AIAA 2003–3708

Start-up Issues in a Newton-Krylov Algorithm for Turbulent Aerodynamic Flows

Todd T. Chisholm, and David W. Zingg
University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, ON, M3H 5T6, Canada

16th AIAA Computational Fluid Dynamics Conference
June 23–26, 2003
Orlando, FL

Start-up Issues in a Newton-Krylov Algorithm for Turbulent Aerodynamic Flows

Todd T. Chisholm,* and David W. Zingg†
University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, ON, M3H 5T6, Canada

A fast Newton-Krylov algorithm is presented for solving the compressible Navier-Stokes equations on structured multi-block grids with application to turbulent aerodynamic flows. The one-equation Spalart-Allmaras model is used to provide the turbulent viscosity. The optimization of the start-up algorithm is discussed. A spatial time step is presented that stabilizes the turbulence model, provides reduced CPU time to converge to the region of convergence for Newton's method, and eliminates the memory requirements of the modified Jacobian suggested by Spalart and Allmaras. Scaling of the turbulence variable and equation improves the efficiency of the linear solver. A modification of the matrix-free method addresses a problem with round-off errors when scaling differences occur in the system. Three test cases are used to demonstrate convergence rates. Single-element cases are solved in less than 60 seconds on a desktop computer, while the solution of a multi-element case can be found in about 10 minutes.

Introduction

Recently, Newton-Krylov methods have been shown to be very effective in reducing the time required to compute numerical solutions to the Navier-Stokes equations. Blanco and Zingg[1] studied the solution of the Euler equations on unstructured grids with a matrix-free Newton-Krylov method. Geuzaine[2] used a similar method with the compressible Navier-Stokes equations, modeling turbulence with the Spalart-Allmaras model. Barth and Linton[3] studied a parallel implemention of a Newton-Krylov solver on unstructured grids for two- and three-dimensional flows. Pueyo and Zingg[4] solved the turbulent, compressible Navier-Stokes equations on structured grids.

Pueyo and Zingg have demonstrated that this approach is competitive with state of the art multigrid methods. However, their work was limited to representing turbulence with the algebraic Baldwin-Lomax model on single block grids. Here we discuss the solution of the Navier-Stokes equations on single- and multi-element airfoils, using the one-equation Spalart-Allmaras turbulence model.[5]

A common problem with this turbulence model is exacerbated by the using Newton's method. That is, during the early stages of convergence, the Spalart-Allmaras equation is quite unstable. This paper discusses methods to efficiently address this problem. We will investigate time stepping methods which bring the solution within the radius of Newton's method, as well as scaling changes which make the linear solver more effective.

Algorithm Description

Governing Equations

We study the solution of the steady compressible thin-layer Navier-Stokes equations on structured grids. A generalized curvilinear coordinate transformation is used to map the physical space to a rectangular computational domain. The use of multiple blocks allows for complex geometries such as multi-element airfoils. A circulation correction is used to reduce the effect of the farfield boundary. The Spalart-Allmaras turbulence model, including trip terms, is implemented as described by Godin et al.,[6] with a small change in the calculation of the modified vorticity factor, first used by Ashford[7]

$$\tilde{S} = S f_{v3} + \frac{\tilde{\nu}_T}{\kappa^2 d^2} f_{v2} \tag{1}$$

$$f_{v2} = \left(1 + \frac{\chi}{c_{v2}}\right)^{-3} \tag{2}$$

$$f_{v3} = \frac{(1 + \chi f_{v1})(1 - f_{v2})}{\chi} \tag{3}$$

with $c_{v2} = 5.0$. The original form, which allowed $\tilde{S}$ to become negative, introduced a local minimum quite close to the solution root at some nodes at the edge of recirculation bubbles. This can cause the residual to hang, despite the majority of the flow being converged. The new form helps avoid this problem.

Spatial Discretization

The spatial discretization follows that used by Nelson et al.[8] Second-order centred differences are used to approximate derivatives. Both Jameson's[9] scalar and

*Ph.D Candidate, todd@oddjob.utias.utoronto.ca

†Professor, Senior Member AIAA, http://goldfinger.utias.utoronto.ca/∼dwz/

Swanson and Turkel's[10] matrix second- and fourth-difference dissipation models can be used to stabilize the centred difference scheme. A pressure switch is used to control the activation of second-difference dissipation. The matrix dissipation model uses two switches V_l and V_n to avoid the effect of overly small eigenvalues in the flux Jacobian matrix. We use $V_l = V_n = 0.025$ for subsonic cases, and $V_l = 0.025$ and $V_n = 0.25$ for transonic cases. The turbulent viscosity convection and diffusion terms are discretized using first-order upwinding and second-order centred differencing, respectively, as suggested by Spalart and Allmaras.[5]

Since a Newton solver is used to solve the resulting nonlinear system, it is important that all of the boundaries be handled fully implicitly. This includes the interfaces between blocks. The Navier-Stokes equations are solved on these interfaces in the same manner as the interior nodes.

Newton-Krylov Algorithm

The Nonlinear System.

After spatial discretization, we have a system of the form

$$R(\hat{Q}^*) = 0 \tag{4}$$

where each block of $\hat{Q}$, the conservative state variables with the turbulence variable, is

$$\hat{Q}_i = J_i^{-1} Q_i = J_i^{-1} [\rho_i, \rho u_i, \rho v_i, e_i, \tilde{\nu}_i]^T$$

To find $\hat{Q}^*$ which satisfies Eq. 4, we apply the implicit Euler method repeatedly until some convergence criterion, typically $\|R\|_2 < 10^{-12}$, is reached:

$$\left[\frac{I}{\Delta t_n} - \frac{\partial R_n}{\partial \hat{Q}_n}\right] \Delta \hat{Q}_n = R_n$$

$$\hat{Q}_{n+1} = \hat{Q}_n + \Delta \hat{Q}_n$$

We call these the *outer* iterations. When the time step is increased towards infinity, Newton's method is approached. If Δt is increased appropriately as $\|R\|$ decreases, the quadratic convergence characteristic of Newton's method can be achieved, while dramatically increasing the region of convergence. Note that, in order for Newton's method to converge quadratically, $\frac{\partial R}{\partial Q}$ must be accurate. This requires that the equations be fully coupled.

The Linear System.

In order for $\Delta \hat{Q}_n$ to be found, a linear system needs to be solved. This system tends to be very large, so that direct solution is prohibitive in both memory and time. Fortunately, finding the exact $\Delta \hat{Q}$ is not necessary, and we may settle for finding an approximation. This is an inexact-Newton method. There are a number of popular methods of finding the approximate solution of the linear system. The proper selection and use of this method is crucial to the success of the overall solver.[11] The most successful class are the Krylov iterative methods. Specifically, the preconditioned Generalized Minimum Residual (GMRES)[12] has proven to be effective for aerodynamic systems. We call these linear iterations the *inner* iterations.

Over-solving the linear system needs to be avoided for efficiency. A stopping criterion is needed for the inner iterations. There are two considerations. First, we use a target reduction in the inner residual. Pueyo and Zingg[13] found a one order of magnitude reduction ideal to balance outer and inner iteration efficiency. The second consideration is setting the maximum number of iterations of GMRES. The amount of memory and CPU time increases with each GMRES iteration, so a limit is prudent. GMRES may be restarted, which keeps the memory requirements lower, while allowing further solution of the linear system. However, this can significantly slow the linear system convergence, due to the very poor conditioning seen in these systems. Typically, we do not use restarting for this reason.

The convergence rate of GMRES is very sensitive to the condition number of the matrix. Since the Jacobian of the equations being solved is typically extremely ill-conditioned, a good preconditioner is required to limit the number of inner iterations. Pueyo and Zingg[4] have shown that an incomplete LU preconditioner (ILU)[14] with two levels of fill minimizes solution time. They also found that a preconditioner based on a first-order Jacobian is more efficient than the exact Jacobian, both in saving memory and CPU time. The first-order Jacobian is formed by using only second-difference dissipation. This reduces the number of entries per equation to five instead of nine. It tends to give a better conditioned matrix, which leads to a more stable LU factorization. The coefficient of the second-difference dissipation used in the approximate Jacobian matrix, ϵ_2^l, is found by

$$\epsilon_2^l = \epsilon_2^r + \sigma \epsilon_4^r$$

where σ is found empirically and ϵ_2^r and ϵ_4^r are the second- and fourth-difference dissipation coefficients used in the evaluation of the residual and the exact Jacobian.

The systems being solved here are significantly more ill-conditioned than those used by Pueyo and Zingg, mostly due to the use of the turbulence model and matrix dissipation. The latter results in a higher optimum value of $\sigma = 10$. The most efficient level of fill for the ILU preconditioner is four, although two is usually close, and reduces memory requirement somewhat.

The GMRES algorithm only requires matrix-vector multiplies, and does not explicitly require the matrix, except in forming the preconditioner. A Jacobian-free implementation of GMRES may be used, which has been found by Pueyo and Zingg[13] to be faster, as well as resulting in significant memory savings.

Case	Mach	Alpha	Re·10^6	Airfoil
1	0.3	6.0°	9.0	NACA0012
2	0.729	2.31°	6.5	RAE2822
3	0.185	6.0°	2.51	NLR

Table 1 Flow conditions

Case	Dimensions	Nodes	Offwall Spacing
1	305× 57	17385	10^{-6}
2	257× 57	14619	$2 \cdot 10^{-6}$
3	–	44059	10^{-6}

Table 2 Grids

Reverse Cuthill-McKee reordering is used to reduce the effect of dropped elements in the preconditioner. We have found that the root node choosen for this method is a very important parameter. A node in the wake is a good choice.

Test Cases

Three test cases are presented. Two are single-element, one subsonic, the other transonic. The third case is an airfoil with a detached flap, at low Mach number and moderate angle of attack. The flow conditions are shown in Table 1, grid details in Table 2. Off-wall spacing is given relative to chord length.

The GMRES iterations for the single-element cases are limited to thirty search directions. This is not sufficient for the larger multiblock case. A limit of sixty search directions allows the linear solver to converge for this case.

Startup

For all but the simplest cases, using the freestream condition for the initial guess causes the equations to be outside the region of convergence of Newton's method. To stabilize the system, we add a time step during start-up, using Euler's method instead of Newton's. The choice of a time step is critical to the efficiency and robustness of the overall algorithm.

Reference time step

Some authors have suggested using a function to define Δt_{ref}, dependent on the norm of the residual. For example, we have had success with

$$\Delta t_{ref} = \min(\alpha \cdot ||R||_2^{-\beta}, \Delta t_{min}) \qquad (5)$$

There are a few interesting details to recognize when using this approach. The interior of the domain is fully converged at the first iteration. The only contribution to the residual is from the equations at the body. The residual from these equations, which do not have a time component, should not be used to set the coefficients in the above equation. Until the residual of the interior is high enough to drive equation 5, Δt_{ref} should be held to the minimum value. Usually, this only takes two or three iterations. This is the purpose of Δt_{min}. This will only work as presented if the boundary condition equations have a magnitude that is large enough that the norm of the residual does not cause the time step to increase past Δt_{min}.

The proper choice of Δt_{min}, α, and β is quite important. The time step will increase very slowly if α or the minimum are choosen, or are required to be, too small. Fortunately, a set of coefficients that give fast results is available. The proper selection of α depends strongly on the scaling of the equations. This is discussed is section . The set of coefficients also needs to be adjusted depending on the type of case being solved. For example, using a turbulence model both raises the residual norm, and requires a more conservative time step schedule. Depending on how it is scaled, the turbulence model normally dominates the residual, and thus determines the time step. This is desirable since the mean flow equations are generally much more stable than the turbulence model equations. The existence of a shock also requires a smaller minimum Δt_{ref}.

Equation 5 is really only useful while Δt_{ref} is below about 10^3. Above this, we are effectively using Newton's method. Hopefully, the time spent going from the minimum value time step to Newton's method is reasonably short, relative to total time to converge. Choosing β strongly affects this. We find that a β of one provides rapid increase of time step, while maintaining stability.

Finally, this method is only efficient when there is a clear connection between the maximum potential time step and the residual. Unfortunately, this is not always true, especially with turbulent flows. In many of these cases, if too high a multiplier, α, is used divergence will occur after the residual drops an order of magnitude. On the other hand, if α is decreased, a long plateau occurs in convergence. During this time, when the residual is not changing significantly, the time step could be ramped up, significantly shortening this phase. We will investigate a number of techniques to overcome this deficiency, including grid sequencing, local time step modifications, and the use of a limited time step for the first few iterations.

When Reynolds number scaling is used, $\alpha = 0.001$ is appropriate for inviscid flows, while $\alpha = 0.1$ is better for turbulent flow. This reflects the higher residual of the turbulence model. Subsonic cases can use an initial time step of 100, while transonic cases should use a lower value of 10 due to the presence of the shock.

Spatial time step choice

We follow Pulliam,[15] who suggests using a local time step dependent on grid metrics for the mean flow equations. This approximates a constant CFL, and was shown to provide a large improvement in convergence rates when compared with a time accurate approach.

This geometric time step is

$$\Delta t = \frac{\Delta t_{ref}}{1+\sqrt{J}} \tag{6}$$

We also need to choose a time step for the turbulence model. The geometric time step is compelling for the same reason that it is for the mean flow equations: an approximately constant CFL improves stability for the convection part of the equations. However, the turbulence model is much less convection dominated than the mean flow equations, and TORNADO uses a spatially constant time step, so we will compare the spatially constant and varying time steps.

In both methods, the experiments where carried out using varying reference time step as described in section . We have to choose a set of parameters (Δt_{min}, α, and β) for each method. Also, we should allow a different Δt_{ref} for the mean flow and turbulent equations. This is especially important when a spatially different time step is used for the mean flow and turbulent equations. We could use an entirely separate parameter set for each, based on the corresponding residual norm. However, to simplify, one set of parameters is used for both sets of equations, but the turbulence reference time step is scaled by a factor τ to allow some flexibility. We do this to reduce the number of parameters to optimize. This also limits the time step of the mean flow equations when the residual of the turbulence model is high, which is usually a desirable characteristic.

To summarize, a spatially varying time step for the turbulence model is

$$\Delta t = \frac{\tau \cdot \Delta t_{ref}}{1+\sqrt{J}} \tag{7}$$

while the time accurate step is simply

$$\Delta t = \tau \cdot \Delta t_{ref} \tag{8}$$

Using $\tau = 1$ is appropriate for a spatially varying system, which is perhaps not surprising. $\tau = 10^{-2}$ is optimal for a constant time step. Both methods used $\alpha = 0.1$, $\beta = 1$, and $\Delta t_{min} = 100$. No grid sequencing was used in order to accentuate the differences, and the turbulent residual was scaled by the Reynolds number, as discussed in section . When scaling modifications are made to the turbulence equations, as discussed in section , the time step, whether spatially constant or varying, has to be appropriately scaled.

Figure 1 compares the geometric time step and a time accurate step for the subsonic single element and the NLR cases. It is interesting that there is little difference in the two methods when the residual is large. In fact, the only difference appears when the limiting time step, described in section becomes inactive. This causes a change of only one or two iterations. We have choosen the geometric time step for the remaining tests.

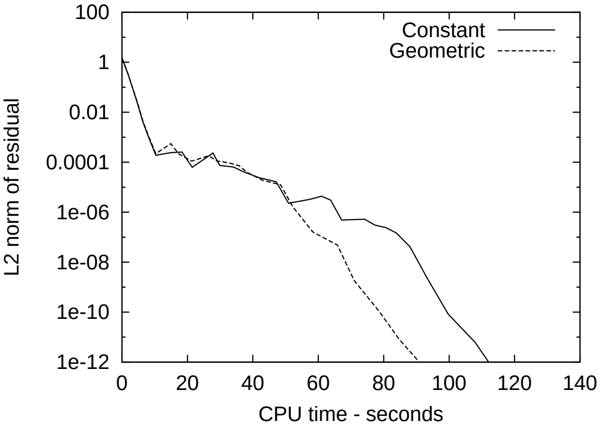

Fig. 1 Spatially varying vs. constant time step

Grid Sequencing

Grid sequencing uses a series of coarse grids to provide a good initial guess on the final grid. This has the advantage of starting much closer to the region of convergence of Newton's method, and reducing the start-up time. Potentially, a larger Δt_{min} can be used on the fine grid.

Each coarse grid is formed by removing every other grid line in both directions from the parent grid. We see modest benefits to using grid sequencing for all but the turbulent multiblock case. Grid sequencing also makes it easier to choose the coefficients for the reference time step equation. A more aggresive sequence can be used on the fine grid, while more conservative values can be use on coarse grids, where the iterations are quicker.

Turbulence Model Stabilization

Examination of the turbulence model with a value of $\tilde{\nu}$ less than zero shows that production, which is roughly proportional to $\tilde{\nu}$, will become negative, while the destruction term, which is proportional to $\tilde{\nu}^2$, remains negative. This shows that the model becomes unstable as $\tilde{\nu}$ decreases from zero. Therefore, it is crucial to take steps to ensure that $\tilde{\nu}$ remains positive. An obvious fix is to clip the turbulence variable after each update. This is useful, but not sufficient to stabilize start-up without the use of an extremely small time step. This is a result of the strong coupling between the turbulence model and mean flow equations. A large update of the turbulence quantity can drive the pressure or density negative. Further measures are required. We present two methods of stabilizing the turbulence model during the critical start-up. Both are aimed at preventing updates to $\tilde{\nu}$ that will make it negative. The first follows Spalart and Allmaras in modifying the Jacobian of the turbulence equations. The second presents a new time step designed to limit updates. Clipping as well as these methods are discussed below.

When the mean flow equations are started from freestream, there are usually large transients present at the body during the first few iterations. The production and trip terms can produce very large residuals because of this. It is generally a good idea to perform a few iterations of the mean flow solver without the turbulence model to establish a reasonable flow at the body.

A common problem occurs when the turbulence model fails to trip in a portion of the flow during the early iterations. The residual can drop a few orders of magnitude with this laminar region. Unfortunately, this means that the time step has increased, and can be approaching Newton's method, when this region finally transitions. The result can be a spike in residual with corresponding slow down in convergence, or can often be divergence because of the large time step. This is a more common problem when fully turbulent flow, with no trip terms to help transition, is used. Fortunately, this is easily treated. By starting with a low level of turbulence where the final solution is expected to have transitioned, and by starting the cases with just the mean flow equations as described above, we can obtain proper tripping. This turbulence viscosity is rapidly convected from regions that lack enough shear to sustain it. By starting with a few mean flow iterations, we establish enough shear near the body to hold the turbulent viscosity. Using $\nu = 10$ is high enough to seed tripped regions, but still convect out quickly.

When no trip terms are used, we are assuming fully turbulent flow, so all nodes start with this small turbulent quantity. When using trip terms, just those regions behind the trip points are set to nonzero. It is important to leave the turbulent viscosity at zero in regions upwind of trip points. Otherwise, it is common that the flow remains fully transitioned, even when fully converged.

Modified Jacobian

Spalart and Allmaras[5] suggest modifying the Jacobian of the turbulence model so that it becomes positive definite. Note that these changes preclude the use of matrix-free GMRES. The rational behind the change is that the updates to the turbulence quantity will always be positive.

The original modifications were derived for a loosely coupled or uncoupled solver, so that there are no derivatives of the turbulence model with respect to the flow state vector. The presence of these coupling derivatives could complicate the modifications significantly. However, experimentation has shown that changes to these entries are not necessary to obtain positive updates.

Of course, these changes are not desirable when using a Newton solver and slow the outer iterations significantly. However it is during start-up when the highest transients are occurring that we are at the most risk of seeing very large negative values. After the residual has dropped, it is possible to return to the unmodified Jacobian without much risk of encountering negative values of $\tilde{\nu}$.

The time step method presented in section does not work well for this method. It requires much higher time steps for efficiency. Also, there is no clear relationship between the maximum stable time step and residual. In fact, a time step of infinity is often possible, after a few iterations to smooth the solution. Convergence is instead limited by the modified Jacobian. For these reasons, we use an alternative method to set the time step when the modified Jacobian is used. A typical time step series using grid sequencing proceeds as follows:

- On the twice-coarsened grid
  - $\Delta t_{ref} = 50.$ until turbulence model trips
  - $\Delta t_{ref} = 500.$ until $||R|| < 10^{-4}$
- Prolong to one-coarsened grid
  - $\Delta t_{ref} = 50.$ for 3 iterations
  - $\Delta t_{ref} = 500.$ until $||R|| < 10^{-4}$
- Prolong to final grid
  - $\Delta t_{ref} = 50.$ for 3 iterations
  - $\Delta t_{ref} = 500.$ until $||R|| < 10^{-4}$
  - Newton stage, with true SA Jacobian

The first stage ends after the turbulent residual has peaked. The value it reaches varies strongly by case, but it is generally safe to choose a minimum of 15 iterations, and switch to the next stage when the residual drops below one. This stage is quite fast because a coarse grid is used, so a conservative number of iterations should be used.

After prolonging, a few iterations are performed at lower time step in order to smooth out prolongation errors. Negative values of $\tilde{\nu}$ often show up here. Clipping is useful to maintain stability. The transition to the true Jacobian should happen as soon as possible, since the modifications to the Jacobian slow convergence significantly.

The above assumes that the turbulent equations are scaled by the Reynolds number, as described in section . If a Jacobian scaling is used, the tolerance of the linear system must be tightened considerably. There seems to be two reasons for this. The modifications to the Jacobian require tighter tolerances to enforce a positive update. Second, since large time steps are being used, the linear residual of the mean flow equations must be reduced, or at least not allowed to grow in order to preserve nonlinear stability. They are much smaller than the turbulence model residuals, so a correspondingly smaller linear tolerance is required.

Local time step

The production and destruction terms are highly nonlinear and exhibit some very undesirable characteristics when used with a Newton solver. Figure 2 shows the local residual of the turbulence model at a problem node. In this particular case, the residual and derivative were evaluated at $\tilde{\nu}$ roughly equal to 17. The slope is positive, but very close to zero. This situation leads to a large negative update. Fortunately, it is also reasonably easy to avoid if we use an appropriate time step. The following applies only locally and does not address coupling effects. However, most coupling results from convection and diffusion, both of which are unlikely to result in large negative updates, and trip terms, which are discussed in section

If Newton's method is used on this uncoupled equation, we get the following update:

$$\Delta\tilde{\nu} = \frac{R}{-J_D} \tag{9}$$

where R and J_D are the residual and diagonal element of the equation, respectively. We want to limit this update so that

$$|\Delta\tilde{\nu}| < |r| \cdot \tilde{\nu} \tag{10}$$

where $|r|$ is a specified ratio. Choosing $|r| = 1$ would keep the updated $\tilde{\nu}$ positive in the uncoupled case, but in practise a smaller value is more robust. To enforce the limit on $\Delta\tilde{\nu}$, we use a local time step, $\Delta t_{\tilde{\nu}}$, determined from

$$\Delta\tilde{\nu} = \frac{R}{\Delta t_{\tilde{\nu}}^{-1} + J_D} \tag{11}$$

Applying the target update gives

$$\Delta t_{\tilde{\nu}} = \left[\frac{R}{r \cdot \tilde{\nu}} - J_D\right]^{-1} \tag{12}$$

$\Delta\tilde{\nu}$ and therefore r should have the same sign as R. This moves the update in the direction of the residual. Figure 3 shows different choices of $|r|$. The method is reasonably insensitive to the value, with a range of about 0.4 through 0.8. Going much higher than 0.8 can occasionally cause large negative updates.

Comparison of Stabilization Method

Figure 4 compares the modified Jacobian method to the modified time step. The new time step shows a modest improvement over the modified Jacobian. There are other advantages to the modified time step that are not evident in the convergence plots. It does not require an explicit Jacobian matrix to be used, which opens the possibility to use entirely matrix-free methods. We do not have to decide where to transition from the modified to exact (or matrix-free) Jacobian. The one or two iterations after this transition can be problematic. Large updates may occur, which can cause instability in both the nonlinear and linear systems. For these reasons, we prefer to use the modified time step method.

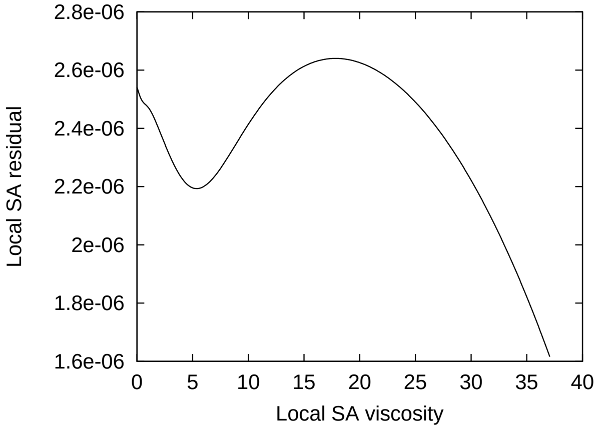

Fig. 2 Local turbulence residual at problem node

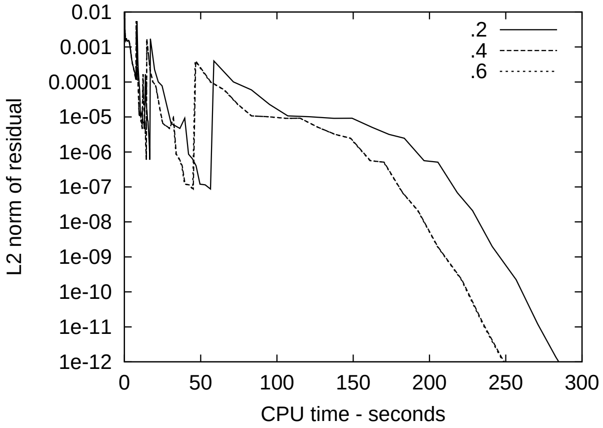

Fig. 3 Local turbulent time step limiting ratio for multi-element case

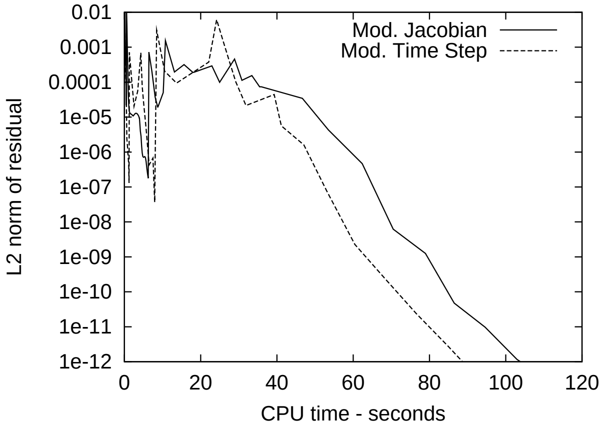

Fig. 4 Modified Jacobian vs time step methods of stabilizing turbulence model

Turbulence Model Scaling

If the turbulence model is discretized as suggested, large scaling differences appear within the blocks of the Jacobian matrix. The off-diagonal entries for the turbulence model are many orders of magnitude higher than the off-diagonal entries of the mean flow equations, especially those from the derivative with respect to the turbulence variable. The diagonal elements are quite close in magnitude. This disparity results in part from the mean flow variables being scaled by J^{-1}. The residual of the turbulence equations is also orders of magnitude higher than that of the mean flow equations. This can cause serious problems with nonlinear stability. The linear system does not need to reduce the linear residual of the mean flow equations in order to meet the convergence criterion and in fact these can increase. For this reason, scaling the residual of the turbulence equations to within about an order of the mean flow is quite important.

A natural decision would be to scale the turbulence variable by J^{-1}, the same factor as the mean flow variables. This improves the derivatives of the mean flow equations with respect to the turbulence variable. A scaling of the turbulence model equation addresses the derivative of the turbulence model with respect to the mean flow variables. Using J^{-1} is one choice. This is compelling because it closely matches that of the mean flow. However, it leads to a strong decrease in the residuals at the body. This causes problems with trip and destruction terms, which may not be resolved until later in the convergence history. Using a progressive Δt_{ref} means there may be quite a high time step when the residual close to the body is resolved, a situation best avoided.

A second choice is scaling the turbulence residual by the inverse of the Reynolds number. This helps scale the Jacobian (although not as well as using J^{-1}), and brings the turbulence model residual norm much closer to that of the mean flow equations, while not affecting the relative turbulence residual between nodes. It also makes a certain amount of physical sense, since the turbulent effects scale roughly by the Reynolds number.

An extension of this is to automatically rescale the equations at each nonlinear iteration. We have to do this at each iteration because we cannot rely on the differences in the first few iterations, as these residuals are strongly dependent on boundary residuals. Note that we calculate one factor which is used for the turbulence model equations at every node, as opposed to a calculating an individual factor for each equation. Applying this method as is does not give good results. Convergence is slow, requiring excessive linear solves, to the point where nonlinear convergence can stall. Watching both mean flow and turbulence model residuals reveals that the difference between them grows as convergence proceeds. To remedy this, we instead only require that the turbulence residual be rescaled to within one order of the mean flow. This ensures that the linear residual of the mean flow equations can't grow significantly (since we reduce the linear system residual by one order), but allows the residual of the turbulence model to decrease to the residual of the mean flow.

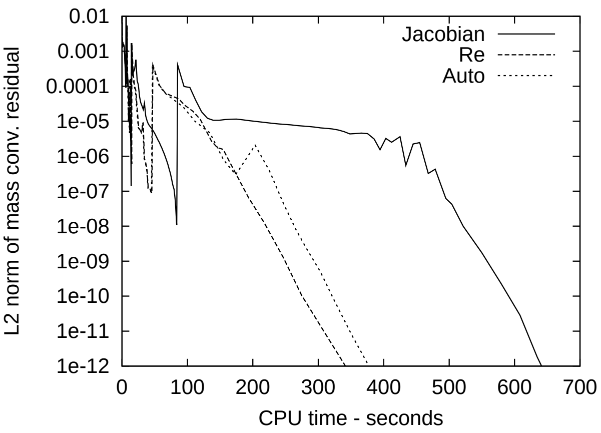

Fig. 5 Effect of turbulence model residual scaling on multi-element case

Figure 5 compares the above three methods of scaling the turbulent residual. Each uses a turbulence variable scaled by J^{-1}. It is quite difficult and slow to converge without this variable scaling. The mean flow residuals are shown instead of the total residual, because the scaling changes in the turbulence residual make comparison difficult. The parameters of equation 5 have to be optimized for the rescaled equations. The results show that using either Reynolds number or automatic scaling is significantly faster than using Jacobian scaling. This difference is even larger when trip terms are used.

Trip terms

The use of trip terms is strongly recommended by Spalart and Allmaras in order to ensure that turbulent transition occurs where desired. We follow their suggestion in using a limited number of nodes 'within a reasonable distance of the trip' for the calculation of the first trip function f_{t1}. This simplifies the coding, and speeds calculation of the residual slightly. If the region is chosen properly, the residual should not be affected because of the exponential decrease in f_{t1} with increasing distance from the trip point.

Using the trip terms provides a challenge for the Newton solver. f_{t1} is quite nonlinear, and can show extreme sensitivity to the velocities at the trip point. This is due to the exponential nature of f_{t1} and the fact that the vorticity is a derivative of the velocities. To further complicate the issue, the residuals of the mean flow equations are scaled by J^{-1}, which means that the linear residual near the body may not be reduced, leading to fluctuations in the velocities at the trip, and a potentially large increase in the trip term.

	CPU time in seconds	
Case	Matrix free	Matrix explicit
1	50	83
2	56	132
3	257	306

Table 3 CPU time with matrix-free and matrix-explicit

The coupling adds extra entries to the Jacobian matrix. These are important to include if an explicit matrix is being used. They are not desirable in the preconditioner, however, since they affect the reordering and increase memory requirements, especially with higher levels of fill.

If trip terms are used with no modifications made to the algorithm presented, convergence can be slowed unpredictably as the trip term fluctuates. Measures need to be taken to damp the changes in the turbulence variable near the trip. This is important for robustness as well as efficiency. The changes in the residual can easily be large enough to destablilize the solution. The approach taken is to drop the time step at the nodes where the trip terms show sensitivity to velocity. We want to keep the rule as simple as possible, and to have it work with the reference time step presented in section . Nodes where the magnitude of the derivative of the trip term are greater than a set limit have the time step by up to two orders according to the following

$$\Delta t^t_{ref} = \begin{cases} \Delta t_{ref} & : \quad \Delta t_{ref} < 10 \\ 10 & : \quad 10 < \Delta t_{ref} < 1000 \\ \frac{\Delta t_{ref}}{100} & : \quad \Delta t_{ref} > 1000 \end{cases} \qquad (13)$$

Choosing 10^{-3} for the derivative limit encompasses all the sensitive nodes. The method is not too sensitive to this value.

Matrix free vs. matrix explicit

As mentioned previously, GMRES does not require the matrix to be explicitly formed. There is a trade-off in speed between matrix-free and matrix-explicit GMRES, which depends on the number of linear iterations. The former requires one residual evaluation per iteration. The latter requires a matrix construction when beginning the linear solve, plus a matrix-vector multiply per iteration. Since the matrix-vector multiply is cheaper than a residual evaluation, matrix-free GMRES becomes less efficient with more difficult systems. Table 3 compares these two methods.

Case 2, the transonic case, shows the largest difference. The matrix here is better conditioned than the other cases, due in part to the higher Mach number, and needs about 10 inner iterations per Newton step. Also, due to the presence of the shock, the pressure switch has activated second-difference dissipation. Adding the linearization of the switch and the dissipation coefficient requires significant amounts of time when calculating the Jacobian and is not done. Without these terms in the Jacobian matrix, the convergence of the outer iterations is adversely affected. The first case also shows a significant advantage. Case 3 has the smallest gain. This is a result of the increased number of inner iterations required. Matrix-free is clearly the best choice as it saves CPU time, especially in the transonic case, and has much lower memory requirements.

The natural scalings found between the mean flow equations and turbulence model (in both the Jacobian and the residual) can be problematic for the matrix-free method. A forward difference approximation, which requires only one extra right hand side evaluation, is used to approximate the matrix–vector multiply

$$\mathcal{A}v \approx \frac{\mathcal{F}(\hat{Q} + \varepsilon v) - \mathcal{F}(\hat{Q})}{\varepsilon} \qquad (14)$$

where ε is a small scalar used to perturb $\hat{Q}$ in the direction of v. Nielsen et al.[16] choose ε as

$$\varepsilon \|v\|_2 \approx \sqrt{\varepsilon_m} \qquad (15)$$

where ε_m is the value of machine zero. Pueyo and Zingg[4] used this method successfully with the mean flow equations and an algebraic turbulence model. There are three potential sources of error in equation 14. The perturbation εv must be small enough to minimize nonlinearities in $\mathcal{F}$. The difference in the numerator must be large enough, compared to $\mathcal{F}$, to minimize round-off error, just as εv must be large enough compared to $\hat{Q}$.

The problem arises when there are large scaling differences in the components of vector v. Equation 15 will choose ε appropriate for the largest components of v. Large round-off errors can then result from the very small components. This is regularly witnessed when solving a system with both the mean flow equations and turbulence model, especially with trip terms. The manifestation can be subtle and is easily attributed to a problem with the nonlinear portion of the algorithm. This is because the linear solver will converge when this problem occurs. Effectively, a different matrix is being solved. The GMRES solver shows no sign of difficulty because the linear residual is not explicitly calculated during iteration. However, if the true linear residual is calculated after the Krylov method finishes, the residual of the mean flow equations is seen to have increased by one or two orders. Another symptom is an increase in number of inner iterations, resulting from the preconditioner no longer representing the Jacobian that is effectively being solved.

The main culprit is the round-off error. Increasing ε will correct this, at risk of increasing nonlinear errors. As it turns out, a value can be found that is still within the linear range, but that minimizes round-off error.

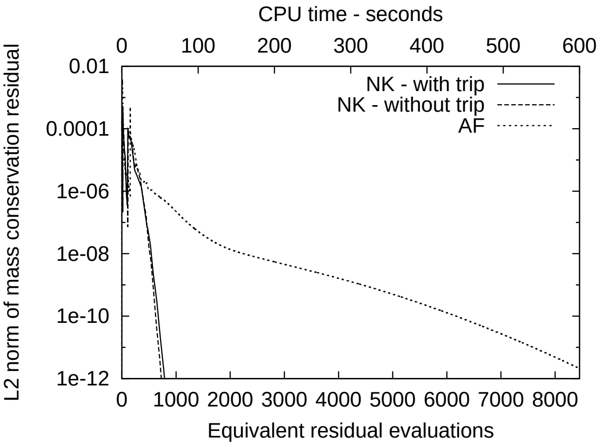

Fig. 6 Case 1 residual history

We use

$$\varepsilon||v||_2 \approx \sqrt{10^{-10}}$$

Results

Figures 6, 7, and 8 compare the convergence histories of the Newton-Krylov solver (NK) with an approximately-factored solver (AF) in diagonal form using grid sequencing. They show the conservation equation residual versus both CPU time and number of equivalent residual evaluations. We include both the evaluation of the mean flow and turbulence model equations in the residual calculation time. The time to evaluate the residual is virtually identical in both the AF and NK solvers.

The Newton-Krylov solver was run with and without trip terms. The AF solver used trip terms, although there is little difference in convergence time without them. All cases used an ILU(4) preconditioner, and were run on an AMD 1800-XP desktop computer. Cases 1 and 3 are considerably faster than the AF solver, while case 2 shows a smaller improvement. This is due to the AF solver already solving the case in 2500 residual evaluations. Case 3 is not fully converged by the AF solver. There is a small recirculation bubble at the trailing edge of the flap. Difficulties arise at the edges of these bubbles with the highly nonlinear destruction term of the turbulence model. Outside of these few nodes, the model is fully converged.

Cases 1 and 2 show very little difference in convergence when run with and without trip terms. Case 3 shows over a factor of two slow down when trip terms are used. This results from the trip vorticity on the underside of the main element converging very slowly. The vorticity is weak here, and so causes high trip sources. It also changes significantly between the coarse grids and the final grid, reducing the effectiveness of grid sequencing.

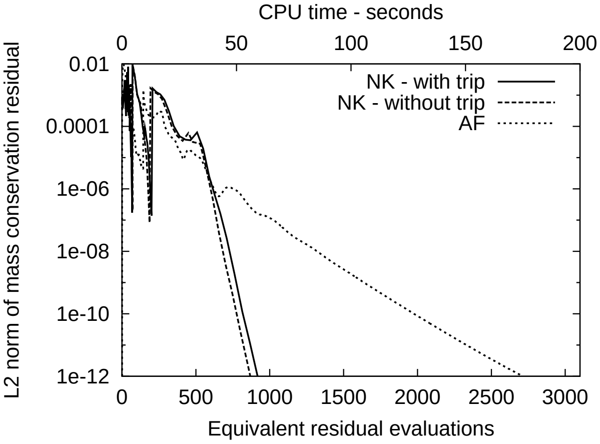

Fig. 7 Case 2 residual history

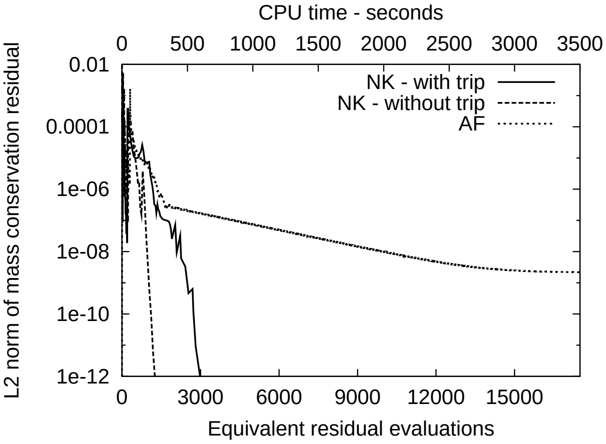

Fig. 8 Case 3 residual history

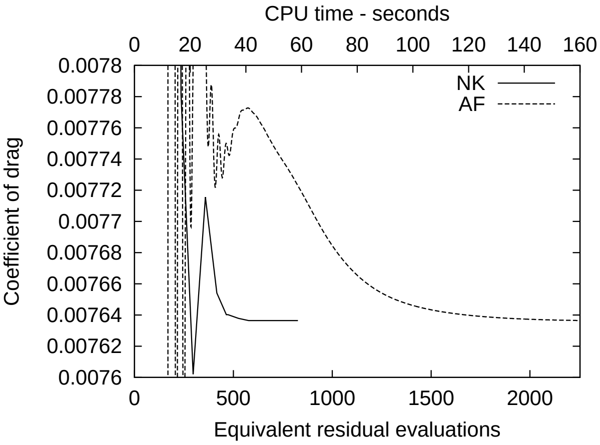

Fig. 9 Case 1 drag history

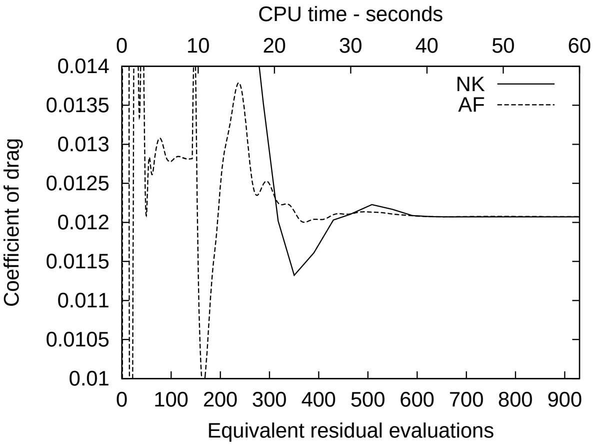

Fig. 10 Case 2 drag history

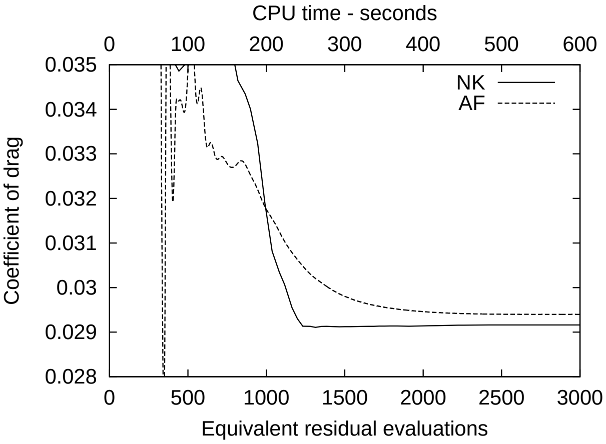

Fig. 11 Case 3 drag history

Conclusions

An efficient Newton-Krylov solver has been presented for the steady compressible Navier-Stokes equations governing turbulent flows over multi-element airfoils. A spatial time step has been presented that stabilizes the turbulence model, and provides reduced CPU time to converge to the region of convergence for Newton's method, as well as eliminating the memory requirements of the modified Jacobian suggested by Spalart and Allmaras. Scaling of the turbulence variable and equation improves the efficiency of the linear solver. Increasing the perturbation of the matrix-free method addresses a problem with round-off errors when scaling differences occur in the system. A reduced time step for the nodes where the trip source terms show sensitivity to trip vorticity is used to ensure that the turbulence model converges smoothly in these areas. With out this modification, large jumps in residual can destabilize the solution. The single-element test cases can be solved in less than 60 seconds, while the complex flow on the multi-element case can be found in about ten minutes. The subsonic cases converge four to eight times faster than an approximately factored algorithm, while the transonic is better than twice as fast.

Acknowledgments

This research was supported by Bombardier Aerospace, and an OGSST grant from the Government of Ontario.

References

[1] Blanco, M. and Zingg, D. W., "Fast Newton-Krylov Method for Unstructured Grids," *AIAA Journal*, Vol. 36, No. 4, April 1998, pp. 607–612.

[2] Geuzaine, P., "Newton-Krylov Strategy for Compressible Turbulent Flows on Unstructured Meshes," *AIAA Journal*, Vol. 39, No. 3, 1997, pp. 528–531.

[3] Barth, T. J. and Linton, S. W., "An Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation," 1995, AIAA-95-0221.

[4] Pueyo, A. and Zingg, D. W., "Efficient Newton-Krylov Solver For Aerodynamic Calculations," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 1991–1997.

[5] Spalart, P. and Allmaras, S., "A One-Equation Turbulence Model for Aerodynamic Flows," 1992, AIAA paper 92-0439.

[6] Godin, P., Zingg, D. W., and Nelson, T., "High-lift Aerodynamics Calculations WIth One- and Two-equation Turbulence Models," *AIAA Journal*, Vol. 35, No. 11, Feb. 1997.

[7] Ashford, G. A., *An Unstructured Grid Generation and Adaptive Solution Technique for High Reynolds Number Compressible Flows*, Ph.D. thesis, University of Michigan, 1996.

[8] Nelson, T. E., Zingg, D. W., and Johnston, G. W., "Compressible Navier-Stokes Computations of Multielement Airfoil Flows Using Multiblock Grids," *AIAA Journal*, Vol. 32, No. 3, 1997, pp. 506–511.

[9] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions Of The Euler Equations By Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," June 1981, AIAA paper 81-1259.

[10] Swanson, R. C. and Turkel, E., "On Central-Difference and Upwind Schemes," *J. Comp. Phys.*, Vol. 101, 1992, pp. 292–306.

[11] Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and van der Vorst, H., "Templates For The Solution Of Linear Systems: Building Blocks For Iterative Methods," Tech. rep., SIAM, 1994.

[12] Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm For Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Computing*, Vol. 7, 1986, pp. 856–869.

[13] Pueyo, A. and Zingg, D. W., "Airfoil Computations Using A Newton-GMRES Method." *Proceedings of the Fourth Annual Conference of the CFD Society of Canada*, CFD96, June 1997.

[14] Chow, E. and Saad, Y., "Experimental Study Of ILU Preconditioners For Indefinite Matrices," *J. of Comp. and Appl. Mathematics*, Vol. 87, 1997, pp. 387–414.

[15] Pulliam, T. H., "Efficient Solution Methods for the Navier-Stokes Equations," Lecture Notes for The Von Karman Institute.

[16] Nielsen, E. J., Anderson, W. K., Walters, R. W., and Keyes, D. E., "Application of Newton-Krylov Methodology to a Three-Dimensional Unstructured Euler Code," June 1995, AIAA Paper 95-1733-CP.