Numerical Algorithms manuscript No.
(will be inserted by the editor)

Matrix-Free Monolithic Homotopy Continuation with Application to Computational Aerodynamics

David A. Brown · David W. Zingg

Received: xxxx / Accepted: xxxx

Abstract A matrix-free monolithic homotopy continuation algorithm is developed which allows for approximate numerical solutions to nonlinear systems of equations without the need to solve a linear system, thereby avoiding the formation of any Jacobian or preconditioner matrices. The algorithm can converge from an arbitrary starting guess, under suitable conditions, and can give a sufficiently accurate approximation to the converged solution such that a rapid locally convergent method such as Newton's method will converge successfully. Several forms of the algorithm are presented, as are augmentations to the algorithms which can lead to improved efficiency or stability. The method is validated and the stability and efficiency are investigated numerically based on a computational aerodynamics flow solver.

Keywords homotopy · continuation · monolithic · computational fluid dynamics · numerical algorithms · explicit method · explicit time marching

1 Introduction

Homotopy continuation [1] can be an efficient and robust globalization strategy for implicit algorithms for solving nonlinear systems of equations. The study of homotopy continuation as a competitive continuation algorithm for solving computational aerodynamics problems was initiated by Hicken and Zingg [14] and continued by Hicken et al. [12], Brown and Zingg [2], and Yu and Wang [25]. Subsequently, Brown and Zingg [3] developed an efficient and robust monolithic homotopy continuation framework based on a dissipation operator and applied it to the

David A. Brown
University of Toronto Institute for Aerospace Studies
Toronto, Ontario, Canada, M3H 5T6
E-mail: utiasdavid.brown@mail.utoronto.ca

David W. Zingg
University of Toronto Institute for Aerospace Studies
Toronto, Ontario, Canada, M3H 5T6
E-mail: dwz@oddjob.utias.utoronto.ca

computational solver for steady aerodynamics developed by Hicken and Zingg [13], demonstrating superior performance than an implementation of the widely-used implicit pseudo-transient method. The study encompassed several wing and airfoil geometries for inviscid and viscous subsonic and transonic flows.

The monolithic homotopy continuation algorithm of Brown and Zingg [3] is based on the dynamic inversion principle described by Getz and Marsden [10] and requires the solution to a linear system of equations at each iteration. The linear system can be solved inexactly using an iterative linear solver such as preconditioned (F)GMRES [22,23]. What is presented in this paper is an algorithm similarly based on the dynamic inversion principle but which does not require the formation of any matrices during the continuation process. The method is designed to give an approximate solution to a given nonlinear system of equations. The residual can then be reduced to machine accuracy using, for example, a quasi-Newton method.

The aim of developing the matrix-free monolithic homotopy continuation algorithms is not cost-competitiveness with the matrix-present algorithm. Further work would need to be performed to achieve this. For one, the homotopy used in the numerical investigations in this paper is designed to be efficient when used with implicit solvers and may not be ideal for the matrix-free case. Additionally, we do not employ any of the convergence acceleration techniques, such as multigrid [16], typically applied to explicit time-marching methods. The three reasons for developing this algorithm are:

1. To reduce the labour intensity involved in investigating the feasibility of candidate homotopies for homotopy continuation;
2. To develop a better understanding and further characterize monolithic homotopy continuation algorithms;
3. To continue our study of the stability and accuracy of monolithic homotopy continuation methods.

The numerical studies in this paper have targeted the third point. We have seen through numerical analysis that the inexact update which results from solving the linear system inexactly can result in instabilities in the matrix-present monolithic algorithm [3]. However, since the algorithm normally converges in a relatively small number of iterations, the instabilities do not often become problematic. The effect can be studied by adjusting the linear solver tolerance; however, this affects performance of the algorithm in other ways (it affects the accuracy with which the homotopy "curve" is being traced) and so it is difficult to study the instabilities directly. The matrix-free monolithic algorithm has been a vehicle for the direct study of the stability of the monolithic homotopy algorithms in general and may guide future research efforts in the development of this class of algorithm.

2 Homotopy Continuation

A homotopy is a continuous deformation between two points. The convention that we adopt is the so-called *convex homotopy* [1], which is defined as the (presumably) continuous solution $\mathbf{q}(\lambda)$ to

$$\mathcal{H}(\mathbf{q},\lambda) = (1-\lambda)\,\mathcal{R}(\mathbf{q}) + \lambda\mathcal{G}(\mathbf{q}) = \mathbf{0}, \tag{1}$$

$$\mathcal{H} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N,\ \mathcal{G} : \mathbb{R}^N \to \mathbb{R}^N,\ \mathcal{R} : \mathbb{R}^N \to \mathbb{R}^N,\ \lambda \in \mathbb{R}.$$

If $\mathcal{R}(\mathbf{q}) = \mathbf{0}$ and $\mathcal{G}(\mathbf{q}) = \mathbf{0}$ both have unique solutions and are continuous, then the deformation, if it exists, is continuous and is therefore a homotopy. If additionally $\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}, \lambda)$ is invertible for all $\lambda \in \{[0, 1]\}$, then the homotopy exists and is said to be regular. The regularity condition also ensures that there are no bifurcation points in the interval $\lambda \in \{[0, 1]\}$ [1].

Interpreting the homotopy as a curve existing in $\mathbb{R}^N$, a continuation method, called convex homotopy continuation, can be developed from this homotopy by discretizing in λ to form a sequence of nonlinear equations:

$$\mathcal{H}(\mathbf{q}, \lambda_k) = (1 - \lambda_k)\mathcal{R}(\mathbf{q}) + \lambda_k \mathcal{G}(\mathbf{q}) = \mathbf{0}, \tag{2}$$

$$k \in [0, m],\ \lambda_k \in \mathbb{R},\ \lambda_0 = 1,\ \lambda_m = 0,\ \lambda_{k+1} < \lambda_k,$$

$$\mathcal{H} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N,\ \mathcal{G} : \mathbb{R}^N \to \mathbb{R}^N,\ \mathcal{R} : \mathbb{R}^N \to \mathbb{R}^N.$$

Approximately solving $\mathcal{H}(\mathbf{q}, \lambda) = \mathbf{0}$ for sequentially increasing k can be used to obtain an approximate solution to $\mathcal{R}(\mathbf{q})$. This is only useful if the solution to $\mathcal{G}(\mathbf{q})$ is known or easily obtainable. Previously, Brown and Zingg [2] have presented a second-difference numerical dissipation operator with far-field pseudo-boundary conditions, based on the earlier work of Hicken and Zingg [12,14], suited to this purpose. This operator is used exclusively for the studies presented in this paper.

3 Dynamic Inversion

A matrix-present version of the monolithic homotopy continuation algorithm class is given by Brown and Zingg [3] and is based on the work of Getz and Marsden [9, 10] and Getz [8]. The method is based on the dynamic inversion principle. A formal definition of the dynamic inverse is given by Getz and Marsden [10] and is adapted to homotopy continuation by Brown and Zingg [3]. The definition is repeated here.

Definition 1 Let $\mathbf{q}_s(\lambda)$ be a regular homotopy defined implicitly by $\mathcal{H}(\mathbf{q}, \lambda) = \mathbf{0}$, $\mathcal{H} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$, $(\mathbf{q}, \lambda) \mapsto \mathcal{H}(\mathbf{q}, \lambda)$, $\mathbf{q} \equiv \mathbf{q}_s(\lambda) + \Delta\mathbf{q}$. Let $\mathcal{H}^* : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$, $(\mathbf{w}, \lambda) \mapsto \mathcal{H}^*(\mathbf{w}, \lambda)$, $\mathbf{w} \equiv \mathcal{H}(\mathbf{q}_s + \Delta\mathbf{q}, \lambda)$ be continuous in λ and Lipschitz continuous on the ball $\mathcal{B}_r = \left\{ \Delta\mathbf{q} \in \mathbb{R}^N \mid \|\Delta\mathbf{q}\| \leq r \right\}$, $r > 0$. Then $\mathcal{H}^*$ is called a *forward dynamic inverse* of $\mathcal{H}$ on $\mathcal{B}_r$ if there exists fixed $\beta \in \mathbb{R}$, $0 < \beta < \infty$, such that

$$\Delta\mathbf{q}^T \mathcal{H}^* \mathcal{H}(\mathbf{q}, \lambda) \geq \beta \|\Delta\mathbf{q}\|^2 \tag{3}$$

for all $\Delta\mathbf{q} \in \mathcal{B}_r$. Similarly, $\mathcal{H}^*$ is called a *reverse mode dynamic inverse* of $\mathcal{H}$ on $\mathcal{B}_r$ if there exists fixed $\beta \in \mathbb{R}$, $0 < \beta < \infty$, such that

$$\Delta\mathbf{q}^T \mathcal{H}^* \mathcal{H}(\mathbf{q}, \lambda) \leq -\beta \|\Delta\mathbf{q}\|^2 \tag{4}$$

for all $\Delta\mathbf{q} \in \mathcal{B}_r$.

Note that we have used the short-hand notation $\mathcal{H}^*\mathcal{H}(\mathbf{q}, \lambda)$ in place of the more precise nut more cumbersome notation $\mathcal{H}^*(\mathcal{H}(\mathbf{q}, \lambda), \lambda)$.

Remark 1 If $\mathcal{H}^*$ is a (forward or reverse mode) dynamic inverse of $\mathcal{H}$ with constant β, then for any $\gamma \in \mathbb{R}$, $\gamma > 0$, $\gamma\mathcal{H}^*$ is a (forward or reverse mode) dynamic inverse of $\mathcal{H}$ with constant $\gamma\beta$.

Remark 2 If $\mathcal{H}^*$ is a forward dynamic inverse of $\mathcal{H}$, then $-\mathcal{H}^*$ is a reverse mode dynamic inverse of $\mathcal{H}$.

If a dynamic inverse can be found then we have the following Theorem [3,8].

Theorem 1 *Let $\mathbf{q}_s(\lambda)$ be a regular homotopy defined implicitly by $\mathcal{H}(\mathbf{q},\lambda) = \mathbf{0}$. Assume that $\mathcal{H}^* : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$; $(\mathbf{w},\lambda) \mapsto \mathcal{H}^*(\mathbf{w},\lambda)$ is a reverse mode dynamic inverse of $\mathcal{H}(\mathbf{q},\lambda)$ on $\mathcal{B}_r = \left\{ \Delta\mathbf{q} \in \mathbb{R}^N \middle| \, \|\Delta\mathbf{q}\| \leq r \right\}$, $r > 0$, $0 < \beta < \infty$. Let $\mathcal{E} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$; $(\mathbf{q},\lambda) \mapsto \mathcal{E}(\mathbf{q},\lambda)$ be locally Lipschitz in $\mathbf{q}$ and piecewise continuous in λ. Assume that for some fixed $\omega \in (0,\infty)$, $\mathcal{E}(\mathbf{q},\lambda)$ satisfies*

$$-\frac{1}{2}\omega \|\Delta\mathbf{q}\|^2 \leq \Delta\mathbf{q}^T \left[\mathcal{E}(\mathbf{q}_s + \Delta\mathbf{q},\lambda) + \dot{\mathbf{q}}_s(\lambda)\right] \leq \frac{1}{2}\omega \|\Delta\mathbf{q}\|^2 \tag{5}$$

for all $\Delta\mathbf{q} \in \mathcal{B}_r$. Let $\mathbf{q}'_s(\lambda)$ denote the solution to the system

$$-\dot{\mathbf{q}} = \gamma\mathcal{H}^*\mathcal{H}(\mathbf{q},\lambda) + \mathcal{E}(\mathbf{q},\lambda), \tag{6}$$

where $\gamma \in \mathbb{R}$, $\gamma > 0$ (see Remark 1). Consider now some $\lambda_k \in \mathbb{R}$ such that

$$\mathbf{q}_s(\lambda_k) - \mathbf{q}'_s(\lambda_k) \in \mathcal{B}_r. \tag{7}$$

Then

$$\left\|\mathbf{q}'_s(\lambda) - \mathbf{q}_s(\lambda)\right\| \leq \left\|\mathbf{q}'_s(\lambda_k) - \mathbf{q}_s(\lambda_k)\right\| \mathrm{e}^{-(\gamma\beta-\omega)|\lambda_k-\lambda|} \tag{8}$$

for all $\lambda < \lambda_k$.

The proof of Theorem 1 is given by Brown and Zingg [3]. As observed from equation (8), convergence of the ODE (6) depends on the values of γ, which is a free parameter, β, which depends on the dynamic inverse, and ω, which depends on $\mathcal{E}$. Brown and Zingg [3] showed, based on Getz and Marsden [9], that $\mathcal{H}^* = -\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)$ is a reverse-mode dynamic inverse of $\mathcal{H}(\mathbf{q},\lambda)$. Clearly, $\mathcal{E} = -\dot{\mathbf{q}}$ will give $\omega = 0$, which is the optimal value of ω for convergence. However, calculating $\dot{\mathbf{q}}$ accurately requires the formation and inversion of a matrix [3]. In this paper, matrix-free constructions of $\mathcal{H}^*$ and $\mathcal{E}$ are investigated.

4 Matrix-Free Dynamic Inverse

Theorem 2 *Let $\mathcal{H} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$, $(\mathbf{q},\lambda) \mapsto \mathcal{H}(\mathbf{q},\lambda)$ be a regular homotopy which is Lipschitz continuous for all $0 \leq \lambda \leq 1$. Let $\mathbf{q}'_s = \mathbf{q}_s + \Delta\mathbf{q}$, $\Delta\mathbf{q} \in \mathcal{B}_r \subset \mathbb{R}^N$, $r > 0$ be a point near the curve. Then if $\mathcal{H}^* : R^N \times \mathbb{R} \to \mathbb{R}^N$ is a linear operator, and $\mathcal{H}^*\nabla_{\mathbf{q}}\mathcal{H}\left(\mathbf{q}'_s,\lambda\right)$ is positive-definite in $\mathcal{B}_r$, then $\mathcal{H}^*$ is a dynamic inverse of $\mathcal{H}$ on $\mathcal{B}_r$.*

Proof The residual $\mathcal{H}\left(\mathbf{q}'_s,\lambda\right)$ can be represented locally by taking the Taylor expansion at $\mathbf{q}_s$:

$$\mathcal{H}(\mathbf{q}_s + \Delta\mathbf{q},\lambda) = \mathcal{H}(\mathbf{q}_s,\lambda) + \nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}_s,\lambda)\Delta\mathbf{q} + \mathcal{O}\left(\|\Delta\mathbf{q}\|^2\right). \tag{9}$$

Similarly, expanding $\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)$ at $\mathbf{q}_s$ gives

$$\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}_s,\lambda) = \nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}_s+\Delta\mathbf{q},\lambda) + \mathcal{O}(\|\Delta\mathbf{q}\|). \tag{10}$$

Using equation (10) and $\mathcal{H}(\mathbf{q}_s,\lambda)=\mathbf{0}$, equation (9) becomes

$$\mathcal{H}(\mathbf{q}_s+\Delta\mathbf{q},\lambda) = \nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}_s+\Delta\mathbf{q},\lambda)\,\Delta\mathbf{q} + \mathcal{O}\left(\|\Delta\mathbf{q}\|^2\right). \tag{11}$$

Assuming that $\mathcal{H}^*$ is a linear operator, applying $\Delta\mathbf{q}^T\mathcal{H}^*$ to both sides of equation (11) gives

$$\Delta\mathbf{q}^T\mathcal{H}^*\mathcal{H}(\mathbf{q}_s+\Delta\mathbf{q},\lambda) = \Delta\mathbf{q}^T\mathcal{H}^*\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}_s+\Delta\mathbf{q},\lambda)\,\Delta\mathbf{q} + \mathcal{O}\left(\|\Delta\mathbf{q}\|^3\right). \tag{12}$$

If $\mathcal{H}^*\nabla_{\mathbf{q}}\mathcal{H}\left(\mathbf{q}_s',\lambda\right)$ is positive-definite, then

$$\Delta\mathbf{q}^T\mathcal{H}^*\nabla_{\mathbf{q}}\mathcal{H}\left(\mathbf{q}_s',\lambda\right)\Delta\mathbf{q} > 0, \tag{13}$$

so there exists $0<\beta<1$ such that

$$\Delta\mathbf{q}^T\mathcal{H}^*\nabla_{\mathbf{q}}\mathcal{H}\left(\mathbf{q}_s',\lambda\right)\Delta\mathbf{q} \geq 2\beta\|\Delta\mathbf{q}\|^2. \tag{14}$$

For sufficiently small $r>0$, there exists fixed $0<\beta<1$ such that the $\mathcal{O}\left(\|\Delta\mathbf{q}\|^3\right)$ terms are upper-bounded by $\beta\|\Delta\mathbf{q}\|^2$ for all $\Delta\mathbf{q}\in\mathcal{B}_r$. Thus, considering equations (12) and (14), as long as $\mathcal{H}^*\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)$ is positive-definite, then for sufficiently small $r>0$ there exists fixed $0<\beta<1$ such that

$$\Delta\mathbf{q}^T\mathcal{H}^*\mathcal{H}(\mathbf{q}_s+\Delta\mathbf{q},\lambda) \geq 2\beta\|\Delta\mathbf{q}\|^2 - \beta\|\Delta\mathbf{q}\|^2 = \beta\|\Delta\mathbf{q}\|^2 \tag{15}$$

for all $\Delta\mathbf{q}\in\mathcal{B}_r$.

By Theorem 2, $[\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)]^T$ is clearly a dynamic inverse of $\mathcal{H}(\mathbf{q},\lambda)$, regardless of whether or not $\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)$ is positive-definite. This formulation would avoid solving any linear systems of equations and could be useful for solving systems of equations whose Jacobian is indefinite. However, the method would not be matrix-free and is not the focus of this paper.

Consider as a candidate for $\mathcal{H}^*$ a diagonal positive-definite matrix $\mathcal{T}$. If the entries of $\mathcal{T}$ are all identical and positive, then, assuming that $\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)$ is positive-definite, $\mathcal{T}\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)$ is also positive-definite, so $\mathcal{T}$ is a dynamic inverse of $\mathcal{H}(\mathbf{q},\lambda)$ if $\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q},\lambda)$ is positive-definite.

Remark 3 By Remark 2, if $\mathcal{T}$ is a forward dynamic inverse of $\mathcal{H}$ then $-\mathcal{T}$ is a reverse-mode dynamic inverse of $\mathcal{H}$.

Ideally, we would like to construct $\mathcal{E}$ such that equation (5) is satisfied. This may be accomplished by setting

$$\mathcal{E}(\mathbf{q}_s,\lambda) = -\dot{\mathbf{q}}_s. \tag{16}$$

However, performing this calculation exactly requires the solution to a linear system of equations [3] and so the resulting algorithm would not be matrix-free.

Since the objective in this paper is to construct a matrix-free algorithm, an approximation to the tangent vector based on backwards-differencing can be used:

$$\mathcal{E}_k = \frac{1}{\lambda_k - \lambda_{k-1}} \left(\mathbf{q}_k - \mathbf{q}_{k-1} \right). \tag{17}$$

Applying a diagonal matrix $\mathcal{T}$ as the dynamic inverse with Remark 3 and constructing $\mathcal{E}$ according to equation (17) gives the discrete update formula

$$\mathbf{q}_{k+1} = \mathbf{q}_k + (\lambda_{k+1} - \lambda_k) \left[\gamma \mathcal{T} \mathcal{H} \left(\mathbf{q}_k, \lambda_k \right) + \frac{1}{\lambda_k - \lambda_{k-1}} \left(\mathbf{q}_k - \mathbf{q}_{k-1} \right) \right]. \tag{18}$$

The use of backwards differencing to approximate the tangent vector violates condition (5), the consequence being that stability of the algorithm is no longer assured. We demonstrate in this paper, however, that even though the algorithm can become unstable as a result of this approximation, it is possible to stabilize the algorithm using an explicit filter.

5 Two-Stage Formulation

Since the predictor and corrector portion of the update are independent, they can be separated. Unlike the matrix-present formulation, applying the predictor and corrector components separately incurs minimal additional computational cost. The benefit of this two-stage formulation is that information from the corrector iteration can be used to improve on the quality of the predictor.

Separating the corrector and predictor portion of the update, the two-stage formulation with Euler corrector and secant predictor is given by:

$$\begin{cases} \mathbf{q}_{k+\frac{1}{2}} = \mathbf{q}_k + \left(\lambda_{k+\frac{1}{2}} - \lambda_k \right) \gamma_k \mathcal{T}_k \mathcal{H} \left(\mathbf{q}_k, \lambda_k \right), \\ \mathbf{q}_{k+1} = \mathbf{q}_{k+\frac{1}{2}} + \frac{\lambda_{k+1} - \lambda_k}{\lambda_k - \lambda_{k-1}} \left(\mathbf{q}_{k+\frac{1}{2}} - \mathbf{q}_{k-\frac{1}{2}} \right). \end{cases} \tag{19}$$

This equation resembles a dual time marching method [21] where a single dual time step iteration is applied as corrector (the upper equation) using explicit Euler integration. By this analogy, it is a logical design choice to set $\gamma = \gamma_{\text{ref}} / \left| \lambda_{k+\frac{1}{2}} - \lambda_k \right|$ where $\gamma_{\text{ref}} \in \mathbb{R}$ is a constant user-defined parameter. This eliminates dependence on the step size $\Delta\lambda$ from the corrector equation.

First-order backwards differencing is not the only estimate that can be used for the predictor. It can be useful to consider performance when using other predictors as well. A "rank 1" predictor refers to a secant predictor, a "rank 0" predictor refers to the case of $\mathcal{E} = 0$, and a "rank $\frac{1}{2}$" predictor refers to the case where the predictor alternates between the rank 0 and rank 1 predictor at successive iterations. A "rank 2" predictor indicates the use of a second degree Lagrange polynomial to extrapolate $\mathbf{q}_{k+1}$. Lagrange interpolation/extrapolation is covered by many textbooks, one example of which is Lomax et al. [18]. The formula for the point predicted using a second degree Lagrange polynomial extrapolation in the context of homotopy continuation is given below:

$$\mathbf{q}_{k+1} = \mathbf{q}_{k+\frac{1}{2}} l_k + \mathbf{q}_{k-\frac{1}{2}} l_{k-1} + \mathbf{q}_{k-\frac{3}{2}} l_{k-2}, \tag{20}$$

$$l_k = \frac{(\lambda_{k+1} - \lambda_{k-2})(\lambda_{k+1} - \lambda_{k-1})}{(\lambda_k - \lambda_{k-2})(\lambda_k - \lambda_{k-1})},$$
$$l_{k-1} = \frac{(\lambda_{k+1} - \lambda_{k-2})(\lambda_{k+1} - \lambda_k)}{(\lambda_{k-1} - \lambda_{k-2})(\lambda_{k-1} - \lambda_k)},$$
$$l_{k-2} = \frac{(\lambda_{k+1} - \lambda_{k-1})(\lambda_{k+1} - \lambda_k)}{(\lambda_{k-2} - \lambda_{k-1})(\lambda_{k-2} - \lambda_k)},$$
$$l_k,\, l_{k-1},\, l_{k-2} \in \mathbb{R}.$$

Since the predictor portion of the update can use information from previous iterative steps, a predictor-corrector algorithm is used to initialize the first interior traversing point. A rank 0 predictor is used from the first point and the corrector problem is solved at the second point by explicit time marching until a suitable user-defined tolerance is reached.

As we will see in Section 7, the algorithm presented in this section is unstable when using a rank $\frac{1}{2}$, 1, or 2 predictor. The expansive class of time-integration methods known as *general linear methods* [15] can be used to construct stable and/or accurate time-integration methods. In Section 6 we apply a general linear method which uses multiple stages based on current information to a rank 0 predictor algorithm. However, because the ODE (6) being integrated is redefined at each iteration of the algorithm, it is not immediately clear how or if a multi-step general linear method, using previous information, is justifiable, and it is not clear how to apply such methods to a formulation which includes a non-zero predictor component.

In lieu of attempting to apply a general linear method to our two-stage matrix-free monolithic homotopy formula, as given by equation (18), we attempt to stabilize the algorithm using an explicit filter to damp out high-frequency oscillations appearing in the λ domain. The most successful explicit filter that we have applied is a λ-direction explicit kernel smoother. The filter uses the Nadaraya-Watson [19, 24] kernel weighted average, applied at the k-th iteration. The general formula for a smoothed point $\tilde{\mathbf{q}}$ at a given parameter value λ^* is

$$\tilde{\mathbf{q}}(\lambda^*) = \frac{\sum_{i=k+1-p}^{k+1} K_b(\lambda^*, \lambda_i)\, \mathbf{q}(\lambda_i)}{\sum_{i=k+1-p}^{k+1} K_b(\lambda^*, \lambda_i)}, \tag{21}$$

with Gaussian kernel function given by

$$K_b(\lambda^*, \lambda_i) = \exp\left(-\frac{(\lambda^* - \lambda_i)^2}{2b^2}\right), \tag{22}$$

where $K_b : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $b \in \mathbb{R}$, $b > 0$. The smoothing is applied to the updated point $\mathbf{q}_{k+1}$ at $\lambda^* = \lambda_{k+1}$ after the update (18) is applied making use of p previous stages. The resulting update takes the form

$$\mathbf{q}_{k+1} \leftarrow \frac{\mathbf{q}_{k+1} + \sum_{k-p+1}^{k} K_b(\lambda_{k+1}, \lambda_i)\, \mathbf{q}_{i+\frac{1}{2}}}{1 + \sum_{k-p+1}^{k} K_b(\lambda_{k+1}, \lambda_i)}. \tag{23}$$

We observe that the filter-augmented algorithm resembles a general linear method with coefficients depending on the parameter b, where $b = |\Delta\lambda_k|\, b_{\text{ref}}$, and

Algorithm 1: The two-stage algorithm with explicit Gaussian kernel filter

Initialize: Set $\lambda = 1$ and solve $\mathcal{G}(\mathbf{q}) = \mathbf{0}$ if necessary. Take a step $\lambda \leftarrow \lambda + \Delta\lambda$ and solve $\mathcal{H}(\mathbf{q}, \lambda) = \mathbf{0}$ at the updated value of λ.

Iterate: while $\lambda > 0$ **do**

Calculate $\mathcal{H}(\mathbf{q}, \lambda)$

Set the diagonal matrix $\mathcal{T}$

Take the corrector portion of the update: $\mathbf{q}_{k+\frac{1}{2}} \leftarrow \mathbf{q}_k + \gamma_{\text{ref}} \mathcal{T}\mathcal{H}$

Step-length adaptation can be applied at this stage if desired

Perform a predictor step: $\mathbf{q}_{k+1} \leftarrow \mathbf{q}_{k+\frac{1}{2}} + \frac{\lambda_{k+1}-\lambda_k}{\lambda_k-\lambda_{k-1}} \left(\mathbf{q}_{k+\frac{1}{2}} - \mathbf{q}_{k-\frac{1}{2}}\right)$

Smooth the update using the Gaussian kernel filter:

$\mathbf{q}_{k+1} \leftarrow \frac{\mathbf{q}_{k+1} + \sum_{k-p+1}^{k} K_b(\lambda_{k+1}, \lambda_i) \mathbf{q}_{i+\frac{1}{2}}}{1 + \sum_{k-p+1}^{k} K_b(\lambda_{k+1}, \lambda_i)}$

end

$b_{\text{ref}} \in \mathbb{R}$, $b_{\text{ref}} > 0$ is a user input. Larger values of b_{ref} will result in more smoothing, the effect of which is increased stability at the cost of reduced curve-tracing accuracy. In this study, values in the range $0.5 \leq b_{\text{ref}} \leq 0.8$ were found to be suitable. For values of b_{ref} in this range, increasing the value of p beyond 2 has negligible effect. A pseudo-code of the two-stage algorithm with explicit filter based on equation (18) is shown in Algorithm 1.

6 Single-Stage Predictor-Free Formulation

The stability concern in equation (18) comes from the fact that $\mathcal{E}(\mathbf{q}, \lambda)$ is approximated using a backwards difference formula. This invalidates condition (5) which is a condition for convergence of the continuous ODE (6) to the curve. No continuous analogue can be made for the backwards-difference formulation, making analysis difficult. We show through numerical testing in Section 7 that the method is unstable.

Let us turn our attention to the case where $\mathcal{E} = \mathbf{0}$. By the Cauchy-Schwarz inequality,

$$\Delta\mathbf{q}^T \left[\mathcal{E}(\mathbf{q}_s + \Delta\mathbf{q}, \lambda) + \dot{\mathbf{q}}_s(\lambda)\right] = \left|\Delta\mathbf{q}^T \dot{\mathbf{q}}_s\right| \leq \|\Delta\mathbf{q}\| \|\dot{\mathbf{q}}_s\|. \tag{24}$$

Setting $\omega = 2\|\dot{\mathbf{q}}_s\|$, we get

$$-\frac{1}{2}\omega \|\Delta\mathbf{q}\| \leq \Delta\mathbf{q}^T \left[\mathcal{E}(\mathbf{q}_s + \Delta\mathbf{q}, \lambda) + \dot{\mathbf{q}}_s(\lambda)\right] \leq \frac{1}{2}\omega \|\Delta\mathbf{q}\|, \tag{25}$$

which is insufficient to show unconditional convergence of the ODE (6) to the curve. However, it is still possible to show conditional convergence for the discrete case by directly analyzing the ODE

$$-\dot{\mathbf{q}} = \gamma \mathcal{H}^* \mathcal{H}(\mathbf{q}, \lambda). \tag{26}$$

Let $\mathbf{z} = \mathbf{q}'_s - \mathbf{q}_s$, $\mathbf{z} \in \mathbb{R}^N$, where $\mathbf{q}'_s$ is the solution to the ODE (6) and $\mathbf{q}_s$ is the solution to the homotopy equation (2). Let $V(\mathbf{z}(\lambda)) = \frac{1}{2}\|\mathbf{z}(\lambda)\|^2$, $V : \mathbb{R}^N \to \mathbb{R}$.

Then we have:

$$\begin{aligned}\dot{\mathbf{z}} &= \dot{\mathbf{q}}_s' - \dot{\mathbf{q}}_s \\ &= -\gamma \mathcal{H}^* \mathcal{H}\left(\mathbf{q}_s', \lambda\right) - \dot{\mathbf{q}}_s \\ &= -\gamma \mathcal{H}^* \mathcal{H}\left(\mathbf{q}_s + \mathbf{z}, \lambda\right) - \dot{\mathbf{q}}_s. \end{aligned} \tag{27}$$

Differentiating $V(\mathbf{z}(\lambda))$ with respect to $-\lambda$ gives:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}(-\lambda)} V(\mathbf{z}(\lambda)) &= -\mathbf{z}^T \dot{\mathbf{z}} \\ &= \mathbf{z}^T \left[\gamma \mathcal{H}^* \left(\mathcal{H}\left(\mathbf{q}_s + \mathbf{z}, \lambda\right)\right) + \dot{\mathbf{q}}_s\right] \\ &\leq -\gamma\beta \left\|\mathbf{z}\right\|^2 + \mathbf{z}^T \dot{\mathbf{q}}_s. \end{aligned} \tag{28}$$

Since the term $\mathbf{z}^T \dot{\mathbf{q}}_s$ is independent of γ and $|\Delta\lambda|$, it is possible to choose γ sufficiently large such that $\frac{\mathrm{d}}{\mathrm{d}(-\lambda)} V(\mathbf{z}(\lambda)) \leq 0$ and hence the continuous ODE (26) is convergent for sufficiently large γ.

Continuing the analysis, let $\omega = \left\|\dot{\mathbf{q}}_s\right\|$. Then, applying the Cauchy-Schwarz inequality again,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}(-\lambda)} V(\mathbf{z}(\lambda)) &\leq -\gamma\beta \left\|\mathbf{z}\right\|^2 + \omega \left\|\mathbf{z}\right\| \\ &= -2\gamma\beta V(\mathbf{z}(\lambda)) + \omega\sqrt{2V(\mathbf{z}(\lambda))}. \end{aligned} \tag{29}$$

Solving this ODE and using the Comparison Theorem [11] gives

$$\begin{aligned}V(\mathbf{z}(\lambda)) \leq &\left[\left(\sqrt{V(\mathbf{z}(\lambda_k))} - \frac{\omega}{\sqrt{2}\gamma\beta}\right) \mathrm{e}^{-\gamma\beta(\lambda_k - \lambda)}\right]^2 \\ &+ \frac{\sqrt{2}\omega}{\gamma\beta}\left(\sqrt{V(\mathbf{z}(\lambda_k))} - \frac{\omega}{\sqrt{2}\gamma\beta}\right) \mathrm{e}^{-\gamma\beta(\lambda_k - \lambda)} + \left(\frac{\omega}{\sqrt{2}\gamma\beta}\right)^2 \end{aligned} \tag{30}$$

where $\lambda < \lambda_k$. The expression on the right-hand side of the equation can be factored to give

$$V(\mathbf{z}(\lambda)) \leq \left[\left(\sqrt{V(\mathbf{z}(\lambda_k))} - \frac{\omega}{\sqrt{2}\gamma\beta}\right) \mathrm{e}^{-\gamma\beta(\lambda_k - \lambda)} + \frac{\omega}{\sqrt{2}\gamma\beta}\right]^2. \tag{31}$$

In order for the ODE (26) to converge, $V(\mathbf{z}(\lambda))$ must decrease to 0 as $\lambda_k - \lambda$ increases, which would indicate that the error vanishes in the limit. Since $\gamma > 0$ and $\beta > 0$ by assumption, the exponential term clearly satisfies this property. However, the constant term $\omega / \left[\sqrt{2}\gamma\beta\right]$ prevents unconditional convergence. There are only two ways in which $V(\mathbf{z}(\lambda))$ can vanish in the limit of $\lambda \to -\infty$:

1. $\omega = 0$, in which case the curve is a stationary point and the class of algorithms becomes equivalent to pseudo-transient continuation;
2. γ is taken arbitrarily large.

Though γ cannot in practice be taken arbitrarily large, γ can indeed be chosen sufficiently large such that any desired error threshold can be achieved. This is guaranteed by the fact that the right-hand side expression of equation (31), as a function of γ, is continuous for all $\gamma > 0$ in the interval $\{(0, \infty)\}$.

If, given some $\mathbf{q}'_{s,k}$, it is desired that the condition

$$V\left(\mathbf{z}\left(\lambda_{k+1}\right)\right) \leq e_V \tag{32}$$

should be met for some specified $e_V \in \mathbb{R}$, then this condition will be met for any λ_{k+1} satisfying

$$\left[\left(\sqrt{V\left(\mathbf{z}\left(\lambda_k\right)\right)} - \frac{\omega}{\sqrt{2}\gamma\beta}\right) \mathrm{e}^{-\gamma\beta(\lambda_k - \lambda_{k+1})} + \frac{\omega}{\sqrt{2}\gamma\beta}\right]^2 \leq e_V. \tag{33}$$

Solving this equation for $\lambda_k - \lambda_{k+1}$ gives

$$\lambda_k - \lambda_{k+1} \leq \frac{1}{\beta\gamma} \ln\left(\frac{\sqrt{2V\left(\mathbf{z}\left(\lambda_k\right)\right)}\beta\gamma - \omega}{\sqrt{2e_V}\beta\gamma - \omega}\right), \tag{34}$$

which is the maximum distance in λ which can be applied to meet condition (32). Equation (34) indicates that existence of such a λ_{k+1} is assured, but only for sufficiently large γ. When integrating this ODE numerically, it may be necessary to take a smaller integration step, depending on the numerical integration scheme, to achieve the accuracy target.

The single-stage algorithm is given explicitly by the equation

$$\dot{\mathbf{q}} = -\gamma\mathcal{T}\mathcal{H}\left(\mathbf{q}, \lambda\right). \tag{35}$$

We have now shown that this equation is stable for sufficiently large γ. Since $|\Delta\lambda|$ is taken inversely proportional to γ, the condition of requiring γ to be sufficiently large is equivalent to requiring that $|\Delta\lambda|$ be taken sufficiently small. It is possible to numerically integrate equation (35) with either an Euler update or a more stable update. In this study, a multi-stage scheme with four stages, which we abbreviate as MS4, is considered as an alternative to the explicit Euler update. Multi-stage methods are discussed in more detail by Pulliam and Zingg [21]. The update is given here in the context of equation (35):

$$\begin{cases} \lambda_{n+m/q} = \lambda_n + \alpha_m \Delta\lambda, \\ \mathbf{q}_{n+m/q} = \mathbf{q}_n - \alpha_m \gamma_{\mathrm{ref}} \mathcal{T}\mathcal{H}\left(\mathbf{q}_{n+(m-1)/q}, \lambda_{n+(m-1)/q}\right), \end{cases} \tag{36}$$

$$m = 1, \ldots, q,$$

with $q = 4$ and

$$\left(\alpha_1, \alpha_2, \alpha_3, \alpha_4\right) = \left(\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\right).$$

A pseudo-code of the single-stage algorithm based on equation (35) is presented as Algorithm 2.

Algorithm 2: Single-stage algorithm

Initialize: Set $\lambda = 1$ and solve $\mathcal{G}(\mathbf{q}) = \mathbf{0}$ if necessary
Iterate: while $\lambda > 0$ **do**
  Calculate $\mathcal{H}(\mathbf{q}, \lambda)$
  Set the diagonal matrix $\mathcal{T}$
  Use equation (35) to update $\mathbf{q}$ applying explicit Euler or MS4 (36)
  Step-length adaptation can be applied at this stage if desired
end

7 Performance Investigation for the Equations Governing Compressible Inviscid Fluid Flow

Test cases were performed in order to investigate the functionality of the algorithms. Since the algorithm variants are intended as continuation algorithms, the objective is not to solve for the solution accurately but to obtain a sufficiently accurate approximation so that a rapidly-convergent local root-finding method such as Newton's method can be applied successfully.

The algorithms are applied to the parallel flow solver for inviscid compressible aerodynamic flows developed by Hicken and Zingg [13]. The Euler equations are discretized using the SBP-SAT [4,5,7,17] approach, which uses Summation-By-Parts (SBP) operators to represent the discrete derivatives and Simultaneous Approximation Terms (SATs) to enforce the boundary conditions and couple the flow equations at block interfaces. Though the flow solver is intended as a Newton-Krylov-Schur three-dimensional flow solver, and has been extended by Osusky and Zingg [20] to also handle viscous flows, we do not employ any elements of the Newton-Krylov-Schur methodology and have found two-dimensional inviscid flows sufficient for the studies in this paper.

The homotopy studied is the convex homotopy given by equation (2), where the second-difference numerical dissipation operator with far-field pseudo-boundary conditions of Brown and Zingg [2] is used as the homotopy system. The specific expression assigned to the diagonal elements of $\mathcal{T}$ for these studies is

$$\mathcal{T}_{[i]} = \frac{\mathcal{J}_{[i]}}{1 + \mathcal{J}_{[i]}^{\frac{1}{D}}}, \tag{37}$$

where $\mathcal{J}$ is the metric Jacobian resulting from the spatial coordinate transformation [13], and D is the number of spatial dimensions (equal to 2 for these studies). Equation (37) is based on the local time step formula in [18], which is derived based on the advection equation, and it should be possible to develop expressions better suited to specific homotopies. However, optimizing performance is not critical for the studies that we are interested in and so we use equation (37) without modification. Though the connection to local time stepping is loose, we have found it to be far more effective in practice than using either $\mathcal{J}^{-1}$ on the diagonals or the identity matrix as $\mathcal{T}$.

The test case is two-dimensional flow over the NACA 0012 airfoil. The mesh consists of $15,390$ nodes divided evenly into 18 blocks for parallelization on 18 processors. The flow is inviscid at a Mach number of 0.3 and an angle of attack of 1°. To assess the curve-tracing accuracy, the two-dimensional lift coefficient C_l is calculated along the trajectory mapped by the explicit method and compared to

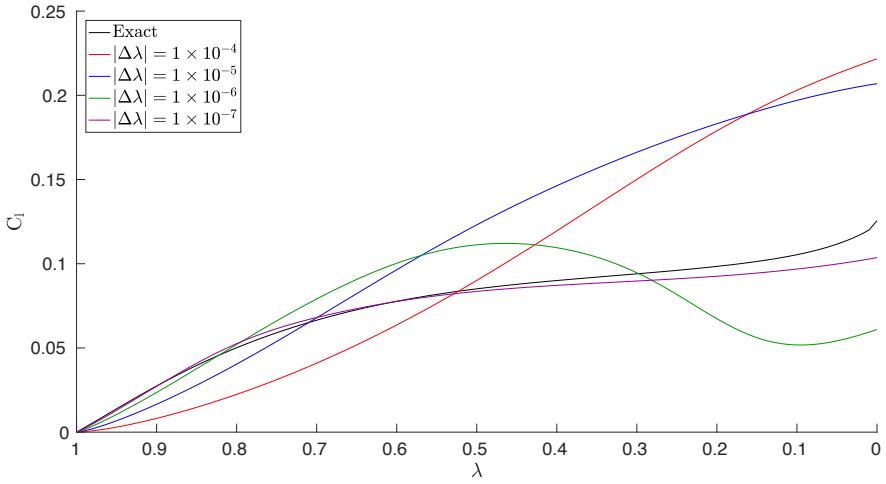


Fig. 1: Two-stage algorithm with Euler corrector, rank 0 predictor; The effect of different $|\Delta\lambda|$ is investigated

an accurate C_l curve generated using the predictor-corrector algorithm presented by Brown and Zingg [2].

7.1 Step Size $\Delta\lambda$

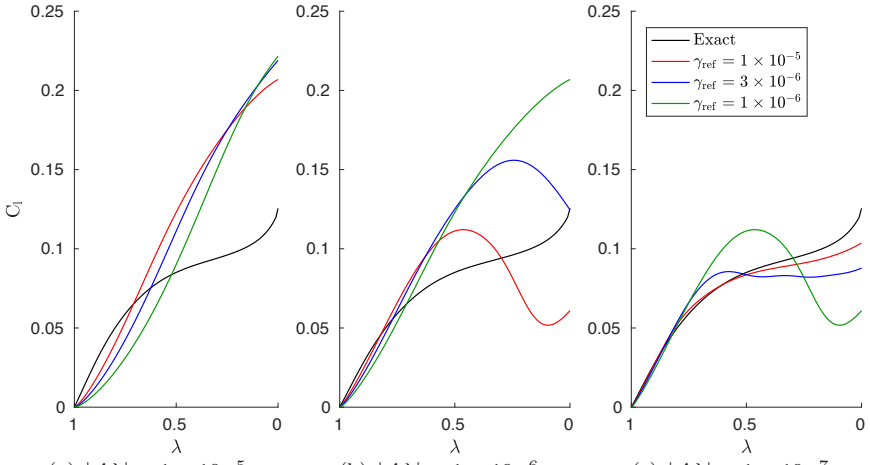
For the first study, the two-stage algorithm is employed with Euler corrector and rank 0 predictor, equivalent to the single-stage algorithm with Euler parameter integration. The accuracy of the curve-tracing algorithm is investigated by varying the constant step size $|\Delta\lambda|$ over the range 1×10^{-4} to 1×10^{-7} with $\gamma_{\text{ref}} = 1\times10^{-5}$. The data collected from the study is shown in Figure 1.

The data show the expected trend that the curve tracing accuracy is improved as the step size $|\Delta\lambda|$ is reduced. The curve is traced very inaccurately at $|\Delta\lambda| = 1 \times 10^{-4}$ and fairly accurately at 1×10^{-7}.

7.2 Parameter γ_{ref}

For the second study, again the two-stage algorithm is employed with Euler corrector and rank 0 predictor. The focus of this study is the effect of γ_{ref} on accuracy and stability. The reference step size γ_{ref} was varied between 1×10^{-5} and 1×10^{-6} while $|\Delta\lambda|$ was varied between 1×10^{-5} and 1×10^{-7}. The data from the study are plotted in Figure 2.

It was found that the algorithm becomes unstable for γ_{ref} values around $\gamma_{\text{ref}} = 3\times10^{-5}$ but demonstrates no symptoms of instability at $\gamma_{\text{ref}} = 1\times10^{-5}$. It is apparent from the plots that the $\gamma_{\text{ref}} = 1 \times 10^{-6}$ cases produce nearly identical



(a) $|\Delta\lambda| = 1 \times 10^{-5}$ (b) $|\Delta\lambda| = 1 \times 10^{-6}$ (c) $|\Delta\lambda| = 1 \times 10^{-7}$

Fig. 2: Two-stage algorithm with Euler corrector, rank 0 predictor; the effect of γ_{ref} is investigated at several values of $|\Delta\lambda|$

curves to the $\gamma_{\text{ref}} = 1 \times 10^{-5}$ case on the next larger $|\Delta\lambda|$. To clarify, taking ten times more integration steps at one tenth the reference value of γ_{ref} has minimal impact on the curve-tracing accuracy. This is an important observation as it indicates that γ_{ref} should be chosen as large as possible so long as the algorithm remains stable.

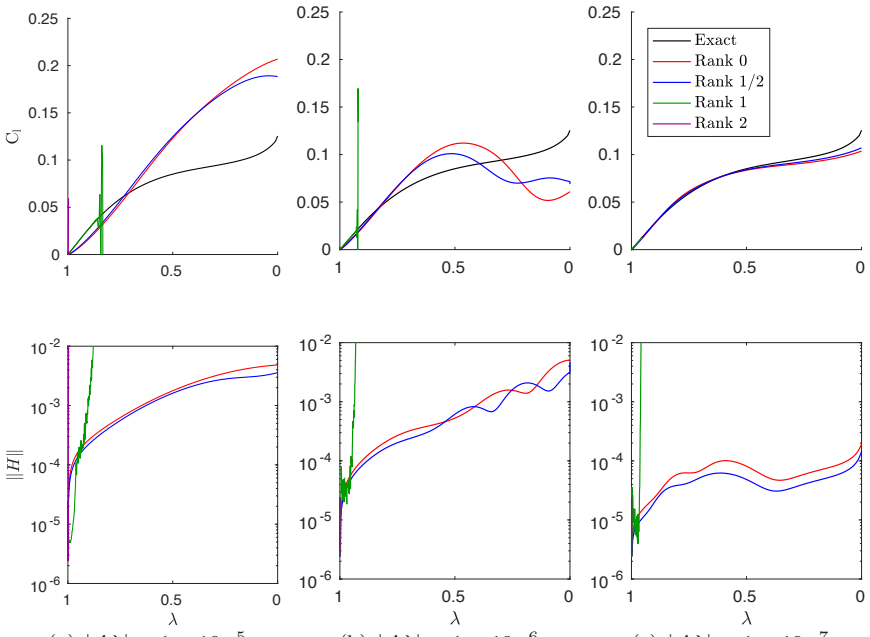
7.3 Predictor Variants for the Two-Stage Algorithm

The third study is an investigation of how the accuracy and stability are affected by predictor choice. The two-stage algorithm was used with Euler corrector and $\gamma_{\text{ref}} = 1 \times 10^{-5}$. The data from this study are plotted in Figure 3.

It is observed that the rank 1 predictor traces the curve much more accurately but results in instability early on. The fact that instability occurs earlier when $|\Delta\lambda|$ is made smaller suggests that the instability is more closely related to the number of iterations performed than the progress in λ. Using the rank $\frac{1}{2}$ predictor alleviates this instability but loses almost all of the additional accuracy incurred by the rank 1 predictor. The rank 2 predictor destabilizes even sooner than the rank 1 predictor. The use of an explicit filter to stabilize this method is investigated in the fifth study.

7.4 Multi-Stage Integration for the Single-Stage Algorithm

For the fourth study, the efficiency gained by augmenting the single-stage algorithm with MS4 parameter integration is studied. The performance investigation



(a) $|\Delta\lambda| = 1 \times 10^{-5}$ (b) $|\Delta\lambda| = 1 \times 10^{-6}$ (c) $|\Delta\lambda| = 1 \times 10^{-7}$

Fig. 3: Two-stage algorithm with Euler corrector, $\gamma_{\text{ref}} = 1 \times 10^{-5}$; the effect of different rank predictors is investigated

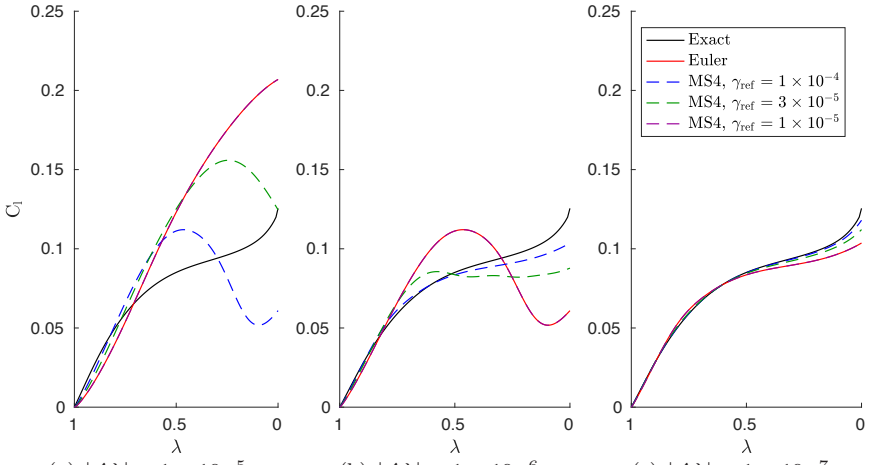
encompasses several γ_{ref} in the range of 1×10^{-4} to 1×10^{-5} for constant step size $|\Delta\lambda|$ ranging from 1×10^{-5} to 1×10^{-7}. The data from this study are shown in Figure 4.

It was found that the MS4 algorithm gives a nearly identical trajectory as the explicit Euler corrector when $\gamma_{\text{ref}} = 1 \times 10^{-5}$ was used for both algorithms. The advantage of MS4 comes from being able to use a larger γ_{ref} values. It was found that γ_{ref} could be increased by an order of magnitude before stability issues were encountered. Since MS4 comes at four times the cost for a given $|\Delta\lambda|$, it is thus about 2.5 times more efficient.

7.5 Filter-Stabilized Two-Stage Algorithm

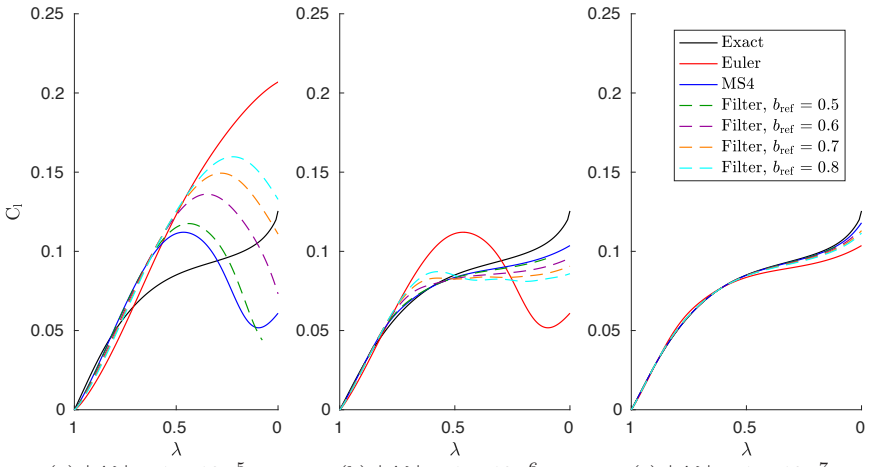
The Gaussian kernel filter-stabilized method with $p = 2$, rank 1 predictor and $\gamma_{\text{ref}} = 1 \times 10^{-5}$ from equation (18) was compared to the explicit Euler method applied to equation (35) with $\gamma_{\text{ref}} = 1 \times 10^{-5}$ and MS4 applied to equation (35) with $\gamma_{\text{ref}} = 1 \times 10^{-4}$. Several values of b were investigated in the range $b = 0.5$ to $b = 0.8$. The data are shown in Figure 5.

The filter-stabilized algorithm, when it completes without becoming unstable, provides significant accuracy improvement over the explicit Euler method but is less accurate than the MS4 method. However, the cost increase of the filter-



(a) $|\Delta\lambda| = 1 \times 10^{-5}$ (b) $|\Delta\lambda| = 1 \times 10^{-6}$ (c) $|\Delta\lambda| = 1 \times 10^{-7}$

Fig. 4: Single-stage algorithm; the effectiveness of MS4 parameter integration is compared to Euler with $\gamma_{\text{ref}} = 1 \times 10^{-5}$; note that the trajectory of the MS4 algorithm with $\gamma_{\text{ref}} = 1 \times 10^{-5}$ is visually indistinguishable from that of the Euler algorithm



(a) $|\Delta\lambda| = 1 \times 10^{-5}$ (b) $|\Delta\lambda| = 1 \times 10^{-6}$ (c) $|\Delta\lambda| = 1 \times 10^{-7}$

Fig. 5: Two-stage algorithm with rank 1 predictor including Gaussian kernel filtering; the filtered algorithm has $\gamma_{\text{ref}} = 1 \times 10^{-5}$ and is compared to MS4 at $\gamma_{\text{ref}} = 1 \times 10^{-4}$ and explicit Euler with rank 0 predictor at $\gamma_{\text{ref}} = 1 \times 10^{-5}$

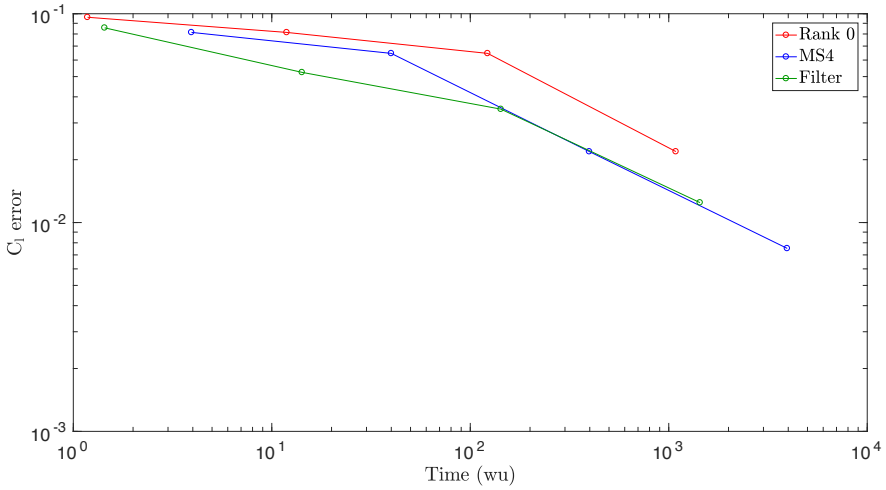


Fig. 6: Globalization error versus work units for the rank 0 method, the two-stage algorithm with MS4, and the single-stage rank 1 filtered algorithm

stabilized algorithm is only about 17% whereas the cost increase of the MS4 algorithm is approximately 300%. It also appears that the minimum value of b_{ref} needed for stability increases as $|\Delta\lambda|$ decreases; the $b_{\text{ref}} = 0.5$ case failed when $|\Delta\lambda| = 1 \times 10^{-5}$, as did the $b_{\text{ref}} = 0.5$ and $b_{\text{ref}} = 0.6$ cases for both $|\Delta\lambda| = 1 \times 10^{-6}$ and $|\Delta\lambda| = 1 \times 10^{-7}$. The rank 2 predictor was also tested with the filter but either became unstable or converged with the same accuracy as the rank 1 case for all $\|\Delta\lambda\|$ that were tested.

7.6 Comparison of Algorithm Variants

Since the single-stage MS4 algorithm was less accurate but also less expensive than the two-stage explicitly filtered rank 1 predictor algorithm, the relative effectiveness of the algorithms is evaluated by plotting the error in C_l at the end of the continuation phase versus the CPU wall time, where the wall time is measured from the beginning of iterations until the end of the continuation phase. The values of b_{ref} used in the filtered algorithm were the smallest which were found to lead to a stable algorithm, spcifically $b_{\text{ref}} = 5$, 6, 7, and 7 for $\|\Delta\lambda\| = 10^{-4}$, 10^{-5}, 10^{-6}, and 10^{-7} respectively.

The data are plotted in Figure 6, measuring CPU time in TauBench "work units." We define one work unit (wu) as the total time divided by the time taken to run the DLR TauBench [6] code using 2.5×10^5 nodes in serial with 10 iterative steps. We have measured one work unit as 9.571s on the SciNet general purpose cluster.

For this study, the two-stage rank 1 filtered algorithm and the single-stage MS4 algorithm give comparable performance and either of these two algorithms

% error in C_l	28%	20%	10%
Homotopy with filter, $\gamma_{\text{ref}} = 10^{-4}$	123	407	1437
Pseudo-time-marching, $\Delta t_{\text{ref}} = 10^{-4}$	286	407	580

Table 1: Time (wu) required to achieve a certain error threshold in C_l

is more efficient than the basic rank 0 predictor algorithm.

7.7 Preliminary Comparison to a Pseudo-Time-Marching Algorithm

For reference, we have also collected some timing data from a pseudo-time-marching algorithm. The algorithm is based on MS4 and uses a spatially varying step-size given by equation (37) times a reference time step Δt_{ref}. Comparative timing data is shown in Table 1 which compares timing results from the pseudo-time algorithm with reference time step $\Delta t_{\text{ref}} = 10^{-4}$ to that of the filtered homotopy algorithm.

It can be seen from the table that in its current form the homotopy algorithm can be faster at obtaining an approximate solution up to a certain error threshold, in this case approximately 20% error in C_l. We mention that while this error seems large, the C_l for this case is very low, and all approximations shown in the table were sufficiently close to the solution for Newton's method to converge when applied.

8 Conclusions

A new matrix-free class of monolithic homotopy continuation algorithm was developed. Convergence and stability were studied analytically and numerically through application to a computational aerodynamics flow solver. The so-called "two-stage" formulation was found to be unstable but accurate. Stability could be achieved at the cost of accuracy by augmenting the algorithm with an explicit filter. The so-called "single-stage" formulation, which included no predictor component, was found to be stable but required very small step size to trace the curve accurately, making the algorithm computationally inefficient. Efficiency was improved by incorporating a more efficient multi-stage integration method into the discretization. The single-stage algorithm using MS4 and the two-stage filter-augmented algorithm exhibited similar computational efficiency when performance was compared for a dissipation-based homotopy on a two-dimensional inviscid subsonic compressible discrete aerodynamic flow problem. Since there are an infinite number of homotopies which can be constructed there is considerable potential to improve on the results demonstrated in this paper.

Other convergence acceleration techniques, such as multi-grid, are expected to be beneficial but were not investigated due to practical considerations. The method can be of practical value to implementations where it is not desirable to implement any matrix inversion, either to reduce implementation time or because it may be difficult to do so for some homotopies. Of more immediate value, the

analysis of the new algorithms has improved our understanding of monolithic homotopy continuation algorithms in general and may guide future development of these methods.

Acknowledgements The authors gratefully acknowledge financial assistance from the Natural Science and Engineering Research Council (NSERC), the Canada Research Chairs program, and the University of Toronto. Computations were performed on the GPC supercomputer 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. Allgower, E.L., Georg, K.: Introduction to Numerical Continuation Methods. Society for Industrial and Applied Mathematics (1990)
2. Brown, D.A., Zingg, D.W.: Advances in homotopy-based globalization strategies in computational fluid dynamics (2013). AIAA-2013-2944
3. Brown, D.A., Zingg, D.W.: A monolithic homotopy continuation algorithm with application to computational fluid dynamics. J. Comp. Phys. **321**, 55–75 (2016)
4. Carpenter, M.H., Gottlieb, D., Abarbanel, S.: Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes. J. Comput. Phys. **111**(2), 220–236 (1994)
5. Del Rey Fernández, D.C., Hicken, J.E., Zingg, D.W.: Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. Comput. Fluids **95**, 171–196 (2014)
6. DLR Germany: Taubench version 1.1, IPACS. http://www.ipacs-benchmark.org (2014). Accessed: 2014-09-20
7. Funaro, D., Gottlieb, D.: A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations. Math. Comput. **51**, 599–613 (1988)
8. Getz, N.H.: Dynamic inversion of nonlinear maps with applications to nonlinear control and robotics. Ph.D. thesis, University of California at Berkeley, Berkeley, California, USA (1995)
9. Getz, N.H., Marsden, J.E.: A dynamic inverse for nonlinear maps. In: Proceedings of the 34th IEEE Conference on Decision and Control, vol. 4, pp. 4218–4223 (1995)
10. Getz, N.H., Marsden, J.E.: Tracking implicit trajectories. In: IFAC Symposium on Nonlinear Control Systems Design. Tahoe City (1995)
11. Hartman, P.: Ordinary Differential Equations, second edn. Birkhauser, Berlin (1982)
12. Hicken, J.E., Buckley, H., Osusky, M., Zingg, D.W.: Dissipation-based continuation: a globalization for inexact-Newton solvers (2011). AIAA 2011-3237
13. Hicken, J.E., Zingg, D.W.: A parallel Newton-Krylov solver for the Euler equations discretized using simultaneous approximation terms. AIAA J. **46**(11), 2773–2786 (2008)
14. Hicken, J.E., Zingg, D.W.: Globalization strategies for inexact-Newton solvers (2009). AIAA-2009-4139
15. Jackiewicz, Z.: General Linear Methods for Ordinary Differential Equations. John Wiley & Sons (2009)
16. Jameson, A., Yoon, S.: Multigrid solution of the Euler equations using implicit schemes (1985). AIAA-85-0293
17. Kreiss, H., Scherer, G.: Finite element and finite difference methods for hyperbolic partial differential equations. In: C. de Boor (ed.) Mathematical Aspects of Finite Elements in Partial Differential Equations: proceedings of a symposium conducted by the Mathematics Research Center, the University of Wisconsin, pp. 195–212. Mathematics Research Center, the University of Wisconsin, Academic Press (1974)
18. Lomax, H., Pulliam, T.H., Zingg, D.W.: Fundamentals of Computational Fluid Dynamics. Springer-Verlag (2001)
19. Nadaraya, E.A.: On estimating regression. Theor. Probab. Appl. **9**(1), 141–142 (1964)
20. Osusky, M., Zingg, D.W.: A parallel Newton-Krylov-Schur flow solver for the Navier-Stokes equations discretized using summation-by-parts operators. AIAA J. **51**(12), 2833–2851 (2013)

21. Pulliam, T.H., Zingg, D.W.: Fundamental Algorithms in Computational Fluid Dynamics. Springer (2014)
22. Saad, Y.: Iterative Methods for Sparse Linear Systems, second edn. SIAM, Philadelphia, PA (2003)
23. Saad, Y., Schultz, M.: GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM J. Sci. Stat. Comp. **7**, 856–869 (1986)
24. Watson, G.S.: Smooth regression analysis. Sankhya Ser. A **26**(4), 359–372 (1964)
25. Yu, M., Wang, Z.J.: Homotopy continuation for correction procedure via reconstruction - discrete Galerkin (CPR-DG) methods (2015). AIAA 2015-0570