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

AIAA 2009-3657

A High-Order Parallel Newton-Krylov flow solver for the Euler Equations

Sydney C. Dias* and David W. Zingg†

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

This work presents a parallel Newton-Krylov flow solver employing third and fourth-order spatial discretizations to solve the three-dimensional Euler equations on structured multi-block meshes. The fluxes are discretized using summation-by-parts operators; boundary and interface conditions are implemented using simultaneous approximation terms. Functionals, drag and lift, are calculated using Simpson's rule. The solver is verified using the method of manufactured solutions and Ringleb flow and validated using the ONERA M6 wing. The results demonstrate that the combination of high-order finite-difference operators with a parallel Newton-Krylov solution technique is an excellent option for efficient computation of aerodynamic flows.

I. Introduction

Fuel consumption and emissions are two major issues when designing aircraft today. Rising fuel costs have caused major airlines to cut operations and in extreme cases shut down. Also, the impact of aircraft emissions on climate change is significant and continues to increase. Hence, there is a great need to find alternate fuel sources and design aircraft that have a lower fuel consumption per passenger-km.[1,2]

In order to meet these design requirements, several research groups around the world are developing numerical methods to optimize aircraft that will aid aircraft manufacturers.[3–16] The process for aerodynamic shape optimization is shown in Figure 1, and the aim of this work is to develop an efficient and robust flow solver using high-order methods, which will then be used as a platform for aerodynamic shape optimization.

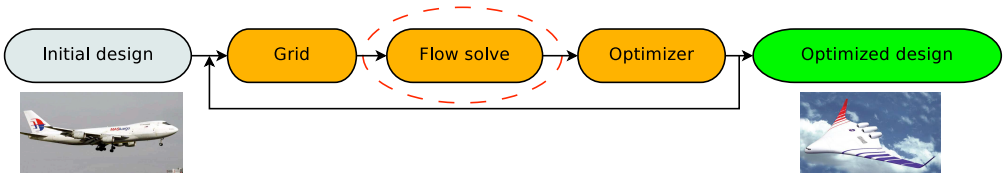


Figure 1. Optimization process

Flow solvers have been developed with high-order finite-element, finite-volume and finite-difference schemes. For example, Gooch et al.[17–21] and Vaassen et al.[22] and many others have developed high-order finite-volume schemes, Darmofal et al.[23–26] and others have developed high-order discontinous Galerkin schemes, and Zingg et al.[27–31] and Svard et al.[32,33] have developed high-order finite-difference schemes.

The reason high-order methods are being pursued is because they generally produce a smaller error per unit cost[18,24,29,34] and are thus efficient. Using a high-order method, one can obtain a solution of a required accuracy using fewer nodes/cells than would be required for a second-order method. Additional benefits of using coarser grids enabled by the use of high-order methods are seen in aerodynamic shape optimization, for example reduced mesh movement cost.

The objective of this work is to develop a parallel Newton-Krylov flow solver employing high-order finite-difference schemes to solve the steady three-dimensional Euler equations on structured multi-block meshes.

*Graduate student, AIAA Student Member

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

The fluxes are discretized using summation-by-parts operators, and boundary and interface conditions are implemented using simultaneous approximation terms. High-order integration is performed using Simpson's rule. The solver is verified using several test cases, including the method of manufactured solutions and Ringleb flow, and is validated using the ONERA M6 wing.

II. Governing Equations

We deal with the three-dimensional Euler equations, which for a Cartesian coordinate system are given by

$$\partial_t \mathsf{Q} + \partial_x \mathsf{E} + \partial_y \mathsf{F} + \partial_z \mathsf{G} = 0 \tag{1}$$

where

$$\mathsf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \quad \mathsf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \rho w u \\ u\,(e+p) - p \end{bmatrix} \quad \mathsf{F} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho w v \\ v\,(e+p) - p \end{bmatrix} \quad \mathsf{G} = \begin{bmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ w\,(e+p) - p \end{bmatrix}$$

The equations comprise the conservation of mass, equations for the conservation of momentum in each of the three coordinate directions, and the conservation of energy. This equation may also be given in a curvilinear coordinate system by[35,36]

$$\partial_\tau \hat{\mathsf{Q}} + \partial_\xi \hat{\mathsf{E}} + \partial_\eta \hat{\mathsf{F}} + \partial_\zeta \hat{\mathsf{G}} = 0 \tag{2}$$

where

$$\hat{\mathsf{Q}} = J^{-1}\mathsf{Q}$$

$$\hat{\mathsf{E}} = J^{-1}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ U\,(e+p) - \xi_t p \end{bmatrix} \quad \hat{\mathsf{F}} = J^{-1}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ V\,(e+p) - \eta_t p \end{bmatrix} \quad \hat{\mathsf{G}} = J^{-1}\begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ W\,(e+p) - \zeta_t p \end{bmatrix}$$

and the contravariant velocities are given by

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \end{aligned}$$

The variables ξ, η, ζ and τ correspond to the new coordinate system and ξ_x, ξ_y, ξ_z, η_x, η_y, etc are the grid metrics. As a consequence of using a generalized coordinate system in conservative form we have the following identities known as the metric invariants that must be satisfied:

$$\begin{aligned} (\xi_x)_\xi + (\eta_x)_\eta + (\zeta_x)_\zeta &= 0 \\ (\xi_y)_\xi + (\eta_y)_\eta + (\zeta_y)_\zeta &= 0 \\ (\xi_z)_\xi + (\eta_z)_\eta + (\zeta_z)_\zeta &= 0 \\ (1/J)_\tau + (\xi_t)_\xi + (\eta_t)_\eta + (\zeta_t)_\zeta &= 0 \end{aligned} \tag{3}$$

Lastly, a closed set of equations is obtained by using the ideal gas approximation for a thermally and calorically perfect gas with γ=1.4 for air.

III. Algorithm

In this section the algorithm used to solve the governing equations is described. The section begins by introducing the spatial discretization scheme and then moves on to describe the iterative technique.

III.A. Spatial discretization

The three-dimensional domain is discretized using structured grids. Additionally, the domain is broken down into a number of blocks to enable the solver to deal with complex geometries. For the purpose of this work assume that each block consists of n_ξ, n_η and n_ζ nodes in the ξ, η and ζ directions, respectively.

III.A.1. Finite-difference schemes

Operators that satisfy the summation-by-parts (SBP) property are used. For a spatial discretization operator $D = H^{-1}P$, applied to vectors u and v such that $x \in [a, b]$, the SBP property is defined as:[37–40]

$$< u, Dv > + < Dv, u >= uv|_a^b \tag{4}$$

where the scalar product is defined by:

$$< u, v >= u^T H v \tag{5}$$

and H is a positive definite matrix. SBP operators are used in this work because they provide an energy estimate.

SBP operators come in two varieties, using diagonal and non-diagonal norms. In this work diagonal norm operators are used because they guarantee stability when using a curvilinear coordinate system in three dimensions.[38,41–43] As an example of SBP operators, H and P for a second-order spatial discretization of the first derivative are shown below:[39,40]

$$H = \begin{pmatrix} \frac{1}{2} & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \frac{1}{2} \end{pmatrix} \qquad P = \frac{1}{2\Delta x} \begin{pmatrix} -1 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 1 \end{pmatrix}$$

The third and fourth-order operators are shown in Section VII. They are centered differences (anti-symmetric) and therefore require dissipation (symmetric), which shall be discussed later.

Now the Euler equations (2) become

$$\partial_\tau \hat{Q} + \delta_\xi^a \hat{E} + \delta_\eta^a \hat{F} + \delta_\zeta^a \hat{G} = 0 \tag{6}$$

where $\hat{Q}$ is a vector of dimensions $n_\xi \times n_\eta \times n_\zeta \times 5$ that contains the vectors $\hat{\mathrm{Q}}$ at every node in the domain. The antisymmetric[a] operators are defined by

$$\begin{aligned} \delta_\xi^a &= D_\xi \otimes I_\eta \otimes I_\zeta \otimes I_5 \\ \delta_\eta^a &= I_\xi \otimes D_\eta \otimes I_\zeta \otimes I_5 \\ \delta_\zeta^a &= I_\xi \otimes I_\eta \otimes D_\zeta \otimes I_5 \end{aligned}$$

where the operator $\otimes$ is the Kronecker product for matrices such that if $A \in M^{m\times n}$ and $B \in M^{p\times q}$ then $C = A \otimes B \in M^{mp\times nq}$ is defined by $C_{p(i-1)+k,q(j-1)+l} = A_{i,j}B_{k,l}$. I_ξ, I_η, I_ζ and I_5 are identity matrices of dimensions $n_\xi \times n_\xi$, $n_\eta \times n_\eta$, $n_\zeta \times n_\zeta$ and 5×5 respectively. D_ξ, D_η and D_ζ are finite difference operators of dimensions $n_\xi \times n_\xi$, $n_\eta \times n_\eta$ and $n_\zeta \times n_\zeta$ respectively.[32]

III.A.2. Grid metrics

The grid metrics are calculated using the spatial discretization outlined in the previous section. However, if the grid metrics are calculated directly, for instance ξ_x is given by

$$\xi_x = J(y_\eta z_\zeta - y_\zeta z_\eta) \tag{7}$$

[a]The interior finite-difference scheme is antisymmetric.

then the metric invariants (3) are not necessarily satisfied. This means that a uniform free stream could fail to satisfy the discrete Euler equations. The approach implemented in this work exploits the fact that when finite differences are used to approximate a first derivative, the product rule is not satisfied numerically. Thus the metrics are calculated using a conservative form. For instance:[44,45]

$$\xi_x = J[(y_\eta z)_\zeta - (y_\zeta z)_\eta] \tag{8}$$

Similar formulas can be obtained for the other metric terms. For this method to work, the finite-difference scheme used to calculate the metrics must be the same as that used to calculate the flux terms. One can now easily verify that the metric invariants are satisfied by substituting the modified metrics into equation (3).

III.A.3. Artificial dissipation

Nonlinear convection problems such as the Euler equations that use central differences require the use of artificial disspation to damp out high frequency modes. In this work, the dissipation operator is implemented so that it meets the following criteria, as stated by Mattson et al.[46] :

1. Efficiently reduce spurious oscillations.
2. Accuracy should be the same as the the finite-difference scheme.
3. Computational work should be comparable to the finite-difference scheme, i.e. the stencil size of the dissipation operator should be similar to the finite-difference scheme if not the same.
4. The dissipation operator should provide an energy estimate.

Matrix dissipation is used because it normally results in lower errors compared to scalar dissipation.[47,48] The matrix dissipation model along the ξ direction is given by:[35,49–51]

$$\delta^s_\xi \hat{Q} = \mathcal{H}^{-1}\tilde{\mathcal{D}}_- J^{-1} \left|\overline{A}\right| \left(\epsilon^{(2)}\tilde{\mathcal{D}}_+ - \epsilon\tilde{\mathcal{D}}_-^{-1}\tilde{\mathcal{D}}_p^T \mathcal{B}\tilde{\mathcal{D}}_p\right) J\hat{Q} \tag{9}$$

where the operators are given by

$$\begin{aligned}
\mathcal{H} &= H \otimes I_\eta \otimes I_\zeta \otimes I_5 \\
\tilde{\mathcal{D}}_- &= \tilde{D}_- \otimes I_\eta \otimes I_\zeta \otimes I_5 \\
\tilde{\mathcal{D}}_+ &= \tilde{D}_+ \otimes I_\eta \otimes I_\zeta \otimes I_5 \\
\tilde{\mathcal{D}}_p &= \tilde{D}_p \otimes I_\eta \otimes I_\zeta \otimes I_5 \\
\mathcal{B} &= B \otimes I_\eta \otimes I_\zeta \otimes I_5 \\
\left|\overline{A}\right| &= \mathrm{diag}\left(\left|\overline{A}\right|_{1+\frac{1}{2},1,1}, ..., \left|\overline{A}\right|_{j+\frac{1}{2},k,m}, ..., \left|\overline{A}\right|_{n_\xi,n_\eta,n_\zeta}\right) \\
\epsilon^{(2)} &= \mathrm{diag}(\epsilon^{(2)}_{1+\frac{1}{2},1,1}, ..., \epsilon^{(2)}_{j+\frac{1}{2},k,m}, ..., \epsilon^{(2)}_{n_\xi,n_\eta,n_\zeta}) \otimes I_5 \\
\epsilon &= \mathrm{diag}(\epsilon_{1+\frac{1}{2},1,1}, ..., \epsilon_{j+\frac{1}{2},k,m}, ..., \epsilon_{n_\xi,n_\eta,n_\zeta}) \otimes I_5 \\
J &= \mathrm{diag}(J_{1+\frac{1}{2},1,1}, ..., J_{j+\frac{1}{2},k,m}, ..., J_{n_\xi,n_\eta,n_\zeta}) \otimes I_5
\end{aligned}$$

$\tilde{D}_-$ and $\tilde{D}_+$ are backward and forward difference operators given by:[51]

$$\tilde{D}_- = \begin{pmatrix} 1 & & & & & & \\ -1 & 1 & & & & & \\ & -1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & -1 & 1 & & \\ & & & & -1 & 1 & \\ & & & & & -1 & 1 \end{pmatrix}$$

$$\tilde{D}_+ = \begin{pmatrix} -1 & 1 & & & & & \\ & -1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & -1 & 1 & & \\ & & & & -1 & 1 & \\ & & & & & -1 & 1 \\ & & & & & & -1 \end{pmatrix}$$

$\tilde{D}_p$ is a consistent difference approximation of $\frac{\partial^p}{\partial \xi^p}$ and is chosen to be $\tilde{D}_2$ for the second and third-order methods and $\tilde{D}_3$ for the fourth-order method. $\tilde{D}_2$ and $\tilde{D}_3$ are given by:[46]

$$\tilde{D}_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$$

$$\tilde{D}_3 = \begin{pmatrix} -1 & 3 & -3 & 1 & & \\ -1 & 3 & -3 & 1 & & \\ -1 & 3 & -3 & 1 & & \\ & -1 & 3 & -3 & 1 & \\ & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

B is a diagonal matrix with a spatial dependence that is chosen to satisfy the accuracy requirements. For the second-order method $B = \text{diag}(0, 1, 1, ..., 1, 1, 0)$. For the third and fourth-order methods, B is chosen such that it increases from $\left(\frac{1}{n_\xi - 1}\right)^{p-1}$ at the boundaries to 1 in the interior, and derivatives up to $p-2$ are 0 at the boundaries and transition points. A typical distribution for the diagonal of B including the transition region is shown in Figure 2. $\left|\overline{A}\right|$ is a block diagonal matrix that is calculated for a Roe averaged state and $|A|$ is given by:[48]

$$|A| = X \left|\Lambda\right| X^{-1} \tag{10}$$

where X is the eigenvector matrix of the flux Jacobian and

$$|\Lambda| = \begin{pmatrix} |\lambda_1| & 0 & 0 & 0 & 0 \\ 0 & |\lambda_2| & 0 & 0 & 0 \\ 0 & 0 & |\lambda_3| & 0 & 0 \\ 0 & 0 & 0 & |\lambda_3| & 0 \\ 0 & 0 & 0 & 0 & |\lambda_3| \end{pmatrix}$$

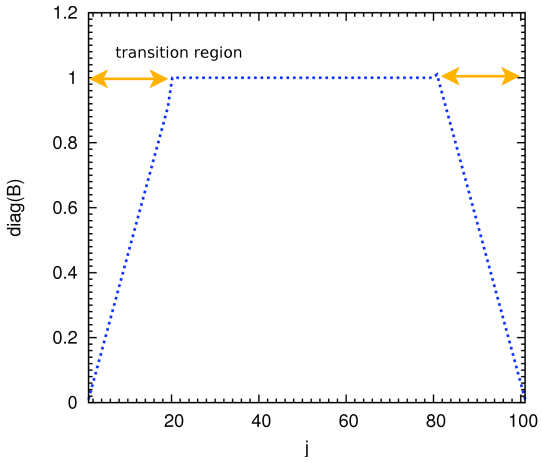


Figure 2. Distribution of the diagonal of B showing the interior and transition regions

where

$$\lambda_1 = U + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}$$
$$\lambda_2 = U - a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}$$
$$\lambda_3 = U$$
$$|\lambda_1| = \max(|\lambda_1|,\ V_n\rho)$$
$$|\lambda_2| = \max(|\lambda_2|,\ V_n\rho)$$
$$|\lambda_3| = \max(|\lambda_3|,\ V_l\rho)$$
$$\rho = J^{-1}\left(|U| + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}\right)$$

The remaining variables are defined as

$$\Upsilon_{j,k,m} = \frac{|p_{j+1,k,m} - 2p_{j,k,m} + p_{j-1,k,m}|}{|p_{j+1,k,m} + 2p_{j,k,m} + p_{j-1,k,m}|}$$
$$\epsilon = \max(0, \kappa - \epsilon_2)$$
$$\epsilon^{(2)}_{j,k,m} = \kappa_2 \max\left[\Upsilon_{j+1,k,m}, \Upsilon_{j,k,m}, \Upsilon_{j-1,k,m}\right]$$

Υ is a pressure switch that turns on second-difference dissipation in the vicinity of a shock. Typical values for κ_2, κ, V_n and V_l are 1, 0.04, 0.25 and 0.25, respectively.

On adding the dissipation terms to the semi-discrete Euler equations (6), we get

$$\partial_\tau \hat{Q} + \delta^a_\xi \hat{E} + \delta^s_\xi \hat{Q} + \delta^a_\eta \hat{F} + \delta^s_\eta \hat{Q} + \delta^a_\zeta \hat{G} + \delta^s_\zeta \hat{Q} = 0 \tag{11}$$

III.A.4. Interfaces and boundaries

The block interfaces and boundaries are dealt with using simultaneous approximation terms (SATs).[52–55] SATs are a penalty method which, when implemented with SBP operators, provide an energy estimate. For convenience we shall consider interfaces and boundaries in the ξ direction. It is quite easy to obtain similar results for the other two directions.

Consider that in the ξ direction the domain has been split into two blocks with boundaries at $j = 0$ and $j = N$ and an interface at $j = s$ and $j = s + 1$, as shown in Figure 3. The semi-discrete Euler equations with SATs for the interface and boundaries for the left and right blocks are given by

$$\partial_\tau \hat{Q} + \delta^a_\xi \hat{E} + \delta^s_\xi \hat{Q} + \delta^a_\eta \hat{F} + \delta^s_\eta \hat{Q} + \delta^a_\zeta \hat{G} + \delta^s_\zeta \hat{Q} = bc_s + bc_0 \tag{12}$$
$$\partial_\tau \hat{Q} + \delta^a_\xi \hat{E} + \delta^s_\xi \hat{Q} + \delta^a_\eta \hat{F} + \delta^s_\eta \hat{Q} + \delta^a_\zeta \hat{G} + \delta^s_\zeta \hat{Q} = bc_{s+1} + bc_N \tag{13}$$

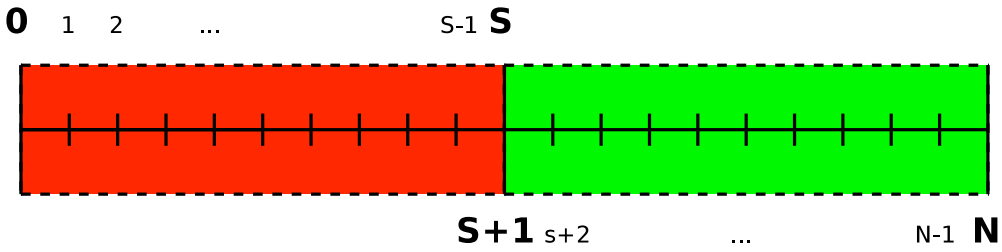


Figure 3. Figure showing two blocks with boundaries and an interface

where the SATs are:

$$bc_0 = -\tau_{bc}\mathcal{H}^{-1}J^{-1}\frac{(|A|+A)}{2}(e_0 \otimes (Q_0 - Q_{bc}))$$
$$bc_s = -\tau_{bc}\mathcal{H}^{-1}J^{-1}\frac{(|A|-A)}{2}(e_s \otimes (Q_s - Q_{s+1}))$$
$$bc_{s+1} = -\tau_{bc}\mathcal{H}^{-1}J^{-1}\frac{(|A|+A)}{2}(e_{s+1} \otimes (Q_{s+1} - Q_s))$$
$$bc_N = -\tau_{bc}\mathcal{H}^{-1}J^{-1}\frac{(|A|-A)}{2}(e_N \otimes (Q_N - Q_{bc}))$$

where

$$|A| = \text{diag}\left(|A|_{1,1,1}, ..., |A|_{j,k,m}, ..., |A|_{n_\xi,n_\eta,n_\zeta}\right)$$
$$A = \text{diag}\left(A_{1,1,1}, ..., A_{j,k,m}, ..., A_{n_\xi,n_\eta,n_\zeta}\right)$$

A is the flux Jacobian and $|A|$ is given by equation (10). The variables $e_0 = [1, 0, ..., 0]^T$, $e_s = [0, 0, ..., 1]^T$, $e_{s+1} = [1, 0, ..., 0]^T$ and $e_N = [0, 0, ..., 1]^T$ are column vectors of dimensions $n_\xi \times 1$, where n_ξ is the number of points in the ξ direction, which maybe different for each block. Q_0, Q_s, Q_{s+1} Q_N and Q_{bc} are column vectors of dimensions $5n_\eta n_\zeta \times 1$. Q_{bc} contains the flow variables that need to be imposed at the boundaries.[47] τ_{bc} is a scalar that scales the boundary and interface terms.

III.A.5. Stability

The operators derived thus far do not guarantee stability for the three-dimensional Euler equations. However, when the same formulation is applied to the linear convection equation, it provides an energy estimate and thus guarantees stability. Stability of the linear convection equation is necessary but not sufficient to guarantee stability for the three-dimensional Euler equations.[37,38,40,43,46]

III.B. Iteration to steady state

Using the spatial discretization from the previous section we obtain

$$\frac{d\hat{Q}}{d\tau} = -R(\hat{Q}) \tag{14}$$

where $R(\hat{Q})$ is the residual given by

$$R(\hat{Q}) = \delta^a_\xi \hat{E} + \delta^s_\xi \hat{Q} + \delta^a_\eta \hat{F} + \delta^s_\eta \hat{Q} + \delta^a_\zeta \hat{G} + \delta^s_\zeta \hat{Q} - bc$$

This work is concerned with steady flows and thus the equation we aim to solve is

$$R(\hat{Q}) = 0 \tag{15}$$

Using the implicit Euler method through local linearization of $R(\hat{Q})$, one obtains:[56]

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}}\right)^n\right] \Delta \hat{Q}^n = -R^n \tag{16}$$

where the superscript n is the outer iteration number and $\tau = t$. If we let $\Delta t \to \infty$, Newton's method, which converges quadratically, is obtained. If the initial guess is not close to the solution, then Newton's method can fail. Hence, the solution process uses two stages, a start-up phase to find a suitable iterate and an inexact-Newton phase. Before we describe the two phases, we shall lay down some of the background.

III.B.1. Relaxation

For each solution update, we have the following equation

$$\hat{Q}^{n+1} = \hat{Q}^n + \theta \Delta \hat{Q}^n \tag{17}$$

where θ is the relaxation factor. Under relaxation ($\theta < 1$) helps in preventing solution updates during the start-up phase that would cause the flow to become non-physical.

III.B.2. Linear system solution

We utilize GMRES to solve the linear system (16).[57,58] GMRES requires matrix-vector products. We use Jacobian-free matrix-vector products, which are approximated with a finite-difference:[17–20,55,59–61]

$$\frac{\partial R(\hat{Q})}{\partial \hat{Q}} v \approx \frac{R(\hat{Q} + \epsilon v) - R(\hat{Q})}{\epsilon} \tag{18}$$

with

$$\epsilon = \sqrt{\frac{N\delta}{v^T v}}$$

where N is the number of unknowns, and $\delta = 10^{-13}$.[55] The performance of GMRES can be improved if the system is preconditioned. When right preconditioned with M the system in equation (16) becomes

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}}\right)^n\right] M^{-1} M \Delta \hat{Q}^n = -R^n \tag{19}$$

M^{-1} is calculated by performing an incomplete lower-upper factorization (ILU) with fill level k. As the fill level k is increased, the factorization becomes more accurate.[62] We choose M to be the ILU factorization of a first-order approximate Jacobian. This is ideal because the first-order Jacobian requires less memory and is faster to calculate and factor.[17–20,60] The high-order dissipation coefficient coefficient ϵ is lumped with the first-order dissipation coefficient $\epsilon^{(2)}$ in the preconditioner using the parameter σ:[55,60,61]

$$\epsilon^{(M)} = \epsilon^{(2)} + \sigma\epsilon \tag{20}$$

Also the additive Schwarz and approximate Schur parallel preconditioners are used.[55,62]

III.B.3. Local time stepping

To speed up convergence a spatially varying time step is used:[55]

$$\Delta t_{j,k,m} = \frac{\Delta t^n_{ref}}{J_{j,k,m}\left(1 + \sqrt[3]{J_{j,k,m}}\right)} \tag{21}$$

This approximately represents a constant CFL number. The reference time step Δt_{ref} is chosen differently for the two phases, as described below.

III.B.4. Reordering of unknowns

The nodes are reordered using the reverse Cuthill-McKee (RCM) method. This reduces the bandwidth of the system and also improves the conditioning of the ILU factors.[60,61,63] The root node is chosen to be the node with indices (n_ξ, n_η, n_ζ) for each block, which lies on the downstream boundary of the block.

III.B.5. Start-up phase

The idea behind the start-up phase is to go through the initial transient as quickly as possible, in other words, to provide a suitable iterate so that we can switch to the inexact-Newton phase. We modify equation (16) and solve the following equation:[17–20,55,60]

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}}\right)_1\right] \Delta\hat{Q} = -R \tag{22}$$

where $\left(\frac{\partial R}{\partial \hat{Q}}\right)_1$ is the first-order Jacobian. Equation (22) is not solved exactly but to a relative tolerance of ω such that:[55]

$$\left|\left|R + \left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}}\right)_1\right] \Delta\hat{Q}\right|\right|_2 \leq \omega \left|\left|R\right|\right|_2 \tag{23}$$

The parameter ω is normally chosen to be 0.5. Also since factoring is expensive, we use a lagged Jacobian update[55] i.e. we update the Jacobian every three iterations. The reference time step is chosen such that,[55]

$$\Delta t^n_{ref} = a(b)^n \tag{24}$$

Typical values for a and b are 0.01 and 1.2 respectively.

III.B.6. Inexact-Newton phase

The switch to the inexact-Newton phase is made once the residual has fallen below a certain value specified by τ such that:[55]

$$\left|\left|R^n\right|\right|_2 \leq \tau \left|\left|R^0\right|\right|_2 \tag{25}$$

A typical value for τ is $\frac{1}{15}$. Equation (16) is now solved with a tolerance ω such that[55]

$$\left|\left|R + \left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}}\right)\right] \Delta\hat{Q}\right|\right|_2 \leq \omega \left|\left|R\right|\right|_2 \tag{26}$$

where,

$$\omega_n = \max\left(0.01, \omega_{n-1}^{\frac{1+\sqrt{5}}{2}}\right) \tag{27}$$

The reference time step is chosen such that:[55]

$$\Delta t^n_{ref} = \max\left(\alpha \left(\frac{\left|\left|R^n\right|\right|_2}{\left|\left|R^0\right|\right|_2}\right)^{-\beta}, \Delta t^{n-1}_{ref}\right) \tag{28}$$

with

$$\alpha = a(b)^n \left(\frac{\left|\left|R^k\right|\right|_2}{\left|\left|R^0\right|\right|_2}\right)^{\beta} \tag{29}$$

where k is the last start-up phase iteration, and β =2.

III.C. Force Integration

Lift and drag coefficients are calculated using Simpson's rule, which is a fourth-order accurate method.[47,64]

IV. Results and Discussion

This section demonstrates the algorithm described in the previous section for various test cases. It begins by describing the method of manufactured solutions and Ringleb flow. Using these tests we can rigorously test the algorithm. The algorithm is then validated for the ONERA M6 wing.

IV.A. Computational hardware

An SMP HP Itanium Beowulf-class cluster is used. Each node consists of 4 Itanium II processors with 6MB L3 cache and a clock speed of 1500MHz, with at least 8GB of RAM. The nodes are connected with a high-bandwidth low-latency Myrinet network.

IV.B. Test cases

IV.B.1. Method of manufactured solutions

The method of manufactured solutions can be used to verify the prescribed order of accuracy for computational codes.[65] It works by manufacturing a solution by adding appropriate source terms to the governing equations.[66]

Let the form of the solution be

$$\phi(x,y,z) = \phi_o + \phi_x f_x\left(\frac{a_{\phi x}\pi x}{L}\right) + \phi_y f_y\left(\frac{a_{\phi y}\pi y}{L}\right) + \phi_z f_z\left(\frac{a_{\phi z}\pi z}{L}\right) \tag{30}$$

where ϕ is ρ, u, v, w, or p. The form of the solution and the constants f_x, f_y, f_z, ϕ_o, ϕ_x, ϕ_y, $a_{\phi x}$, $a_{\phi y}$, $a_{\phi z}$ and L are given in Table 1. These constants have been chosen in order to keep the flow physical, i.e. $p > 0$ and $\rho > 0$. Now that the form of the solution has been specified, the governing equations are applied to the manufactured solution to generate the analytical source terms. Thus equation (2) becomes

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = \hat{\psi} \tag{31}$$

where $\hat{\psi}$ is the source term.

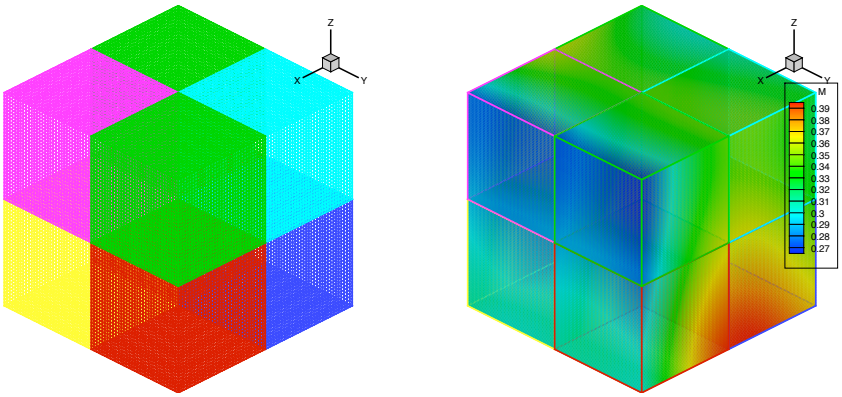
Quantity ϕ	f_x	f_y	f_z	ϕ_o	ϕ_x	ϕ_y	ϕ_z	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$	L
ρ	sin	cos	cos	1	0.1	0.15	0.15	0.75	1.0	1.0	1.0
u	sin	cos	cos	$\frac{70}{100\sqrt{14}}$	$\frac{4}{100\sqrt{14}}$	$\frac{-12}{100\sqrt{14}}$	$\frac{-8}{100\sqrt{14}}$	$\frac{5}{3}$	1.5	1.0	1.0
v	cos	sin	cos	$\frac{90}{100\sqrt{14}}$	$\frac{-20}{100\sqrt{14}}$	$\frac{4}{100\sqrt{14}}$	$\frac{3}{100\sqrt{14}}$	1.5	1.0	0.75	1.0
w	cos	sin	cos	$\frac{40}{100\sqrt{14}}$	$\frac{-2}{100\sqrt{14}}$	$\frac{2}{100\sqrt{14}}$	$\frac{3}{100\sqrt{14}}$	1.0	1.5	1.0	1.0
p	cos	sin	sin	$\frac{1}{1.4}$	$\frac{-0.3}{1.4}$	$\frac{0.2}{1.4}$	$\frac{0.05}{1.4}$	1.0	1.25	1.0	1.0

Table 1. Constants and form of the solution for three-dimensional flow using the method of manufactured solutions

The domain for the problem consists of eight blocks such that $0 \le x \le 1$, $0 \le y \le 1$ and $0 \le z \le 1$, as shown in Figure 4. Each block has a mesh density ranging from 13×13×13 nodes to 65×65×65 nodes. The boundary conditions are implemented by setting Q_{bc} to the exact solution.

IV.B.2. Ringleb flow

Ringleb flow is an exact solution to the two-dimensional Euler equations.[67–70] The flow lies between two streamlines. Hence, in addition to testing the order of accuracy of the flow solver, Ringleb flow may also be used to test the wall boundary conditions. The solution is obtained using a hodograph transformation and



(a) Domain (b) Solution (Mach number)

Figure 4. Domain and solution for the three-dimensional method of manufactured solutions

is given by:[67–70]

$$x(q,\mu) = \frac{1}{2\rho}\left(\frac{2}{\mu^2} - \frac{1}{q^2}\right) - \frac{C}{2} \tag{32}$$

$$y(q,\mu) = \pm\frac{1}{\mu\rho q}\sqrt{1-\left(\frac{q}{\mu}\right)^2} \tag{33}$$

$$a = \sqrt{1 - \frac{\gamma-1}{2}q^2} \tag{34}$$

$$\rho = a^{\frac{2}{\gamma-1}} \tag{35}$$

$$C = \frac{1}{a} + \frac{1}{3a^3} + \frac{1}{5a^5} - \frac{1}{2}\log\frac{1+a}{1-a} \tag{36}$$

where ρ is the density, x and y are the Cartesian coordinates, q is the nondimensional velocity magnitude, a is the sound speed, and μ is the streamline constant. The velocity is nondimensionalized with respect to the stagnation sound speed. The flow angle θ is by,

$$\theta = 2\pi - \sin^{-1}\left(\frac{q}{\mu}\right) \tag{37}$$

Lastly, the energy is obtained using isentropic relationships.

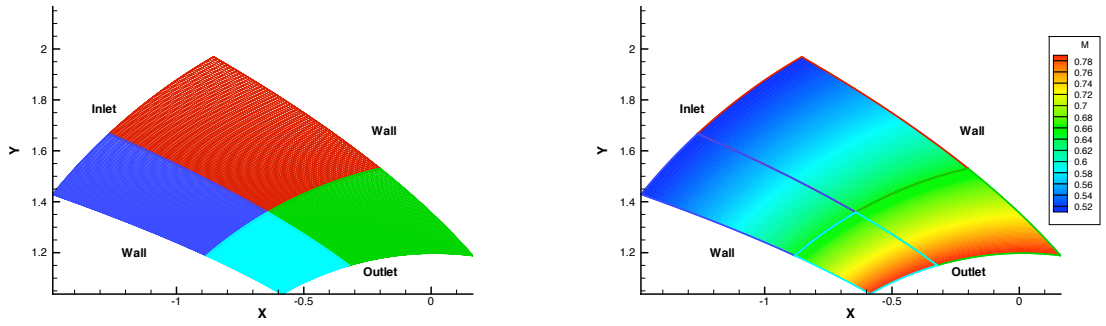
The entire flow can be parameterized using μ and q. In this work μ and q were chosen such that $1 \leq \mu \leq 1.5$ and $0.5 \leq q \leq 0.75$. The flow domain and solution are shown in Figure 5. The domain consists of four blocks with a mesh density ranging from 13×13 nodes to 67×67 nodes per block. Inlet, outflow and wall boundary conditions are imposed as shown in Figure 5. The inlet and outlet boundary conditions are imposed by setting Q_{bc} to the exact solution. Since the flow solver in this work is three-dimensional, symmetry boundary conditions are imposed on the $x-y$ planes.

IV.C. Order of accuracy

The order of accuracy of each scheme is tested using the method of manufactured solutions and Ringleb flow. The order of accuracy for the different test cases is shown in Figure 6, and the slopes for the lines of best fit to the data are shown in Table 2. The errors shown in Table 2 are standard asymptotic errors in the slopes for the lines of best fit. It can be seen that the observed order of accuracy agrees well with the formal order of accuracy. In addition to establishing the order of accuracy, these results verify that there are no bugs in the spatial discretization.

IV.D. Efficiency

Efficiency may be studied quantitatively using the Ringleb flow. Normally two factors are taken into consideration when one performs a flow solve, the time to achieve convergence and the accuracy of the solution.

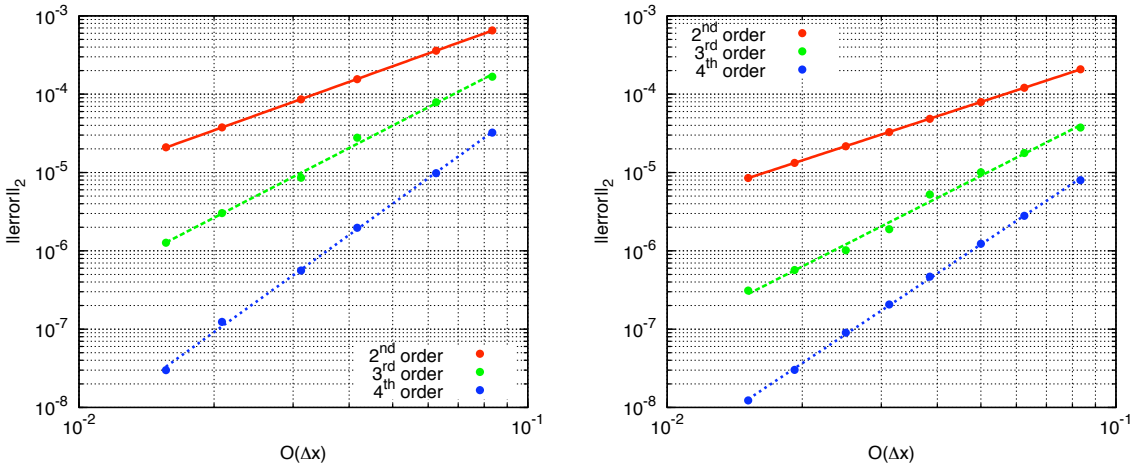


(a) Domain (b) Solution (Mach Number)

Figure 5. Domain and exact solution for Ringleb flow

Method	Slope
MMS	
2^{nd} order	2.053 ± 0.004
3^{rd} order	3.0 ± 0.1
4^{th} order	4.1 ± 0.1
Ringleb flow	
2^{nd} order	1.874 ± 0.002
3^{rd} order	2.9 ± 0.1
4^{th} order	3.81 ± 0.03

Table 2. Order of accuracy for different test cases, slopes of the lines of best fit



(a) MMS (b) Ringleb flow

Figure 6. Order of accuracy for different test cases

Ideally one would want the most accurate solution in the shortest possible time. One may then specify the desired accuracy or the time, if time is the limiting factor. If the accuracy is specified, the most efficient method would provide a solution of this accuracy in the shortest possible time. If time is the limiting factor, the most efficient method would provide the most accurate solution in this given time.

In Figure 7 the variation of the convergence time with accuracy of the solution is shown. It is quite easy to see that with the above definition of efficiency the high-order methods are indeed efficient compared to the second-order method. For an error level of 10^{-5}, for example, the third-order method requires roughly 10% of the CPU time of the second-order method while the fourth-order method requires roughly 4%.

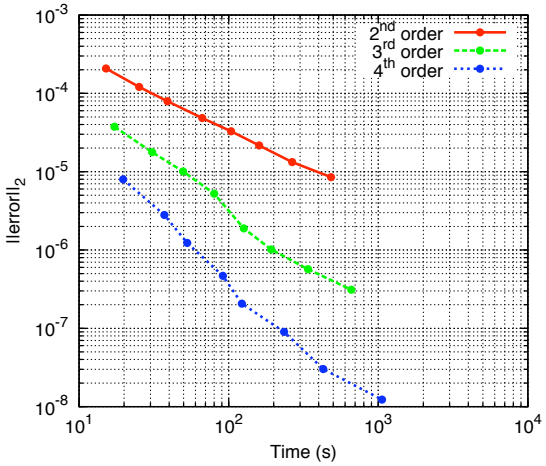


Figure 7. Efficiency study using Ringleb flow

IV.E. Schwarz and Schur preconditioners

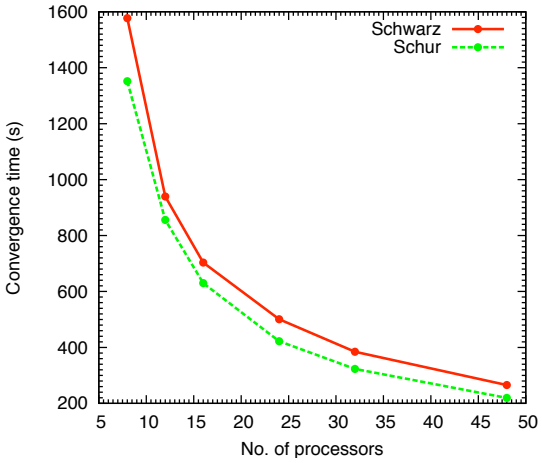
For this comparison the ONERA M6 wing at M=0.699 & α=3.06° is used on a grid with 96 blocks×17×17×17 nodes. The solution is obtained using both the Schwarz and Schur parallel preconditioners on 8, 12, 16, 32 and 48 processors for all three spatial discretizations. The data are shown in Figure 8. From the data we see that the Schur preconditioner performs consistently better for all three spatial discretizations. Both preconditioners show good parallel efficiency.

IV.F. ONERA M6 Wing

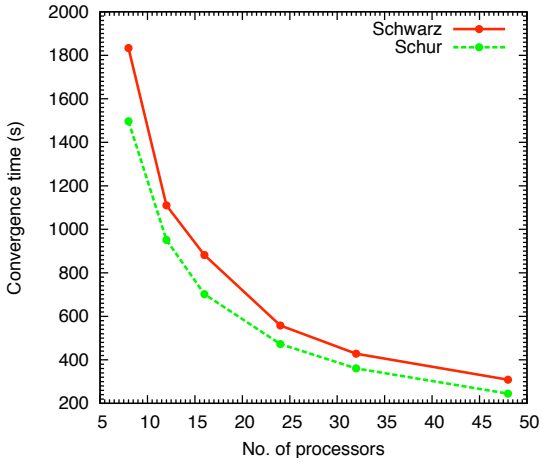
In this section, we study the ONERA M6 wing at $M = 0.3$ & α=4.0° and M=0.699 & α=3.06°. The first test case is purely subsonic and the second test case has a weak leading-edge shock. We use five grids ranging from 96 blocks×17×17×17 nodes to 96 blocks×33×33×33 nodes. The far field is placed 25 chord lengths away from the wing. The mesh for the 96 blocks×33×33×33 nodes grid is shown in Figure 9. The solution is obtained using 48 processors. The dissipation coefficients are set to $\kappa_2 = 0$ and $\kappa = 0.04$ for the first case. For the second case $\kappa_2 = 1$. V_n and V_l are set to 0.25. The transition region in the dissipation model is set to 8%. Far field and wall boundary conditions are used and the boundary and interface conditions are scaled by unity, i.e. $\tau_{bc} = 1$. ILU(0) and ILU(1) is used during the start-up and inexact-Newton phases. The first-order preconditioner is used and the dissipation coefficients are lumped with $\sigma = 10$. The coefficients during the start-up phase are $a = 0.01$, $b = 1.2$ and $\omega = 0.05$. Also the preconditioner is updated every three iterations during the start-up phase. Relaxation is used only during the start-up phase and is set at $\theta = 0.6$. During the inexact-Newton phase, β is chosen to be 2, and the parameter that decides when to switch to the inexact-Newton phase, τ, is set to 1/15. The number of GMRES iterations is limited to 80 (no restarts), and the solution is said to be converged when the residual norm has dropped below $1E-11$.

The convergence summary for the ONERA M6 wing at M=0.699 and α=3.06° is shown in Table 3, and the convergence history is shown in Figure 10. As expected, the high-order methods take longer to converge and require more inner iterations. Also the time for each residual evaluation is slightly higher for the high-order methods, since the high-order methods have a larger stencil.[b]

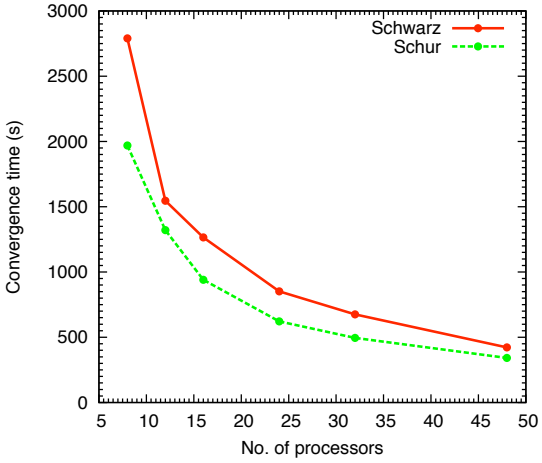
[b]See Section VII for the high-order finite-difference operators.



(a) 2^{nd} order



(b) 3^{rd} order



(c) 4^{th} order

Figure 8. Comparison of Schwarz and Schur precondtioners

The lift and drag coefficients are shown in Table 4. The lift and drag coefficients agree within 0.2% and 4% of each other, respectively. The C_p plots at different cross sections on a coarse and fine grid are shown in Figure 11. The coarse grid consists of 96 blocks$\times$25$\times$25$\times$25 nodes and the fine grid consists of 96 blocks$\times$33$\times$33$\times$33 nodes. The experimental C_p data is included as well, and even though viscous effects are neglected, the agreement is quite good.

A zoomed-in C_p plot for the 20% cross section is shown in Figure 12. We see that near the leading edge the high-order solutions display improved accuracy, where we assume that the fourth-order solution on the fine mesh is the most accurate solution available. Also at the mesh block interface on the wing, the discontinuity in C_p associated with the use of SATs is reduced noticeably by the higher-order methods. However, at the aftmost location shown the coarse mesh second-order solution is closer to the fine mesh solution.

The downstream preservation of the wing tip vortices is another measure of the accuracy of the solution. For the subsonic test case we examine the magnitude of vorticity 22 chord lengths behing the wing. Data for a coarse (96 blocks$\times$25$\times$25$\times$25 nodes) and a fine (96 blocks$\times$33$\times$33$\times$33 nodes) mesh for the three orders of accuracy are shown in Figure 13. The data demonstrate that the higher-order methods preserve the trailing vortices better than the second-order method.

Based on the lift and drag coefficients, there seems to be little advantage to the high-order methods, in contrast to the results obtained for the Ringleb flow (Figure 7) and the vorticity study (Figure 13). Further study is needed to clarify this.

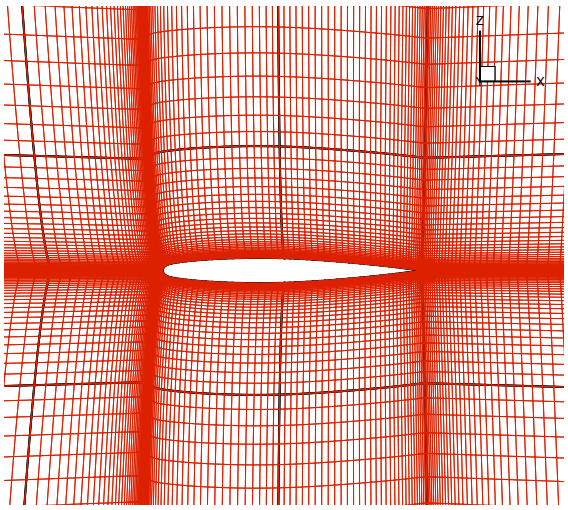


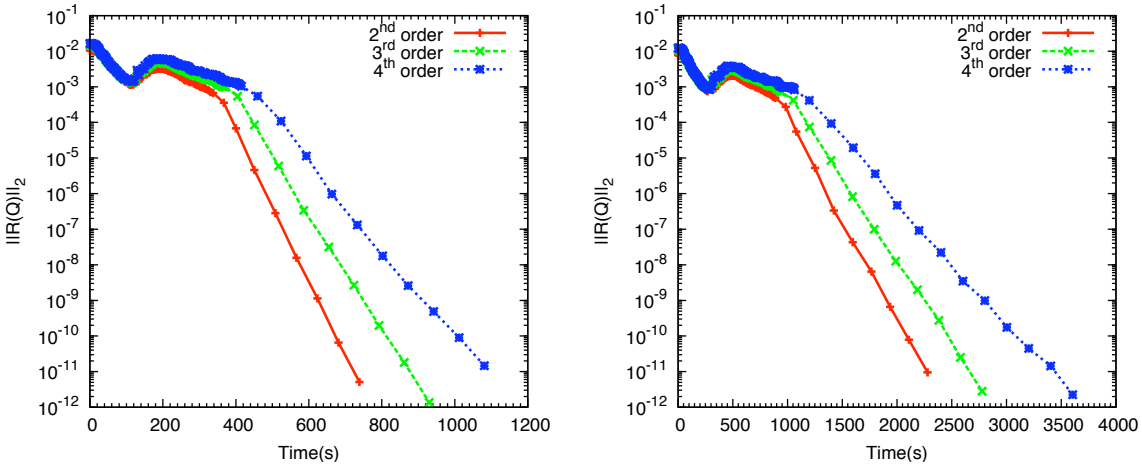
Figure 9. Mesh for the ONERA M6 wing (Symmetry plane)

V. Conclusions

A high-order finite-difference flow solver is presented for the Euler equations. Spatial discretization is carried out using summation-by-parts operators; the boundary and interface conditions are implemented using simultaneous approximation terms. The solution is marched to steady state using a two-phase Newton-Krylov approach. The first phase uses a first-order accurate flow Jacobian in an approximate-Newton approach. Once a suitable iterate is found, we use the high-order Jacobian in the inexact-Newton phase. Force integration is performed using Simpson's rule. The code is verified using the method of manufactured solutions and Ringleb flow. It is found that the observed order of accuracy agrees with the prescribed order of accuracy. An efficiency study for the Ringleb flow demonstrates that the high-order methods are indeed efficient in that they produce a low error per computational cost. Lastly, the code was validated for the ONERA M6 wing. The combination of a high-order spatial discretization using SAT's with a parallel Newton-Krylov solution technique is an efficient approach to computing aerodynamic flows. Extension of the algorithm to viscous and turbulent flows is an avenue for future work.

Test case	Total time(s)	Residual evaluation time(s)	Inner iterations	Equivalent residual evaluations
96 blocks×17×17×17 nodes				
2^{nd} order	220.3	0.06009	876	3666
3^{rd} order	244.3	0.10480	871	2331
4^{th} order	342.0	0.10879	1136	3143
96 blocks×21×21×21 nodes				
2^{nd} order	441.2	0.11335	1006	3892
3^{rd} order	506.6	0.19353	1021	2617
4^{th} order	643.9	0.20467	1242	3146
96 blocks×25×25×25 nodes				
2^{nd} order	738.6	0.19383	1064	3810
3^{rd} order	930.1	0.33273	1179	2795
4^{th} order	1081	0.34858	1325	3101
96 blocks×29×29×29 nodes				
2^{nd} order	1301	0.30694	1212	4238
3^{rd} order	1497	0.56579	1234	2645
4^{th} order	1962	0.55285	1520	3548
96 blocks×33×33×33 nodes				
2^{nd} order	2279	0.46370	1285	4914
3^{rd} order	2777	0.79640	1379	3486
4^{th} order	3604	0.83765	1704	4302

Table 3. Convergence summary for the ONERA M6 wing - M=0.699, α=3.06°

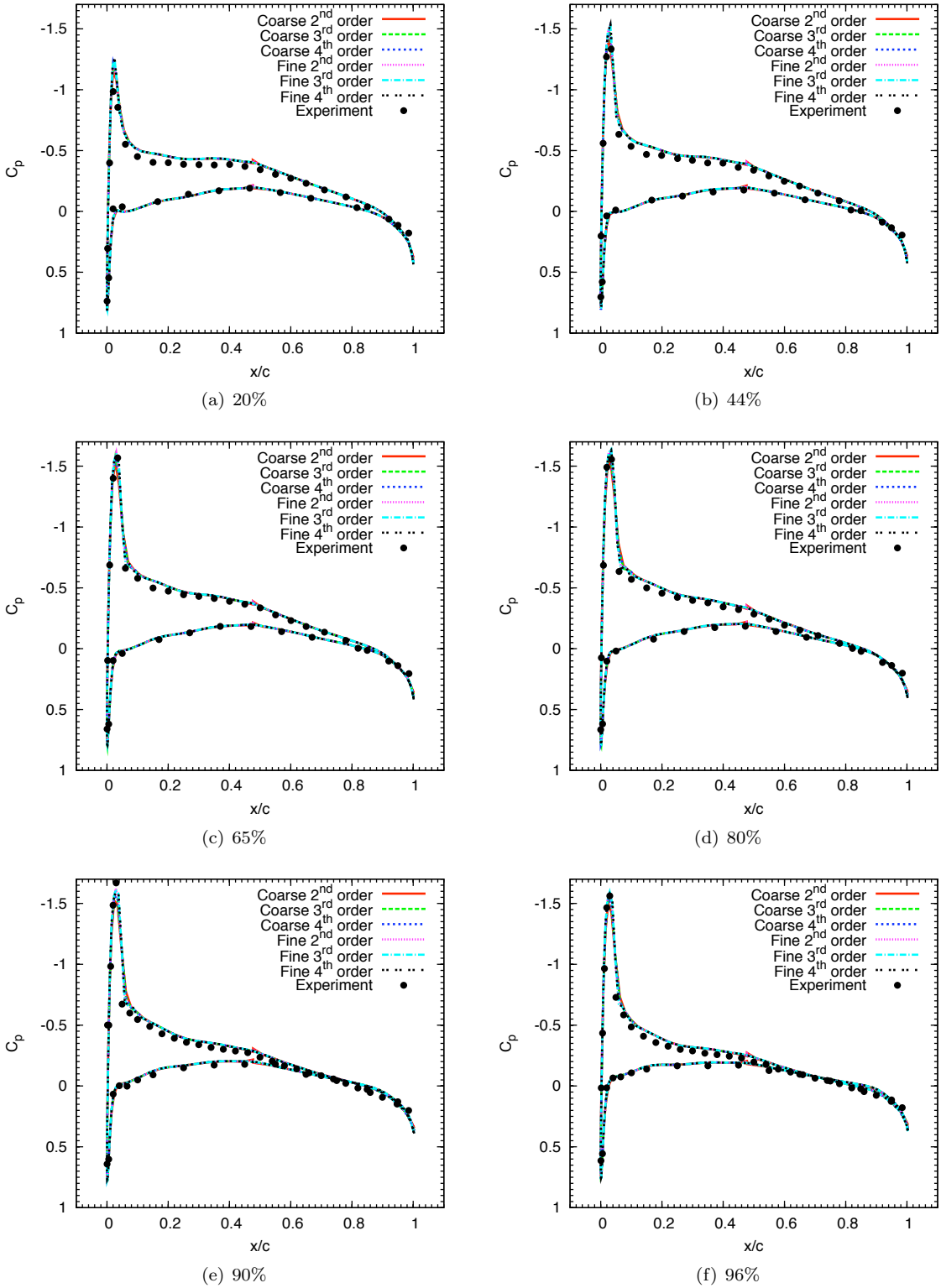


(a) 96 blocks×25×25×25 (b) 96 blocks×33×33×33

Figure 10. Convergence history for the ONERA M6 wing - M=0.699, α=3.06°

Test case	Lift coefficient	Drag coefficient
96 blocks×17×17×17 nodes		
2^{nd} order	0.27763455	0.0065825086
3^{rd} order	0.27567148	0.0075857959
4^{th} order	0.27322270	0.0074356907
96 blocks×21×21×21 nodes		
2^{nd} order	0.27848939	0.0063827045
3^{rd} order	0.27709814	0.0070536488
4^{th} order	0.27642214	0.0068295743
96 blocks×25×25×25 nodes		
2^{nd} order	0.27921397	0.0062418883
3^{rd} order	0.27821599	0.0067114483
4^{th} order	0.27779249	0.0066318836
96 blocks×29×29×29 nodes		
2^{nd} order	0.27967114	0.0061734778
3^{rd} order	0.27892535	0.0065268452
4^{th} order	0.27872654	0.0064820850
96 blocks×33×33×33 nodes		
2^{nd} order	0.27997232	0.0061391219
3^{rd} order	0.27940949	0.0064173196
4^{th} order	0.27924200	0.0063872851

Table 4. Drag and lift coefficients for the ONERA M6 wing - M=0.699, α=3.06°



(a) 20%

(b) 44%

(c) 65%

(d) 80%

(e) 90%

(f) 96%

Figure 11. Pressure coefficient at different cross sections for the ONERA M6 wing - M=0.699, α=3.06°

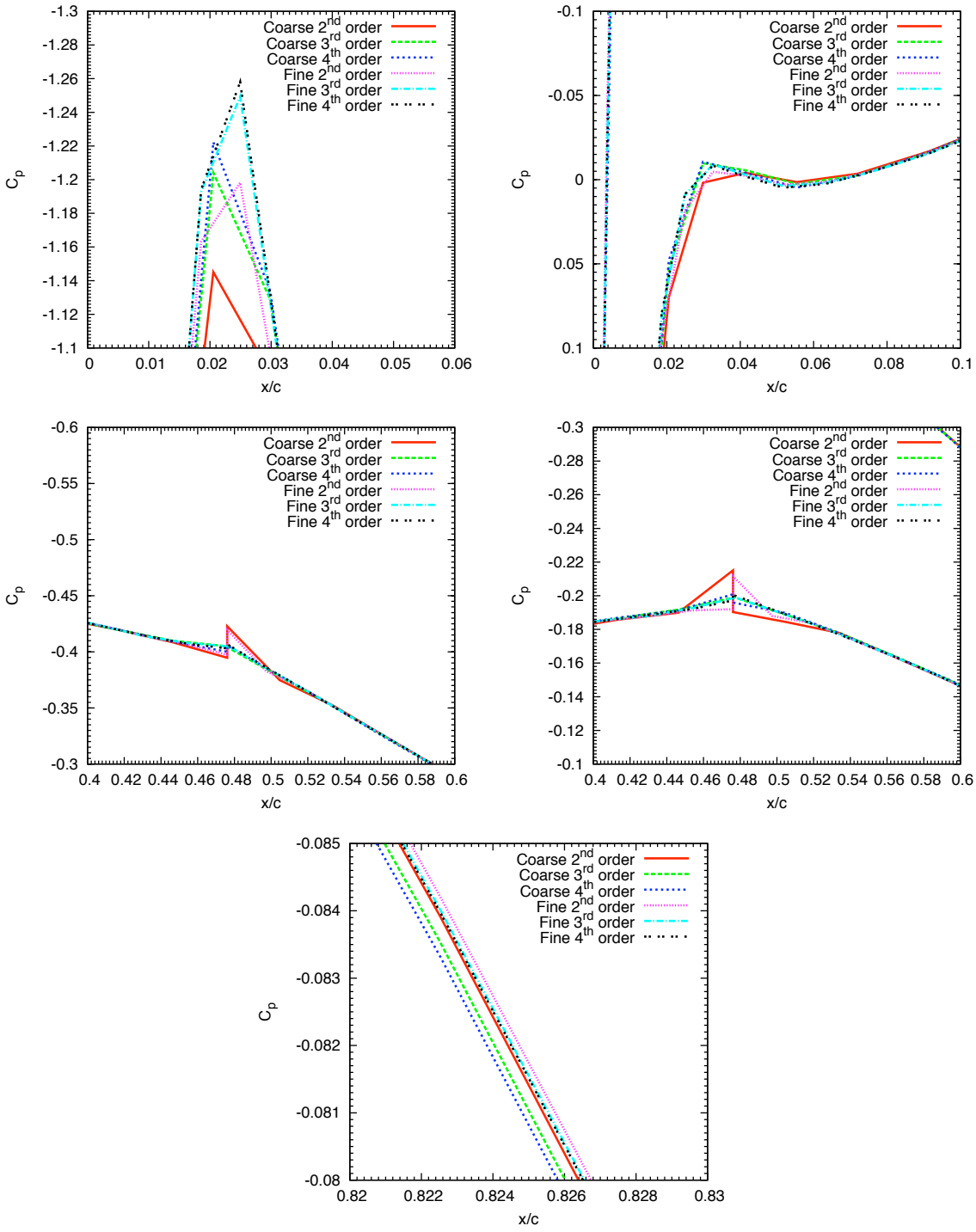
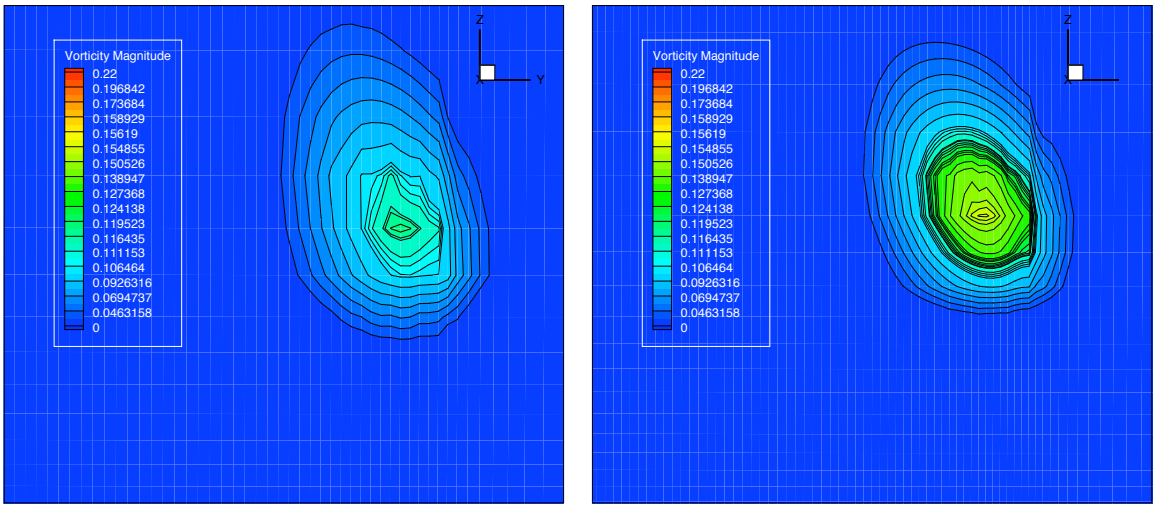


Figure 12. Zoomed in pressure coefficient plots at the 20% cross section for the ONERA M6 wing - M=0.699, α=3.06°



(a) 2nd order, 96 blocks×25×25×25

(b) 2nd order, 96 blocks×33×33×33

(c) 3rd order, 96 blocks×25×25×25

(d) 3rd order, 96 blocks×33×33×33

(e) 4th order, 96 blocks×25×25×25

(f) 4th order, 96 blocks×33×33×33

Figure 13. Vorticity contours 22 chord lengths behind the wing - M=0.3, α=4.0°

VI. Acknowledgements

The first author would like to thank Jason Hicken and Ken Mattsson for explaining their implementation of summation-by-parts and dissipation operators. The authors gratefully acknowledge financial assistance from the Mathematics of Information Technology and Complex Systems (MITACS), Bombardier Aerospace, Natural Sciences and Engineering Research Council (NSERC), the Canada Research Chairs program and the University of Toronto.

VII. High-order finite-difference operators

VII.A. Third-order method

$$H = \begin{pmatrix} \frac{17}{48} & & & & & & \\ & \frac{59}{48} & & & & & \\ & & \frac{43}{48} & & & & \\ & & & \frac{49}{48} & & & \\ & & & & 1 & & \\ & & & & & \ddots & \end{pmatrix}$$

$$D = \frac{1}{\Delta x} \begin{pmatrix} -\frac{24}{17} & \frac{59}{34} & -\frac{4}{17} & -\frac{3}{34} & & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & & & & \\ \frac{4}{43} & -\frac{59}{86} & 0 & \frac{59}{86} & -\frac{4}{43} & & & \\ \frac{3}{98} & 0 & -\frac{59}{98} & 0 & \frac{32}{49} & -\frac{4}{49} & & \\ 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

VII.B. Fourth-order method

$$H = \begin{pmatrix} \frac{13649}{43200} & & & & & & & \\ & \frac{12013}{8640} & & & & & & \\ & & \frac{2711}{4320} & & & & & \\ & & & \frac{5359}{4320} & & & & \\ & & & & \frac{7877}{8640} & & & \\ & & & & & \frac{43801}{43200} & & \\ & & & & & & 1 & \\ & & & & & & & \ddots \end{pmatrix}$$

$$D = \frac{1}{\Delta x} \begin{pmatrix} -\frac{21600}{13649} & \frac{81763}{40947} & \frac{131}{27298} & -\frac{9143}{13649} & \frac{20539}{81894} & & & & & \\ -\frac{81763}{180195} & 0 & \frac{7357}{36039} & \frac{30637}{72078} & -\frac{2328}{12013} & \frac{6611}{360390} & & & & \\ -\frac{131}{54220} & -\frac{7357}{16266} & 0 & \frac{645}{2711} & \frac{11237}{32532} & -\frac{3487}{27110} & & & & \\ -\frac{9143}{53590} & -\frac{30637}{64308} & -\frac{645}{5359} & 0 & \frac{13733}{32154} & -\frac{67}{4660} & \frac{72}{5359} & & & \\ -\frac{20539}{236310} & \frac{2328}{7877} & -\frac{11237}{47262} & -\frac{13733}{23631} & 0 & \frac{89387}{118155} & -\frac{1296}{7877} & \frac{144}{7877} & & \\ 0 & -\frac{6611}{262806} & \frac{3487}{43801} & \frac{1541}{87602} & -\frac{89387}{131403} & 0 & \frac{32400}{43801} & -\frac{6480}{43801} & \frac{720}{43801} & \\ & & & -\frac{1}{60} & \frac{3}{20} & -\frac{3}{4} & 0 & \frac{3}{4} & -\frac{3}{20} & \frac{1}{60} \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

References

[1]Brasseur, G., Cox, R., Hauglustaine, D., Isaksen, I., Lelieveld, J., Lister, D., Sausen, R., Schumann, U., Wahner, A., and Wiesen, P., "European scientific assessment of the atmospheric effects of aircraft emissions," *Atmospheric Environment*, Vol. 32, No. 13, July 1998, pp. 2329–2418.

[2]Penner, J. E., *Aviation and the Global Atmosphere*, Cambridge University Press, 1999.

[3]Hicken, J. E. and Zingg, D. W., "An investigation of induced drag minimization using a Newton-Krylov algorithm," *AIAA paper 2008-5807*, 2008.

[4]Zingg, D. W., Nemec, M., and Pulliam, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, 2002, pp. 1057–1065.

[5]Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum aerodynamic design using the Navier-Stokes equations," *Theoretical and Computational Fluid Dynamics*, , No. 10, 1998, pp. 213–237.

[6]Nielsen, E. J. and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations," *AIAA paper 98-4809*, 1998.

[7]Zingg, D. W. and Elias, S., "On Aerodynamic Optimization Under a Range of Operating Conditions," *AIAA Journal*, Vol. 44, 2006, pp. 2787–2792.

[8]Zingg, D. W., Diosady, L., and Billing, L., "Adaptive Airfoils for Drag Reduction at Transonic Speeds," *AIAA paper 2006-3656*, 2006.

[9]Nemec, M. and Zingg, D., "Multipoint and Multi-Objective aerodynamics shape optimization," *AIAA Journal*, Vol. 42, 2004, pp. 1057–1065.

[10]Zingg, D. W., Leung, T., Diosady, L., Truong, A. H., Elias, S., and Nemec, M., "Improvements to a Newton-Krylov algorithm for aerodynamic optimization," *AIAA paper 2005-4857*, 2005.

[11]Zingg, D. W. and Billing, L., "Toward practical aerodynamic design through numerical optimization," *AIAA paper 2007-3950*, 2007.

[12]Rumpfkeil, M. P. and Zingg, D. W., "Unsteady optimization using a discrete adjoing approach applied to aeroacoustis shape design," *AIAA paper 2008-18*, 2008.

[13]Hicks, R. M. and Henne, P. A., "Wing design by numerical optimization," *Journal of Aircraft*, Vol. 15, No. 7, July 1978, pp. 407–412.

[14]Jameson, A., "Aerodynamic design via control theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260.

[15]Burgreen, G. W. and Baysal, O., "Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis," Vol. 34, No. 9, Sept. 1996, pp. 1761–1770.

[16]Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., "Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation," *AIAA paper 1996-0094*, 1996.

[17]Nejat, A. and Ollivier-Gooch, C., "Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations," *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2366–2386.

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

[19]Nejat, A. and Ollivier-Gooch, C., "A High-Order Accurate unstructured Newton-Krylov solver for inviscid compressible flows," *AIAA paper 2006-3711*, 2006.

[20]Nejat, A. and Ollivier-Gooch, C., "On preconditioning of Newton-GMRES algorithm for a high-order accurate unstructured solver," *Proceedings of the Fourteenth Annual Conference of the Computational Fluid Dynamics Society of Canada*, 2006.

[21]Nejat, A., Ollivier-Gooch, C., and Michalak, K., "Accuracy Assessment Methodology for a Higher-Order Unstructured Finite Volume Solver," *AIAA paper 2007-4194*, 2007.

[22]Vaassen, J. M., Vigneron, D., and Essers, J. A., "An implicit high order finite volume scheme for the solution of 3D Navier-Stokes equations with new discretization of diffusive terms," *Journal of Computational and Applied Mathematics*, Vol. 215, No. 2, 2008, pp. 595–601.

[23]Fidkowski, K. J. and Darmofal, D. L., "A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 225, No. 2, 2007, pp. 1653–1672.

[24]Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, No. 1, 2005, pp. 92–113.

[25]Fidkowski, K. J., *A High-Order Discontinuous Galerkin Multigrid Solver for aerodynamic applications*, Master's thesis, MIT, 2004.

[26]Fidkowski, K. J., *A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations*, Ph.D. thesis, MIT, 2007.

[27]De Rango, S. and Zingg, D. W., "A High-Order Spatial Discretization for Turbulent Aerodynamic Computations," *AIAA Journal*, Vol. 39, 2001, pp. 1296–1304.

[28]Zingg, D. W., De Rango, S., Nemec, M., and Pulliam, T. H., "Comparison of several spatial discretizations for the Navier–Stokes equations," *AIAA paper 99-3269*, 1999.

[29]Jurgens, H. M. and Zingg, D. W., "Numerical Solution of the Time-Domain Maxwell Equations Using High-Accuracy Finite-Difference Methods," *SIAM Journal on Scientific Computing*, Vol. 22, No. 5, 2001, pp. 1675–1696.

[30]Zingg, D. W., "Comparison of High-Accuracy Finite-Difference Schemes for Linear Wave Propagation," *SIAM Journal on Scientific Computing*, Vol. 22, No. 2, 2000, pp. 476–502.

[31]Zingg, D. W., Lomax, H., and Jurgens, H., "High-Accuracy Finite-Difference Schemes for Linear Wave Propagation," *SIAM Journal on Scientific Computing*, Vol. 17, No. 2, 1996, pp. 328–346.

[32]Svärd, M., Carpenter, M. H., and Nordström, J., "A stable high-order finite difference scheme for the compressible Navier-Stokes equations, far-field boundary conditions," *Journal of Computational Physics*, Vol. 225, No. 1, 2007, pp. 1020–1038.

[33]Mattsson, K., Svärd, M., Carpenter, M., and Nordström, J., "High-order accurate computations for unsteady aerodynamics," *Computers & Fluids*, Vol. 36, No. 3, March 2007, pp. 636–649.

[34]De Rango, S. and Zingg, D. W., "Further investigation of a higher-order algorithm for aerodynamic computations," *AIAA paper 2000-0823*, 2000.

[35]Pulliam, T. H., *Efficient solution methods for the Navier-Stokes equations-Lecture notes for the Von Karman institute for fluid dynamics lecture series: Numerical techniques for viscous flow computation in turbomachinery bladings*, 1986.

[36]Visbal, M. R. and Gaitonde, D. V., "On the use of higher-order finite-difference schemes on curvilinear and deforming meshes," *Journal of Computational Physics*, Vol. 181, No. 1, 2002, pp. 155–185.

[37]Lehner, L., Reula, O., and Tiglio, M., "Multi-block simulations in general relativity: high-order discretizations, numerical stability and applications," *Classical and Quantum Gravity*, Vol. 22, No. 24, 2005, pp. 5283–5321.

[38]Diener, P., Dorband, E. N., Schnetter, E., and Tiglio, M., "Optimized High-Order Derivative and Dissipation Operators Satisfying Summation by Parts, and Applications in Three-dimensional Multi-block Evolutions," *Journal of Scientific Computing*, Vol. 32, No. 1, 2007, pp. 109–145.

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

[40]Mattsson, K. and Nordström, J., "Summation by parts operators for finite difference approximations of second derivatives," *Journal of Computational Physics*, Vol. 199, No. 2, 2004, pp. 503–540.

[41]Olsson, P., "Summation by parts, projections, and stability I," *Mathematics of Computation*, Vol. 64, No. 211, 1995, pp. 1035–1065.

[42]Olsson, P., "Summation by parts, projections, and stability. II," *Mathematics of Computation*, Vol. 64, No. 212, 1995, pp. 1473–1493.

[43]Svärd, M., "On Coordinate Transformations for Summation-by-Parts Operators," *Journal of Scientific Computing*, Vol. 20, No. 1, 2004, pp. 29–42.

[44]Hixon, R., "Numerically Consistent Strong Conservation Grid Motion for Finite Difference Schemes," *AIAA Journal*, Vol. 38, 2000, pp. 1586–1593.

[45]Thomas, P. D. and Lombard, C. K., "Geometric Conservation Law and its Application to Flow Computations on Moving Grids," *AIAA Journal.*, Vol. 17, 1979, pp. 1030–1037.

[46]Mattsson, K., Svärd, M., and Nordström, J., "Stable and Accurate Artificial Dissipation," *Journal of Scientific Computing*, Vol. 21, No. 1, 2004, pp. 57–79.

[47]Dias, S. C., *A high-order, parallel, Newton-Krylov flow solver for the Euler equations*, Master's thesis, University of Toronto, 2008.

[48]Swanson, R. C. and Turkel, E., "On Central-Difference and Upwind Schemes," *Journal of Computational Physics*, Vol. 101, 1992, pp. 292–306.

[49]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," *AIAA paper 81-1259*, 1981.

[50]Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations," *AIAA Journal*, Vol. 24, 1986, pp. 1931–1940.

[51]Olsson, P. and Johnsson, S. L., "A study of dissipation operators for the Euler equations and a three-dimensional channel flow," *Supercomputing '89: Proceedings of the 1989 ACM/IEEE conference on Supercomputing*, ACM, New York, NY, USA, 1989, pp. 141–151.

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

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

[54]Nordström, J. and Carpenter, M. H., "Boundary and interface conditions for high-order finite-difference methods applied to the Euler and Navier-Stokes equations," *Journal of Computational Physics*, Vol. 148, No. 2, 1999, pp. 621–645.

[55]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.

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

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

[58]Saad, Y. and Schultz, M. H., "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *Journal of Scientific Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[59]Barth, T. J. and Linton, S. W., "An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation," *AIAA paper 95-0221*, 1995.

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

[61]Nichols, J. C., *A three-multidimensional multi-block Newton-Krylov flow solver for the Euler equations*, Master's thesis, University of Toronto, 2004.

[62]Saad, Y., *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[63]Cuthill, E. and McKee, J., "Reducing the bandwidth of sparse symmetric matrices," *Proceedings of the 1969 24th national conference*, 1969, pp. 157–172.

[64]Bregains, J. C., Coleman, I. C., Ares, F., and Moreno, E., "Calculating Directivities with the Two-Dimensional Simpson's Rule," *IEEE Antennas and Propagation Magazine*, Vol. 46, 2004, pp. 106–112.

[65]Roache, P. J., *Verification and validation in computational science and engineering*, Hermosa, 1998.

[66]Roy, C. J., Nelson, C. C., Smith, T., and Ober, C. C., "Verification of Euler/Naver-Stokes codes using the method of manufactured solutions," *International Journal for Numerical Methods in Fluids*, Vol. 44, 2004, pp. 599–620.

[67]Salas, M. D., "Some observations on grid convergence," *Computers & Fluids*, Vol. 35, No. 7, Aug. 2006, pp. 688–692.

[68]Ivan, L. and Groth, C., "High-order central ENO finite-volume scheme with adaptive mesh refinement," *AIAA paper 2007-4323*, 2007.

[69]Coirier, W. J. and Powell, K. G., "An accuracy assessment of Cartesian-mesh approaches for the Euler equations," *Journal of Computational Physics*, Vol. 117, No. 1, 1995, pp. 121–131.

[70]Chiocchia, G., "Exact solutions to transonic and supersonic flows," *AGARD*, Vol. AGARDograph No. 211, 1985.