. The algorithm employs internal Newton iterations, at each time level, to achieve time accuracy and computational efficiency. For steady flow computations, an "artificial" time-dependent derivative term is introduced into the SSD Equation to incorporate temporal numerical dissipation. This term is implemented for variable time stepping, to allow for step size cycling to accelerate convergence to steady-state. In the AF algorithm, the reduced potential is determined via an iterated finite difference scheme, in which the coefficient matrix acting on the unknown reduced potential difference is approximately factored. The reduced potential is then determined via the solution of two tridiagonal linear systems. The size of the time step is cycled in a predetermined fashion, with the minimum and maximum time steps based on the spatial grid spacing. Results for steady subsonic flow over an aerofoil, with a 10% thick double parabolic arc profile, inclined at 0° and 1° angle of attack (AOA) are presented. To demonstrate that the method works for nonlinear partial differential equations, results for three-dimensional steady subsonic and transonic flow over a F5 wing are presented.

Introduction

The purpose of this paper is to describe the development of an AF algorithm, that solves the steady SSD Equation with nonreflecting far-field boundary conditions. The AF algorithm was originally developed by Ballhaus, Jameson and Albert[1] in 1977 for steady transonic flow computations. It was shown by Batina[2,3] (see also Gear[10,11] and Gear, Ly and Phillips[12]) to be very robust for either steady or oscillatory transonic flow problems. Our aim is to develop an improved version of this algorithm, in the sence that this improved algorithm should lead to a better convergence rate.

An "artificial" time-dependent term is introduced into the steady SSD Equation to incorporate temporal numerical dissipation, this is sometimes referred to as the method of false transients.[7] In the AF algorithm, the reduced potential is determined via an iterated finite difference scheme, in which the coefficient matrix acting on the unknown reduced potential difference is approximately factored. The reduced potential is then determined via the solution of two tridiagonal linear systems. The size of the time step is cycled in a predetermined fashion, with the minimum and maximum time steps based on the spatial grid spacing, in order to achieve the fastest convergence rate.[6,14] A numerical stability analysis of the iterated finite difference scheme is carried out. This allows the choice of an appropriate time differencing strategy. The proposed method also works for nonlinear partial differential equations.

Steady results, generated by the use of the scheme, for the subsonic flow past a double parabolic arc aerofoil are presented, and comparison with the analytic solution is made. To demonstrate that this method also works for nonlinear partial differential equations, results for three-dimensional steady transonic flow over a F5 wing are also generated.

A nearly planar wing is immersed in a steady, isentropic and inviscid flow. The steady SSD Equation for the reduced potential in three dimensions, $\phi(x, y, z)$, may be written in conservation form as

$$\frac{\partial}{\partial x}\Big(\left(1 - M^2\right)\phi_x\Big) + \frac{\partial}{\partial y}\phi_y + \frac{\partial}{\partial z}\phi_z = 0, \qquad (1)$$

where $\phi_x = \partial\phi/\partial x$, similarly applies to ϕ_y and ϕ_z derivative terms. Here (x, y, z) represents a nondimensional rectangular Cartesian coordinate system with the coordinates based on an aerofoil chord length as the characteristic lengthscale, and M is the freestream Mach number. In nondimensional terms the total velocity potential, $\Phi(x, y, z)$, and the fluid velocity vector, $\mathbf{v} = (u, v, w)$, are given by $\Phi = x + \phi$ and $\mathbf{v} = \nabla(x + \phi)$, respectively.

Since numerical computation of steady flow in an unbounded region is performed on grids with finite dimensions, nonreflecting far-field boundary conditions[8,9,13] must be employed, so that the unphysical effects created by the far-field computational boundaries can be minimised. For Equation (1) the following boundary conditions are imposed,

$$\begin{array}{rll}
\phi_x = 0 & \text{streamwise boundaries,} & (2)\\
\phi_y = 0 & \text{right spanwise boundary,} & (3)\\
\phi_z = 0 & \text{vertical boundaries,} & (4)\\
\phi_y = 0 & \text{symmetry plane,} & (5)\\
\langle\phi_x\rangle = 0 & \text{wake,} & (6)\\
\langle\phi_z\rangle = 0 & \text{wake,} & (7)
\end{array}$$

where $\langle \circ \rangle$ indicates the jump in the indicated quantity across the wake. On the wing the flow must be tangent to the surfaces at all points,[5,16] thus the following physical boundary condition is imposed at the mean plane of the wing,

$$\phi_z\left(x, y, 0^{\pm}\right) = h_x^{\pm}\left(x, y\right). \qquad (8)$$

The upper and lower wing surfaces are defined as $z = h^{\pm}(x, y)$, respectively.

Method of False Transients

The steady-state solution is obtained by the method of false transients, see Beam and Warming,[4,17] Davis[7] ,and Ly, Gear and Phillips,[14] in which an artificial time-dependent term, ϕ_τ, is added into Equation (1) to incorporate the temporal numerical dissipation. Since the purpose of this paper is to investigate ways to enchance convergence rate of the numerical scheme to steady-state, we consider only two-dimensional flow. That is we assume the wing to have an infinite aspect ratio, so that the flow field around the aerofoil is the same for any cross section perpendicular to the wing. Thus the steady problem is now governed by

$$\phi_\tau = \nabla^2\phi\left(\overline{x}, z\right), \qquad (9)$$

where ∇^2 is the Laplacian in the $(\overline{x}, z)$ domain and $\overline{x} = x/\sqrt{1-M^2}$. The boundary conditions for (9) are given by (2), (4), (6) and (7), with the physical boundary condition on the wing given by

$$\phi_z\left(x, 0^{\pm}\right) = h_x^{\pm}\left(x\right). \qquad (10)$$

Here we consider a rectangular computational region with all the computational boundaries as depicted in Figure 1. The locations of these boundaries are determined mainly by computer storage and accuracy requirements. The mean plane of the aerofoil lies along the x-axis from $x = 0$ to $x = 1$.

Since the boundary conditions (2) to (8) are time-independent for the steady problem and provided the solution converges, we expect that $\phi_\tau \to 0$. This indicates that solving the boundary value problem governed by Equation (9) will generate solutions that also satisfy the two-dimensional form of (1). The price paid is the loss of the true transient solution, but this is not significant to us, since the objective of this work is to develop a numerical scheme that will generate the steady-state solution at an enchanced convergence rate.

Factored Scheme

Let the "artificial" time scale, τ, be discretised as $\tau = n\Delta\tau$ where $\Delta\tau$ is the discrete increment of τ, and n denotes the time-level (or iteration number). Thus $\phi(\tau) = \phi(n\Delta\tau) = \phi^n$. Here the spatial dependence has been temporarily suppressed. The time difference formula, written in a general Padé form[17], is given by

$$\Delta\tau\frac{\partial\phi^n}{\partial\tau} = \frac{(1+a)\overrightarrow{\Delta_\tau} - a\overleftarrow{\Delta_\tau}}{1+b\overrightarrow{\Delta_\tau}}\phi^n + (b-a-1/2)\,O\left(\Delta\tau\right)^2 + O\left(\Delta\tau\right)^3, \qquad (11)$$

where $\overrightarrow{\Delta_\tau}$ is defined as the foward time difference operator

$$\overrightarrow{\Delta_\tau}\phi^n = \phi^{n+1} - \phi^n, \qquad (12)$$

and $\overleftarrow{\Delta_\tau}$ the backward time difference operator

$$\overleftarrow{\Delta_\tau}\phi^n = \phi^n - \phi^{n-1}. \qquad (13)$$

Formula (11) includes the following well-known rules,

$a = 0,$	$b = 1/2,$	trapezoidal formula,
$a = 0,$	$b = 1,$	Euler implict,
$a = 1/2,$	$b = 1,$	three-point backward,
$a = 0,$	$b = 0,$	Euler explicit,
$a = -1/2,$	$b = 0,$	leap frog.

In the next section we will show that the Euler explicit and leap frog rules are inappropriate for the solution process as they will cause scheme instabilities.

Inserting formula (11) into (9), and after clearing fractions and rearranging gives

$$\left(1 - \Delta\overline{\tau}\nabla^2\right)\overrightarrow{\Delta_\tau}\phi^n = \alpha\overleftarrow{\Delta_\tau}\phi^n + \Delta\overline{\tau}\lambda\mathcal{R} + (b-a-1/2)\,O\left(\Delta\tau\right)^2 + O\left(\Delta\tau\right)^3, \qquad (14)$$

where the residual $\mathcal{R} = \nabla^2\phi^n$. The scheme constants are $\Delta\overline{\tau} = b\Delta\tau/(1+a)$, $\lambda = 1/b$ and $\alpha = a/(1+a)$ with $a \neq -1$. The form of (14) suggests an approximate factorisation of the terms on the left, omitting second-order terms in $\Delta\tau$, we obtain

$$\left[1 - \Delta\overline{\tau}\frac{\partial^2}{\partial\overline{x}^2}\right]\left[1 - \Delta\overline{\tau}\frac{\partial^2}{\partial z^2}\right]\overrightarrow{\Delta_\tau}\phi^n = \alpha\overleftarrow{\Delta_\tau}\phi^n + \Delta\overline{\tau}\lambda\mathcal{R}. \qquad (15)$$

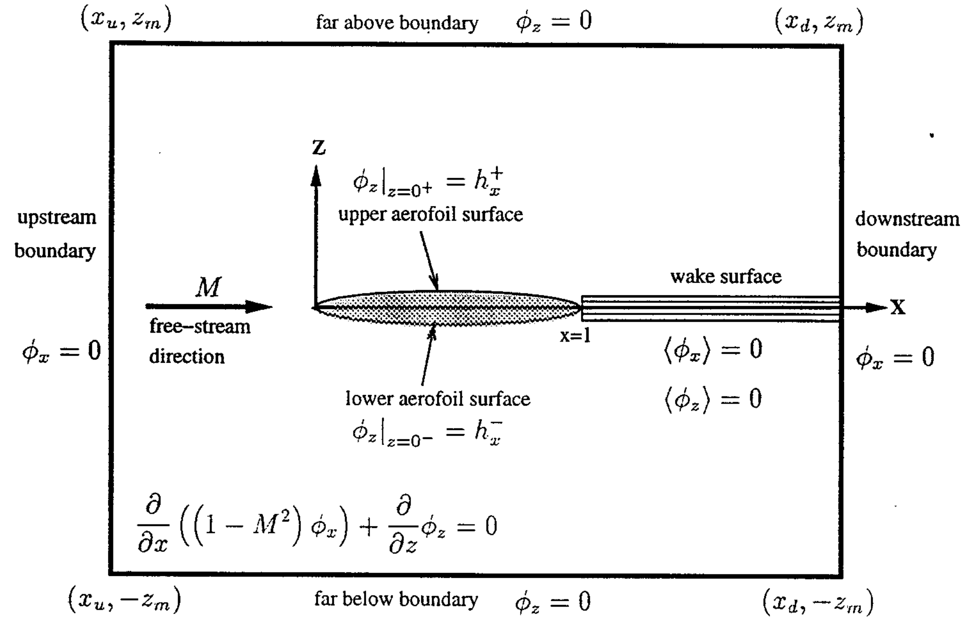
(x_u, z_m)
far above boundary $\phi_z = 0$
(x_d, z_m)
Z
$\phi_z|_{z=0^+} = h_x^+$
upper aerofoil surface
upstream boundary
downstream boundary
M
free–stream direction
wake surface
X
$\phi_x = 0$
x=1
$\langle\phi_x\rangle = 0$
$\phi_x = 0$
lower aerofoil surface
$\phi_z|_{z=0^-} = h_x^-$
$\langle\phi_z\rangle = 0$
$\frac{\partial}{\partial x}\left(\left(1-M^2\right)\phi_x\right)+\frac{\partial}{\partial z}\phi_z = 0$
$(x_u, -z_m)$
far below boundary $\phi_z = 0$
$(x_d, -z_m)$

FIGURE 1: A mathematical model for two-dimenional steady flow computations.

The factored scheme (15) can then be solved in the following manner, iteratively,

$$\left[1-\Delta\overline{\tau}\frac{\partial^2}{\partial\overline{x}^2}\right]\psi = \alpha\overleftarrow{\Delta_\tau}\phi^n + \Delta\overline{\tau}\lambda\mathcal{R}, \qquad (16)$$

$$\left[1-\Delta\overline{\tau}\frac{\partial^2}{\partial z^2}\right]\overrightarrow{\Delta_\tau}\phi^n = \psi, \qquad (17)$$

$$\phi^{n+1} = \phi^n + \overrightarrow{\Delta_\tau}\phi^n, \qquad (18)$$

where ψ is a dummy temporal difference. In each iteration, a new approximation to ϕ is found by systematically solving Equation (16) for ψ, Equation (17) for $\overrightarrow{\Delta_\tau}\phi^n$, and then using (18) to obtain ϕ^{n+1}. When the solution converges, $\overrightarrow{\Delta_\tau}\phi^n \to 0$, and the numerical solution is then given by $\phi \approx \phi^n$. The method is potentially fast since the solution process is fully vectorised, and variable time stepping is incorporated.

In general Equations (16) to (18) form a three time-level iterative scheme. In order to further simplify the current problem (and make it more manageable in the context of stability analysis in the next section), only a two-time level scheme is considered (setting $\alpha = 0$). All the spatial derivatives in Equations (16) and (17) are discretised using second-order accurate central differencing approximations, since the two-dimensional steady SSD Equation (that is Equation (1) without the y-derivative term) is of elliptic type. This then leads to Equations (16) and (17) forming two tridiagonal systems of equations. Usually Equation (14) forms a linear, block tridiagonal system of equations. However the factored scheme (15) reduces a formidable matrix inversion problem to a series of small bandwidth (tridiagonal in this case) matrix inversion problems that have efficient solution algorithms. For an accurate factorisation the time step used in the iterative scheme must be small relative to the spatial grid spacings. This ensures that each linear system is strongly diagonally dominant. For a detailed description of how the boundary conditions are incorporated into the finite difference scheme, see Gear$^{(10)}$ and Murman$^{(16)}$.

Stability Analysis and Improved Algorithm

Stability Analysis of Factored Scheme

In this subsection, we will determine the stability characteristics of the factored scheme using the von Neumann method of stability analysis. In the analysis, the errors distributed along the grid points at one time-level are expanded as a finite Fourier series. The stability or instability of the finite difference equation is then determined by considering whether a Fourier component of the error distribution decays or amplifies on progressing to the next time-level.

For simplicity we assume the solution of the difference equation is spatially periodic of the form

$$\begin{aligned}\phi^n_{jk} &\equiv \phi\left(n\Delta\overline{\tau}, j\Delta\overline{x}, k\Delta z\right)\\ &= \varphi^n\left(\kappa_1,\kappa_2\right)\exp\left\{\mathrm{i}\left(\kappa_1 j\Delta\overline{x}+\kappa_2 k\Delta z\right)\right\}, \qquad (19)\end{aligned}$$

where $\mathrm{i} = \sqrt{-1}$, φ^n is the Fourier coefficient and κ_1, κ_2 are the Fourier variables. The streamwise and vertical indices of the grid points, respectively, are denoted by j and k. Upon substitution of (19) into the discretised form of (15) (recalls that $\alpha = 0$), we find that in order for

$|\varphi^{n+1}| \leq |\varphi^n|$, the following two inequalities must hold,

$$|\alpha| < (1+\Delta\overline{\tau}P)(1+\Delta\overline{\tau}Q), \qquad (20)$$

$$\lambda \leq 2+\frac{\alpha+1}{2\Delta\overline{\tau}_{\max}}\min\left((\Delta\overline{x})^2,(\Delta z)^2\right), \qquad (21)$$

where the so-called "frequencies" are defined by

$$P = \left[\frac{2}{\Delta\overline{x}}\sin\left(\kappa_1\frac{\Delta\overline{x}}{2}\right)\right]^2, \qquad (22)$$

$$Q = \left[\frac{2}{\Delta z}\sin\left(\kappa_2\frac{\Delta z}{2}\right)\right]^2. \qquad (23)$$

These inequalities are satisfied if $a > -1/2$ and $b \geq 1/2$.

It is interesting to note that the unfactored scheme, represented by Equation (14), also imposes the same restrictions on the values of a and b for a stable solution process. This suggests that the stability characteristics are not deteriorated when the factored scheme is used, and in fact has the advantage that the solution process can be fully vectorised.

Improved Algorithm

Let ϕ be the converged solution and define the error after the nth iteration as $e^n = \phi^n - \phi$. The error distribution is considered to behave in a spatially periodic fashion, in a similar form given by (19). If the error amplification factor, β, is defined to be $\beta = \varphi^{n+1}/\varphi^n$, then by substituting this into (15) we arrive at

$$\beta = 1-\frac{\lambda(P+Q)\Delta\overline{\tau}}{PQ(\Delta\overline{\tau})^2+(P+Q)\Delta\overline{\tau}+1}. \qquad (24)$$

For rapid convergence the time step sequence must be carefully determined. We wish to minimise β for a possible range of P and Q values.(6) Setting $\beta = 0$ and for real values of $\Delta\overline{\tau}$, we must have

$$\lambda \geq 1+\frac{2\sqrt{PQ}}{P+Q}. \qquad (25)$$

We chose $\lambda = 2$, so that this inequality and inequality given by (21) are satisfied. In this case, β will be zero if $\Delta\overline{\tau} = 1/P$ or $\Delta\overline{\tau} = 1/Q$. For each root chosen the error e^n will be reduced as the solution proceeds, since $|\beta| < 1$ for all possible P and Q values.

Suppose we can construct a discrete set of variable time steps in the form $\Delta\overline{\tau}_j = 1/P_j$ ($j = 1,2,\ldots,N$). From now on, readers should be aware that j does not represent the streamwise index of the grid points. Since $|Q-P_j| < |Q+P_j|$ we find that for each j value β takes the maximum value given by

$$\beta_j(P) = \frac{|P-P_j|}{P+Pj}. \qquad (26)$$

This represents the worst case in the context of error reduction. When the algorithm is executed for N iterations, the error should be reduced by a factor of $\prod_{j=1}^{N}\beta_j$, thus the mean reduction of error per iteration is approximately given by

$$\overline{\beta} = \left[\prod_{j=1}^{N}\beta_j\right]^{1/N}. \qquad (27)$$

For convergence we require the error corresponding to all frequencies to decrease successively, and the speed of convergence will depend on how small β is. Thus the aim is to choose a sequence of time steps which minimise $\overline{\beta}$. If P values are ranged from P_1 to P_N (from (22), $P \in [1,(2/\Delta\overline{x})^2]$) and using the relationship $\Delta\overline{\tau}_j = 1/P_j$, the following geometric time step sequence is derived,

$$\Delta\overline{\tau}_j = \left[\frac{\Delta\overline{x}}{2}\right]^{\frac{2(j-1)}{N-1}} \quad \text{for } j = 1,2,\ldots,N, \qquad (28)$$

which cycles through the full range of possible $\Delta\overline{\tau}_j$ values.

The results for two-dimensional subsonic flow over a double parabolic arc aerofoil are computed using a computational region of the size $(x_u, x_d, \pm z_m) = (-1,2,\pm 2)$, see Figure 1 for more details. The grid spacings are $\Delta x = 1/50$ and $\Delta z = 1/60$, that is using 50 and 60 grid points per chord length in the streamwise and vertical directions, respectively. In this case it was determined that $N = 13$ gives the smallest $\overline{\beta}$ values of 0.708482. The distribution of $\overline{\beta}$ versus frequency P is represented by the thin line in Figure 2. Thus using the time step sequence given by (28), the mean reduction of error per iteration in all frequencies is less than 0.708482. Note that with a constant time step some frequencies in the error reduce at a rate just less than one, and hence die out very slowly. However using (28) we expect the sequence to produce a dramatic speed up in the convergence of the solution process.

Sequence With Repeated Endpoints. It is clearly shown in Figure 2 (thin line) that large $\overline{\beta}$ values occur at the ends of the frequency range. This unfavourable behaviour can be eliminated by repeating the endpoints of the sequence, since if P lies in between, say P_j and P_{j+1}, then the major reduction of error will come from β_j and β_{j+1}, and β will be largest if P is such that $\beta_j = \beta_{j+1}$. Using (26) this requires $P^2 = P_jP_{j+1}$, and hence β_j and β_{j+1} become

$$\beta_j = \beta_{j+1} = \frac{\left|\sqrt{P_{j+1}/P_j}-1\right|}{\sqrt{P_{j+1}/P_j}+1}. \qquad (29)$$

So we enforce $P_1 = P_2$ and $P_{N-1} = P_N$, and hence

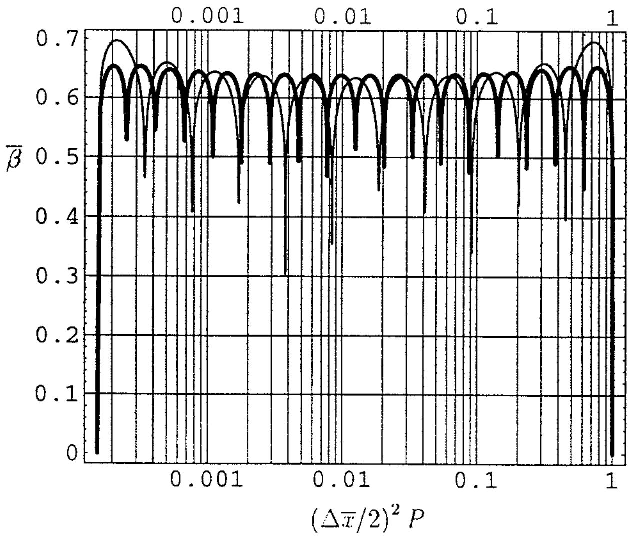

FIGURE 2: Distributions of $\overline{\beta}$ for sequence without (thin line) and with (thick line) repeated endpoints using $\Delta x = 1/50$.

obtaining the following time step sequence,

$$\begin{aligned} \Delta\overline{\tau}_1 &= 1, \\ \Delta\overline{\tau}_j &= \left[\frac{\Delta\overline{x}}{2}\right]^{\frac{2(j-2)}{N-3}} \quad \text{for } j = 2, 3, \ldots, N-1, \qquad (30) \\ \Delta\overline{\tau}_N &= (\Delta\overline{x})^2/4. \end{aligned}$$

Using this sequence we can obtain a small $\overline{\beta}$ value of 0.665443 with $N = 22$, as represented by the thick line in Figure 2.

We also obtain similar results to those represented by sequences (28) and (30) when $\Delta\overline{\tau}_j = 1/Q_j$ and $\Delta z = 1/60$ are used. By this stage we have four different smallest $\overline{\beta}$ values to choose from, as tabulated in Table 1, and we obviously select the smallest one to construct the time step sequence for our computations.

Computed Results

Two-Dimensional Subsonic Flow Computations

The steady subsonic results for flow, with a freestream Mach number 0.25, past a 10% thick double parabolic arc aerofoil inclined at 0° and 1° AOA are presented here. The aerofoil profile is given by

$$h^{\pm}(x) = \pm 0.2x(1-x) \quad \text{for } x \in [0,1]. \qquad (31)$$

A FORTRAN 90 code has been written implementing the proposed improved AF algorithm. The computations are carried out employing the Euler implicit time difference formula.

For nonlift-generating computations, numerical experiments were performed using the geometric time step sequence (with $N = 22$), Equation (30) ($N = 22, \overline{\beta} = 0.665443$) and with a constant time step, $\Delta\overline{\tau} = 0.025$. If $\mathcal{R}$ and $\mathcal{R}_0$ are the current and initial residual values computed at aerofoil (upper surface) quarter-chord point, the solution is said to converge when $\log_{10}|\mathcal{R}/\mathcal{R}_0|_{1/4}$ is less than -5.5. The convergence history for the time step cycling curves is presented in Figure 3 with all the final details tabulated in Table 2. In this figure the vertical scale is logarithmic, so that a value of $-m$ indicates that the current residual has descreased to 10^{-m} of its initial value. The experiment with repeated endpoints requires less iterations for convergence, and in the case without time step cycling, 122 iterations are required to obtain similar accuracy. Note that the constant $\Delta\overline{\tau}$ value was arbitrarily chosen, and a smaller value could be used, but this would then lead to more iterations to obtain similar accuracy. This value must be chosen to ensure stability, and in this case, $\Delta\overline{\tau} = 0.025$, is the largest possible value that could be used without causing an instability.

For aerofoils with a simple profile, like the one used here, an analytic solution for the reduced potential on the aerofoil surface can be derived using the integral form of the steady subsonic equations. The comparison between the analytical and numerical solutions for the case of 0° AOA is excellent, as illustrated in Figure 4. The steady pressure coefficient,

$$C_{\mathrm{p}} = -2\phi_x, \qquad (32)$$

computed for the cases of 0° and 1° AOA are collected in Figure 5. In this figure the pressure coefficient has been scaled with the critical pressure coefficient; $C_{\mathrm{p}}^{*} = -2\overline{u}$ where $\overline{u}$ is given by (37). Values of $C_{\mathrm{p}}/C_{\mathrm{p}}^{*}$ greater than one indicate locally supersonic flow.

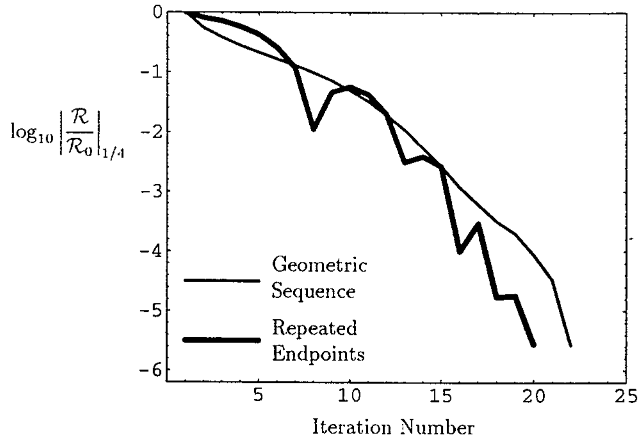

FIGURE 3: Convergence history. Note $|\mathcal{R}_0|_{1/4} = 6.4800$.

TABLE 1: Smallest possible $\overline{\beta}$ values.

Repeated	$\Delta\overline{\tau} = 1/P$ $(\Delta x = 1/50)$		$\Delta\overline{\tau} = 1/Q$ $(\Delta z = 1/60)$	
Endpoints	$\overline{\beta}$	N	$\overline{\beta}$	N
without	0.708482	13	0.717058	13
with	0.665443	22	0.674453	22

TABLE 2: Convergence history data for Figure 3. Note $|\mathcal{R}_0|_{1/4} = 6.4800$.

Code	Iterations	$\lvert\mathcal{R}\rvert_{1/4}$	$\log_{10}\lvert\mathcal{R}/\mathcal{R}_0\rvert_{1/4}$	$\lvert\overrightarrow{\Delta_\tau}\phi^n\rvert_{\max}$
Geometric Sequence $N = 22$	22	$1.7518(10^{-5})$	-5.5681	$1.3742(10^{-4})$
Repeated Endpoints Equation (30), $N = 22$, $\overline{\beta} = 0.665443$	20	$1.8232(10^{-5})$	-5.5508	$1.0111(10^{-6})$
Constant Time Step $\Delta\overline{\tau} = 0.025$,	122	$9.5740(10^{-6})$	-5.8305	$2.8210(10^{-6})$

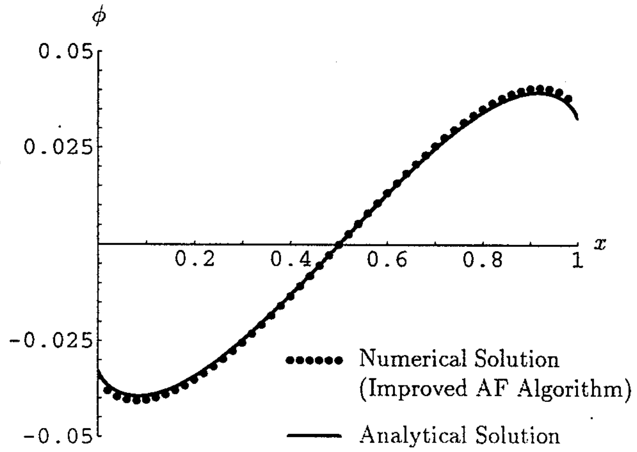

FIGURE 4: Comparison between the numerical and analytical solutions for the reduced potential on the aerofoil surface at 0° AOA.

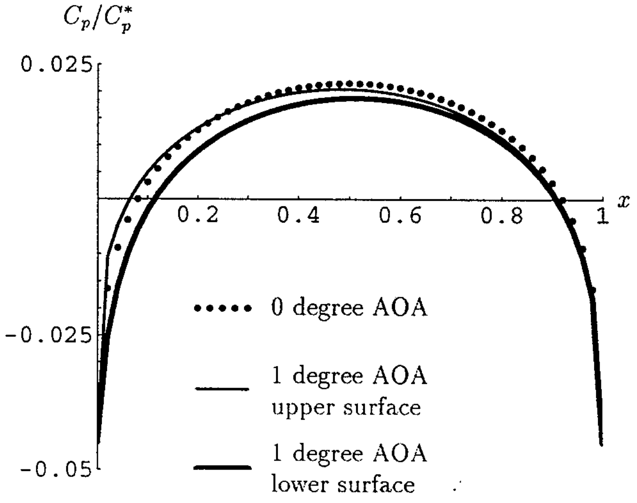

FIGURE 5: Scaled pressure coefficient on the aerofoil surfaces.

Three-Dimensional Transonic Flow Computations

The governing equation for the steady transonic flow, in three dimensions, is the nonlinear modified Transonic Small Disturbance (TSD) Equation. The modified TSD Equation(2,3,12) for the reduced potential, $\phi(x, y, z)$, may be written in conservation form as

$$\frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} = 0, \tag{33}$$

where

$$f_1 = \frac{1}{2\overline{u}}\left(M^2 - 1\right)\left(\phi_x - \overline{u}\right)^2 + G\phi_y^2, \tag{34}$$

$$f_2 = \phi_y\left(1 + H\phi_x\right), \tag{35}$$

$$f_3 = \phi_z. \tag{36}$$

The constants $\overline{u}$, G and H are

$$\overline{u} = \left(1/M^2 - 1\right)/\left(\gamma + 1\right), \tag{37}$$

$$G = M^2\left(\gamma - 3\right)/2, \tag{38}$$

$$H = M^2\left(1 - \gamma\right). \tag{39}$$

Here γ represents the ratio of specific heats. Equation (33) is of hyperbolic type representing supersonic flow for $\phi_x > \overline{u}$ and of elliptic type representing subsonic flow for $\phi_x < \overline{u}$. The boundary conditions imposed at the far-field boundaries and on the wing are those given by Equations (2) to (8).

Computations were performed for the steady flow over a swept and tapered F5 wing at freestream Mach numbers 0.5 and 0.9. The cross section of the wing is a symmetric NACA 65A004.8 section with a 7° droop nose over the first 10% of the chord. The semispan of the wing is one, with its chord length measuring one at the root section and tapering down to 0.2875 at the wing tip. The location of the leading and trailing edges of the wing (with respect to the wing span) are given by $x_{\rm le} = 0.625y$ and $x_{\rm te} = 1 - 0.0875y$, respectively. The size of the computational region used in this study is $(x_u, x_d, y_m, \pm z_m) = (-1, 2, 2, \pm 3)$, with 80, 30 and 180 grid points in the streamwise, spanwise and vertical directions, respectively. On the wing we have 40 grid points along each cross section, and there are 15 such sections along the semispan of the wing.

The approximate solution was obtained by the use of a type-dependent, finite difference scheme incorporating the time step sequences developed in the previous section. The full details of this scheme can be found in Gear[10,11] and Gear, Ly and Phillips.[12,15] The emphasis on this part of the paper is to demonstrate that the convergence rate of the scheme can be enchanced by the uses of the time step sequence with repeated endpoints, see Table 3. In each case presented, in Table 3, we see that the scheme with repeated endpoints converges faster. Using only about 80% of the number of iterations used by the scheme without repeated endpoints. The computed scaled pressure coefficient on the surfaces of the F5 wing are presented in Figure 6. In this figure, the shock surfaces are represented by the discontinuous jumps in the scaled pressure coefficient. The shock position and strength vary along the wing semispan, with greatest variation in strength near the wing tip.

General Remarks and Conclusion

In the preceding sections, the development of the improved AF algorithm for the solution of the two-dimensional steady SSD Equation is presented. The algorithm has also been implemented to solve the two-dimensional steady TSD Equation by the author.[12,15] It is evident from the numerical experiments and the analysis presented here that the method of false transients with time step cycling can dramatically enchance the rate of convergence to the steady-state solution.

The method also works for nonlinear partial differential equations, as evident in Table 3 and Figure 6. The present analysis gives us a technique for determining the minimum and maximum time steps based on the spatial grid spacing. But for nonlinear partial differential equations the maximum time step should be chosen to ensure stability. The present analysis shows us that by repeating the endpoints of the time step sequence, at least twice, we should be able to reduce the error at the extreme frequencies at a faster rate.

References

[1] Ballhaus, W. F., Jameson, A., and Albert, J. Implicit Approximate-Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems. AIAA Paper 77-634, 1977.

[2] Batina, J. T. Efficient Algorithm for Solution of the Unsteady Transonic Small-Disturbance Equation. *J. Aircraft 25*, 7, July 1988, 598–605.

[3] Batina, J. T. Unsteady Transonic Algorithm Improvements for Realistic Aircraft Application. *J. Aircraft 26*, 2, February 1989, 131–139.

[4] Beam, R. M., and Warming, R. F. An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form. *Journal of Computational Physics 22*, 1976, 87–110.

[5] Bertin, J. J., and Smith, M. L. *Aerodynamics for Engineers*, second ed. Prentice-Hall International, Inc., 1989, ch. 11. Prentice-Hall International Editions.

[6] Catherall, D. Optimum Approximate-Factorization Schemes for Two-Dimensional Steady Potential Flows. *AIAA Journal 20*, 8, August 1982, 1057–1063.

[7] Davis, G. d. V. Finite Difference Methods for Viscous Flow and Heat Transfer. In *Computational Techniques & Applications: CTAC85* (North-Holland, 1986), B. J. Noye and R. L. May, editors, Elsevier Science Publishers B.V., pp. 71–89.

[8] Engquist, B. Numerical Radiation Boundary Conditions for Unsteady Transonic Flow. *Journal of Computational Physics 40*, 1981, 91–103.

[9] Fung, K. Y. Far Field Boundary Conditions for Unsteady Transonic Flows. *AIAA Journal 19*, 7, February 1981, 180–183.

[10] Gear, J. A. Approximate Factorization Algorithm for the Steady TSD Equation. Research Report 10, Department of Mathematics, Faculty of Applied Science, RMIT University, GPO Box 2476V, Melbourne, Victoria 3001, Australia, May 1994.

[11] Gear, J. A. Algorithm Improvements for Steady Transonic Flow Calculations. Research Report 10, Department of Mathematics, Faculty of Applied Science, RMIT University, GPO Box 2476V, Melbourne, Victoria 3001, Australia, March 1995.

[12] Gear, J. A., Ly, E., and Phillips, N. J. T. Time Marching Finite Difference Solution of the Modified Transonic Small Disturbance Equation. In *Computational Techniques and Applications: CTAC97* (Singapore, 1997), B. J. Noye, M. D. Teubner, and A. W. Gill, editors, World Scientific, pp. 209–216.

TABLE 3: Details of numerical experiments for the F5 wing.

Time Step Sequence	M	Iterations	$\Delta\tau_{\text{min}}$	$\Delta\tau_{\text{max}}$
Geometric Sequence $N = 22$	0.5 0.9	107 591	0.001 0.001	0.050 0.040
Repeated Endpoints $N = 22$	0.5 0.9	84 475	0.001 0.001	0.050 0.040

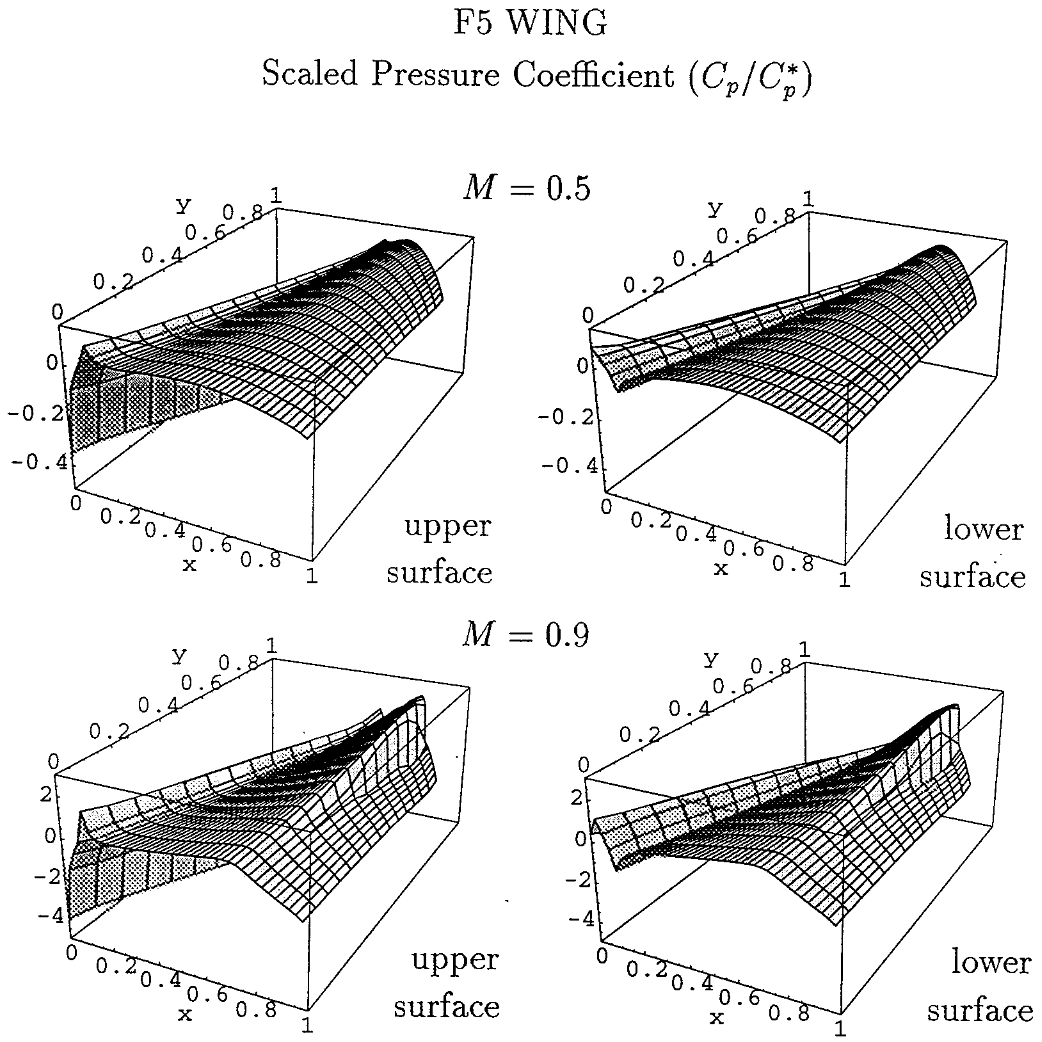

FIGURE 6: Scaled pressure coefficient on the surfaces of the F5 wing.

[13] Kwak, D. Nonreflecting Far-Field Boundary Conditions for Unsteady Transonic Flow Computation. *AIAA Journal* ***19***, 11, November 1981, 1401–1407.

[14] Ly, E., Gear, J. A., and Phillips, N. J. T. Improved Approximate Factorisation Algorithm. In *Computational Techniques and Applications: CTAC97* (Singapore, 1997), B. J. Noye, M. D. Teubner, and A. W. Gill, editors, World Scientific, pp. 393–400.

[15] Ly, E., Gear, J. A., and Phillips, N. J. T. Simulated Shock Motion Using a Time-Linearised Transonic Code. In *Engineering Mathematics and Applications Conference: EMAC98* (1998). To appear.

[16] Murman, E. M. Computational Methods for Inviscid Transonic Flows with Imbedded Shock Waves. In *AGARD Lecture Series Number 48 on Numerical Methods in Fluid Dynamics*, J. J. Smolderen, editor, AGARD-LS-48. AGARD, Advisory Group for Aerospace Research and Development, May 1972, ch. 13, pp. 1–34.

[17] Warming, R. F., and Beam, R. M. On the Construction and Application of Implicit Factored Schemes for Conservation Laws. In *SIAM-AMS Proceedings* (1978), vol. 11, pp. 85–129.