20th AIAA Computational Fluid Dynamics Conference
27 - 30 June 2011, Honolulu, Hawaii

AIAA 2011-3237

Dissipation-based continuation: a globalization for inexact-Newton solvers

Jason E. Hicken* Howard Buckley† Michal Osusky‡
David W. Zingg §
Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, M3H 5T6, Canada

Dissipation-based continuation (DBC) is a form of globalization suitable for inexact-Newton flow solvers and a robust alternative to pseudo-transient continuation. DBC uses a sequence of modified equations, each one a perturbation of the previous one. The modified equations are obtained by adding numerical dissipation to the discrete governing equations, with a continuation parameter controlling the magnitude of the dissipation. DBC begins with significant numerical dissipation, which increases the basin of attraction for Newton's method to converge using the free-stream as the initial iterate. The continuation parameter is then reduced, and the next modified equation in the sequence is inexactly solved using the previous solution as the initial iterate. The process is repeated until the dissipation is removed, the original equations are recovered, and the desired solution is obtained. We describe DBC in detail and its implementation in a multi-block finite-difference discretization. DBC is benchmarked against pseudo-transient continuation on a number of numerical experiments to quantify its robustness and efficiency, and it is shown to be generally superior.

I. Introduction

Although many research challenges remain, the field of computational fluid dynamics (CFD) is sufficiently mature that flow solvers are routinely used by industry for analyses. In this role, a CFD solution is used to assess a design or process, much like the results from a wind tunnel experiment. Although not yet routine, simulation-based optimization represents the next significant application of CFD in industry. In this context, CFD goes beyond informing the designer and helps automate the design process.

If a flow solver experiences convergence difficulties during a single analysis, the user can intervene and adjust parameters accordingly. Such user intervention is untenable during a simulation-based optimization, where hundreds or thousands of flow evaluations are necessary. A similar argument holds for probabilistic uncertainty quantification, where many samples are necessary to estimate the mean and variance in a quantity of interest. These applications motivate robust and efficient solution strategies for CFD.

Our focus in this paper is on the class of inexact-Newton solution algorithms, which have proven to be efficient for a wide range of steady[1–6] and unsteady[7–9] problems in CFD. In simple terms, these algorithms apply Newton's method to the nonlinear algebraic system that results from discretizing the flow equations; however, because they are based on Newton's method, these schemes are sensitive to the choice of initial iterate. When applied to unsteady problems, the solution from the previous time step may provide a suitable initial iterate. No such solution is available in steady-state problems, where a globalization strategy is needed to find an initial iterate that lies in the basin of attraction of Newton's method.

The globalization strategy has a significant impact on the efficiency and robustness of an inexact-Newton solver. The impact on robustness is clear: unless the globalization finds a suitable initial iterate, the flow solver will fail. Globalization can also represent a substantial percentage (50% or more) of the total solution

[*] Postdoctoral Fellow, AIAA Member

[†] Research Engineer, AIAA Member

[‡] PhD candidate, AIAA Student Member

[§] Professor and Director, J. Armand Bombardier Foundation Chair in Aerospace Flight, Tier 1 Canada Research Chair in Computational Aerodynamics, Associate Fellow AIAA time; thus, the rapid convergence of an inexact-Newton scheme in its terminal phase may be overshadowed by an inefficient globalization strategy.

Pseudo-transient continuation is a popular form of globalization in CFD applications and is based on a time-marching analogy; it is essentially an implicit-Euler time-marching method that gradually increases the time-step until Newton's method is recovered. Although pseudo-transient continuation has proven effective, it can fail when the flow is nearly unstable, i.e. the Jacobian has eigenvalues close to the imaginary axis. In addition, the method can introduce numerous parameters that require tuning for optimal performance.

Line-search and trust-region methods, developed for nonlinear optimization, can also be used to globalize Newton's method. These methods attempt to minimize the L^2 norm of the residual by selecting appropriate search directions and step lengths. Because they target the norm of the residual, optimization-based globalizations may converge to a local minimum that does not correspond to the solution. Nevertheless, Pawlowski *et al.*[10] compared several optimization-based strategies and demonstrated that they can be effective, at least for the incompressible flows they investigated.

Grid sequencing can be used as a globalization method when a set of nested grids is available. When two grid levels are available, for example, the nonlinear problem is first solved on the coarse grid. Subsequently, the coarse solution is interpolated onto the fine grid where it becomes the initial iterate. In finite-element methods the coarse solution can come from a lower-order polynomial basis. Grid sequencing can be highly effective,[6] but requires a nested family of grids that may not be available.

In this paper we continue our investigation of a dissipation-based globalization method.[11] In dissipation-based continuation (DBC), numerical dissipation is introduced into the discrete equations using a parameter λ that controls the amount of dissipation. For sufficiently large values of λ the modified nonlinear system is typically much easier to solve than the target system. Once the first modified system is solved, the value of λ can be decreased (reducing the dissipation), and a new nonlinear system can be initiated using the solution from the first problem. Repeating this process, a sequence of nonlinear systems results, with the solution of one being used as the initial iterate for the next.

In Ref. 11 we proposed and compared boundary-condition continuation and DBC for globalization of steady CFD problems. DBC was found to be especially robust and efficient relative to pseudo-transient and boundary-condition continuation. Indeed, in a recent investigation of induced-drag minimization,[12] we relied on DBC exclusively during the optimization runs. This study alone constitutes on the order 10^3 successful flow solutions on distinct geometries, since each iteration in an optimization involves a change in the aerodynamic shape.

We begin in Section II with a review of inexact-Newton methods and the subclass of Newton-Krylov algorithms. We consider two globalization strategies, which we describe in Section III: the proposed dissipation-based continuation and the benchmark pseudo-transient continuation. Results from inviscid, laminar, and turbulent cases are presented in Section IV. Conclusions can be found in Section V.

II. Inexact-Newton-Krylov Methods

In this section we review Newton's method and introduce our notation. We also discuss inexact-Newton methods and, in particular, Newton-Krylov algorithms.

A. Newton's Method

Consider a nonlinear algebraic system of the form $\boldsymbol{\mathcal{R}}(\mathbf{q}) = \mathbf{0}$, where $\boldsymbol{\mathcal{R}} : \mathbb{R}^N \rightarrow \mathbb{R}^N$. In CFD applications, the vector $\mathbf{q} \in \mathbb{R}^N$ denotes a discrete representation of the flow variables. For example, it may hold the variable values at nodal or cell-centered locations, or the coefficients in a polynomial expansion. The function $\boldsymbol{\mathcal{R}}$ is the residual representing the discretized flow equations.

At iteration n of Newton's method, the estimated root is given by $\mathbf{q}^{(n)}$, and we seek a perturbation $\boldsymbol{\Delta}\mathbf{q}^{(n)} = \mathbf{q}^{(n+1)} - \mathbf{q}^{(n)}$ such that the first-order Taylor series vanishes at the perturbed state. The perturbation is given by the solution of the linear system

$$\mathcal{A}^{(n)} \boldsymbol{\Delta}\mathbf{q}^{(n)} = -\boldsymbol{\mathcal{R}}^{(n)}, \tag{1}$$

where $\mathcal{R}^{(n)} = \mathcal{R}\left(\mathbf{q}^{(n)}\right)$ and

$$\mathcal{A}^{(n)} = \left.\frac{\partial \mathcal{R}}{\partial \mathbf{q}}\right|_{\mathbf{q}^{(n)}} \tag{2}$$

is the Jacobian evaluated at the state $\mathbf{q}^{(n)}$.

Newton's method converges provided $\mathcal{A}^{(n)}$ is non-singular and the initial iterate $\mathbf{q}^{(0)}$ is sufficiently close to the solution. Furthermore, the convergence rate will be quadratic in n if $\mathcal{A} = \partial\mathcal{R}/\partial\mathbf{q}$ is Lipschitz continuous near the solution.[13] The Jacobian is usually invertible for well-posed CFD discretizations, although it is often ill-conditioned.

B. Inexact-Newton

For the moment, suppose that a suitable initial iterate is available. Then it is easy to see that the efficiency of Newton's method depends on the algorithm used to find $\mathbf{\Delta q}^{(n)}$. An obvious choice is to solve the linear system (1). However, during the early iterations, the update $\mathbf{q}^{(n+1)} = \mathbf{q}^{(n)} + \mathbf{\Delta q}^{(n)}$ will not significantly reduce the norm of the nonlinear residual, because the linearized model of $\mathcal{R}(\mathbf{q}) = 0$ is not sufficiently accurate. Thus, solving for $\mathbf{\Delta q}^{(n)}$ exactly is unnecessary.

Inexact-Newton methods[14] take advantage of this observation that an approximate solution to (1) is more efficient than an exact solution during the early iterations. Specifically, the class of inexact-Newton methods seeks an update that satisfies

$$\|\mathcal{R}^{(n)} + \mathcal{A}^{(n)}\mathbf{\Delta q}^{(n)}\| \leq \eta\|\mathcal{R}^{(n)}\|, \tag{3}$$

where $\eta \in [0, 1)$ is the forcing parameter, and $\|\cdot\|$ denotes the 2-norm. Inexact-Newton methods were developed to take advantage of iterative linear solvers, which, unlike direct linear solvers, can solve the Newton update equation to the tolerance defined by η.

C. Jacobian-Free Newton-Krylov

Although an inexact-Newton algorithm can use any iterative linear solver, Krylov-subspace methods are particularly attractive. Krylov methods tend to be more efficient and robust than classical stationary methods like Gauss-Seidel or SOR. In addition, many Krylov-iterative methods, including GMRES[15] and BiCGStab,[16] do not require the Jacobian $\mathcal{A}^{(n)}$ to be explicitly formed. Instead, these methods only need matrix-vector products of the form $\mathcal{A}^{(n)}\mathbf{v}$, which can be approximated using first-order forward differences:

$$\mathcal{A}^{(n)}\mathbf{v} \approx \frac{\mathcal{R}(\mathbf{q}^{(n)} + \epsilon\mathbf{v}) - \mathcal{R}(\mathbf{q}^{(n)})}{\epsilon}. \tag{4}$$

We remark that some Krylov methods also require transposed-matrix-vector products; however, even these products can be evaluated without explicitly forming $\mathcal{A}$ by using reverse-mode automatic differentiation.[17]

Inexact-Newton-Krylov algorithms that make use of the approximation (4) are called Jacobian-free methods; see Knoll and Keyes[18] for a review. This does not mean that these solvers are matrix-free. For most practical problems, Krylov solvers must be preconditioned, and the preconditioning operation often requires an approximate Jacobian; however, this approximate Jacobian is cheaper to form and store than the full Jacobian.

A naive inexact-Newton-Krylov strategy may fail when solving the Reynolds-averaged Navier-Stokes (RANS) equations with the Spalart-Allmaras turbulence model, because the Krylov solver can reduce the residual of the turbulence model while allowing the mean flow residual to increase. This problem arises because, depending on the details of the implementation, the residual norm of the turbulence equation can be orders of magnitude larger than that of the mean flow equations; thus, it is possible for the Krylov solver to reduce the 2-norm of the global residual by targeting the turbulence model residual alone. A solution to this problem is to introduce equation scaling at the beginning of each nonlinear iteration that ensures the 2-norm of the residuals are the same order of magnitude. This approach is implemented for both globalization methods considered in this work. For further details on the scaling, see Ref. 6.

III. Globalization

This section describes the two globalization strategies that we consider. We begin with the well-established pseudo-transient continuation, which is used as our benchmark. Subsequently, we review the dissipation-based continuation proposed in Ref. 11. Globalization strategies address the nonlinearity of the system, but they also dictate the properties of the Newton update equation for the perturbed nonlinear problem. Therefore, we conclude this section by analyzing the impact that the globalization strategies have on the linear subproblems of Newton's method.

A. Pseudo-transient Continuation

If a solution is steady, then the eigenvalues of the Jacobian should have strictly positive real parts[a]. In this case, it may be possible to reach the steady solution by time-marching until the transient terms decay sufficiently.[19] Pseudo-transient continuation (PTC) uses this idea to find an initial iterate for Newton's method.

In the context of finding the steady solution (or an initial iterate), only the stability of the time marching scheme is important, not its time accuracy. In addition, while explicit schemes can be used, implicit schemes generally permit much larger time steps that quickly eliminate the transient terms. For these reasons, the implicit Euler method is the basis for PTC.

All formulations of PTC begin by adding a diagonal matrix $\mathcal{T}^{(n)}$, with strictly positive entries, to the Jacobian:

$$\left(\mathcal{T}^{(n)} + \mathcal{A}^{(n)}\right) \boldsymbol{\Delta}\mathbf{q}^{(n)} = -\boldsymbol{\mathcal{R}}^{(n)}. \tag{5}$$

Two cases are noteworthy. First, as $\mathcal{T}^{(n)} \to 0$ we recover Newton's method (1). Second, if $\mathcal{T}^{(n)} = \frac{1}{\Delta t}\mathcal{I}$, where $\mathcal{I}$ is the identity matrix, we obtain the implicit-Euler time marching scheme.

1. *Implementation of Pseudo-transient Continuation*

The general PTC algorithm described above is often modified and tuned for a specific discretization and application. We summarize these modifications for our implementation below.

Experience suggests that convergence can be accelerated if the elements of $\mathcal{T}^{(n)}$ have a spatial dependence, i.e. the "pure" implicit-Euler method is rarely used. Typically, a global time step is scaled by a function of the local mesh spacing:

$$\left(\mathcal{T}^{(n)}\right)_{ii} = \frac{1}{\Delta t_i^{(n)}} = \frac{1}{T_i \Delta t_{\text{ref}}^{(n)}}, \tag{6}$$

where $\Delta t_i^{(n)}$ is the local time step for variable i, $\Delta t_{\text{ref}}^{(n)}$ is the global reference time step, and T_i is the local scaling for Δt_i.

During PTC, the exact Jacobian $\mathcal{A}$ is replaced with a first-order Jacobian. The first-order Jacobian is obtained by neglecting the high-order dissipation and increasing the second-difference dissipation. Forming this approximate Jacobian does not introduce additional work, since the same matrix is factored using ILU(k)[20] to build the preconditioner. Factoring the matrix is one of the most expensive tasks, so the approximate Jacobian may not be updated and factored every iteration, but rather every m iterations. We use $m = 3$ for inviscid flows and $m = 1$ otherwise.

The reference time step appearing in (6) is given by

$$\Delta t_{\text{ref}}^{(n)} = a(b)^{m\lfloor \frac{n}{m} \rfloor},$$

where $\lfloor \cdot \rfloor$ is the floor operator[b]. This definition ensures that updates to $\Delta t_{\text{ref}}^{(n)}$ are consistent with the Jacobian-update period m. Typical ranges for a and b are $a \in [10^{-4}, 10^{-1}]$ and $b \in [1.2, 1.7]$. The default values of $a = 0.1$ and $b = 1.5$ are used unless otherwise noted. The local scaling for variable i is defined by

$$T_i = \left[J(1 + \sqrt[3]{J})\right]^{-1},$$

where J is the metric Jacobian of the coordinate mapping at the node where variable i is located.

[a] The eigenvalues might also have strictly negative parts, depending on the definition of the residual

[b] $\lfloor x \rfloor$ gives the largest integer less than or equal to x

PTC is terminated, and the inexact-Newton phase begins, when the relative residual is reduced by a user-specified tolerance:

$$\mathrm{R}_{\text{rel}}^{(n)} \equiv \frac{\|\boldsymbol{\mathcal{R}}^{(n)}\|}{\|\boldsymbol{\mathcal{R}}^{(0)}\|} \leq \mu, \tag{7}$$

where $\mu \in [10^{-4}, 10^{-1}]$. The default value for this tolerance is $\mu = 0.1$. Recall that during the inexact-Newton phase, the Jacobian-vector products are obtained using equation (4).

B. Dissipation-based Continuation

Numerical dissipation is often added to the discrete equations to maintain stability and capture shocks. It appears either explicitly through the addition of artificial dissipation or implicitly through the use of an upwind scheme. The contribution of numerical dissipation to the flux balance should be minimized to obtain an accurate solution; however, practitioners sometimes increase the dissipation to assist the solution algorithm at the expense of accuracy (see, e.g., Ref. 21). This practice motivates dissipation-based continuation (DBC).

DBC is a form of parameter continuation and is closely related to probability-one homotopy[c] methods.[22,23] The idea behind homotopy methods is simple. Suppose we wish to solve $\boldsymbol{\mathcal{R}}(\mathbf{q}) = \mathbf{0}$. We introduce a parameter $\lambda \in [0, \lambda_{\max}]$ and a modified residual $\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda)$, called the homotopy mapping, such that $\boldsymbol{\mathcal{F}}(\mathbf{q}, 0) = \boldsymbol{\mathcal{R}}(\mathbf{q})$. To be useful, the homotopy map should be chosen such that $\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda_{\max}) = \mathbf{0}$ is significantly easier to solve than the target problem $\boldsymbol{\mathcal{R}}(\mathbf{q}) = \mathbf{0}$. The algorithm begins with $\lambda_{\max}$ and decreases this parameter incrementally, solving a sequence of modified problems until $\lambda = 0$. The solution path defined by $\mathbf{g}(\lambda) = \{(\mathbf{q}, \lambda) \in \mathbb{R}^{N+1} : \boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda) = \mathbf{0}\}$ is followed approximately, using the previous solution as the initial guess for the next.

In the case of DBC, we adopt a modified residual (homotopy map) of the form

$$\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda) = \boldsymbol{\mathcal{R}}(\mathbf{q}) + \lambda \boldsymbol{\mathcal{D}}(\mathbf{q}),$$

where $\boldsymbol{\mathcal{D}}(\mathbf{q})$ is an appropriate numerical dissipation. We use a second-difference (first-order) scalar dissipation, because it is symmetric positive-definite for fixed $\mathbf{q}$ and uses only nearest-neighbours in the stencil. No pressure switch or limiter is required, or even desired, in the construction of $\boldsymbol{\mathcal{D}}$, since this would introduce additional nonlinearities into the modified residual. In practice, the residual $\boldsymbol{\mathcal{R}}$ is computed first to track convergence before adding the dissipation $\lambda\boldsymbol{\mathcal{D}}$. Although we consider finite-difference discretizations, it should be straightforward to implement DBC in most forms of discretization.

DBC is similar to continuation in the Reynolds number,[18] i.e. beginning with a small Reynolds number and increasing it gradually to the desired value. However, in the proposed method dissipation is added to all flow equations, while Reynolds-number continuation affects only the momentum and energy equations. Moreover, the influence of the Reynolds number is different in the momentum and energy equations. These distinctions may be important, since folds — points where the Jacobian is singular — can be introduced when using Reynolds-number continuation, see, for example, Ref. 24. In addition, DBC is well suited to hyperbolic equations like the Euler equations, where Reynolds-number continuation will cause instabilities as the limiting case of infinite Reynolds number is approached.

1. Implementation of Dissipation-based Continuation

Like the linear problems in Newton's method, solving the modified nonlinear problems $\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda) = 0$ exactly is inefficient. Instead, the modified subproblems are solved to a relative tolerance of $\tau_\lambda = 0.1$ with an upper bound of $n_{\max} = 30$ iterations for each value of λ. Note that, unlike PTC, the exact Jacobian is used throughout convergence.

The efficiency and robustness of DBC depends on the initial value of the continuation parameter and the schedule used to decrease λ to zero. Here, we vary the continuation parameter according to the formula

$$\lambda_i = \min\left(\frac{\lambda_{i-1}}{2}, \lambda_0 \mathrm{R}_{\text{rel}}^{\beta}\right),$$

where the relative residual, R_{rel}, is defined in equation (7). The initial value of the continuation parameter depends on the type of flow. For inviscid flows we typically use $\lambda_0 = 2$, while for laminar and turbulent

[c] Continuation, path-following, and homotopy are often used interchangeably in the literature.

flows a larger value of $\lambda_0 = 20$ is required. The parameter β controls the rate at which λ is decreased in the later stages of the globalization. For the results presented below, the value $\beta = 2$ has been adopted unless otherwise indicated.

When using DBC to solve the RANS equations with the Spalart-Allmaras turbulence model, second-difference dissipation is also added to the equation for the working variable $\tilde{\nu}$. No other modifications are considered in this work, although subsequent work may investigate more complex homotopy maps for the turbulence model.

C. Impact of Continuation Strategies on Linear Subproblems

Adding the dissipation operator $\lambda\boldsymbol{\mathcal{D}}$ to the governing equations has a regularizing effect on the nonlinear solution. Increasing the parameter λ has the qualitative effect of "smearing" the solution, which tends to make the free-stream more suitable as an initial iterate for Newton's method. This achieves our objective of globalizing the nonlinear iterations; however, the addition of dissipation also has an impact on the linear subproblems. Is this impact detrimental or beneficial? In this section, we investigate how DBC affects the linear subproblems, and compare its impact with that of PTC.

We will use a simple one-dimensional advection equation to model the impact of PTC and DBC on the linear Newton subproblems. In particular, consider the boundary-value problem

$$\begin{aligned} \frac{\partial}{\partial x}\left(\mathcal{U}\right) &= \mathcal{F}, \qquad \forall\, x \in \Omega = [0,1], \\ \mathcal{U}(0) &= \mathcal{U}_L, \end{aligned}. \tag{8}$$

Note that the actual choice for $\mathcal{F}$ and $\mathcal{U}_L$ will not be important in our study of the discrete system matrix.

Consider a uniform grid on the domain $[0,1]$ with node locations $x_i = hi$, where $i = 0, 1, 2, \ldots, n$ and $h = 1/n$. For this grid, a second-order accurate discretization of (8) is given by

$$Q\left(u_h\right) = Hf - e_0\lambda_0\left(e_0^T u_h - \mathcal{U}_L\right), \tag{9}$$

where $e_0 = [1\ 0\ 0\ \cdots\ 0]^T$ and

$$\begin{aligned} f &= \begin{bmatrix} \mathcal{F}(x_0) & \mathcal{F}(x_1) & \cdots & \mathcal{F}(x_n) \end{bmatrix}^T, \\ H &= h\,\mathrm{diag}\left(\frac{1}{2}, 1, 1, \ldots, 1, \frac{1}{2}\right), \\ Q &= \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2} & 0 & \frac{1}{2} \\ & & & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}. \end{aligned}$$

The matrix operators Q and H define the standard second-order accurate summation-by-parts (SBP) operator $D = H^{-1}Q$; see Ref. 25. The discretization (9) imposes the boundary conditions weakly using penalty terms.[26,27]

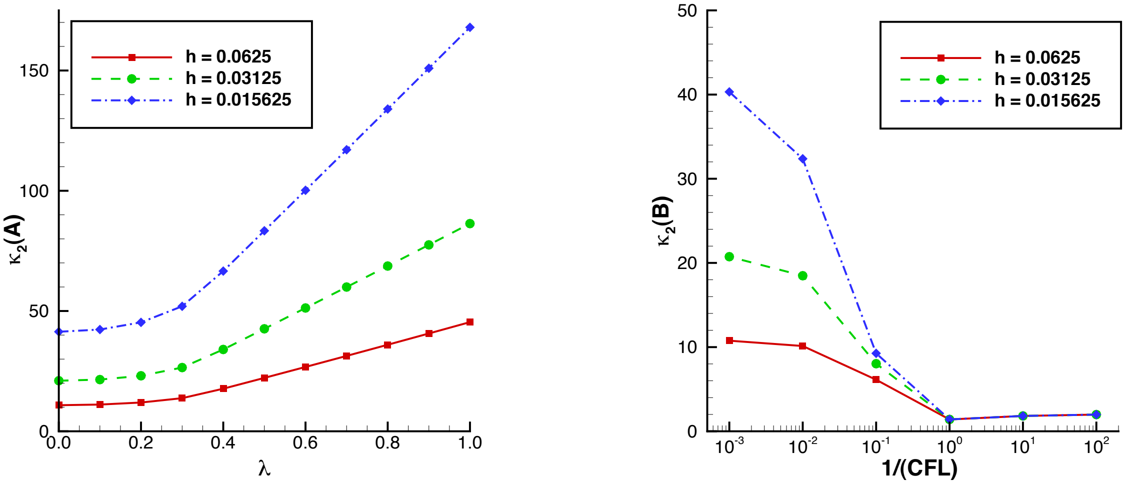
For the present model problem, the dissipation operator can be defined by $\lambda\boldsymbol{\mathcal{D}} \equiv \lambda\Delta^T\Delta$, where Δ is the undivided forward difference operator.[d] Including the dissipation operator in the discretization and rearranging, we arrive at

$$Au_h = Hf + e_0\mathcal{U}_L$$

where

$$A = Q + e_0\, e_0^T + \lambda\Delta^T\Delta.$$

[d] The forward difference operator is defined by a rectangular matrix with $n+1$ columns and n rows.



(a) dissipation-based continuation

(b) pseudo-transient continuation

Figure 1. Condition number for the model advection problem as a function of λ in DBC (left) or inverse Courant number in PTC (right) for $h \in \{0.0625, 0.03125, 0.015625\}$.

The matrix A models the system matrix of the modified residual in DBC. To model PTC, we simply include a diagonal matrix on the left hand side of (9):

$$Bu_h = Hf + e_0\mathcal{U}_L$$

where

$$B = \frac{1}{h\,\text{CFL}} H + Q + e_0\, e_0^T, w$$

where $(h\,\text{CFL})^{-1}H$ plays the role of $\boldsymbol{\mathcal{T}}$, and CFL denotes the Courant number.

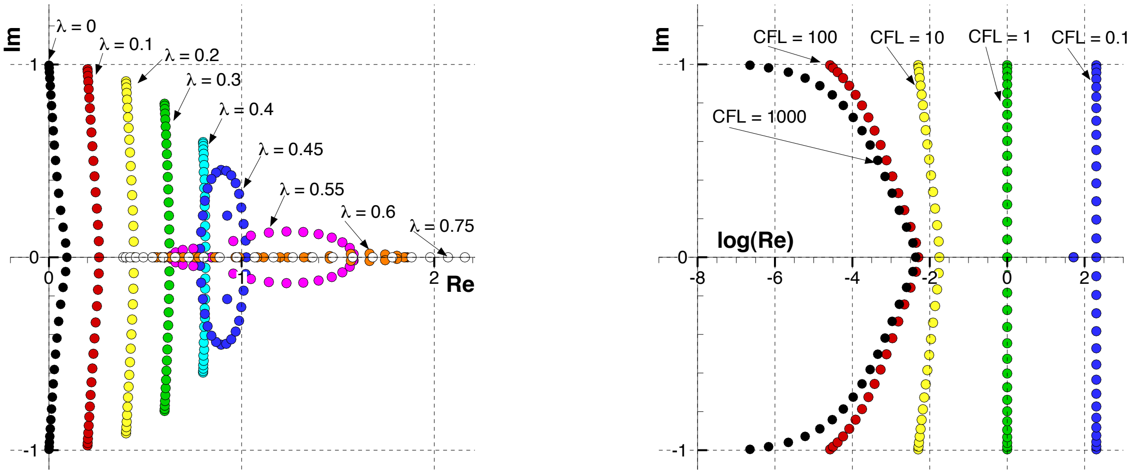
Figure 1 plots the 2-norm condition number[e] of A and B as a function of λ and $(\text{CFL})^{-1}$, respectively. Results are plotted for grid sizes corresponding to $n = 17, 33$, and 65 (i.e. $h = 0.0625, 0.03125$, and 0.015625). For sufficiently large λ, there is a linear relationship between the condition number of A and λ. Increasing the magnitude of the dissipation leads to ill-conditioning, because the operator $\boldsymbol{\mathcal{D}}$ is singular — all constant vectors are in the null space of Δ. In contrast, decreasing CFL in PTC improves the conditioning of the matrix B; as CFL $\to 0$, B tends toward a diagonal matrix.

The relationship between λ and the conditioning of A suggests that DBC may lead to stiff linear subproblems in Newton's method. However, our experience has been that the subproblems in DBC are well behaved. To understand why the condition number provides an incomplete picture here, we plot the eigenvalue spectrum of the matrices A and B in Figure 2 for several values of λ and CFL. The spectra correspond to a fixed grid size of $h = 0.03125$, although the results are similar for other mesh spacings. Note that Figure 2(b) plots the natural logarithm of the real part.

As λ increases, the spectrum of A is transformed from one that is clustered along the imaginary axis, to one that is real and positive. Essentially, the hyperbolic problem is transformed into an elliptic one. This is not surprising, since $\lambda\Delta^T\Delta$ is a symmetric positive-definite matrix. In the case of PTC, the spectrum of B is shifted to the right as CFL decreases ($(\text{CFL})^{-1}$ increases), but, unlike the spectrum of A, the vertical extent of the spectrum is unchanged.

The change in the spectrum of A hints at why the linear subproblems in DBC are well behaved. First, for $\lambda \geq \frac{1}{2}$, one can easily show that the matrix A is irreducibly diagonally dominant. Besides guaranteeing invertibility,[28] this diagonal dominance helps improve the robustness of preconditioners based on incomplete lower-upper factorizations, because of increased pivot sizes. Second, the eigenvalues of A become strictly real and positive as λ increases past some critical value λ^*: the matrix A becomes positive definite.

[e] The 2-norm condition number of an arbitrary matrix A is defined by $\kappa_2(A) \equiv \|A\|_2\|A^{-1}\|_2$



(a) dissipation-based continuation

(b) pseudo-transient continuation

Figure 2. Discrete spectrum for various values of λ and CFL for a uniform grid with $h = 0.03125$. The x-axis in Figure (b) is the natural logarithm of the real part of the spectrum. The black-filled circles in both figures can be regarded as the spectrum of the system matrix for the unmodified problem (9).

Krylov-subspace methods generally solve systems involving positive-definite matrices much more readily than systems with indefinite matrices. For example, restarted GMRES(m) is guaranteed to converge for any $m \geq 1$ if A is positive definite.[28]

In summary, although the condition number of the system matrix A in DBC grows as λ increases, this growth is countered by improved diagonal dominance and definiteness.

IV. Results

In the results presented below, we apply PTC and DBC to a summation-by-parts[25] finite-difference discretization. Boundary conditions are imposed weakly using penalty terms called simultaneous approximation terms (SATs)[26,27] analogous to those found in the model problem discretization (9). The SAT methodology is also used at block interfaces to couple the solution across blocks. See Refs. 29 and 30 for further details on the discretization.

A. Inviscid Test Case

We begin with a comparison of PTC and DBC over a range of inviscid operating conditions around the NACA 0012 airfoil[f]. Additional results for inviscid flows can be found in Ref. 11.

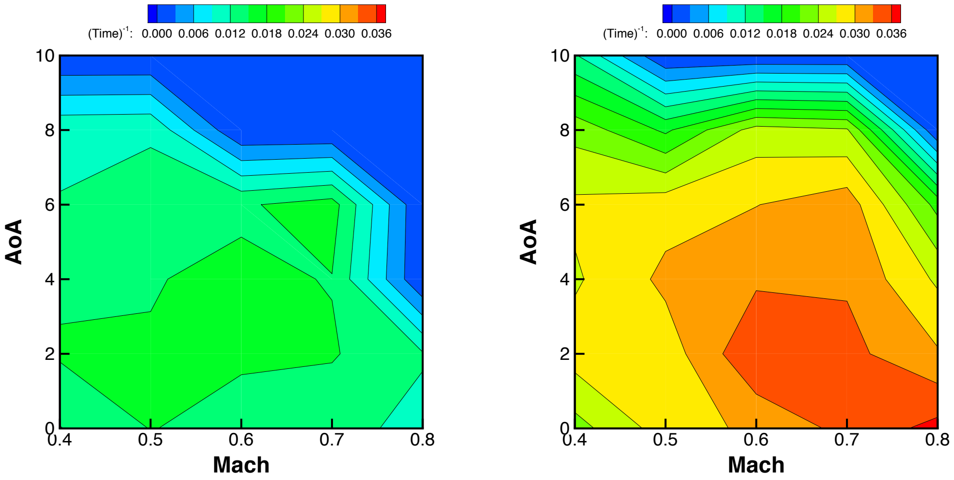
The Mach number is varied in increments of 0.1 over the range $[0.4, 0.8]$, and the angle of attack is varied in increments of 2° over the range $[0^\circ, 10^\circ]$. A 3-block C-grid is used, with 18 395 nodes and an off-wall spacing of approximately 10^{-4}.

The two globalizations are applied with their default parameters. We emphasize that if a globalization fails at a particular operating condition, then it may be possible to converge the solver if the default parameters are changed; however, to illustrate the robustness of DBC, we avoid such tuning of the parameters here.

The success rate of DBC on the 30 operating conditions is 83%, while the success rate of PTC is 66.7%. Note, there is no case for which DBC failed and PTC succeeded. Although the failure rate for even DBC seems high (17%), some of the operating conditions do not produce steady flow; this is discussed further below.

Figure 3 depicts the qualitative performance of the two globalization strategies. Each contour plot shows the inverse time required by the respective methods to reduce the relative residual (7) by 10 orders of magnitude. Because the metric plotted is $1/T$, failure to converge is indicated by the zero contour, and

[f] With the usual sharp trailing edge modification



(a) pseudo-transient continuation

(b) dissipation-based continuation

Figure 3. Contour plots of $1/T$, where T is the time required to reduce the relative residual ten orders.

faster convergence is indicated by larger values.

From the two plots in Figure 3, we can see that both globalizations have difficulty at high angles of attack, where the flow is potentially unsteady and convergence should not be expected (the Jacobian has eigenvalues with positive or very small negative real parts). Nevertheless, DBC has a significantly smaller region of failure. The figures also show that, on average, DBC converges in less time than PTC. For cases in which both methods converged, DBC converged in 48% of the time required by PTC.

B. Laminar-flow Test Cases

1. RAE 2822 Angle of Attack Sweep

The first laminar-flow test case is an angle of attack sweep of the RAE 2822 airfoil geometry. The sweep is performed at Reynolds numbers of Re $=$ 200 and Re $=$ 1000. This case is intended to compare the globalization strategies as the laminar flow changes from low-lift conditions to high-lift conditions.

The geometry is surrounded by a single block C-grid composed of 18785 nodes. There are 65 nodes in the off-wall direction, and 193 nodes along the airfoil. The grid is suitable for turbulent flows modelled with the Reynolds-average Navier-Stokes equations, and the off-wall normal spacing is approximately 2.29×10^{-6} chord units.

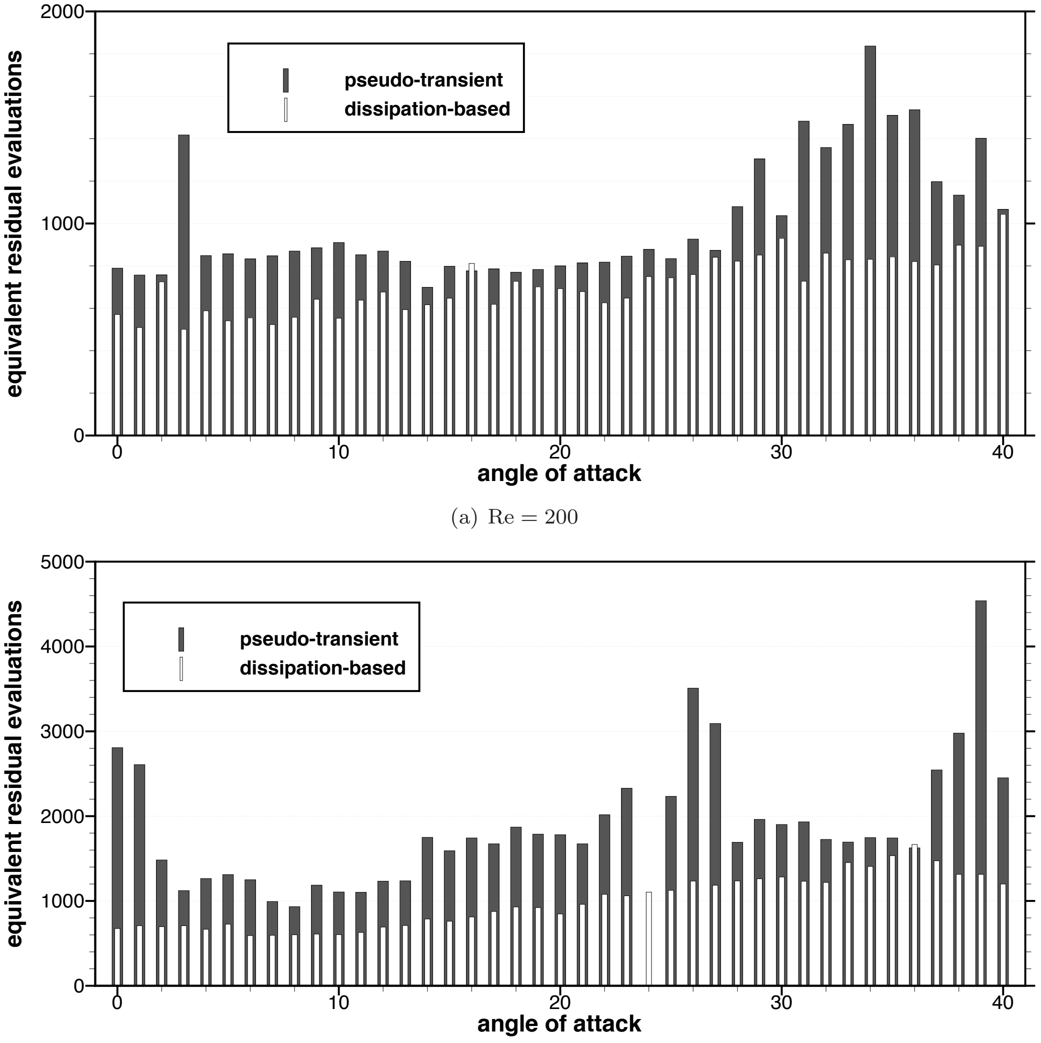
The default solver parameters are adopted, with the following exceptions. PTC is continued until the 2-norm of the relative residual has been decreased by $\mu = 5 \times 10^{-5}$. For DBC, the formula that controls λ uses $\beta = 1.5$, and the modified subproblems are solved to $\tau_\lambda = 0.01$.

Figure 4 compares the run-time performance of PTC and DBC over the range of angles $\alpha = [0°, 40°]$. CPU time is normalized by the time required to evaluate the nonlinear residual $\boldsymbol{\mathcal{R}}$; we refer to this normalized CPU time as equivalent residual evaluations. Results for Re $=$ 200 and Re $=$ 1000 are plotted in the upper and lower figures, respectively. The two methods are considered converged when the 2-norm of the residual has been reduced 10 orders from its initial value.

The figures clearly show that DBC outperforms PTC on this test case. For Re $=$ 200, DBC is on average 26% faster than PTC, which is faster for only one case out of 41. For Re $=$ 1000, DBC performs even better, with run-times 45% faster on average; this average excludes the case at $\alpha = 24°$, for which PTC did not converge. Again, the PTC method is faster in only one case.

2. ONERA M6 Wing at High-Lift Conditions

The second laminar-flow test case involves the ONERA M6 wing geometry at several high-lift conditions. The Mach and Reynolds numbers are fixed at 0.25 and 1000, respectively. The seven high-lift cases correspond to particular angles of attack, which are listed in Table 1. The grid used for the ONERA M6 wing has 16



(a) Re = 200

(b) Re = 1000

Figure 4. Number of equivalent residual evaluations required by the two globalization strategies to reduce the L2-norm of the residual 10 orders for the RAE 2822 angle of attack sweeps.

Table 1. **High-lift cases and defining angles of attack for the ONERA M6 Wing**

Case	1	2	3	4	5	6	7
A.O.A	12°	15°	17°	18°	20°	22°	24°

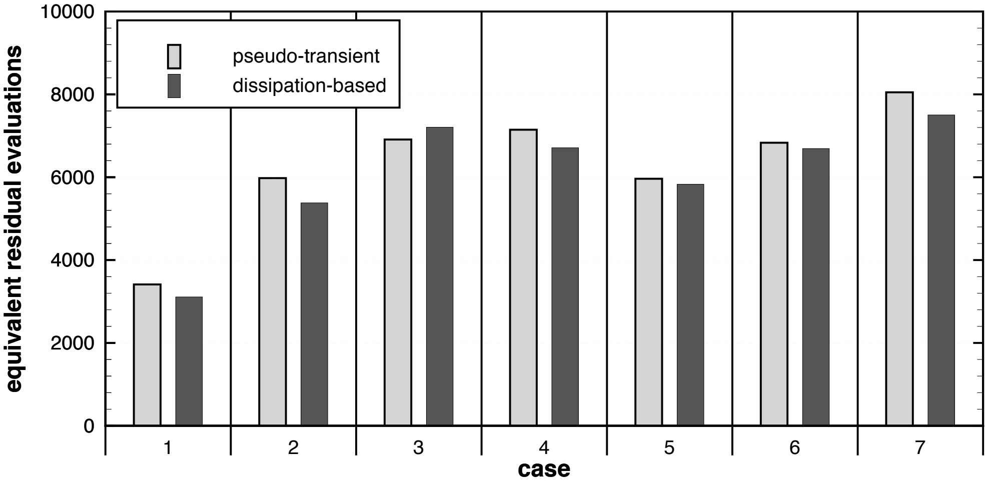


Figure 5. Number of equivalent residual evaluations required by the two globalization strategies to solve the ONERA M6 angle of attack cases listed in Table 1.

blocks with $49 \times 49 \times 49$ nodes per block, for a total of 1.88×10^6 nodes. Mesh spacing in the normal direction at the surface of the wing is approximately 7×10^{-5}.

The default PTC and DBC parameters are used with the following modifications. The transition from PTC to full inexact-Newton occurs when the relative residual has been reduced below $\mu = 5 \times 10^{-2}$. In DBC, the initial value $\lambda_0 = 15$ is used and $\beta = 1$ in the λ update formula.

For each case in Table 1, Figure 5 plots the number of equivalent residual evaluations required by PTC and DBC to reduce the relative residual 10 orders of magnitude. We see that there is no significant difference in run-time between the two globalization strategies. Thus, for this test case, DBC is a competitive alternative to PTC.

C. Preliminary Results for Turbulent Test Case

In this section, we present preliminary results using DBC to globalize the solution of the RANS equations with the one-equation Spalart-Allmaras turbulence model. The RAE 2822 airfoil geometry is adopted for the test, and a range of Mach numbers and flow angles are considered. For each Mach number, a Reynolds number is calculated based on the mean-aerodynamic chord of a B737 flying at 41,000 ft. For reference, Table 2 lists the set of Mach numbers and corresponding Reynolds numbers used for the tests. The grid is the same one used for the laminar RAE 2822 angle of attack sweeps and is described in Section IV.B.1.

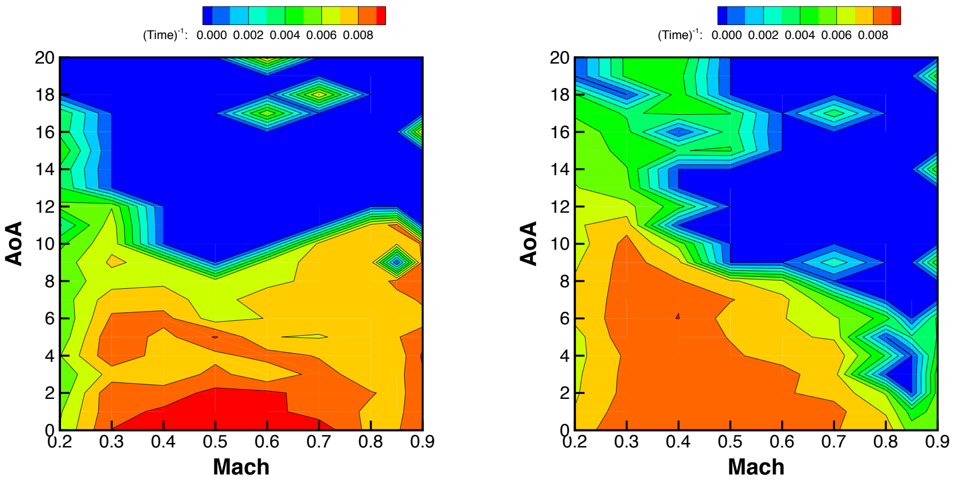
In PTC, a significantly smaller initial time step must used relative to the inviscid and laminar cases: we use $a = 10^{-4}$. Moreover, PTC is continued until the relative residual has been reduced below $\mu = 10^{-4}$. For DBC globalization, $\lambda_0 = 20$, $\beta = 1$, and $\tau_\lambda = 0.01$. All other parameters for PTC and DBC are assigned their default values.

As with the inviscid NACA0012 results, we use the inverse time metric to compare the two globalization schemes. Figure 6 plots the inverse of the CPU time for the set of operating conditions considered. Blue contours correspond to runs that failed to converge, either because of time constraints or non-physical densities or pressures. Larger contour values (e.g. red) correspond to faster run-times.

Comparing Figures 6(a) and 6(b), we see that there are conditions for which PTC converges and DBC does not, and vice versa. In particular, DBC is more robust for low Mach numbers and high angles of attack. PTC appears to be more robust for high Mach numbers and $\alpha \in [0, 10]$. The high Mach number, high angle

Table 2. Operating conditions for the turbulent angle-of-attack sweep.

Mach	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.85	0.9
Re $\times 10^7$	1.85	0.748	0.997	1.25	1.50	1.74	1.99	2.12	2.24



(a) pseudo-transient continuation

(b) dissipation-based continuation

Figure 6. Contour plots of $1/T$, where T is the time required to reduce the relative residual ten orders.

of attack cases correspond to flows with shock-induced boundary-layer separation; it is this characteristic of the flow that we believe is leading to convergence difficulties for DBC.

We emphasize that these results are preliminary. For the Spalart-Allmaras turbulence model, the performance of DBC is respectable compared with PTC; however, there remains significant scope to improve its speed and robustness. The Spalart-Allmaras model is highly nonlinear, and several terms in the turbulence model may benefit from some form of parameter continuation. We have only begun to explore these possibilities.

V. Conclusions

Future industrial applications of CFD, including simulation-based optimization and uncertainty quantification, will require robust and efficient solution algorithms that can handle a variety of geometries and operating conditions without user intervention. Dissipation-based continuation has been developed with such applications in mind.

We have investigated the impact of dissipation-based continuation on the linear subproblems that arise in Newton's method. Although the continuation strategy leads to modest increases in the condition number, this is offset by the increased diagonal dominance and positive-definiteness of the system matrix.

For inviscid and laminar flows, the results indicate that dissipation-based continuation is a robust and efficient alternative to the popular pseudo-transient continuation. In the context of the Reynolds-averaged Navier-Stokes equations with the one-equation Spalart-Allmaras turbulence model, preliminary results suggest that dissipation-based continuation is competitive with pseudo-transient continuation. Future work will consider parameter continuation within the Spalart-Allmaras model itself.

Acknowledgments

The authors gratefully acknowledge financial assistance from Bombardier Aerospace, the Natural Sciences and Engineering Research Council (NSERC), the Canada Research Chairs program, Mathematics of Information Technology and Complex systems (MITACS), and the University of Toronto.

Computations were performed on the general purpose cluster at the SciNet HPC Consortium. SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund - Research Excellence; and the University of Toronto.

References

[1] Keyes, D. E., “Aerodynamic applications of Newton-Krylov-Schwarz solvers,” *Proceedings of the 14th International Conference on Numerical Methods in Fluid Dynamics*, Springer, New York, 1995, pp. 1–20.

[2] Pueyo, A. and Zingg, D. W., “Efficient Newton-Krylov solver for aerodynamic computations,” *AIAA Journal*, Vol. 36, No. 11, Nov. 1998, pp. 1991–1997.

[3] Blanco, M. and Zingg, D. W., “Fast Newton-Krylov method for unstructured grids,” *AIAA Journal*, Vol. 36, No. 4, April 1998, pp. 607–612.

[4] Groth, C. P. and Northrup, S. A., “Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh,” *17th AIAA Computational Fluid Dynamics Conference*, No. AIAA–2005–5333, Toronto, Ontario, Canada, June 2005.

[5] Nejat, A. and Ollivier-Gooch, C., “A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows,” *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2582–2609.

[6] Chisholm, T. T. and Zingg, D. W., “A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows,” *Journal of Computational Physics*, Vol. 228, No. 9, 2009, pp. 3490–3507.

[7] Lucas, P., Bijl, H., and van Zuijlen, A. H., “Efficient unsteady high Reynolds number flow computations on unstructured grids,” *Computers and Fluids*, Vol. 39, No. 2, 2010, pp. 271–282.

[8] Rumpfkeil, M. P. and Zingg, D. W., “The optimal control of unsteady flows with a discrete adjoint method,” *Optimization and Engineering*, Vol. 11, No. 1, 2010, pp. 5–22.

[9] Tabesh, M. and Zingg, D. W., “Efficient implicit time-marching methods using a Newton-Krylov algorithm,” *47th AIAA Aerospace Science Meeting and Exhibit*, No. AIAA–2009–0164, Orlando, Florida, Jan. 2009.

[10] Pawlowski, R. P., Shadid, J. N., Simonis, J. P., and Walker, H. F., “Globalization techniques for Newton-Krylov methods and applications to the fully coupled solution of the Navier– Stokes equations,” *SIAM Review*, Vol. 48, No. 4, 2006, pp. 700–721.

[11] Hicken, J. E. and Zingg, D. W., “Globalization strategies for inexact-Newton solvers,” *19th AIAA Computational Fluid Dynamics Conference*, No. AIAA-2009-4139, San Antonio, Texas, United States, June 2009.

[12] Hicken, J. E. and Zingg, D. W., “Induced-drag minimization of nonplanar geometries based on the Euler equations,” *AIAA Journal*, Vol. 48, No. 11, Nov. 2010, pp. 2564–2575.

[13] Kelley, C. T., *Solving Nonlinear Equations with Newton’s Method*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[14] Dembo, R. S., Eisenstat, S. C., and Steihaug, T., “Inexact Newton methods,” *SIAM Journal on Numerical Analysis*, Vol. 19, No. 2, 1982, pp. 400–408.

[15] Saad, Y. and Schultz, M. H., “GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, July 1986, pp. 856–869.

[16] van der Vorst, H. A., “BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 13, No. 2, March 1992, pp. 631–644.

[17] Griewank, A., *Evaluating Derivatives*, SIAM, Philadelphia, PA, 2000.

[18] Knoll, D. and Keyes, D., “Jacobian-free Newton-Krylov methods: a survey of approaches and applications,” *Journal of Computational Physics*, Vol. 193, No. 2, 2004, pp. 357–397.

[19] Lomax, H., Pulliam, T. H., and Zingg, D. W., *Fundamentals of Computational Fluid Dynamics*, Springer–Verlag, Berlin, Germany, 2001.

[20] Meijerink, J. A. and van der Vorst, H. A., “An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix,” *Mathematics of Computation*, Vol. 31, No. 137, Jan. 1977, pp. 148–162.

[21] Liersch, C. M., Streit, T., and Visser, K. D., “Numerical implications of spanwise camber on minimum induced drag configurations,” *47th AIAA Aerospace Science Meeting and Exhibit*, No. AIAA–2009–898, Orlando, Florida, Jan. 2009.

[22] Watson, L. T., “Globally convergent homotopy algorithms for nonlinear systems of equations,” *Nonlinear Dynamics*, Vol. 1, No. 2, March 1990, pp. 143–191.

[23] Allgower, E. L. and Georg, K., *Acta Numerica*, chap. Continuation and Path Following, Cambridge University Press, 1993, pp. 1–64.

[24] Walker, H. F., “An adaptation of Krylov subspace methods to path following problems,” *SIAM Journal on Scientific Computing*, Vol. 21, No. 3, 1999, pp. 1191–1198.

[25] Kreiss, H.-O. and Scherer, G., “Finite element and finite difference methods for hyperbolic partial differential equations,” *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. de Boor, Mathematics Research Center, the University of Wisconsin, Academic Press, 1974.

[26] Funaro, D. and Gottlieb, D., “A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations,” *Mathematics of Computation*, Vol. 51, No. 184, Oct. 1988, pp. 599–613.

[27] Carpenter, M. H., Gottlieb, D., and Abarbanel, S., “Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes,” *Journal of Computational Physics*, Vol. 111, No. 2, 1994, pp. 220–236.

[28] Saad, Y., *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, 2nd ed., 2003.

[29] Hicken, J. E. and Zingg, D. W., “A parallel Newton-Krylov solver for the Euler equations discretized using simultaneous approximation terms,” *AIAA Journal*, Vol. 46, No. 11, Nov. 2008, pp. 2773–2786.

[30] Osusky, M., Hicken, J. E., and Zingg, D. W., “A parallel Newton-Krylov-Schur flow solver for the Navier-Stokes equations using the SBP-SAT approach,” *48th AIAA Aerospace Sciences Meeting*, No. AIAA–2010–0116, Orlando, Florida, Jan. 2010.