AIAA-2013-2370

21st AIAA Computational Fluid Dynamics Conference, 24-27 June 2013, San Diego, USA

Advances in Homotopy Continuation Methods in Computational Fluid Dynamics

David Brown* and David W. Zingg†

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

Progress in the development of a homotopy continuation method to globalize the spatially-discretized steady flow equations in computational fluid dynamics (CFD) is presented. The algorithm contains many new features developed specifically for application to the large sparse highly parallelized systems encountered in modern CFD. A second-difference dissipation operator common in finite-difference CFD algorithms is used as the homotopy function. Performance is investigated for compressible subsonic and transonic flow over an ONERA M6 wing geometry. Performance of the homotopy algorithm is demonstrated to be comparable to or better than the well-established pseudo-transient continuation method for all inviscid and laminar cases investigated. The algorithm also shows comparable performance to pseudo-transient continuation for turbulent flows (using Reynolds-averaging).

Abbreviations

RANS	Reynolds-averaged Navier-Stokes
AMVs	Approximate matrix-vector products
FDMVs	Finite-difference matrix-vector products
PTC	Pseudo-transient continuation
CHC	Convex homotopy continuation
INP	Inexact Newton phase

Notation

$\mathbf{Q}^{(n)}$	The vector $\mathbf{Q} \in \mathbb{R}^N$ at the n-th Newton iterate
$\mathcal{D}^{(2)}$, $\mathcal{D}^{(4)}$	Second- and fourth-difference dissipation operators (an exception to the above notation)
$\mathbf{Q}_k$	$\mathbf{Q}$ at the k-th homotopy sub-problem
$\mathbf{Q}[i]$	The i-th component of the vector $\mathbf{Q}$

I. Introduction

Computational fluid dynamics (CFD) involves the numerical solution to systems of partial differential equations that model fluid flow. Typically, CFD algorithms approximate the continuous physical spatial domain with a finite number of grid points. The partial differential equations modelling the flow physics are then discretized over this grid. In doing so, the system of nonlinear partial differential equations is approximated by a system of ordinary time differential equations. Specializing to the case of steady flows, the system further reduces to a system of nonlinear algebraic equations. We will refer to this algebraic system as $\mathcal{R}(\mathbf{Q}) = \mathbf{0}$, where $\mathbf{Q} \in \mathbb{R}^N$ are the dependent variables at the grid points and $\mathcal{R} : \mathbb{R}^N \to \mathbb{R}^N$ is the flow residual.

*PhD Candidate, AIAA Student Member

†Professor and Director, Tier 1 Canada Research Chair in Computational Aerodynamics and Environmentally Friendly Aircraft Design, J. Armand Bombardier Foundation Chair in Aerospace Flight, Associate Fellow AIAA

Newton's method is a popular fixed-point iterative method[a] due to the fact that its convergence rate is usually quadratic. However, it is well known that Newton's method will only converge if the initial guess is sufficiently close to the solution. To obtain a suitable initial iterate for Newton's method, we often employ a slower but more reliable method initially and switch to Newton's method when we believe that a suitable initial iterate has been obtained.

Obtaining a suitable starting point for Newton's method is known as *globalization*, and the process used to obtain this point is referred to as *continuation*. Pseudo-Transient Continuation (PTC) methods imitate some form of time marching and are common and well-established globalization strategies. Though PTC gives less than quadratic convergence, convergence can still be super-linear.[1] The relatively high convergence rate and robustness along with the simplicity of the method make PTC a popular globalization technique. Other techniques that have been employed successfully in CFD include line search, trust region, and grid sequencing methods.[2]

We introduce the concept of homotopy continuation by presenting two systems of equations: the first being the system that we ultimately wish to solve, represented by $\mathcal{R}(\mathbf{Q}) = \mathbf{0}$, and the second, some other system that we can solve much more easily, given by $\mathcal{G}(\mathbf{Q}) = \mathbf{0}$. We will refer to $\mathcal{G}(\mathbf{Q})$ as the *homotopy function* and $\mathcal{R}(\mathbf{Q})$ will, in this paper, correspond to the discretized flow equations. Under certain conditions, $\mathcal{R}^{-1}(\mathbf{0})$ and $\mathcal{G}^{-1}(\mathbf{0})$ are homotopic; that is, they are connected by a continuous deformation. This continuous deformation. or *homotopy*, defines a curve (1-manifold) in $\mathbb{R}^n$. By solving $\mathcal{G}(\mathbf{Q}) = \mathbf{0}$ and tracing this curve, we can globalize the system $\mathcal{R}(\mathbf{Q}) = \mathbf{0}$.

As a further introduction to homotopy continuation methods, we quote an excerpt from Ref. 3: "Their use can be traced back at least to such venerated works as those of Poincaré[4] (1881), Klein[5] (1882), and Bernstein[6] (1910). Leray and Schauder[7] (1934) refined the tool and presented it as a global result in topology viz. the homotopy invariance of degree." Lahaye[8] (1934) is perhaps the first to use homotopy methods to solve nonlinear equations, and later, systems of equations[9] (1948). Ortega and Rheinboldt[10] (1970) provide a framework for the implementation of homotopy methods as numerical algorithms. Contributions and refinements to these algorithms over the next two decades are presented in Allgower and Georg[3] (1990). We refer to the survey article of Allgower and Georg[11] for an extensive literature review and bibliography pertaining to homotopy methods prior to 1992.

Several researchers have applied homotopy continuation to CFD problems. Carey and Krishnan[12] investigated using the Reynolds number as the continuation parameter for a simple incompressible viscous lid-driven cavity problem. There has also been some research interest in using the critical point detection properties of homotopy methods in CFD; for example, Refs. 13, 14, 15, 16. Riley and Winters[13] studied the bifurcation points that arise from rotating a problem with a unique solution into a problem with multiple solutions, treating the angle of rotation as the continuation parameter. More recently, Wales et al.[17] have applied continuation to two-dimensional turbulent flow around a NACA 0012 airfoil. By treating the angle of attack as the continuation parameter, they are able to acquire flow solutions for dynamically unstable flow at high Reynolds number and high angle of attack unobtainable with PTC.

In contrast to the literature, our interest is in applying homotopy continuation as a general and cost-competitive globalization strategy. Hicken and Zingg[18] applied homotopy continuation in two ways: by varying the boundary conditions and using the second-order artificial dissipation operator of Pulliam.[19] The dissipation-based continuation algorithm was investigated further by Hicken et al.[20] for inviscid, viscous, and turbulent (RANS) cases. The algorithm was found to be competitive with PTC in all three applications, particularly for the inviscid case. Motivated by these results, we have decided to formalize the method and to improve the algorithm towards its maximum potential as a general-purpose globalization strategy. In this paper, we present our current contributions to the algorithm, including the development of a versatile homotopy toolbox including predictor, corrector, and steplength adaptation.

Currently, we employ a form of homotopy continuation known as *convex homotopy continuation*. The algorithm is applied to our group's in-house Newton-Krylov flow solver.[21, 22] This is a parallel implicit finite-difference[23] multi-block three-dimensional external aerodynamic flow solver, applicable to the compressible Euler, Navier-Stokes,[24] and Reynolds-averaged Navier-Stokes[25] (RANS) equations using the Spalart and Allmaras[26] one-equation turbulence model. Summation By Parts (SBP) operators are used with Simultaneous Approximation Terms (SATs)[27, 28, 29, 30] to enforce boundary conditions and couple the equations across domain interfaces. We use an approximate Schur[31] preconditioner with block ILU(p)[32] factorization and the linear systems are solved with FGMRES.[33, 34]

[a] Fixed point methods are iterative methods of the form $\mathbf{Q}^{(n+1)} \leftarrow f : \mathbf{Q}^{(n)}$

II. Inexact Newton Method

Newton's method is a fixed-point iterative method given by:

$$\mathcal{A}^{(n)}\,\Delta\mathbf{Q}^{(n)} = -\mathcal{R}\left(\mathbf{Q}^{(n)}\right), \tag{1}$$

$$\Delta\mathbf{Q}^{(n)} = \mathbf{Q}^{(n+1)} - \mathbf{Q}^{(n)},\ \mathcal{A}^{(n)} = \nabla\mathcal{R}\left(\mathbf{Q}^{(n)}\right),$$

where $\mathcal{A}^{(n)} : \mathbb{R}^N \to \mathbb{R}^N$ is the Jacobian of $\mathcal{R}(\mathbf{Q})$ and has a square matrix form. Since the linear system is being solved inexactly using a Krylov method to some relative tolerance $\eta_n \in \mathbb{R}$, the actual Newton step is taken inexactly:

$$\left\|\mathcal{R}\left(\mathbf{Q}^{(n)}\right) + \mathcal{A}^{(n)}\,\Delta\mathbf{Q}^{(n)}\right\| \leq \eta_n \left\|\mathcal{R}\left(\mathbf{Q}^{(n)}\right)\right\|. \tag{2}$$

III. Jacobian-Free Matrix-Vector Products

It is not necessary calculate and store $\mathcal{A}^{(n)}$ since at no point in the algorithm is an explicit expression for $\mathcal{A}^{(n)}$ required, and this matrix can be very expensive in terms of data storage. The Krylov solver does, however, require an approximation to the matrix-vector product $\mathcal{A}^{(n)}\mathbf{v}$, $\mathbf{v} \in \mathbb{R}^N$.

This matrix-vector product can be approximated either using finite-differences (FDMVs):

$$\mathcal{A}^{(n)}\mathbf{v} \approx \frac{\mathcal{R}\left(\mathbf{Q}^{(n)} + \epsilon\mathbf{v}\right) - \mathcal{R}\left(\mathbf{Q}^{(n)}\right)}{\epsilon}, \tag{3}$$

$$\epsilon \in \mathbb{R},$$

or by recycling the smaller-stencil approximate Jacobian used to form the ILU preconditioner (AMVs).

FDMVs require an additional residual calculation per matrix-vector product and are more expensive than AMVs, but generally result in greater accuracy for appropriate choice of ϵ. Since the approximate Jacobian used for the AMVs is also used for the ILU factorization, the linear system when solved with AMVs will generally be better conditioned. Thus, the AMVs will typically result in cheaper matrix-vector products and fewer matrix-vector products for each call to the Krylov solver than FDMVs, but more outer Newton iterations, which will require more ILU factorizations which can be expensive. Thus, we cannot always have a priori knowledge which of the two matrix-vector products will be more cost-effective during the globalization phase. During the inexact Newton phase, FDMVs are the more efficient option.

IV. Pseudo-Transient Continuation (PTC)

PTC is another fixed-point method and can be interpreted as an approximation to a time-marching method or as a relaxation on Newton's method. The update formula is given by:

$$\left(\mathcal{T}^{(n)} + \mathcal{A}^{(n)}\right)\Delta\mathbf{Q}^{(n)} = -\mathcal{R}\left(\mathbf{Q}^{(n)}\right), \tag{4}$$

where $\mathcal{T}^{(n)} = \frac{1}{\Delta t}\mathcal{I}$, and $\mathcal{I}$ is the identity matrix. If Δt is chosen to to be sufficiently small, a time-accurate implicit-Euler time marching scheme with local time linearization[23] is returned.

Since we are not interested in time-accuracy, Δt can be permitted to take large values and can even be permitted to vary spatially. In this study, we use the time step given by:

$$\Delta t_i^{(n)} = T_i a\,(b)^{m\left\lfloor\frac{n}{m}\right\rfloor},\ T_i = \frac{1}{1 + J_i^{\frac{1}{3}}}, \tag{5}$$

where J is the geometric Jacobian resulting from the coordinate transformation on the mesh, i is the grid point index, and $\lfloor\cdot\rfloor$ is the floor operator[b]. Typical values of a and b will be discussed in the results section

[b] $\lfloor x\rfloor = y$, where y is the largest integer less than or equal to x.

as they can take different values in different cases.

PTC is terminated and the inexact-Newton phase (INP) initiated when the relative residual

$$\mathcal{R}_{\text{rel}}^{(n)} \equiv \frac{\left\|\mathcal{R}\left(\mathbf{Q}^{(n)}\right)\right\|}{\left\|\mathcal{R}\left(\mathbf{Q}^{(0)}\right)\right\|} \tag{6}$$

is reduced below some user-specified tolerance μ_{rel}. Typical values of μ_{rel} are $\mu_{\text{rel}} \approx 10^{-1}$ for the Euler equations, $\mu_{\text{rel}} \approx 10^{-2}$ to 10^{-3} for the Navier-Stokes equations (laminar flow), and $\mu_{\text{rel}} \approx 10^{-3}$ to 10^{-5} for the RANS equations.

V. Convex Homotopy Continuation (CHC)

In homotopy continuation methods, we deform $\mathcal{G}^{-1}(\mathbf{0})$ into $\mathcal{R}^{-1}(\mathbf{0})$ by varying some continuation parameter. In this paper, we focus on a class of continuation methods known as *convex homotopy continuation*.[3] This process requires solutions to the sequence of equations defined by

$$\mathcal{H}(\mathbf{Q}, \lambda_k) = (1 - \lambda_k)\,\mathcal{R}(\mathbf{Q}) + \lambda_k \mathcal{G}(\mathbf{Q}) = \mathbf{0}, \tag{7}$$

$$k \in [0, p],\ \lambda_k \in \mathbb{R},\ \lambda_0 = 1,\ \lambda_p = 0,\ \lambda_{k+1} < \lambda_k,$$
$$\mathcal{H} : \mathbb{R}^{N+1} \to \mathbb{R}^N,\ \mathcal{G} : \mathbb{R}^N \to \mathbb{R}^N,\ \mathcal{R} : \mathbb{R}^N \to \mathbb{R}^N.$$

Solving this sequence of problems is referred to as *traversing*. The sequence λ_k is not generally predetermined, but established during traversing using steplength adaptation.

V.A. Constructing the Homotopy Function

In theory, the homotopy function $\mathcal{G}(\mathbf{q})$ can be any operator of our choosing. However, for the numerical algorithm to be of any practical use, the following requirements should be met:

1. The operator is well-conditioned, and improves the conditioning of the flow residual.
2. The solution to $\mathcal{G}(\mathbf{Q}) = \mathbf{0}$ is known or easily obtainable.
3. The solution to $\mathcal{G}(\mathbf{Q}) = \mathbf{0}$ is unique.
4. The homotopy connecting $\mathcal{G}^{-1}(\mathbf{0})$ and $\mathcal{R}^{-1}(\mathbf{0})$ should exhibit modest curvature. That is, the direction of the vector tangent to the curve should change relatively slowly with respect to λ.

If we have existence and uniqueness for solutions of both $\mathcal{G}(\mathbf{Q}) = \mathbf{0}$ and $\mathcal{R}(\mathbf{Q}) = \mathbf{0}$, then local existence and uniqueness of solutions for (7) at intermediate λ_k is guaranteed by the Implicit Function Theorem,[35] contingent on regularity ($\nabla_{\mathbf{Q}}\mathcal{H}(\mathbf{Q}_k, \lambda_k)$ non-singular for all $\lambda \in [0, 1]$). In this case, the homotopy deformation is invertible and is called a *homeomorphism*, and $\mathcal{G}$ and $\mathcal{R}$ are said to be *homeomorphic*. This is important to guarantee that tracing the implicitly-defined curve beginning at a solution to $\mathcal{G}(\mathbf{Q}) = \mathbf{0}$ will indeed lead to the (presumably unique) solution to $\mathcal{R}^{-1}(\mathbf{0})$, and also gives some assurance that the curve will not be simply bifurcated, which would result in poor conditioning and potentially other curve-tracing difficulties. Since $\mathcal{R}(\mathbf{Q}) = \mathbf{0}$ has a unique solution by assumption, it remains to construct $\mathcal{G}$ to also be invertible. This will have the added benefit of improving the conditioning of the linear system near $\lambda = 1$. This reasoning leads to the inclusion of the third item in the list of requirements.

The fourth item listed above is, as far as we know, impossible to ascertain without experimentation, but greatly impacts our ability to trace the curve numerically and is thus very important. All four of these items are essential in selecting our operator.

V.B. The Dissipation Operator as Homotopy Function

For convection-dominated flows, it is important to introduce some form of numerical dissipation to stabilize the numerical algorithm, either through the discretization scheme itself or as an additional term, known as *artificial dissipation*.[19,23] Since we use an anti-symmetric centered-difference scheme for all first derivatives, we need to introduce a symmetric dissipation term for stability.

For second-order accurate finite differences, as are used in the current study, the dissipation operator typically employed is the fourth-difference dissipation operator $\mathcal{D}^{(4)} : \mathbb{R}^N \to \mathbb{R}^N$, which has third-order accuracy. The second-difference dissipation operator $\mathcal{D}^{(2)} : \mathbb{R}^N \to \mathbb{R}^N$ is only first-order accurate and introduces too much error into the solution for a second-order accurate discretization. However, many finite-difference algorithms employ the extra dissipation for shock-capturing; it is often activated only in the vicinity of shocks by means of a pressure switch which locally reduces the coefficient of $\mathcal{D}^{(4)}\mathbf{Q}$ at nodes where $\mathcal{D}^{(2)}\mathbf{Q}$ is large.

Though we do not apply the operator $\mathcal{D}^{(2)}$ at all for subsonic flows, and only in a small part of the physical domain for transonic flows, this operator has a highly stabilizing effect on convection-dominated flows. Hicken et al.[20] have used this operator successfully as the homotopy function in their homotopy continuation algorithm, and we use it in this study as well.

For simplicity, we give only the version of the operator that would correspond to a scalar one-dimensional problem. Extension to higher dimensional systems is straightforward, and we refer to Pulliam[19] for a two-dimensional version. Explicitly, the operator is:

$$\mathcal{D}^{(2)} = \bar{\Delta}^T \Lambda \bar{\Delta}, \tag{8}$$

$$\bar{\Delta} : \mathbb{R}^N \to \mathbb{R}^{N-1}, \ \Lambda : \mathbb{R}^N \to \mathbb{R}^N,$$

$$\bar{\Delta} = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}, \ \Lambda = \begin{bmatrix} d_{1\frac{1}{2}} & & & \\ & d_{2\frac{1}{2}} & & \\ & & \ddots & \\ & & & d_{N-\frac{1}{2}} \end{bmatrix},$$

$$d_{i+\frac{1}{2}} = \frac{1}{2}\left(d_i + d_{i+1}\right), \ d_i = \frac{1}{\Delta x_i}\left(|u_i| + a_i\right),$$

where $\Delta x_i \in \mathbb{R}$ is the local grid spacing, $u_i \in \mathbb{R}$ is the local fluid velocity, and $a_i \in \mathbb{R}$ is the local speed of sound.

The fact that $\bar{\Delta} : \mathbb{R}^N \to \mathbb{R}^{N-1}$ is sufficient to identify that the rank of $\mathcal{D}^{(2)} : \mathbb{R}^N \to \mathbb{R}^N$ is at most $N-1$. Therefore $\mathcal{D}^{(2)}$ is singular and hence fails to meet the third item listed in the requirements for $\mathcal{G}$. This can be mended by applying pseudo boundary conditions to $\mathcal{D}^{(2)}$.

Pseudo boundary conditions for the dissipation operator are formulated using the SAT approach to be consistent with the SAT treatment of the flow equations. In this context, we assume the following form for the homotopy function including boundary conditions:

$$\mathcal{G}\left(\mathbf{Q}\right) = \mathcal{D}^{(2)}\mathbf{Q} + \Sigma\left(\mathbf{Q}\right), \tag{9}$$

$$\Sigma\left(\mathbf{Q}\right) = \mathrm{diag}\left(\sigma_L\left(\mathbf{Q}\left[1\right] - Q_L\right), 0, \ldots, 0, \sigma_R\left(\mathbf{Q}\left[N\right] - Q_R\right)\right), \ \sigma_L, \sigma_R, Q_L, Q_R \in \mathbb{R}.$$

Above, $\Sigma\left(\mathbf{Q}\right)$ is the operator applying the SATs and $\mathbf{Q}\left[i\right]$ denotes the i-th component of $\mathbf{Q}$. At a domain boundary, Q_L and Q_R are boundary conditions. At block interfaces, they are the flow values at the same point in physical space but corresponding to an adjacent block. It can be shown (see Appendix A) that sufficient conditions for local invertibility of $\mathcal{G}$ are $\sigma_R \geq 0, \ \sigma_L \geq 0, \ \sigma_L + \sigma_R > 0$.

By symmetry, and comparison with the well-known diffusion operator, we infer that the proper conditions for well-posedness are given by $\sigma_L = d_1$, $\sigma_R = d_N$. (For the three-dimensional case: $\sigma_L = d_1 I_5$, $\sigma_R = d_N I_5$, where I_5 is the 5×5 identity operator.)

Since $\mathcal{D}^{(2)}$ is non-physical, the boundary conditions themselves can take any value of our choosing. One benefit to using freestream conditions globally as boundary conditions for $\mathcal{D}^{(2)}$ is that the vector that is set to freestream conditions everywhere is a solution to $\mathcal{G}\left(\mathbf{Q}\right) = \mathbf{0}$. Some numerical experimentation suggests that this choice results in relatively modest curvature as well, and thus satisfies the fourth item on our list of requirements. All results presented in this study use this form of boundary condition, though we also note that a configuration that imitates the boundary conditions of the flow residual has also worked fairly well in practice, and $\mathcal{G}\left(\mathbf{Q}\right) = \mathbf{0}$ can be easily solved with Newton's method at low cost.

V.C. The Predictor-Corrector Algorithm

The efficiency of the algorithm depends on our ability to trace the curve connecting $\mathcal{G}^{-1}(\mathbf{0})$ and $\mathcal{R}^{-1}(\mathbf{0})$. To do so, we break the traversing process into two phases, which are performed iteratively:

- The corrector phase: The objective is to solve the nonlinear sub-problem at λ_k. The sub-problems are solved inexactly using the inexact Newton method. We iterate until $\frac{\left\|\mathcal{H}\left(\mathbf{Q}^{(n)},\lambda_k\right)\right\|}{\left\|\mathcal{H}\left(\mathbf{Q}^{(0)},\lambda_k\right)\right\|} < \mu_k$, where μ_k is user-defined. Typically, we use $\mu_k \in [0.1, 0.5]$. We denote by $p_k \in \mathbb{Z}$ the number of iterations required to converge the k-th sub-problem.

- The predictor phase: The objective is to obtain a suitable starting guess for the $k+1$st sub-problem using the estimated solution at the k-th sub-problem, a step direction, and a distance (steplength) to travel in that direction. We use a first-order explicit Euler predictor. At the k-th step:

$$\mathbf{u}_{k+1}^{(1)} = \mathbf{u}_k^{(p_k)} + h_k \mathbf{d}_k, \tag{10}$$

$$h_k \in \mathbb{R},\ \mathbf{u}_k \in \mathbb{R}^{N+1},\ \mathbf{d}_k \in \mathbb{R}^{N+1},$$

  where $\mathbf{u}_k = [\mathbf{Q}_k; \lambda_k]$, $h_k > 0$ is the steplength, and $\mathbf{d}_k$ is the step direction. Note that equation (10) updates both λ and $\mathbf{Q}$. We have found that performance is better with an Euler predictor than with higher-order predictors (such as the second-order Adams-Bashforth method) due to relatively large step sizes. Choice of $\mathbf{d}_k$ and h_k will be the topic of the following two subsections.

A verbal description of the algorithm is as follows:

1. Initialize the solution field at $\lambda_0 = 1$ and solve the nonlinear sub-problem $\mathcal{G}(\mathbf{Q}) = \mathbf{0}$ if necessary;
2. Apply a predictor step;
3. Apply a corrector step;
4. Repeat steps 2 and 3 until $\lambda = 0$ is attained;
5. Solve $\mathcal{R}(\mathbf{Q}) = \mathbf{0}$ using the inexact Newton method.

V.D. The Predictor Step Direction

Since we are using the inexact Newton method exclusively in the sub-problems, the choice of predictor is very important for both speed and robustness, as improving the starting guess will not only allow us to solve the sub-problems in fewer iterations in the corrector phase but will improve the success rate of Newton's method for the sub-problems as well. The most accurate step direction that we can construct with the limited information we have of the curve is to take $\mathbf{d}$ as an approximation to the vector tangent to the curve. However, this can be a relatively expensive calculation. As with most numerical algorithms, we will need to compromise between speed and accuracy of the tangent calculation, which will affect the efficiency of the predictor-corrector algorithm.

We consider several possible choices for the step direction $\mathbf{d}$:

- *Embedding step:* $\mathbf{d} = (0, 0, \ldots, 0, -1)$, which results in $\lambda_{k+1} \leftarrow \lambda_k - h_k$, and $\mathbf{Q}_{k+1}^{(1)} \leftarrow \mathbf{Q}_k^{(p_k)}$. Simply put, the solution to $\mathcal{H}(\mathbf{Q}, \lambda_k) = \mathbf{0}$ is used as the initial guess at λ_{k+1}.

- *Secant method:* This is an approximation to the tangent vector using first-order backwards finite-differencing:

$$\mathbf{d}_k = \frac{1}{h_k}\left(\mathbf{u}_k^{p_k} - \mathbf{u}_{k-1}^{p_{k-1}}\right). \tag{11}$$

- *Algebraic tangent calculation:* This is a special case of the algebraic method presented by Rheinboldt[36] and is the most efficient exact tangent calculation for large sparse systems that we are aware of. It is presented here without explanation:

$$\mathbf{d} = \frac{\tau}{\|\tau\|}, \qquad \text{where} \quad \tau = \begin{pmatrix} \mathbf{z} \\ -1 \end{pmatrix}, \tag{12}$$

and $\mathbf{z} \in \mathcal{R}^N$ is determined by solving the linear system of equations:

$$\nabla_{\mathbf{Q}} \mathcal{H}(\mathbf{Q}, \lambda)\, \mathbf{z} = \mathcal{G}(\mathbf{Q}) - \mathcal{R}(\mathbf{Q}). \tag{13}$$

The derivation is provided in Appendix B.

The embedding step has the advantage that it is very simple and essentially costless, but it is the least accurate predictor. The secant method is also essentially costless relative to the cost of the corrector step. Since it is based on finite differences, the secant method will more accurately approximate the tangent when the steplength is small. Though the steplength is generally not small enough to approximate the tangent very accurately with this method, we have found that it still does generally provide some benefit over the basic embedding step and we have included an option to use it instead of the algebraic method if the steplength is sufficiently small.

The algebraic tangent calculation is an exact calculation of the tangent, the only sources of error being relaxation in the linear solver and rounding error, the former being the more significant of the two. Though it is the only predictor method requiring significant CPU time, the cost usually pays for itself due to the increased quality of the starting point of the following corrector phase. Since improving the starting point of the corrector phase results in improved robustness of the inexact Newton solver, the algebraic tangent is typically our method of choice.

V.E. Steplength Adaptation

Steplength adaptation algorithms are used to automatically adjust the steplength during traversing. The steplength can either be reduced to improve robustness or increased to increase the rate of traversing. These methods do not provide an explicit expression for the steplength. Rather, we make adjustments to the current steplength value based on collected data.

In conformity with the notation in Ref. 3, we will denote the output of the steplength method as the factor f. Once we determine f, we check that $f_{\min} \leq f \leq f_{\max}$ and adjust it if necessary. We then calculate $h_k = \frac{h_{k-1}}{f}$ and then check $h_{\min} \leq h \leq h_{\max}$ and adjust it if necessary. These upper and lower bounds on f and h are user-supplied. However, since h is taken in the $[\mathbf{Q}; \lambda]$ direction, values of $h_{\max}$ and $h_{\min}$ are not intuitive and so we generally enforce $\Delta\lambda_{\min} \leq \Delta\lambda \leq \Delta\lambda_{\max}$ instead.

The method of *asymptotic expansions*, first presented by Georg,[37] is a relatively simple method based on some of the physical properties of consecutive predictor and corrector points. We employ a slightly modified version of the algorithm presented in Allgower and Georg[3] using two criteria: the distance from the predictor point to the corrector point δ and the angle ϕ between consecutive tangent vectors. The factor f is calculated: $f = \max\left\{\sqrt{\frac{\delta}{\tilde{\delta}}}, \frac{\phi}{\tilde{\phi}}\right\}$. One difficulty that we observe with this algorithm is that suitable values for $\tilde{\delta}$ and $\tilde{\phi}$ are not intuitively obvious. We will address this in Section V.G.

We have also given some consideration to Newton-Kantorovich methods, which adapt the steplength based on error models for Newton's method; see, for example, Ref. 38. However, the relaxation that we apply to both the linear system and nonlinear sub-problems reduces the effectiveness of these error models. Furthermore, the method requires a target number of iterations for convergence as user input, which can be difficult to guess, resulting in consistent under- or over-relaxation of the steplength. For these reasons, we have not found this method to be very effective for our applications.

V.F. μ-Scaling

We have developed μ-scaling as a means to balance the relative scaling of $\mathcal{G}$ and $\mathcal{R}$. From a purely mathematical perspective, this relative scaling is irrelevant. From a numerical perspective, this has an effect on curvature and affects the performance of the predictor-corrector tools. We consider the following modified version to the CHC deformation (7):

$$\mathcal{H}(\mathbf{Q}, \lambda_k) = (1 - \lambda_k)\, \mathcal{R}(\mathbf{Q}) + \lambda_k \mu \mathcal{G}(\mathbf{Q}) = \mathbf{0},\ k \in [0, p],\ \lambda_k \in \mathbb{R},\ \lambda_0 = 1,\ \lambda_p = 0,\ \lambda_{k+1} < \lambda_k; \tag{14}$$

$$\mathcal{H} : \mathbb{R}^{N+1} \to \mathbb{R}^N,\ \mathcal{G} : \mathbb{R}^N \to \mathbb{R}^N,\ \mathcal{R} : \mathbb{R}^N \to \mathbb{R}^N,\ \mu \in \mathbb{R}.$$

To explain the effects of μ, we recognize that the deformation (14) is equivalent to (7) under the following change of coordinates:

$$\lambda \leftarrow \frac{\lambda\mu}{1-\lambda+\mu\lambda}. \tag{15}$$

When λ is close to 0, this change of coordinates results approximately in $\lambda \leftarrow \mu\lambda$. So, if $\mu > 1$, the equivalent step in the λ-direction will be larger near $\lambda = 0$. In effect, $\mu > 1$ compresses the domain near $\lambda = 0$ and stretches it near $\lambda = 1$, and $\mu < 1$ has the reverse effect.

Since we do not have any method to predict the curvature of the homeomorphism a priori, we rely on experience to choose μ. Since we are typically dealing exclusively with external aerodynamic flows, we can still automate this process by determining a suitable value of μ for some typical flows and assuming that the value will still be reasonable for other flows of interest. We do this by performing a flow solve with the CHC method using a relatively small and constant $\Delta\lambda$, and some initial guess for μ. We over-solve the nonlinear sub-problems so that the error at the p_k-th step is very small. When the flow solve is complete, we minimize the H^2-norm of some quantification of curvature. The H^2-norm is used to place additional emphasis on regions where the curvature is particularly high. To quantify curvature in the present work, we have used the derivative of the error with respect to λ:

$$\min_{\mu} r(\mu), \; r(\mu) = \int_{\lambda=1}^{0} \left[\frac{de}{d\lambda}\right]^2 d\lambda \rightarrow \sum_k \left[\frac{e_{k+1}^{(1)} - e_k^{(p_k)}}{\lambda_{k+1}-\lambda_k}\right]^2 \Delta\lambda \approx \sum_k \frac{\left[e_{k+1}^{(1)}\right]^2}{\lambda_{k+1}-\lambda_k}, \tag{16}$$

$$e_k^{(n)} \approx \left\| \mathbf{Q}_k^{(n)} - \mathbf{Q}_k^{(p_k)} \right\|, \; e_k^{(p_k)} \in \mathbb{R}, \; r : \mathbb{R} \rightarrow \mathbb{R}.$$

To minimize $r(\mu)$, it would appear that $r(\mu)$ would need to be calculated for a range of μ, requiring numerous flow solves. However, this is not the case. Considering equation (15), we can in fact determine the value of $r(\mu)$ for any μ from a single flow solve by applying the change of coordinates to λ. Of course, for large changes in μ, the Riemann sum approximation to the H^2-norm will becomes less accurate, so if μ changes by a large amount (eg. factor of 5 or more) then we should reapply the algorithm starting with the updated value of μ.

For implementation, we apply μ-scaling as two components:

$$\mu = \mu_r \mu_i, \; \mu_r, \mu_i \in \mathbb{R}, \tag{17}$$

where μ_r is determined from the algorithm (and is treated as part of $\mathcal{G}$), and μ_i is user-supplied. This way, the user can supply a value of μ_i that is always relative to the default $\mu_i = 1$ for all homotopy functions and for all flows. In practice, since our algorithm for determining μ_r is based purely on curvature, we may choose to use $\mu_i \approx 0.8$ to stretch out traversing near $\lambda = 0$, where the nonlinear sub-problems become more difficult to solve.

Using this algorithm, we have set μ_r to 0.7 for inviscid flows, 0.5 for laminar flow, and 0.1 for RANS cases. In the present study, we use $\mu_i = 1$ for all test cases.

V.G. κ-Scaling

The scaling of $\Delta\lambda$ relative to $\|\Delta\mathbf{Q}\|$ is an important consideration for steplength adaptation, as it affects how much $\Delta\mathbf{Q}$ is emphasized relative to $\Delta\lambda$.

To study this problem, we consider a predictor steplength $\Delta\lambda$ and a following corrector step with size $\left\|\Delta\mathbf{Q}_k^{(1)}\right\|$. We then develop a relationship between these two quantities and θ, which is the angle between the constant-λ corrector step $\Delta\mathbf{Q}_k^{(1)}$ and a minimum-norm corrector step[c] $\Delta\mathbf{Q}_k^{(1_c)}$ in the case of an embedding algorithm. We formulate the problem in this way because θ is an intuitive quantity that we might propose to be around 5° if $\Delta\lambda$ and $\left\|\Delta\mathbf{Q}_k^{(1)}\right\|$ are scaled properly relative to each other. The relationship works out to the following:

$$\Delta\lambda = \pm \left\|\Delta\mathbf{Q}_k^{(1)}\right\| \cot\theta. \tag{18}$$

[c]The minimum-norm corrector allows λ to vary and applies a corrector step in the minimum-norm sense

The derivation is provided in Appendix C.

Equation (18) is rather informative. In particular, if we consider the case of $\left\|\Delta\mathbf{Q}_k^{(1)}\right\| \gg \Delta\lambda$, we see that $\theta_k \approx \pm\frac{\pi}{2}$, and equation (38) further indicates that $\left\|\Delta\mathbf{u}^{(c)}\right\| \approx \Delta\lambda$. Hence, if the state variables are scaled too large with respect to λ, then the minimum-norm corrector will act in the opposite (considering equation (37)) direction of an embedding step and with the same magnitude. In other words, the first corrector step will actually reverse the previous embedding step under these circumstances. Though we do not apply a minimum-norm corrector in this study, the impact on steplength adaptation is clear.

To correct this problem, we scale λ by a factor κ, which we can determine from a numerical algorithm. Considering equation (18), we use the following construction:

$$\Delta\lambda_{\text{ideal}} = \|\Delta\mathbf{Q}_{\text{ref}}\| \cot\theta_{\text{ref}},\ \theta_{\text{ref}} \in \mathbb{R}, \tag{19}$$

where θ_{ref} can be user-supplied. The scaling is implemented by modifying the convex homotopy continuation algorithm (7) to the following:

$$\mathcal{H}(\mathbf{Q}, \lambda_k) = (\kappa - \lambda_k)\,\mathcal{R}(\mathbf{Q}) + \lambda_k \mu \mathcal{G}(\mathbf{Q}) = \mathbf{0}, \tag{20}$$

$$k \in [0, p],\ \lambda_k, \mu, \kappa \in \mathbb{R},\ \lambda_0 = \kappa,\ \lambda_p = 0,\ \lambda_{k+1} < \lambda_k,$$

$$\kappa = \frac{\Delta\lambda_{\text{ideal}}}{\Delta\lambda} = C_\kappa \kappa_f \cot\theta_{\text{ref}},\ C_\kappa = \frac{\|\Delta\mathbf{Q}\|}{\Delta\lambda}, \tag{21}$$

$$\mathcal{H} : \mathbb{R}^{N+1} \to \mathbb{R}^N,\ \mathcal{G} : \mathbb{R}^N \to \mathbb{R}^N,\ \mathcal{R} : \mathbb{R}^N \to \mathbb{R}^N, C_\kappa, \kappa_f, \Delta\lambda \in \mathbb{R},$$

where we have supplied the scaling factor κ_f for additional user input, since θ_{ref} may be unintuitive, and $\Delta\lambda$ is the change in λ that accompanies $\Delta\mathbf{Q}$. Generally, this formula is intended to be applied without user intervention, since user intervention is highly unintuitive and fine-tuning of κ is really unimportant; we are satisfied with any reasonable value so that the homotopy features can work properly.

To use equation (21), we need an estimate for C_κ, which is clearly going to be problem-dependent. One possible solution is to calibrate the flow solver by running several test cases, and correlating C_κ to the flow variables and grid parameters according to some target value for θ_{ref}. Some numerical testing has shown that the relationship between C_κ and Mach number is fairly linear and that C_κ is relatively insensitive to angle of attack and Reynolds number. Some analysis indicates that C_κ should also be taken approximately proportional to $\sqrt{n}$ for our discretization.

One further issue that we consider regarding κ-scaling is the distance to the curve criterion for steplength adaptation. Before this analysis, it was unintuitive how the target distance $\tilde{\delta}$ should be chosen. If we consider now that the length of a corrector step after an embedding step $\Delta\lambda$ is given by equation (38), then the interpolated distance at some reference predictor length $\Delta\lambda_{\text{ref}}$ for the standard constant-λ corrector is given by:

$$\tilde{\delta} = C_\kappa \delta_f \Delta\lambda_{\text{ref}}, \tag{22}$$

where $\Delta\lambda_{\text{ref}} \in [0, 1]$ can be taken as the initial steplength, and $\delta_f \in \mathbb{R}$ is a user-suppliable factor. It is far more intuitive to choose this factor than to choose $\tilde{\delta}$ directly.

As a final note, we do not use a minimum-norm corrector in this study, but we provide the equation for $\tilde{\delta}$ for the minimum-norm corrector for possible future reference:

$$\tilde{\delta} = \frac{\Delta\lambda\,\|\Delta\mathbf{Q}\|\,\delta_f}{\sqrt{\|\Delta\mathbf{Q}\|^2 + \Delta\lambda^2}} \frac{\Delta\lambda_{\text{ref}}}{\Delta\lambda} = \frac{C_\kappa \delta_f \kappa \Delta\lambda_{\text{ref}}}{\sqrt{C_\kappa^2 + \kappa^2}}. \tag{23}$$

Though we have modified the problem so that λ is now in the interval $[0, \kappa]$ instead of $[0, 1]$, it is more intuitive to think of λ as taking values within the interval $[0, 1]$ and so we will continue to refer to it as such throughout the rest of this paper, and in particular in the results section.

V.H. Extension to the RANS Equations

To extend the homotopy method to the RANS equations, we need to append a sixth equation to the operator, since the Spalart-Allmaras turbulence model[26] is a sixth equation that is coupled to the Navier-Stokes equations. This sixth equation contains a convective term which requires some form of dissipation

for numerical stability. This is accomplished in our flow solver[25] by appending a sixth dissipative term to the dissipation operator of the form:

$$\mathcal{D}^{(2)}_{\mathrm{SA}}\tilde{\nu} = |U_n|\,\bar{\Delta}^T_\xi \bar{\Delta}_\xi \tilde{\nu}. \tag{24}$$

We do not use equation (24) in the homotopy function because it can introduce very small eigenvalues into the eigensystem since $|U_n|$ can be small at some grid points. One solution to this problem might be to add a positive non-zero term to $|U_n|$ to ensure that it does not become small. One logical choice for this term might be a, the speed of sound, resulting in an operator very similar to (8). For simplicity, the function that we use for $\mathcal{G}$ for the RANS equation is just the application of (8) to a 6×6 system.

One issue that has presented as a major obstacle for curve-tracing with the RANS equations is the scaling of the flow variables. It is well-known that the turbulent variable $\tilde{\nu}$ present in the Spalart-Allmaras model can take values similar in magnitude to the mean flow equations in some regions of the domain but can be roughly 10^3 times greater in other regions. This can adversely affect the performance of the linear solver, and previous researchers[25,39] have mitigated this effect by applying a factor of 10^3 to the columns of the Jacobian corresponding to $\tilde{\nu}$, and correcting the solution when the linear solve is complete. This is equivalent to (but simpler to implement than) non-dimensionalizing $\tilde{\nu}$.

Since it is more complicated to perform this local scaling properly for steplength adaptation, we have instead decided to non-dimensionalize $\tilde{\nu}$ directly. However, we have found that the factor of 10^3 de-emphasizes $\tilde{\nu}$ too much and that a factor of 10 is sufficient. To ensure that the linear system remains well-scaled, we apply an additional column scaling factor of 10^2.

VI. Results

The numerical flow solver contains many parameters that can be used to balance between robustness and speed. To demonstrate the usefulness of CHC as a globalization strategy, we must either show that the algorithm demonstrates superior speed than PTC with comparable or superior robustness, or superior robustness at comparable or superior speed. While it is impossible to conduct a "perfect" comparison between the two methods since we are always limited to running a finite number of test cases, we provide in this section a reasonably quantitative comparison at a large number of Mach numbers (Ma) and angles of attack (AoA) on a variety of meshes for the laminar Navier-Stokes, Euler, and RANS equations.

VI.A. Laminar flow

To ensure that the comparison is "fair", some investigation of the effect of different solver parameters was performed on mesh 3 (see Table 1). The flow solution is initialized using freestream conditions and is considered converged when the residual norm has dropped 10 orders of magnitude relative to its value at initialization. We exclusively use FGMRES and ILU(2) as the linear solver and preconditioner respectively. We have found that a minimum fill level of 2 is generally required to solve the linear system effectively for laminar flow. We use finite-differencing for matrix-vector products in the inexact Newton phase (INP) in all cases. Since there are many cases where we do not know if the solution will be subsonic or transonic, we apply the second-order dissipation with the pressure switch in all cases.

For each parameter tested, the algorithm is applied for 40 operating conditions: Mach numbers (Ma) between 0.2 and 0.9 in increments of 0.1, and angles of attack (α) between 0^o and 20^o in intervals of 5^o. Reynolds number was set to 1000 for all cases. We have found that increasing the Reynolds number beyond this can greatly reduce the number of cases that converge as the flow starts to become dynamically unstable in many cases. This wide range of operating conditions includes many cases that do not have time-stable steady solutions, and hence are not expected to converge. This is done to ensure that we have captured the entire region that does converge in order to identify if there are specific conditions under which one of the algorithms performs better.

Performance is compared by averaging the number of equivalent residual evaluations required to converge the flow for all cases that converged, where relative residual evaluations are defined as the total amount of CPU time required for a complete flow solve divided by the CPU time required to evaluate $\mathcal{R}(\mathbf{Q})$ once. To conduct the comparison, we construct a baseline version of PTC and CHC and test the effect of changing individual parameters or combinations of parameters.

Mesh	Geometry	Flow	Topology	Blocks	Nodes	Max. off-wall spacing
1a	ONERA M6	Inviscid	H-C	32	1.9×10^6	2.00×10^{-3}
1b	ONERA M6	Inviscid	H-C	256	1.9×10^6	1.09×10^{-3}
2	ONERA M6	Laminar	H-C	16	1.9×10^6	2.01×10^{-4}
3	ONERA M6	Laminar	H-H	48	2.1×10^6	2.18×10^{-4}
4	NACA 0012	RANS	H	8	1.9×10^4	1.00×10^{-6}
5	ONERA M6	RANS	H-H	192	2.2×10^6	1.25×10^{-6}

Table 1: Meshes and flow conditions used in this study. The number of nodes is the total over all blocks.

Algorithm		Description	CPU time	Converged
PTC	A1	Baseline	1.00	28/40
	A2	Approximate matrix-vector products	1.62	26/40
	A3	$\mu_{\text{rel}} = 0.01$	1.09	29/40
	A4	ILU factorization updated every 2 iterations	0.96	28/40
	A5	ILU factorization updated every 3 iterations	0.98	29/40
CHC	B1	Baseline	1.00	30/40
	B2	Finite-differencing used for all matrix-vector products	1.14	27/40
	B3	ILU factorization updated every 2 iterations	0.83	29/40
	B4	B3, but with subproblems solved to $\mu_{\text{rel}} = 0.1$	0.67	27/40
	B5	B4, using an embedding algorithm	0.67	27/40
	B6	ILU factorization updated every 3 iterations	0.78	30/40
	B7	B6, but with subproblems solved to $\mu_{\text{rel}} = 0.1$	0.67	29/40
	B8	B7, using an embedding algorithm	0.64	29/40

Table 2: Performance of the PTC and CHC algorithms under different algorithm parameters for laminar flow on mesh 3. Unless otherwise specified, the description describes the algorithm relative to the baseline. CPU time is taken relative to the baseline CPU time for each algorithm and is calculated as an average over all operating conditions that converged for both the algorithm and the baseline.

- Baseline PTC: $a = 0.01$, $b = 1.5$, $\mu_{\text{rel}} = 0.001$, matrix-vector products are calculated using finite-differences, and the preconditioner is updated at every nonlinear iteration.
- Baseline CHC: Algebraic tangent applied for $\Delta\lambda > 0.05$ and secant otherwise, $\Delta\lambda$ initialized at 0.2, AMVs used for the subproblems but FDMVs used for the algebraic tangent, and the preconditioner is updated at every nonlinear iteration. The nonlinear subproblems are solved to a relative tolerance of $\mu_{\text{rel}} = 0.01$, where the relative residual is always calculated relative to the embedding point. No relative residual requirement is enforced for globalization; the flow residual is assumed to be globalized when $\lambda = 0$ is achieved.

Table 2 summarizes the relative performance of both algorithms for several choices of algorithm parameters. Figure 1 shows a comparison of the performance of algorithms A4 and B7 which, as indicated in Table 2, are PTC and CHC algorithms respectively using "good" parameters. However, for the final comparison, the angles of attack used were in a more realistic range. Of the cases investigated, PTC converged in 24/40 cases and CHC converged in 28/40 cases. Of the cases that converged for both algorithms, the CHC algorithm fully converged the flow residual in 88% of the CPU time taken by PTC.

An apparent oddity that must be addressed is that we observe from Figure 1 that it is consistently mid-range Mach numbers that fail to converge, whereas convergence can still be achieved in many cases for higher Ma and AoA. This is a phenomenon that we have only observed for laminar flow and may be the result of local dynamical instabilities due to local mesh skewness or coarseness, which could be leading to

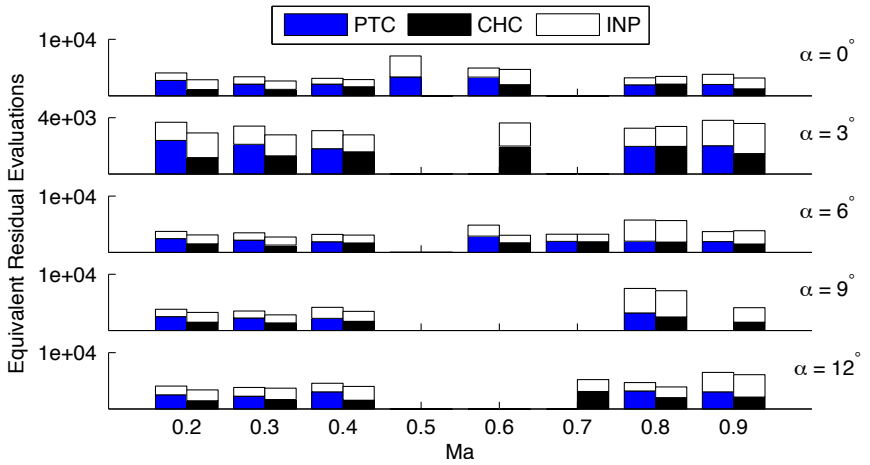

Figure 1: Algorithm performance for laminar flow on mesh 3. Cases not shown did not converge.

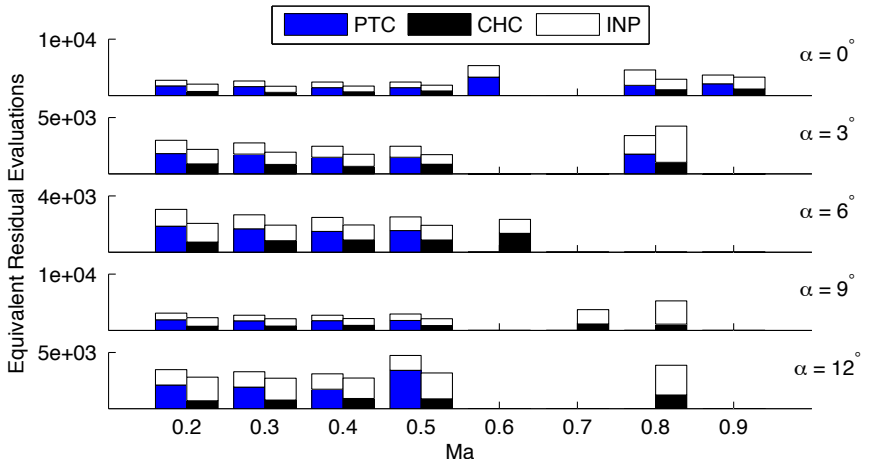

Figure 2: Algorithm performance for laminar flow on mesh 2. Cases not shown did not converge.

additional entropy generation and inducing instability. However, this is not the focus of the current study as it is not directly related to the globalization strategy, and so we do not pursue it further at present.

The dissipation operator used as the homotopy function in this study is a grid-dependent operator. As such, it will "see" different mesh topologies differently and we might expect different performance on different topologies. This was investigated by applying algorithms A4 and B7 to mesh 2, which is of similar size and quality as mesh 3 but with an H-C topology, whereas mesh 3 has an H-H topology. In this case, CHC converged in 27/40 of the operating conditions investigated whereas PTC has only converged in 24/40. Out of the cases that converged for both algorithms, the CHC algorithm fully converged the flow residual in 76% of the CPU time taken by PTC. Figure 2 displays the outcome of the comparison.

VI.B. Inviscid Flow

Some analysis of optimal algorithm parameters for PTC for inviscid flows has been performed in the past.[22,40] Our own analysis of these parameters on mesh 1a also indicates that the solver is fast and robust under the following parameters: $a = 0.01$, $b = 1.5$, ILU(0) for globalization phase updated ever 3 iterations, ILU(1) in the inexact Newton phase (INP) updated every iteration, linear solver relative tolerance of 0.05, $\mu_{\text{rel}} = 1/15$. Furthermore, the solution update at each nonlinear iteration of the globalization phase is relaxed by a factor of 0.6 for additional stability.

We have performed an analysis of optimal algorithm parameters for the CHC algorithm similar to the analysis shown earlier for laminar flow. We have found that the additional cost of the algebraic tangent calculation is not as cost-effective for the Euler equations as the secant method and the additional robustness

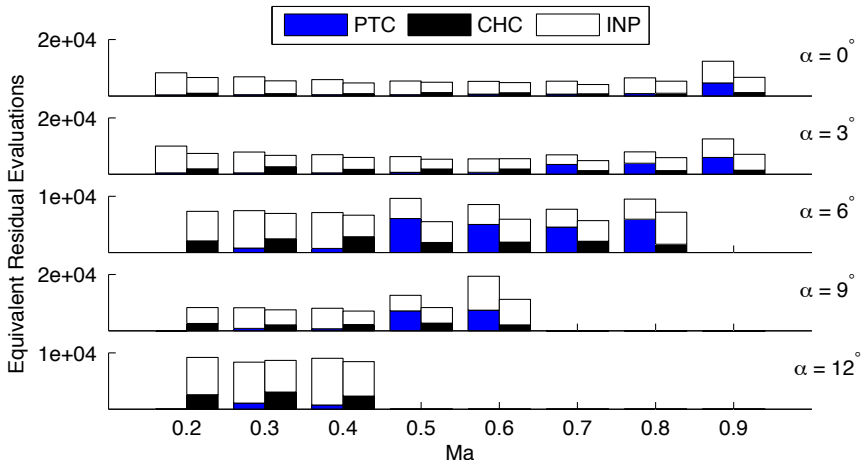

Figure 3: Algorithm performance for inviscid flow on mesh 1a. Cases not shown did not converge.

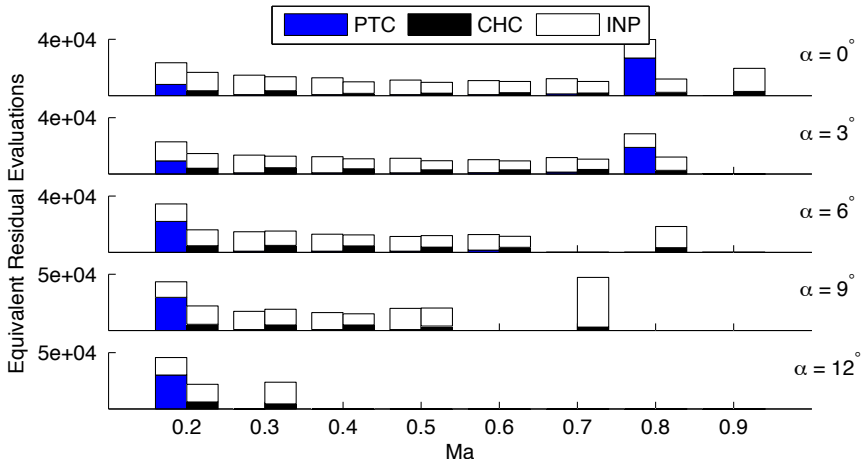

Figure 4: Algorithm performance for inviscid flow on mesh 1b. Cases not shown did not converge.

is not needed. We have also determined that we can solve the nonlinear subproblems to a tolerance of only $\mu_{\text{rel}} = 0.5$ without sacrificing robustness. Aside from these two changes, the algorithm that we apply is algorithm B7.

The comparison is performed on mesh 1a and is shown in Figure 3. We again emphasize that many of the operating conditions investigated are not expected to converge, but are included to ensure that the entire range of stable operating conditions is included in the study. The success rate of CHC was 31/40 and the success rate of PTC was 28/40. On average, considering only the operating conditions for which both algorithms converged, CHC converged in 79% of the CPU time taken by PTC.

To investigate the effects of mesh refinement, the performance of the inviscid CHC and PTC algorithms were compared on mesh 1b. Mesh 1b was generated from mesh 1a by refining the mesh by a factor of 2 in each direction and then splitting the blocks once in each direction, resulting in a mesh with 8 times as many nodes and 8 times as many blocks, but the same number of nodes per block. Relative performance of the two algorithms is shown in Figure 4. The success rate of CHC was 28/40 and the success rate of PTC was 24/40. On average, considering only the operating conditions for which both algorithms converged, CHC converged in 81% of the CPU time taken by PTC.

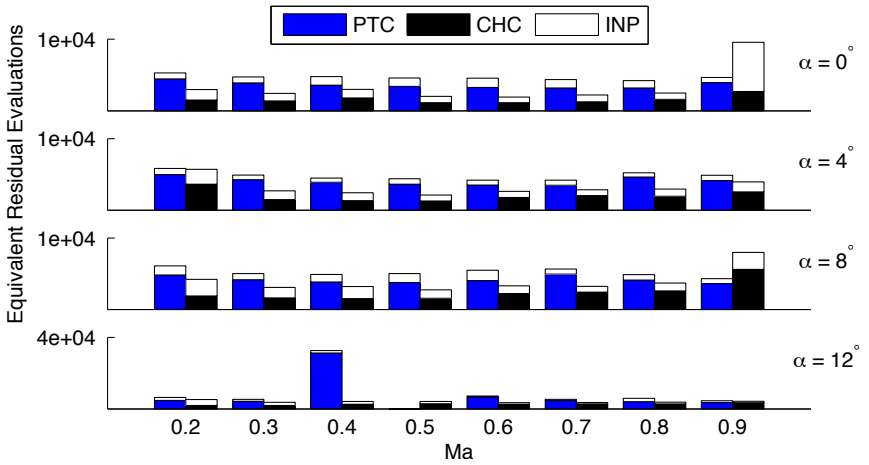

Figure 5: Algorithm performance for the RANS equations on mesh 4. Cases not shown did not converge.

VI.C. Turbulent Flow

Due to the higher CPU cost of RANS problems, parameter testing was performed on a two-dimensional NACA 0012 aerofoil (Table 1, mesh 4). From this analysis, we have found that the optimal parameters for CHC are similar to the optimal parameters for the laminar case. However, we have found that the curvature of the homotopy deformation in RANS solves is higher than with the inviscid and laminar cases. As such, we impose $\Delta\lambda_{\min} = 0.03$, $\Delta\lambda_{\max} = 0.1$, and initialize with $\Delta\lambda = 0.03$. We may interpret this as a "longer" curve that requires more steps to trace accurately. We have also observed that failure to globalize the sub-problems results in higher risk of instability than the laminar or inviscid cases and setting $\mu_{\text{rel}} = 0.5$ can be significantly less robust than $\mu_{\text{rel}} = 0.1$. However, using $\mu_{\text{rel}} = 0.5$ in conjunction with an option to repeat the previous predictor step at reduced steplength if a sub-problem diverges can be just as robust as the $\mu_{\text{rel}} = 0.1$ case and significantly faster in most cases. We have also found that the AMVs can sometimes give unpredictable performance for RANS sub-problems and so we use FDMVs.

PTC also takes different parameters for RANS cases, typically we use $a \in \left[10^{-3}, 10^{-4}\right]$, $b \in [1.1, 1.3]$, $\mu_{\text{rel}} \in \left[10^{-5}, 10^{-3}\right]$. Small μ_{rel} is required for turbulent flows due to the small off-wall spacing and large initial fluctuations in $\tilde{\nu}$, which can lead to a large residual drop initially.[25] For this reason, finer meshes are found to require smaller values of μ_{rel}.

Figure 5 shows a comparison between CHC and PTC on mesh 4 at $\text{Re} = 4 \times 10^7$. PTC uses $a = 0.0001$, $b = 1.1$, and $\mu_{\text{rel}} = 10^{-4}$. The success rate of CHC was 32/32 and the success rate of PTC was 31/32. On average, considering only the operating conditions for which both algorithms converged, CHC converged in 69% of the CPU time taken by PTC. We note that increasing b to 1.35 results in about the same CPU time for CHC and PTC but reduces the success rate of PTC to 29/32. The combination $a = 0.001$, $b = 1.1$ reduces the success rate of PTC to 19/32, verifying that the small value used for a is indeed required.

RANS test cases were also carried out on mesh 5. We use $a = 0.001$, $b = 1.1$, $\mu_{\text{rel}} = 10^{-4}$ for PTC. We have found that larger values of b greatly reduces robustness in this case. For CHC, we use the same parameters as we did on mesh 4, except that we solve the sub-problems with AMVs instead of FDMVs due to higher speed. While not apparent from these test cases, we have found that for RANS cases, solving the sub-problems with AMVs is not always reliable as it can sometimes lead to instability due to poor predictor performance. If a wider range of operating conditions were considered, it is possible that FDMVs would prove to be a better choice for this test case.

Figure 6 shows a comparison between CHC and PTC on mesh 5 at $\text{Re} = 1.172 \times 10^7$. Due to the high cost of these cases, fewer operating conditions were considered. The success rate of both CHC and PTC was 16/16. On average, considering only the operating conditions for which both algorithms converged, CHC converged in 111% of the CPU time taken by PTC.

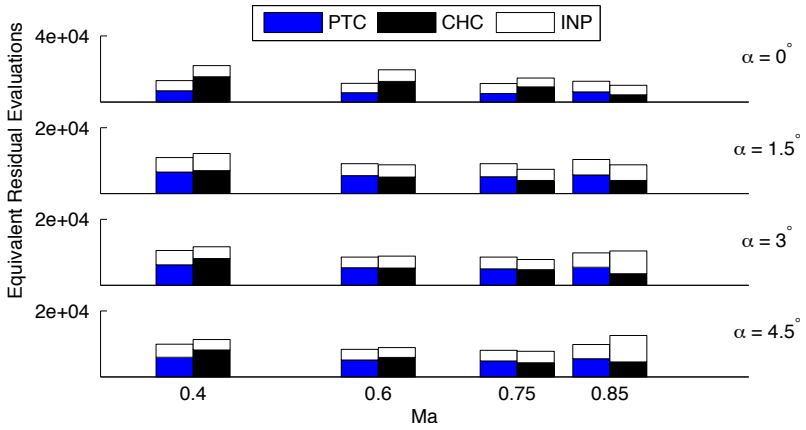

Figure 6: Algorithm performance for the RANS equations on mesh 5. Cases not shown did not converge.

VII. Conclusions

A convex homotopy continuation algorithm has been presented for application to the steady Euler, Navier-Stokes, and Reynolds-Averaged Navier-Stokes equations. Using a second-order dissipation operator as the homotopy function, performance was shown to be at least as good as the established pseudo-transient continuation algorithm for the inviscid and laminar cases on several mesh topologies and levels of mesh refinement. While we were able to demonstrate feasibility of the CHC algorithm for RANS flows with reasonable performance, the efficiency of the algorithm was not demonstrated to be better than PTC. This can mainly be attributed to the high curvature of the deformation, requiring a large number of steps to achieve globalization, as well as high cost of solving the sub-problems. It may be possible to reduce these costs by considering alternate homotopy functions, or at least an alternative to the component corresponding to the turbulence model. This will be investigated in the future.

Appendix A

Purpose: Develop sufficient conditions for local invertibility of the nonlinear system given by equation (9).

Treating $\mathcal{G}$ as a quasi-linear operator on $\mathbf{Q}$:

$$\mathcal{G}^{(l)} = \begin{cases} \left(\sigma_L + d_{1\frac{1}{2}}, -d_{1\frac{1}{2}}, 0, 0, \ldots, 0\right) & i = 1, \\ \text{trid}\left(-d_{i-\frac{1}{2}}, d_{j-\frac{1}{2}} + d_{i+\frac{1}{2}}, -d_{i+\frac{1}{2}}\right) & i \in [2, n-1], \\ \left(0, \ldots, 0, 0, -d_{N-\frac{1}{2}}, d_{N-\frac{1}{2}} + \sigma_R\right) & i = N, \end{cases} \tag{25}$$

$$\text{where } \mathcal{G}(\mathbf{Q}) = \mathcal{G}^{(l)}\mathbf{Q} - (\sigma_L \mathbf{Q}_L, 0, \ldots, 0, \sigma_R \mathbf{Q}_R).$$

We will now need to make use of the following definition:

Definition
A matrix $\mathcal{A}$ with ij-th component $\mathcal{A}_{ij}$ is *strictly irreducibly row-diagonally dominant* if:

$$\sum_{i \neq j} |\mathcal{A}_{ij}| \leq |\mathcal{A}_{ii}| \ \forall i, \ \sum_{i \neq j} |\mathcal{A}_{ij}| < |\mathcal{A}_{ii}| \text{ for some } i,$$

and $\mathcal{A}$ is irreducible.

We clearly have:

$$\left|\mathcal{G}_{ii}^{(l)}\right| - \sum_j \left|\mathcal{G}_{ij}^{(l)}\right| = \begin{cases} \sigma_L & i = 1 \\ 0 & i \in [2, n-1] \\ \sigma_R & i = N \end{cases} \tag{26}$$

It is clear from the above equation that

$$\sigma_R \geq 0,\ \sigma_L \geq 0,\ \sigma_L + \sigma_R > 0, \tag{27}$$

are necessary and sufficient conditions for $\mathcal{G}^{(l)}$ to be strictly row-diagonally dominant. Since $\mathcal{G}^{(l)}$ is irreducible, it then follows that $\mathcal{G}^{(l)}$ is nonsingular, since irreducibly row-diagonally dominant operators are nonsingular.[32] Hence, condition (27) is sufficient to guarantee that the system is nonsingular.

Appendix B

Purpose: Derive the algebraic tangent method.

A formal definition of the tangent vector is given in Allgower and Georg,[3] and reproduced here:

Definition
Let $\mathcal{A} \in \mathbb{R}^{N\times(N+1)}$ with $\text{rank}\,(\mathcal{A}) = N$. The unique vector $t\,(\mathcal{A}) \in \mathbb{R}^{N+1}$ satisfying the three conditions:

(1) $\mathcal{A}t = \mathbf{0}$; (2) $\|t\| = 1$; (3) $\det \begin{pmatrix} \mathcal{A} \\ t^T \end{pmatrix} > 0$;

is called the *tangent vector induced by* $\mathcal{A}$.

Consider the system of equations:

$$\nabla\mathcal{H}\,(\mathbf{u})\,\mathbf{z} = \nabla\mathcal{H}\,(\mathbf{u})\,e_i^T, \qquad e_i\mathbf{z} = 0, \qquad e_i \in \mathbb{R}^{1\times N}. \tag{28}$$

Assuming that $\mathbf{z} \in \mathbb{R}^n$ satisfies the system (28), then it can easily be shown that

$$t\,(\nabla\mathcal{H}\,(\mathbf{u})) = \pm\frac{\tau}{\|\tau\|}, \qquad \text{where} \quad \tau = e_i^T - \mathbf{z}, \tag{29}$$

by substituting (29) into (28). However, there is still some question of the optimal way to select e_i.

For the system (28) to be nonsingular, e_i must not be orthogonal to $\ker\,(\nabla\mathcal{H}\,(\mathbf{u}))$. In fact, to get a well-conditioned system, we would like to have e_i to be as parallel to $\ker\,(\nabla\mathcal{H}\,(\mathbf{u}))$ as possible. For simplicity, Rheinboldt[36] takes e_i as the dual operator $e_i\mathbf{z} = \mathbf{z}\,[i]$. For large sparse systems, we notice that there is significant benefit to exclusively choosing $i = N+1$. In this case, $e_i\mathbf{z} = 0$ reduces to $\Delta\lambda = 0$, thus making it unnecessary to solve for λ in equation (28). Thus we can delete the last column of $\nabla\mathcal{H}\,(\mathbf{u})$, which is critical, since this column is of the form:

$$\nabla_\lambda\mathcal{H}\,(\mathbf{Q},\lambda) = \mathcal{G}\,(\mathbf{Q}) - \mathcal{R}\,(\mathbf{Q})\,, \tag{30}$$

and the system (28) reduces to:

$$\nabla_{\mathbf{Q}}\mathcal{H}\,(\mathbf{Q},\lambda)\,\mathbf{z} = \mathcal{G}\,(\mathbf{Q}) - \mathcal{R}\,(\mathbf{Q})\,. \tag{31}$$

Furthermore, we observe that equation (29) reduces to:

$$t\,(\nabla\mathcal{H}\,(\mathbf{u})) = \pm\frac{\tau}{\|\tau\|}, \qquad \text{where} \quad \tau = \begin{pmatrix} \mathbf{z} \\ -1 \end{pmatrix}. \tag{32}$$

The sign of t is chosen to be positive, since this will give $\lambda_{k+1} < \lambda_k$, and hence there is no need to estimate the determinant.

Appendix C

Purpose: Derive equation (18).

Assume that the $k-1$st sub-problem has been solved to some suitable tolerance in p_{k-1} iterations. We then rearrange:

$$\begin{aligned}\mathcal{H}\left(\mathbf{Q}_{k-1}^{(p_{k-1})},\lambda_{k-1}^{(p_{k-1})}\right) &= \lambda_{k-1}^{(p_{k-1})}\mathcal{G}\left(\mathbf{Q}_{k-1}^{(p_{k-1})}\right)+\left(1-\lambda_{k-1}^{(p_{k-1})}\right)\mathcal{R}\left(\mathbf{Q}_{k-1}^{(p_{k-1})}\right)\approx\mathbf{0}\\ \Rightarrow\mathcal{R}\left(\mathbf{Q}_{k-1}^{(p_{k-1})}\right) &\approx -\lambda_{k-1}^{(p_{k-1})}\left(\mathcal{G}\left(\mathbf{Q}_{k-1}^{(p_{k-1})}\right)-\mathcal{R}\left(\mathbf{Q}_{k-1}^{(p_{k-1})}\right)\right).\end{aligned} \tag{33}$$

Then, we take an embedding step defined by $\mathbf{Q}_k^{(1)}\leftarrow\mathbf{Q}_{k-1}^{(p_{k-1})}$, $\lambda_k^{(1)}\leftarrow\lambda_{k-1}^{(p_{k-1})}+\Delta\lambda$. The following corrector step is given by:

$$\begin{aligned}\mathcal{H}\left(\mathbf{Q}_k^{(1)},\lambda_k^{(1)}\right) &= \lambda_k^{(1)}\left(\mathcal{G}\left(\mathbf{Q}_k^{(1)}\right)-\mathcal{R}\left(\mathbf{Q}_k^{(1)}\right)\right)+\mathcal{R}\left(\mathbf{Q}_k^{(1)}\right)\\ &= \Delta\lambda\left(\mathcal{R}\left(\mathbf{Q}_k^{(1)}\right)-\mathcal{G}\left(\mathbf{Q}_k^{(1)}\right)\right).\end{aligned} \tag{34}$$

From this point forward, we will be dropping the superscript (1) and the subscript k in the interest of clarity.

Consider now the following linear system of equations:

$$\begin{aligned}\nabla\mathcal{H}\left(\mathbf{Q},\lambda\right)\mathbf{z} &= \mathcal{G}\left(\mathbf{Q}\right)-\mathcal{R}\left(\mathbf{Q}\right)=\frac{1}{\Delta\lambda}\mathcal{H}\left(\mathbf{Q},\lambda\right)\\ \Rightarrow\nabla\mathcal{H}\left(\mathbf{Q},\lambda\right)\Delta\lambda\mathbf{z} &= \mathcal{H}\left(\mathbf{Q},\lambda\right).\end{aligned} \tag{35}$$

From this, we can see that $\Delta\lambda\mathbf{z}=\Delta\mathbf{Q}$, the solution to the linear system resulting from the first Newton iteration for a constant-λ corrector. Considering the algebraic method presented for the tangent calculation (see Appendix B), the tangent vector is given by:

$$t\left(\mathcal{H}\left(\mathbf{Q},\lambda\right)\right)=\frac{1}{\sqrt{\left\|\Delta\mathbf{Q}\right\|^2+\Delta\lambda^2}}\left[\Delta\mathbf{Q};\Delta\lambda\right]. \tag{36}$$

Let us now consider the minimum-norm corrector (see Appendix D). Applying equation (44) and performing some algebra gives the modified corrector step:

$$\Delta\mathbf{Q}^{(c)}=\Delta\mathbf{Q}\left(\frac{\Delta\lambda^2}{\left\|\Delta\mathbf{Q}\right\|^2+\Delta\lambda^2}\right),\quad\Delta\lambda^{(c)}=-\Delta\lambda\frac{\left\|\Delta\mathbf{Q}\right\|^2}{\left\|\Delta\mathbf{Q}\right\|^2+\Delta\lambda^2}. \tag{37}$$

Defining the full corrector vector as $\mathbf{u}^{(c)}=\left[\mathbf{Q}^{(c)},\lambda^{(c)}\right]$, we can calculate the norm:

$$\left\|\Delta\mathbf{u}^{(c)}\right\|\equiv\sqrt{\left\|\Delta\mathbf{Q}^{(c)}\right\|^2+\Delta\lambda^2}=\frac{\Delta\lambda\left\|\Delta\mathbf{Q}\right\|}{\sqrt{\left\|\Delta\mathbf{Q}\right\|^2+\Delta\lambda^2}}. \tag{38}$$

We now calculate the unit vector $\Delta\hat{\mathbf{u}}^{(c)}$, though we will only need the first N terms (i.e. the part relating to the flow variables):

$$\Delta\hat{\mathbf{u}}^{(c)}\left[1:N\right]\equiv\frac{1}{\left\|\Delta\mathbf{u}^{(c)}\right\|}\Delta\mathbf{Q}^{(c)}=\Delta\mathbf{Q}\left(\frac{\Delta\lambda}{\sqrt{\left\|\Delta\mathbf{Q}\right\|^2+\Delta\lambda^2}\left\|\Delta\mathbf{Q}\right\|}\right). \tag{39}$$

Finally, we can calculate θ, the angle between $\Delta\mathbf{Q}$ and $\Delta\mathbf{Q}^{(c)}$:

$$\cos\theta\equiv\left\langle\Delta\hat{\mathbf{u}},\Delta\hat{\mathbf{u}}^{(c)}\right\rangle=\left\langle\Delta\hat{\mathbf{Q}},\Delta\hat{\mathbf{u}}^{(c)}\left[1:N\right]\right\rangle, \tag{40}$$

where $\langle\cdot,\cdot\rangle$ denotes the standard inner product. This can be simplified to:

$$\Delta\lambda=\pm\left\|\Delta\mathbf{Q}\right\|\cot\theta. \tag{41}$$

Appendix D

Purpose: Derive an algebraic expression for the minimum-norm corrector step.

The nonlinear sub-problems can be solved in the minimum-norm sense by treating λ as a variable during the corrector phase. This is done by modifying Newton's method.[3] In summary, we first solve the system:

$$\nabla\mathcal{H}\left(\mathbf{u}^{(n)}\right)\mathbf{z}^{(n)} = -\mathcal{H}\left(\mathbf{u}^{(n)}\right), \tag{42}$$

$$e_i\mathbf{z}^{(n)} = 0, \tag{43}$$

where $e_i\mathbf{z} = \mathbf{z}\,[i]$. We then obtain the minimum-norm solution by removing the component of $\mathbf{z}^{(n)}$ that lies parallel to the tangent curve:

$$\Delta\mathbf{u}^{(n)} = \mathbf{z}^{(n)} - \left\langle t\left(\nabla\mathcal{H}\left(\mathbf{u}^{(n)}\right)\right), \mathbf{z}^{(n)}\right\rangle t\left(\nabla\mathcal{H}\left(\mathbf{u}^{(n)}\right)\right), \tag{44}$$

where $\langle\cdot,\cdot\rangle$ denotes the standard inner product. Furthermore, choosing $i = N+1$ for e_i, the system (42), (43) reduces to:

$$\nabla_{\mathbf{Q}}\mathcal{H}\left(\mathbf{Q}^{(n)}, \lambda_k^{(n)}\right)\mathbf{z}^{(n)} = -\mathcal{H}\left(\mathbf{Q}^{(n)}, \lambda_k^{(n)}\right). \tag{45}$$

The minimum-norm corrector is thus given by solving equation (45) for $\mathbf{z}^{(n)}$ and substituting this into equation (44).

Acknowledgements

The authors gratefully acknowledge financial assistance from the National Science and Engineering Research Council (NSERC), MITACS, Bombardier, the Canada Research Chairs program, and the University of Toronto.

All 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]Kelley, C. T. and Keyes, D. E., "Convergence Analysis of Pseudo-Transient Continuation," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 2, 1998, pp. 508–523.

[2]Knoll, D. A. and Keyes, D. E., "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, Vol. 193, 2003, pp. 357–397.

[3]Allgower, E. L. and Georg, K., *Introduction to Numerical Continuation Methods*, Society for Industrial and Applied Mathematics, 1990.

[4]Poincaré, H., "Sur les courbes definé par une équation differentielle. I-IV," *Oevres*, Vol. 1, Gauthier-Villars, Paris, 1881-1886.

[5]Klein, F., "Neue Beiträge zur Riemannschen Funktionentheorie," *Mathematische Annalen*, Vol. 21, 1882-1883.

[6]Bernstein, S., "Sur la généralisation du problème de Dirichlet," *Mathematische Annalen*, Vol. 69, 1910, pp. 82–136.

[7]Leray, J. and Schauder, J., "Topologies et équations fonctionelles," *Annales Scientifiques de l'École Normale Supérieure*, Vol. 51, 1934, pp. 45–78.

[8]Lahaye, E., "Une méthode de résolution d'une categorie d'équations transcendantes," *Comptes Rendus Hebdomataires Séances de l'Académie de Sciences*, Vol. 198, 1934, pp. 1840–1842.

[9]Lahaye, E., "Sur la résolution des systèmes d'équations transcendantes," *Académie Royale de Belgique. Bulletin de la Classe de Sciences*, Vol. 5, 1948, pp. 805–822.

[10]Ortega, J. M. and Rheinboldt, W. C., *Iterative Solution of Nonlinear Equations in Several Variables*, SIAM, 1970.

[11]Allgower, E. L. and Georg, K., "Continuation and Path Following," *Acta Numerica*, Vol. 2, 1993, pp. 1–64.

[12]Carey, G. F. and Krishnan, R., "Continuation techniques for a penalty approximation of the Navier-Stokes equations," *Computer Methods for Applied Mechanics and Engineering*, Vol. 48, 1985, pp. 265–282.

[13]Riley, D. S. and Winters, K. H., "A numerical bifurcation of natural convection in a tilted two-dimensional porous cavity," *Journal of Fluid Mechanics*, Vol. 215, 1990, pp. 309–329.

[14]Sanchez, J., Marques, F., and Lopez, J. M., "A continuation and bifurcation technique for Navier-Stokes flows," *Journal of Computational Physics*, Vol. 180, 2002, pp. 78–98.

[15]Winters, K. H., "A bifurcation study of laminar flow in a curved rectangular cross-section," *Journal of Fluid Mechanics*, Vol. 180, 1987, pp. 343–369.

[16]Winters, K. H. and Cliffe, K. A., "The prediction of critical points for thermal explosions in a finite volume," *Combustion and Flame*, Vol. 62, 1985, pp. 13–20.

[17]Wales, C., Gaitonde, A. L., Jones, D. P., Avitabile, D., and Champneys, A. R., "Numerical continuation of high Reynolds number external flows," *International Journal for Numerical Methods in Fluids*, Vol. 68, 2012, pp. 135–159.

[18]Hicken, J. E. and Zingg, D. W., "Globalization Strategies for Inexact-Newton Solvers," *19th AIAA Computational Fluid Dynamics Conference and Exhibit*, San Antonia, Texas, United States, June 2009, AIAA-2009-4139.

[19]Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations," *AIAA Journal*, Vol. 24, No. 12, December 1986, pp. 1931–1940.

[20]Hicken, J. E., Buckley, H., Osusky, M., and Zingg, D. W., "Dissipation-based continuation: a globalization for inexact-Newton solvers," *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, United States, June 2011, AIAA-2011-3237.

[21]Hicken, J. E., *Efficient Algorithms for Aircraft Design: Contributions to Aerodynamic Shape Optimization*, Ph.D. thesis, University of Toronto, Toronto, Ontario, Canada, 2009.

[22]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, 2008, pp. 2773–2786.

[23]Lomax, H., Pulliam, T. H., and Zingg, D. W., *Fundamentals of Computational Fluid Dynamics*, Springer-Verlag, 2001.

[24]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 and Aerospace Exposition*, Orlando, Florida, United States, January 2010, AIAA-2010-116.

[25]Osusky, M. and Zingg, D. W., "A parallel Newton-Krylov-Schur flow solver for the Reynolds-Averaged Navier-Stokes equations," *50th AIAA Aerospace Sciences Meeting and Aerospace Exposition*, Nashville, Tennessee, United States, January 2012, AIAA-2012-0442.

[26]Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *30th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, January 1992, AIAA-92-0439.

[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, 1994, pp. 220–236.

[28]Funaro, D. and Gottlieb, D., "A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations," *Mathematics of Computation*, Vol. 51, 1988, pp. 599–613.

[29]Kreiss, H. and Scherer, G., "Finite element and finite difference methods for hyperbolic partial differential equations," *Mathematical Aspects of Finite Elements in Partial Differential Equations: proceedings of a symposium conducted by the Mathematics Research Center, the University of Wisconsin*, edited by C. de Boor, Mathematics Research Centre, the University of Wisconsin, Academic Press, 1974.

[30]Strand, B., "Summation by parts for finite difference approximations for d/dx," *Journal of Computational Physics*, Vol. 110, 1994, pp. 47–67.

[31]Saad, Y. and Sosonkina, M., "Distributed Schur complement techniques for general sparse linear systems," *SIAM Journal of Scientific Computing*, Vol. 21, 1999, pp. 1337–1357.

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

[33]Saad, Y. and Schultz, M., "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856–869.

[34]Saad, Y., "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM Journal on Scientific and Statistical Computing*, Vol. 14, No. 2, 1993, pp. 461–469.

[35]Hirsch, M. W., *Differential Topology*, Springer-Verlag, Berlin, Heidelberg, New York, 1976.

[36]Rheinboldt, W. C., "Solution Fields of Nonlinear Equations and Continuation Methods," *SIAM Journal on Numerical Analysis*, Vol. 17, No. 2, 1980, pp. 221–237.

[37]Georg, K., "A Note on Stepsize Control for Numerical Curve Following," *Homotopy Methods and Global Convergence*, edited by B. C. Eaves, F. J. Gould, H. Peitgen, and M. J. Todd, Plenum Press, New York, 1983, pp. 145–154.

[38]Den Heijer, C. and Rheinboldt, W. C., "On Steplength Algorithms for a Class of Continuation Methods," *SIAM Journal on Numerical Analysis*, Vol. 18, No. 5, 1981, pp. 925–948.

[39]Chisholm, T. T. and Zingg, D. W., "A Jacobian-free Newton-Krylov Algorithm for Compressible Turbulent Fluid Flows," *Journal of Computational Physics*, Vol. 228, 2009, pp. 3490–3507.

[40]Dias, S. and Zingg, D. W., "A High-Order Parallel Newton-Krylov Flow Solver for the Euler Equations," *19th AIAA Fluid Dynamics Conference and Exhibit*, San Antonio, Texas, United States, June 2009, AIAA-2009-3657.