19th AIAA Computational Fluid Dynamics
22 - 25 June 2009, San Antonio, Texas

AIAA 2009-4139

Globalization strategies for inexact-Newton solvers

Jason E. Hicken* and David W. Zingg †

Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, M3H 5T6, Canada

Globalization strategies are necessary in practical inexact-Newton flow solvers to ensure convergence when the initial iterate is far from the solution. In this work, we present two novel globalizations based on parameter continuation. The first continuation method parameterizes the boundary conditions while the second parameterizes the numerical dissipation. In both cases, a continuation parameter is used to create a sequence of modified nonlinear equations. The solution of each equation in the sequence provides an initial estimate for the subsequent problem until the desired convergence tolerance is reached. When applied to benign inviscid flows, the proposed globalization methods have similar efficiency compared with the more common pseudo-transient continuation. They are significantly more robust on difficult inviscid problems. Moreover, the continuation-based methods require less tuning.

I. Introduction

Newton's method is a powerful and general procedure for solving nonlinear algebraic equations. In the context of computational fluid dynamics (CFD), the method is frequently used to solve steady-state[1–6] and transient problems.[7,8] Newton's method is attractive in these applications, and in general, because of its rapid convergence: under suitable conditions it will converge quadratically. Unfortunately, the basic method is very sensitive to the initial iterate and may diverge if it is not sufficiently close to the solution.

In transient applications, the solution from the previous time-step is usually an excellent initial iterate. In contrast, there is generally no acceptable initial iterate available for steady flows. Hence, practitioners must use a globalization strategy when applying Newton's method to steady problems. The globalization seeks to move arbitrary initial iterates into the basin of attraction of Newton's method. The globalized algorithm should be robust and efficient.

Pseudo-transient continuation is often used in CFD to globalize Newton's method.[9] The strategy is based on time-marching the flow equations until Newton's method can be initiated. Most algorithms in this class resemble implicit-Euler time marching but use spatially varying time steps. Although often successful as a globalization strategy, pseudo-transient continuation 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.

Newton's method can also be globalized using line-search and trust-region approaches popular in optimization. Generally, these methods seek directions and step lengths that reduce the norm of the residual, and, consequently, may converge to a local minimum of the residual norm. Nevertheless, Pawlowski *et al.* compared several such approaches and demonstrated that they can be effective as globalization strategies, at least in the context of the incompressible Navier-Stokes equations.[10]

The globalization methods we present in this work use parameter continuation and are related to probability-one homotopy[11] and path-following algorithms.[12] Homotopy and path-following methods are applied in fixed-point problems, root finding, bifurcation problems, and interior-point optimization algorithms, to name a few examples. These methods seek to find a set of solutions as a parameter varies; for example, the roots of $f(z) = z^2 + 2z + \lambda$ as λ varies. In the proposed methods, a parameter is introduced to modify the boundary conditions or numerical dissipation, with the aim of making a sequence of nonlinear problems that are easier to solve.

[*] PhD Candidate, AIAA Student Member

[†] Professor and Director, Tier 1 Canada Research Chair in Computational Aerodynamics, Associate Fellow AIAA

There are few examples of parameter continuation in the computational aerodynamics literature. A notable exception is the early work of Bailey and Beam.[13] Rather than globalization, they used arclength continuation to obtain a continuous lift versus angle-of-attack curve.

The paper begins with a brief review of Newton's method applied to CFD problems. We then discuss the globalization strategies considered in this work, beginning with the popular pseudo-transient method of continuation, which is used as a benchmark for the proposed globalization methods. We introduce parameter continuation and provide some general theoretical results for this globalization strategy. We specialize to the cases of boundary-condition and dissipation-based continuation and discuss implementation details. Finally, the proposed methods are demonstrated and compared using several challenging steady-state flow problems.

II. Newton's Method and Inexact-Newton Methods

We assume that most readers are familiar with Newton's method; nevertheless, a brief review is useful to introduce notation and definitions. We consider nonlinear algebraic systems of the form $\boldsymbol{\mathcal{R}}(\mathbf{q}) = \mathbf{0}$, where $\boldsymbol{\mathcal{R}} : \mathbb{R}^N \rightarrow \mathbb{R}^N$. In the context of CFD, the vector $\mathbf{q} \in \mathbb{R}^N$ is a discrete representation of the flow variables; it may hold the variable values at nodal or cell-centered locations, or it may hold coefficients in a polynomial or Fourier 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. Hence, 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 $\boldsymbol{\mathcal{R}}^{(n)} = \boldsymbol{\mathcal{R}}\left(\mathbf{q}^{(n)}\right)$ and

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

is the Jacobian matrix 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. In addition, convergence will be quadratic in n if $\mathcal{A} = \partial \boldsymbol{\mathcal{R}} / \partial \mathbf{q}$ is Lipschitz continuous near the solution.[14] The Jacobian matrix is usually invertible for well-posed CFD discretizations, although it is often ill-conditioned. The initial iterate is a more significant issue.

The efficiency of Newton's method is closely related to the algorithm used to solve the linear systems (1). An important observation is that exact solutions are unnecessary during the early iterations of Newton's method. Rather than solve equation (1) exactly, the class of inexact-Newton methods seeks an update that satisfies

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

where $\eta \in [0, 1)$ is the so-called forcing parameter. 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.

A Newton-Krylov algorithm is an inexact-Newton method that uses a Krylov iterative solver, such as GMRES[15] or BiCGStab.[16] An advantage of using Krylov solvers in Newton's method is that these solvers do not need the Jacobian matrix to be explicitly calculated. Krylov iterative methods only need matrix-vector products $\mathcal{A}^{(n)}\mathbf{v}$, which can be approximated using first-order forward differences:

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

We should 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]

The globalizations presented in this work have been implemented in a Newton-Krylov framework to solve the discrete Euler equations. The Euler equations are discretized on multi-block grids using summation-by-parts (SBP) finite-difference operators, and simultaneous approximation terms (SATs) are used to couple blocks and impose boundary conditions. The flexible generalized minimal residual (FGMRES) method[18] is adopted as the Krylov solver and preconditioned with a parallel approximate-Schur preconditioner.[19] See reference 20 for further details on the discretization and the solution of the linear-update equation.

III. Pseudo-transient Continuation

If the residual is independent of time, and the eigenvalues of its Jacobian have negative real parts, then a solution to $\boldsymbol{\mathcal{R}} = \mathbf{0}$ can be sought by introducing a time derivative and integrating to steady state.[21] This motivates the use of time-marching schemes to find steady-state solutions of the Euler and Navier-Stokes equations and is the basis for pseudo-transient continuation.

When choosing a time-marching method to globalize Newton's method, the stability properties of the method are paramount: time-accuracy is not important in this application. The implicit-Euler time-marching scheme is attractive, because it is unconditionally stable, and its time linearization becomes Newton's method in the limit of infinite step sizes. Thus, the solution algorithm can smoothly transition from the continuation method to Newton's method.

Pseudo-transient continuation is defined by the linear system

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

where $\mathcal{T}^{(n)}$ is a diagonal matrix with positive entries on the diagonal. Note that as $\mathcal{T}^{(n)} \rightarrow 0$ we recover Newton's method, equation (1). On the other hand, 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. However, pseudo-transient continuation typically uses a spatially varying time step, so it is not a time-marching scheme in the strict sense. For example, it is common to scale a global time step by the local mesh spacing to accelerate convergence. In light of this, we express the elements in the diagonal matrix $\mathcal{T}^{(n)}$ in the form

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

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

A. An implementation of pseudo-transient continuation

This section provides a brief summary of the pseudo-transient continuation algorithm used as a benchmark in this work. Additional details can be found in Ref. 20.

The algorithm is broken into two phases: an approximate-Newton start-up phase and a terminal inexact-Newton phase. During the approximate-Newton phase, the exact Jacobian matrix $\mathcal{A}$ is replaced with a first-order Jacobian, which 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(1) to build the Schur preconditioner. However, since factoring the matrix is one of the most expensive tasks, the approximate Jacobian is updated and factored only every m iterations; $m = 3$ is used here.

The reference time step during the approximate-Newton phase is given by

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

where $\lfloor \cdot \rfloor$ is the floor operator[a]. This operator ensures that updates to $\Delta t_{\text{ref}}^{(n)}$ are consistent with the Jacobian-update period m. Typical ranges for a and b are $a \in [0.01, 0.1]$ and $b \in [1.2, 1.7]$. 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 at the node where variable i is located.

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

The algorithm switches to the terminal inexact-Newton phase when the relative residual is reduced one order of magnitude:

$$\mathrm{R}_{\mathrm{rel}}^{(n)} \equiv \frac{\|\boldsymbol{\mathcal{R}}^{(n)}\|}{\|\boldsymbol{\mathcal{R}}^{(0)}\|} \leq 0.1. \tag{6}$$

During the terminal phase, the diagonal matrix $\mathcal{T}^{(n)}$ is set to zero, and the Jacobian-vector products are obtained using equation (4).

IV. Parameter Continuation

We consider continuation-based globalizations in the class of probability-one homotopy[b] methods.[11,12] The mathematical theory behind these methods is well developed, and, as mentioned in the introduction, they are useful in a wide range of applications. However, homotopy methods are not widely used in the computational aerodynamics community, so one of the aims of this paper is to encourage experimentation with these methods.

The idea behind homotopy methods is simple. Suppose we wish to solve $\boldsymbol{\mathcal{R}}(\mathbf{q}) = \mathbf{0}$. Introduce a parameter $\lambda \in [0, \lambda_{\max}]$ and a modified residual $\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda)$ such that $\boldsymbol{\mathcal{F}}(\mathbf{q}, 0) = \boldsymbol{\mathcal{R}}(\mathbf{q})$. The modified residual $\boldsymbol{\mathcal{F}}$ is called a homotopy mapping in the literature. Moreover, assume 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.

The above parameter continuation will converge to the desired root provided the following conditions are met.[11]

1. $\boldsymbol{\mathcal{F}} : \mathbb{R}^{N+1} \to \mathbb{R}^N$ is a C^2 map.
2. The Jacobian matrix $D\boldsymbol{\mathcal{F}} \in \mathbb{R}^{N\times(N+1)}$ has full rank along the solution path $\mathbf{g}$.
3. $\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda_{\max}) = \mathbf{0}$ has a unique solution.
4. The Jacobian $\partial\boldsymbol{\mathcal{R}}/\partial\mathbf{q}$ is nonsingular at the solution of $\boldsymbol{\mathcal{R}}(\mathbf{q}) = \mathbf{0}$.

Conditions 2–4 are not particularly onerous. Condition 1 is more difficult to guarantee in CFD discretizations that involve nonsmooth functions (e.g. limiters). Nevertheless, algorithms can often be applied more widely than their underlying theorems suggest, and the numerical experiments indicate parameter continuation can be applied to steady-state flows with shocks.

Modern probability-one homotopy algorithms can follow the solution curve around folds, points where the Jacobian $\partial\boldsymbol{\mathcal{F}}/\partial\mathbf{q}$ is singular: recall, condition 2 requires only that the $N \times (N+1)$ Jacobian be of full rank along the solution path. Walker[22] has shown that a certain class of path-following algorithms can be implemented in an inexact-Newton-Krylov framework such that folds can be accommodated. Despite the availability of such algorithms, we avoid homotopy maps that introduce folds into the solution curve; intuitively, following a complex curve with folds would require more steps and more computational effort. Therefore, the challenge in CFD applications is finding a homotopy map $\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda)$ with no folds in the solution curve.

Algorithm 1 describes a generic parameter continuation that is suitable for homotopy maps without folds. In the following sections, we present two globalization strategies that are special cases of this algorithm. The generic algorithm requires several inputs.

$\lambda_{\max}$:	the initial value of the continuation parameter.
$n_{\max}$:	the maximum number of Newton iterations for each λ_i.
$\tau,\ \tau_\lambda$:	the tolerances for the residual and modified residual, respectively.
η:	the forcing parameter for the solution of the linear system.
λ_i:	a function defining how the continuation parameter is decreased to zero.

All of the algorithms considered here are converged to a relative tolerance $\tau = 10^{-10}$. Suitable choices for the remaining inputs depend on the homotopy map used and will be specified for each algorithm below.

[b] In the literature, the words continuation, path-following, and homotopy are often used interchangeably.

Algorithm 1: Generic parameter continuation for an inexact-Newton solver.

```
    Input: $\lambda_{\max},\ n_{\max},\ \tau,\ \tau_\lambda,\ \eta$
    set $\lambda_0 = \lambda_{\max}$, and $\boldsymbol{\mathcal{R}}_0 = \boldsymbol{\mathcal{R}}(\mathbf{q}^{(0)})$
    for $i = 0, 1, 2, \ldots$ do
        set $\boldsymbol{\mathcal{F}}_0 = \boldsymbol{\mathcal{F}}(\mathbf{q}^{(i)}, \lambda_i)$
        for $n = 0, 1, 2, \ldots, n_{\max}$ do
            compute residual $\boldsymbol{\mathcal{R}}(\mathbf{q})$
            if $\|\boldsymbol{\mathcal{R}}\| \leq \tau \|\boldsymbol{\mathcal{R}}_0\|$ then done

            compute homotopy map $\boldsymbol{\mathcal{F}}(\mathbf{q}, \lambda_i)$
            if $\|\boldsymbol{\mathcal{F}}\| \leq \tau_\lambda \|\boldsymbol{\mathcal{F}}_0\|$ then exit loop

            inexact solve:
(*)         find $\Delta q$ such that $\|\boldsymbol{\mathcal{F}} + \mathcal{A}\Delta\mathbf{q}\| \leq \eta \|\boldsymbol{\mathcal{F}}\|$
            $\mathbf{q} := \mathbf{q} + \Delta\mathbf{q}$
        endfor
        set $\lambda_{i+1} < \lambda_i$
    endfor
```

A. Boundary-Condition Continuation

In this section we describe a boundary-condition continuation. The implementation is based on a finite-difference discretization of the Euler equations, with boundary conditions imposed weakly through penalty terms. Very similar implementations should be possible with any discretization that uses weakly imposed boundary conditions.

We use summation-by-parts (SBP) finite-difference operators on multi-block structured grids. The SBP operators provide a linearly-stable discretization of the partial differential equations; however, the discretization must be modified to account for the boundary conditions and inter-block coupling. This is accomplished by adding simultaneous approximation terms (SATs) to the nodes along the block boundaries. The SATs penalize variables associated with waves entering the block that do not match the desired boundary condition or neighbouring node value. Consequently, the residual equation for an arbitrary node i on a block boundary has the form

$$\begin{aligned}[\boldsymbol{\mathcal{R}}(\mathbf{q})]_i &= \mathcal{R}_i(\mathbf{q}) \\ &= \mathcal{R}_i^{(\mathrm{E})}(\mathbf{q}) + A_i^{+}\left(\mathbf{Q}_i - \mathbf{Q}_{\mathrm{bc}}\right), \end{aligned} \tag{7}$$

where $\mathcal{R}_i^{(\mathrm{E})}$ denotes the discrete Euler fluxes, including the numerical dissipation, and $A_i^{+}\left(\mathbf{Q}_i - \mathbf{Q}_{\mathrm{bc}}\right)$ is the SAT penalty. The flow state at node i is given by $\mathbf{Q}_i$. The vector $\mathbf{Q}_{\mathrm{bc}}$ denotes the value of the desired boundary state, for nodes on a boundary, or the value at a coincident node on a neighbouring block, for nodes on a block interface. The matrix A_i^{+} is equal to the flux Jacobian matrix for the normal flux with the eigenvalues for exiting waves set to zero. Further details on the SBP-SAT approach can be found in references 23–29, and 20.

For a node on a solid surface, the boundary state is determined by flow tangency:

$$\mathbf{Q}_{\mathrm{bc}} = \mathbf{Q}_{\mathrm{wall}} = \left(\rho, \rho u_1^{\parallel}, \rho u_2^{\parallel}, \rho u_3^{\parallel}, \rho H_\infty - p\right)^T,$$

where p is the pressure, H_∞ is the free-stream enthalpy, and

$$\left(u_1^{\parallel}, u_2^{\parallel}, u_3^{\parallel}\right)^T = \mathbf{u}^{\parallel} \equiv \mathcal{P}\mathbf{u} = (\mathcal{I}_3 - \hat{\mathbf{n}}\hat{\mathbf{n}}^T)\mathbf{u}.$$

The vector $\hat{\mathbf{n}}$ is a unit normal to the surface, and the 3×3 matrix $\mathcal{P}$ projects out the normal component of the velocity at the surface. $\mathcal{I}_3$ is the 3×3 identity matrix.

The boundary-condition continuation replaces $\mathbf{Q}_{\mathrm{wall}}$ with $\mathbf{Q}_\lambda$, a boundary state parameterized using λ. If we use the free-stream value as the initial iterate, a suitable modified boundary state should satisfy

$$\mathbf{Q}_{\lambda=1} = \mathbf{Q}_\infty, \qquad \mathbf{Q}_{\lambda=0} = \mathbf{Q}_{\mathrm{wall}},$$

where $\mathbf{Q}_\infty$ is the free-stream flow state. Other than these two limiting conditions, there appears to be considerable flexibility in choosing $\mathbf{Q}_\lambda$. In practice, $\mathbf{Q}_\lambda$ cannot be chosen arbitrarily, since some states will lead to ill-posed problems, solution-curve folds, or intermediate flows more complicated than the desired solution. For this exploratory work, we have chosen a convex combination of the free-stream state and the wall boundary state.

$$\mathbf{Q}_{\text{bc}} = \mathbf{Q}_\lambda = (1-\lambda)\mathbf{Q}_{\text{wall}} + \lambda\mathbf{Q}_\infty. \tag{8}$$

This boundary state has performed adequately in the proposed continuation method, but we suspect there are better choices.

When $\mathbf{Q}_{\text{wall}}$ is replaced with $\mathbf{Q}_\lambda$, we obtain the modified residual $\boldsymbol{\mathcal{F}}(\mathbf{q},\lambda)$. The modified residual is equivalent to taking a convex combination of two residual equations: one where the flow-tangency condition is applied at the surface and one where the free-stream condition is applied at the surface. In each nonlinear iteration, the true residual, i.e. with $\lambda = 0$, can be formed first and its L^2-norm evaluated to track convergence. Subsequently, $\boldsymbol{\mathcal{R}}$ can be modified to obtain $\boldsymbol{\mathcal{F}}$; the modification occurs only at boundary nodes on the aerodynamic surface.

The continuation parameter λ is updated according to the formula

$$\lambda_i = \frac{\lambda_{i-1}}{i} = \frac{\lambda_0}{i!}, \tag{9}$$

with $\lambda_0 = \lambda_{\max} = 0.95$. We also investigated a formula relating λ_i to the residual, but did not find this to be effective for the boundary-condition continuation.

When $\lambda > 0$, it is inefficient to solve $\boldsymbol{\mathcal{F}}(\mathbf{q},\lambda)$ to a small tolerance. This is analogous to oversolving the linear problem in inexact-Newton algorithms. Therefore, to avoid oversolving the modified system, we perform only one inexact-Newton iteration per λ value ($n_{\max} = 1$), using a fixed forcing parameter of $\eta = 0.01$. The Schur preconditioner for each nonlinear sub-problem is formed using ILU(1).

Similar to pseudo-transient continuation, the exact Jacobian is approximated until the relative residual drops below 0.1. Specifically, matrix $\mathcal{A}$ in line $(*)$ of Algorithm 1 is replaced with an approximate Jacobian that uses first-order dissipation. The approximate Jacobian regularizes the solution updates, smoothing convergence until the exact Jacobian can be used. Indeed, the effectiveness of this smoothing behaviour inspired the globalization method described next.

B. Dissipation-Based Continuation

Numerical dissipation must be added to the discrete equations, either explicitly or implicitly, to prevent oscillations and to capture shocks. Ideally, 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.[30] This practice suggests that a dissipation-based continuation strategy may be useful.

Consider 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, which is symmetric positive-definite for fixed $\mathbf{q}$ and uses only nearest-neighbours in the stencil. Note, no pressure switch or limiter is required, or even desired, for this dissipation. Hence, it should be straightforward to implement this continuation in most discretizations. Like boundary-condition continuation, the residual $\boldsymbol{\mathcal{R}}$ can be computed first to track convergence before adding the dissipation $\lambda\boldsymbol{\mathcal{D}}$.

Dissipation-based continuation is closely related to continuation in the Reynolds number,[9] i.e. beginning with a small Reynolds number and increasing it gradually to the desired value. 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 can be introduced when using Reynolds-number continuation, see, for example, Ref. 22. In addition, dissipation-based continuation can be used for hyperbolic equations like the Euler equations, where the limiting case of infinite Reynolds number may cause numerical difficulties.

When using dissipation-based continuation, each sub-problem is solved to a relative tolerance of $\tau_\lambda = 0.1$ with an upper bound on iterations of $n_{\max} = 30$. The continuation parameter is varied according to the formula

$$\lambda_i = \min\left(\frac{\lambda_{i-1}}{2}, \frac{9}{10}\mathrm{R}_{\mathrm{rel}}^2\right),$$

where the relative residual $\mathrm{R}_{\mathrm{rel}}$ is defined in equation (6). The initial value of the continuation parameter can be varied as necessary depending on the grid and flow conditions. We have found that an initial value of $\lambda_{\max} = 2$ provides a good compromise between a well-behaved initial problem and rapid convergence.

Dissipation-based continuation can use the exact Jacobian throughout convergence, unlike the pseudo-transient and boundary-condition continuation algorithms. In addition, the Schur preconditioner is constructed using ILU(2) at the first iteration of each sub-problem, rather than ILU(1), and kept frozen thereafter. The expense of forming this more accurate preconditioner is warranted, since multiple iterations are used to solve the sub-problems.

V. Results

We have chosen 6 test cases to demonstrate and compare the parameter-continuation algorithms. These cases are described below and summarized in Table 1. In the table, grid size indicates the number of nodes and M indicates the Mach number. The second-difference and fourth-difference dissipation coefficients are denoted by κ_2 and κ_4, respectively.

Case 1: A spherical geometry is solved at a Mach number of 0.5. The grid consists of 5 blocks with nearly isotropic mesh spacing. The solver may produce separated flow and fail to converge.

Case 2: The ONERA M6 wing geometry[31] is solved at a Mach number of 0.84 and an angle of attack of 3.06 degrees. The blunt trailing edge on the experimental geometry is replaced with a sharp edge. The grid has a 12 block HH-grid topology with an off-wall spacing of 0.001.

Case 3: This case considers the ONERA M6 wing geometry at a high-lift condition. The grid is the same as Case 2. The Mach number is 0.2 and the angle of attack is 20.0 degrees. Numerical dissipation encourages separation, which may lead to convergence issues.

Case 4: The DPW-W1 wing from the third drag prediction workshop[32] is modelled at a Mach number of 0.78 and an angle of attack of 3 degrees. This geometry has a blunt trailing edge, which can pose difficulties for the solution algorithm. An 8-block O-topology grid is used with a maximum stretching ratio of 1.6.

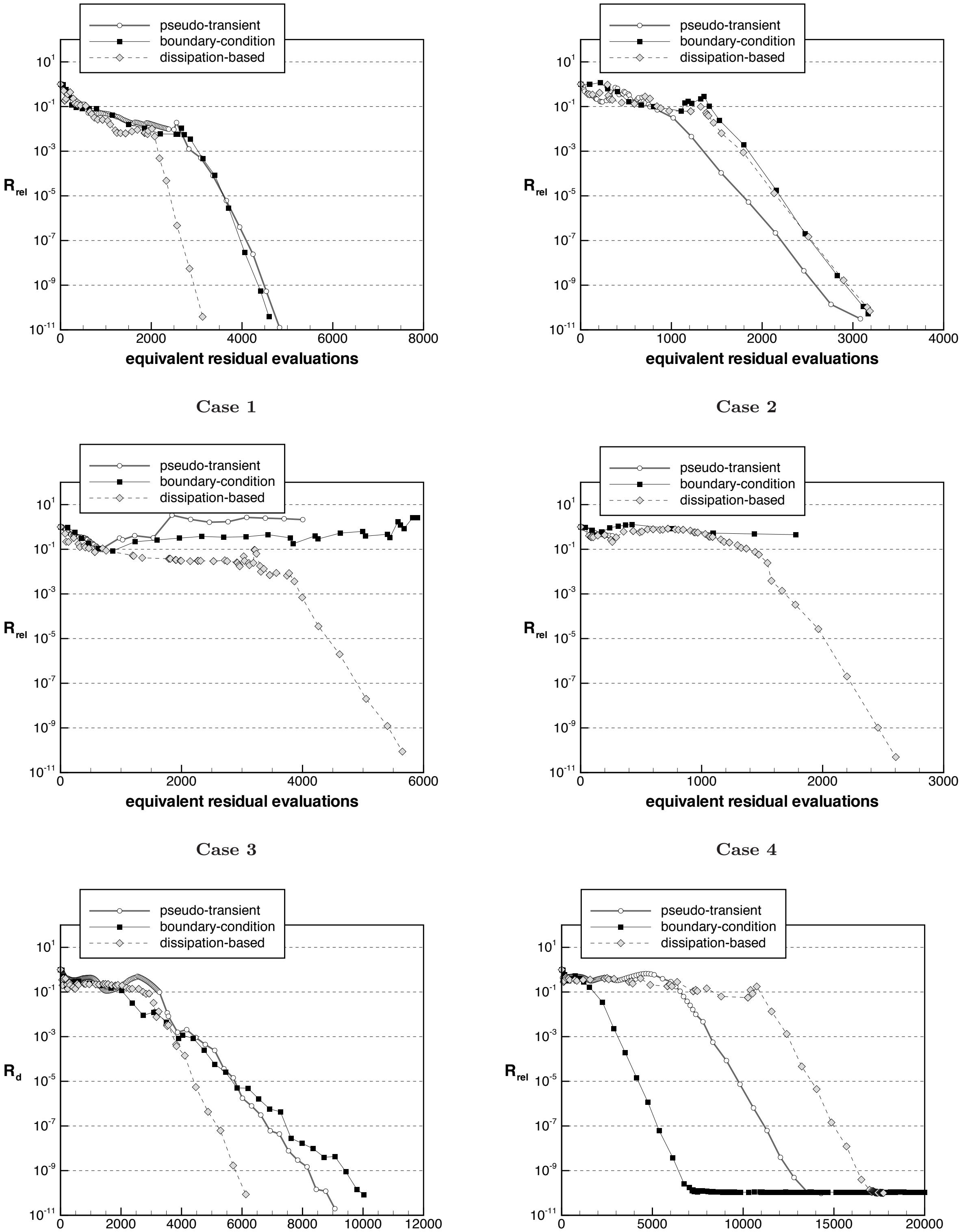
Case 5: This case consists of the DWP-W1 wing geometry at a high-lift condition. The Mach number is 0.2 and the angle of attack is 20.0 degrees. Hence, this case is similar to case 3, but with a different geometry.

Case 6: The last case considers the DLR-F6 wing-body geometry from the third drag prediction workshop.[32] A very coarse grid is obtained by removing every other grid point from the coarse grid used in the workshop. The original grid was intended for turbulent flows, so the resulting grid has a maximum stretching ratio of approximately 6, and a maximum-minimum metric Jacobian ratio of 2×10^{19}; such extreme values typically cause convergence problems. The flow is subsonic with a Mach number of 0.5 and an angle of attack of 3.0.

The cases are intended to be challenging and include geometries with blunt trailing edges (cases 1, 4–6), high lift conditions (cases 3 and 5), and shocks (cases 2 and 4). The geometries with sharp corners are particularly difficult, because the flow around such geometries is typically singular or unsteady.

Individual convergence histories, for each case and globalization method, are shown in Figure 1. The figure displays the history of the relative residual, $\mathrm{R}_{\mathrm{rel}}$, versus equivalent residual evaluations. An equivalent residual evaluation is obtained by normalized the CPU time by the time required to compute the nonlinear residual $\boldsymbol{\mathcal{R}}(\mathbf{q})$. Note that the range on the horizontal axis is different for each problem.

Figure 2 summarizes the equivalent residual evaluations required by the different globalization strategies. Each vertical bar denotes the total number of evaluations required to reduce the relative residual by ten orders of magnitude. An exception is case 6, which stalled slightly above ten orders of magnitude, so the



Case 1

Case 2

Case 3

Case 4

Case 5

Case 6

Figure 1: Convergence histories for the 6 test cases

Table 1: Cases used to test the globalization methods.

case	geometry	grid size	$\mathbf{M}$	$\mathbf{C_L}$	$\boldsymbol{\kappa}_2$	$\boldsymbol{\kappa}_4$
1	sphere	4.35×10^5	0.50	0.000	0.00	0.03
2	ONERA M6	1.16×10^6	0.84	0.278	0.50	0.03
3	ONERA M6	1.16×10^6	0.20	1.247	0.00	0.03
4	DPW-W1	6.55×10^5	0.78	0.937	0.50	0.04
5	DPW-W1	6.55×10^5	0.20	2.02	0.00	0.04
6	DLR F6	6.76×10^5	0.50	0.306	0.05	0.03

bars indicate the evaluations necessary to reach a reduction of nine orders of magnitude. If a bar is absent, then the method failed to converge for that case.

The dissipation-based continuation stands out as the most robust globalization. Indeed, we have not yet found a case where pseudo-transient or boundary-condition continuation converged and dissipation-based continuation did not. Moreover, we suspect that when dissipation-based continuation fails, the Jacobian matrix at $\lambda = 0$ is highly ill-conditioned or perhaps even singular (i.e. condition 4 is not satisfied). Dissipation-based continuation also compares well in terms of CPU time, except for case six (DLR-F6 geometry) where this strategy performs poorly. For this subsonic case, a small amount of second-difference dissipation ($\kappa_2 = 0.05$) was necessary to obtain a converged solution for all three globalization methods. The presence of second-difference dissipation seems to delay convergence of this continuation strategy.

Boundary-condition continuation performs well on four of the six cases. This method establishes the general shape of the pressure field within the first few nonlinear iterations, even when the velocity at the wall is far from tangential. It is not clear why the method sometimes fails, but solutions at intermediate values of λ suggest that the boundary value $\mathbf{Q}_\lambda$ may induce complex flow features downwind of the geometry, which then lead to ill-conditioned linear problems.

Pseudo-transient continuation performs well in terms of CPU time on the more benign cases; however, it has difficulty solving the cases that involve blunt trailing edges and/or high angles of attack. Pseudo-transient continuation is susceptible to producing separated flow during the early iterations with these cases. Once the flow is separated, the globalization seems to have difficulty recovering, and the iterates remain outside the basin of attraction for Newton's method.

VI. Summary and Conclusions

We have described two parameter-based continuations that can be used to globalize Newton's method; one using boundary-condition continuation, the other using dissipation-based continuation. In both methods, a parameter is used to create a sequence of nonlinear problems, where the solution of one problem provides the initial iterate for the next. We have implemented these globalization strategies in a finite-difference discretization, but the method is general enough to be applied to other discretizations.

One difficulty that we foresee with boundary-condition continuation is the possibility that the modified equations fail to have a solution, or that folds are introduced. Carefully constructed boundary conditions are needed to guarantee that these issues are avoided; however, since fully converged intermediate solutions are not necessary, this point may be overstated.

Boundary-condition and dissipation-based continuation have shown considerable promise as globalization strategies for inexact-Newton methods. Dissipation-based continuation, in particular, appears to be especially robust and efficient. However, before these methods can supersede pseudo-transient continuation as the method of choice, they must be applied to a broader spectrum of problems. Therefore, future work will focus on applying these globalizations to turbulent flows and high-order discretizations.

Acknowledgements

The authors gratefully acknowledge financial assistance from Bombardier Aerospace, the Natural Sciences and Engineering Research Council (NSERC), the Canada Research Chairs program, Mathematics of

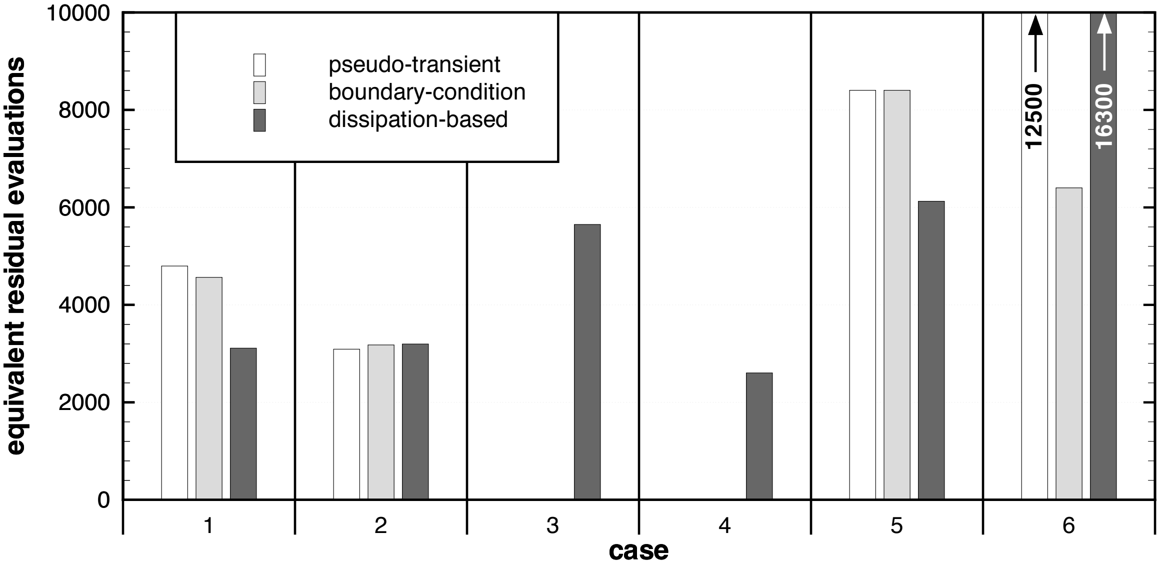


Figure 2: Equivalent residual evaluations (normalized CPU time) required by each of the globalizations to converge the six test cases 10 orders of magnitude.

Information Technology and Complex systems (MITACS), 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] Rumpfkeil, M. P. and Zingg, D. W., "The optimal control of unsteady flows with a discrete adjoint method," *Optimization and Engineering*, 2008, pp. 18.

[8] 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.

[9] 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.

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

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

[13] Bailey, H. E. and Beam, R. M., "Newton's method applied to finite-difference approximations for the steady-state compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 93, No. 1, 1991, pp. 108–127.

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

[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] Saad, Y., "A flexible inner-outer preconditioned GMRES algorithm," *SIAM Journal on Scientific and Statistical Computing*, Vol. 14, No. 2, 1993, pp. 461–469.

[19] Saad, Y. and Sosonkina, M., "Distributed Schur complement techniques for general sparse linear systems," *SIAM Journal of Scientific Computing*, Vol. 21, No. 4, 1999, pp. 1337–1357.

[20] 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.

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

[22] 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.

[23] 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.

[24] Strand, B., "Summation by parts for finite difference approximations for d/dx," *Journal of Computational Physics*, Vol. 110, No. 1, 1994, pp. 47–67.

[25] 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.

[26] Carpenter, M. H., Nordström, J., and Gottlieb, D., "A stable and conservative interface treatment of arbitrary spatial accuracy," *Journal of Computational Physics*, Vol. 148, No. 2, 1999, pp. 341–365.

[27] Nordström, J. and Carpenter, M. H., "High-order finite difference methods, multidimensional linear problems, and curvilinear coordinates," *Journal of Computational Physics*, Vol. 173, No. 1, 2001, pp. 149–174.

[28] Mattsson, K., Svärd, M., and Nordström, J., "Stable and accurate artificial dissipation," *Journal of Scientific Computing*, Vol. 21, No. 1, 2004, pp. 57–79.

[29] Svärd, M., *Stable High-Order Finite Difference Methods for Aerodynamics*, Ph.D. thesis, Uppsala University, Uppsala, Sweden, 2004.

[30] 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.

[31] Schmitt, V. and Charpin, F., "Pressure distributions on the ONERA-M6-wing at transonic mach numbers," Tech. rep., Office National d'Etudes et Recherches Aerospatiales, 92320, Chatillon, France, 1979.

[32] "Third AIAA drag prediction workshop," San Francisco, CA, 2006, http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3/workshop3.html.