AIAA Aviation
13-17 June 2016, Washington, D.C.
46th AIAA Fluid Dynamics Conference

AIAA 2016-3335

A fully-implicit, Giles-type nonreflecting boundary condition in a DG-Chimera turbomachinery solver

Nathan A. Wukie[*], Paul D. Orkwis[†], and Mark G. Turner[‡]
University of Cincinnati, Cincinnati, Ohio, 45221

Development of a fully-implicit, Giles-type nonreflecting boundary condition in a computational fluid dynamics solver using a discontinuous Galerkin method on Chimera overset grids is demonstrated. The implicit formulation is in the context of Newton's method. The implementation was completed without the use of ghost elements or extra degrees of freedom in the system of equations and the boundary condition is not lagged. Rather, the elements associated with the nonreflecting boundary are fully-coupled with each other in the Newton linearization. The linearization of the governing equations and boundary conditions is achieved via an automatic differentiation technique based on operator overloading. Calculations of a turbine cascade geometry using truncated and extended domains were computed. Comparison of pressure and flow angles at the exit plane location between the truncated and extended domains were very well matched. All calculations converged in 4-7 Newton iterations up through 4^{th}-order accuracy.

Nomenclature

$\boldsymbol{Q}$	Solution vector
$\vec{F^a}$	Advective flux vector
$\vec{n}$	Normal vector
ψ	Legendre basis polynomial
Ω	Element volume
Φ	Characteristic variables
c	Speed of sound
T	Primitive to characteristic variable transformation
P	Periodicity

I. Introduction

The discontinuous Galerkin(DG) method for solving partial differential equations has been applied in many fields including acoustics, numerical weather prediction, and aerodynamics.[1–3] Part of the strength of the method exists in its ability to provide arbitrarily high-order accuracy, while maintaining a local computational stencil. It also benefits from taking advantage of a great body of work from the Finite Volume community in its use of numerical fluxes to couple the solution between discontinuous elements.

Turbomachinery is an integral part of the aviation and power generation industries and their designers face constant pressure to continuously improve metrics such as efficiency, pressure ratio, and noise radiation. These improvements are increasingly realized by applying advanced analysis capabilities to better inform designers on the impacts of design changes and provide direction for improving future designs. This requires that analysis tools be continuously improved in their accuracy, speed, and usability. The present work supports these goals in further development of an implicit, high-order analysis capability for turbomachinery and acoustics applications.

[*] PhD Student, Dept. of Aerospace Engineering, ML 70, Cincinnati, Ohio 45221, AIAA Student Member.
[†] Bradley Jones Professor, Dept. of Aerospace Engineering, ML 70, Cincinnati, Ohio 45221, AIAA Associate Fellow.
[‡] Associate Professor, Dept. of Aerospace Engineering, ML 70, Cincinnati, Ohio 45221, AIAA Associate Fellow.

I.A. Implicit, DG-Chimera framework

Previous work has been completed on an implicit, DG framework that uses Chimera overset grids for representing complex geometry and Newton's method for solving nonlinear systems of equations.[4] The framework was implemented with an intrinsic automatic differentiation capability which greatly reduces the effort required to implement new equation sets and boundary conditions while still maintaining Newton convergence of the implicit scheme.

I.A.1. Chimera gridding capability

The DG method is often used with unstructured computational grids because of its local scheme. However, the properties of DG are also very attractive for Chimera overset grids. Galbraith[5] developed an implicit, DG-Chimera solver that highlighted the benefits of DG applied to Chimera overset grids. This included the elimination of orphan nodes and fringe points that traditionally exist in Finite Volume-based Chimera solvers. Additionally, the ability of DG to accommodate curved geometry solved a Finite Volume issue with overlapping grids, where a boundary from one grid can exist inside the computational domain of a neighbor grid. This makes the DG-Chimera method quite attractive as a framework for a design tool, as it brings together the desired traits of a high-order accurate solution with straight-forward geometry representation and modification.

I.A.2. Automatic differentiation capability

Newton's method requires a linearization of the spatial scheme to compute the update in the solution state. In order to achieve quadratic convergence the linearization must be computed exactly, including a linearization of the boundary conditions. Such expressions can be derived by-hand or by using symbolic manipulation tools to simplify the process.[6] Galbraith[5] provides an extensive discussion of the process for deriving and constructing the linearization in the context of a discontinuous Galerkin framework. However, the manual method of deriving and implementing the linearizations is a time-consuming and error-prone process.

The previously developed framework used in the present work computes the linearization of the spatial scheme using an automatic differentiation technique based on operator-overloading advocated by Galbraith et al.[7] As a result, new equation sets and boundary conditions only require that their contributions to the spatial residual, $R(Q)$, be implemented. The linearization of those functions $\frac{\partial R}{\partial Q}$ is computed via the chain rule in the overloaded arithmetic operators and automatically stored to the linearization matrix. This greatly reduces the demand on developers for implementing new features, such as boundary conditions, while maintaining valuable properties of the scheme, such as Newton convergence.

I.B. Implicit, nonreflecting boundary conditions

Solving sets of partial differential equations requires boundary conditions that specify the solution behavior along the boundaries of the computational domain. In the case of waves from the interior solution interacting with a boundary condition, the possibility exists that those waves will be reflected in a non-physical way; creating spurious reflections and contaminating the solution. Nonreflecting boundary conditions attempt to address this issue by recognizing outgoing waves and allowing them to be transmitted from the domain. Some applications allow a form of absorbing layer, such as a Perfectly Matched Layer[8] or grid stretching to dampen reflections. However, for internal flows and particularly in the case of turbomachinery blade-row calculations, boundary conditions must often be placed very close to leading/trailing edge regions and must also transmit information to neighboring blade-rows, in which case these techniques are either inconvenient or impossible to use.

Engquist and Majda[9] performed some of the first work on nonreflecting boundary conditions and Giles[10] extended those concepts for the Euler equations, while presenting them in a more friendly form for researchers with an engineering background. Saxer and Giles[11] extended Giles' original formulation, applying the technique independently at different radial locations on the boundary, which acted as a quasi three-dimensional nonreflecting boundary condition. Hall et al.[12] took a slightly different approach to developing a three-dimensional nonreflecting boundary condition for the Euler equations by considering the eigenmodes allowed by the governing equations and domain geometry. Moinier et al.[13] in-turn extended Hall's 3D, eigenmode technique for the Navier-Stokes equations.

These techniques, based on and extended from Giles' initial work, currently represent the most widely utilized methods for nonreflecting boundary conditions in the turbomachinery industry. However, a current issue with many fluid dynamics solvers for turbomachinery is that either the scheme is explicit and requires many iterations to achieve convergence, or the scheme is implicit but the boundary conditions are updated explicitly; crippling the convergence rate of the implicit scheme.

I.C. Present work

The present work addresses the issue of poor convergence by implementing a fully-implicit, Giles-type nonreflecting boundary condition into a DG-Chimera framework to enable accurate and efficient calculation of turbomachinery flows. This is enabled by the intrinsic automatic differentiation capability in the solver framework.

II. Numerical methodology

The following discussion details the numerical methodology for this investigation, including the discontinuous Galerkin discretization, solution technique, governing equation set, and boundary conditions. See Wukie et al.[4] for a more complete discussion of the framework.

II.A. The discontinuous Galerkin discretization

The framework is designed to handle equation sets in conservation form as

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \nabla \cdot \vec{F}(\boldsymbol{Q}) + S(\boldsymbol{Q}) = 0 \tag{1}$$

The solution variables are expressed as an expansion in basis functions constructed using a tensor product of one-dimensional Legendre polynomials as

$$\boldsymbol{Q}(t,x,y,z) = \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=0}^{N} \boldsymbol{Q}_{ijk}(t)\psi_i(x)\psi_j(y)\psi_k(z) \tag{2}$$

Multiplying by a column of test functions $\boldsymbol{\psi}$ and applying Gauss' Divergence Theorem gives our working form of the discontinuous Galerkin method as

$$\int_{\Omega_e} \boldsymbol{\psi}\frac{\partial \boldsymbol{Q}}{\partial t} d\Omega + \int_{\Gamma_e} \boldsymbol{\psi}\vec{F}^{*}\cdot \vec{n} d\Gamma - \int_{\Omega_e} \nabla\boldsymbol{\psi}\cdot \vec{F} d\Omega + \int_{\Omega_e} \boldsymbol{\psi}\vec{S} d\Omega = 0 \tag{3}$$

Numerical Gauss-Legendre quadrature is used to compute the spatial integrals in Equation 3. Specialization of this method for a particular set of equations requires the definition of solution variables $\boldsymbol{Q}$, the flux vector $\vec{F}$, and a choice of numerical flux $\vec{F}^{*}$. These components are defined for the Euler equations in the following sections.

II.B. Nonlinear solver

The governing equations are solved using a Quasi-Newton method, which provides quadratic convergence when solving systems of nonlinear equations. The formulation of the Quasi-Newton method is

$$\int_{\Omega_e} \psi \frac{\Delta Q}{\Delta \tau_e} d\Omega + \frac{\partial R(Q^n)}{\partial Q}\Delta Q = -R(Q^n) \tag{4}$$

where $R(Q)$ is the spatial residual from Equation 3 and $\frac{\partial R(Q)}{\partial Q}$ is the matrix containing the linearization of the spatial scheme. The Quasi-Newton method adds a pseudo-time term, where the step $\Delta\tau$ is computed element-wise as

$$\Delta\tau_e = \frac{CFL^n h_e}{\bar{\lambda_e}} \tag{5}$$

Here, $h_e = \sqrt[3]{\Omega_e}$, and $\bar{\lambda}_e = |\bar{U}_e| + \bar{c}$ is the mean characteristic speed. The CFL term is computed from the ratio of the initial and current residual norms as

$$CFL^n = CFL^0 \frac{||R(Q^0)||_2}{||R(Q^n)||_2} \tag{6}$$

This approach was originally proposed by Orkwis and McRae.[14] In this way, as the spatial residual converges to zero, the time-step increases to infinity and the original full Newton method is recovered.

The general method for computing the linearization utilizes an automatic differentiation technique that was detailed previously by Wukie et al.[4] The linearization for fully-coupled boundary conditions is discussed later in the present work. Solving for ΔQ, the solution is updated as

$$Q^{n+1} = Q^n + \Delta Q \tag{7}$$

Computing the update vector ΔQ requires the solution of a linear system of equations. This is accomplished with an implementation of the Generalized Minimum Residual(GMRES)[15,16] algorithm coupled with an ILU0 preconditioner.

II.C. Euler equations for nonlinear, inviscid flows

The Euler equations were implemented for this effort, where the solution and advective flux vectors are defined as

$$\boldsymbol{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \qquad \vec{F}(\boldsymbol{Q}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix}, \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{bmatrix}, \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho wH \end{bmatrix} \tag{8}$$

where $H = \frac{\rho E + p}{\rho}$ is the total enthalpy and $[u, v, w]$ correspond to the Cartesian velocity components $[U_x, U_y, U_z]$. The fluid is assumed to be an ideal gas and the pressure is computed as

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho \vec{U}^2) \tag{9}$$

The Roe scheme[17] is used to compute the numerical advective boundary fluxes $\vec{F}^*$.

II.D. Giles' 2D nonreflecting boundary condition

In this work, boundary conditions are imposed weakly through the boundary flux integral in Equation (3). The approach taken here is to construct a boundary condition solution state $\boldsymbol{Q}_{bc}$ and then the boundary flux is computed as $\vec{F}(\boldsymbol{Q}_{bc})$ using the definition in Equation (8).

Giles developed a two-dimensional, steady boundary condition that is well-suited for turbomachinery applications.[10] The analysis here is carried out in primitive variable form and the solution vector of primitive variables is given as $\boldsymbol{U} = [\rho, u, v, p]$. Giles' analysis performs a Fourier decomposition of the solution on the boundary giving

$$\boldsymbol{U}(x_b, y, t) = \sum_{-\infty}^{\infty} \hat{\boldsymbol{U}}(t) e^{i l_m y} \tag{10}$$

where $\hat{\boldsymbol{U}}$ are the complex Fourier coefficients of the Fourier transform and $l_m = 2\pi m/P$. Giles constructed a nonreflecting condition on the Fourier modes of the primitive variables and then reformulated it into a condition on Fourier modes of the characteristic variables $\hat{\Phi} = [\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3, \hat{\phi}_4]$ for implementation purposes. The transformation between primitive and characteristic variables is

$$\boldsymbol{U}' = T\Phi \tag{11}$$

where the transformation matrices are given as

$$T^{-1} = \begin{bmatrix} -\bar{c}^2 & 0 & 0 & 1 \\ 0 & 0 & \bar{\rho}\bar{c} & 0 \\ 0 & \bar{\rho}\bar{c} & 0 & 1 \\ 0 & -\bar{\rho}\bar{c} & 0 & 1 \end{bmatrix} \qquad T = \begin{bmatrix} -1/\bar{c}^2 & 0 & 1/(2\bar{c}^2) & 1/(2\bar{c}^2) \\ 0 & 0 & 1/(2\bar{\rho}\bar{c}) & 1/(2\bar{\rho}\bar{c}) \\ 0 & 1/(\bar{\rho}\bar{c}) & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix} \tag{12}$$

The present work focused only on an implementation of the outlet boundary condition. The inlet boundary condition may be considered in a later work. The nonreflecting condition for an outlet boundary is given in terms of the Fourier modes of the characteristic variables across the boundary as

$$\begin{bmatrix} 0 & -u & \frac{1}{2}(\beta+v) & \frac{1}{2}(\beta-v) \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \\ \hat{\phi}_4 \end{bmatrix} = 0 \tag{13}$$

where β is complex and is defined as

$$\beta = \begin{cases} i\ sign(l)\sqrt{1-u^2-v^2} & u^2+v^2<1 \\ -sign(v)\sqrt{u^2+v^2-1} & u^2+v^2>1 \end{cases} \tag{14}$$

A boundary solution state should be constructed from the incoming and outgoing characteristics. The outgoing characteristics are computed from the interior solution and should not be modified. As such, to impose the nonreflecting conditions of Equation (13) we solve the system for the characteristic that may be modified. This gives

$$\hat{\phi}_4 = \left(\frac{2u}{\beta-v}\right)\hat{\phi}_2 - \left(\frac{\beta+v}{\beta-v}\right)\hat{\phi}_3 \tag{15}$$

which is the practical form of the nonreflecting outlet boundary condition. The implementation of this condition often takes a slightly different form and it is the implementation details that differentiate the present work from previous studies.

The traditional implementation of the condition in Equation (15) is in the context of either a fully-explicit solver or an implicit solver where the boundary conditions are updated explicitly after each step. These techniques often use the concept of ghost cells to facilitate a 'lagging' of the update for the characteristic values that are used to compute the boundary solution state. Giles gives a recommended 'lagged' formulation as

$$\frac{\partial\hat{\phi}_4}{\partial t} = \alpha\left[\left(\frac{2u}{\beta-v}\right)\hat{\phi}_2 - \left(\frac{\beta+v}{\beta-v}\right)\hat{\phi}_3 - \hat{\phi}_4\right] \tag{16}$$

using the parameter α to maintain stability by scaling the update.

The present work solves the governing equations using Newton's method; including the boundary conditions in the Newton linearization. As such, all elements associated with the nonreflecting boundary are fully-coupled with each other while solving the linear system for the solution update, instead of being only locally coupled with the nearest neighboring element through the interior flux. As a result, no ghost elements or ghost solution states are used to lag the update of the characteristic values as would be done using Equation (16). Rather, Equation (15) is used directly. Stability is maintained because the solution is coupled across the boundary, which allows the Newton method to find the correct solution that satisfies the boundary condition.

The linearization of the coupled nonreflecting boundary condition is facilitated by an intrinsic automatic differentiation capability. As such, only the contribution of the boundary condition to the spatial residual $R(\boldsymbol{Q})$ is required and the linearization $\partial R/\partial \boldsymbol{Q}$ is computed automatically by the framework. The implementation of the boundary condition for the spatial residual $R(\boldsymbol{Q})$ will be discussed along with the automatic differentiation method used to compute the linearization.

II.D.1. Implementation methodology

First the solution along the nonreflecting outlet boundary is sampled at multiple points and a Discrete Fourier Transform(DFT) of the solution variables is computed giving the Fourier modes $\hat{\boldsymbol{U}}$. Since the DG method provides a continuous representation of the solution, there is no interpolation error in evaluating the solution at arbitrary points along the boundary in order to compute the DFT. The mean components of the variables over the boundary are taken from the first component of the Fourier transform giving $\bar{\boldsymbol{U}}$. The goal of the implementation is to allow a user to impose an average quantity that sets the boundary condition, while allowing the state across the boundary condition to vary. Here, this is accomplished by a user-specified value for the average pressure on the boundary. This is used to compute an update in the mean solution state across the boundary. Following Giles' implementation,[18] a ϕ_4 characteristic component is computed due to the update in the boundary mean pressure from 1D characteristic theory as

$$\phi_4^{1D} = -2(\bar{p} - p_{user}) \tag{17}$$

The update in the boundary mean solution is then computed as

$$\boldsymbol{U}'_{1D-char} = \begin{bmatrix} \rho' \\ u' \\ v' \\ p' \end{bmatrix} = \begin{bmatrix} \dfrac{1}{2\bar{c}^2} \\ -\dfrac{1}{2\bar{\rho}\bar{c}} \\ 0 \\ \dfrac{1}{2} \end{bmatrix} \phi_4^{1D} \tag{18}$$

Next the total perturbation component of the solution over the boundary was computed as

$$\boldsymbol{U}' = \boldsymbol{U} - \bar{\boldsymbol{U}} \tag{19}$$

The characteristic variables associated with the perturbations were then computed point-wise at DFT nodes as

$$\Phi = T^{-1}\boldsymbol{U}' \tag{20}$$

A Fourier transform of the characteristic variables was then taken giving $\hat{\Phi}$. Each mode of the $\hat{\phi}_4$ characteristic is then recomputed using Equation (15) as

$$\hat{\phi}_{4,i}^{+} = \left(\frac{2u}{\beta - v}\right)\hat{\phi}_{2,i} - \left(\frac{\beta + v}{\beta - v}\right)\hat{\phi}_{3,i} \qquad i = 1, 2, \ldots i_{max} \tag{21}$$

where i indicates the i^{th} Fourier mode and + indicates the modified mode due to the nonreflecting condition. This gives us a set of modes for the incoming characteristic ϕ_4 that satisfy the nonreflecting condition. The inverse transform of $\hat{\phi}_4^{+}$ is then computed to obtain discrete values of ϕ_4^{+} at boundary quadrature nodes. The characteristics $[\phi_1, \phi_2, \phi_3]$ are computed at quadrature nodes in the same manner as Equation (20). The perturbations in the solution variables due to Giles' nonreflecting condition on the modes are then computed as

$$\boldsymbol{U}'_{Giles} = T\Phi \tag{22}$$

Φ is composed as

$$\Phi = \begin{bmatrix} \phi_1^{-} \\ \phi_2^{-} \\ \phi_3^{-} \\ \phi_4^{+} \end{bmatrix} \tag{23}$$

where $-$ indicates the characteristic was computed unmodified from the interior solution and + indicates the characteristic was computed from the nonreflecting condition.

To compute the flux over an element face, as the DG method requires, a solution state $\boldsymbol{U}_{bc}$ must be constructed on the face. This is composed from the boundary-global mean value, the update in the boundary-global mean value from 1D characteristic theory, and the update in the local values across the boundary due to Giles' nonreflecting condition as

$$\boldsymbol{U}_{bc} = \bar{\boldsymbol{U}} + \boldsymbol{U}'_{1D-char} + \boldsymbol{U}'_{Giles} \tag{24}$$

II.D.2. Linearization of coupled boundary condition

The linearization of the boundary condition is computed in the context of the automatic differentiation capability. In this context, the linearization of a flux integral with respect to another element is computed automatically during the evaluation of the integral itself. In the case that an integral is influenced by multiple separate elements, the integral is evaluated multiple times; once with respect to each element.

As an example, a 3x3 grid is shown in Figure 1 with the fully-coupled Giles boundary condition applied at the right boundary. Considering the evaluation of the boundary condition flux for a single element, Figure 2 diagrams the process for computing the flux residual and flux linearization for Element 3 of the sample grid. The boundary condition flux for Element 3 is computed three times; allowing the automatic differentiation capability to differentiate the residual with respect to the solution in each element on the boundary. The resulting matrix structure is shown in Figure 3, where the boundary condition coupling results in additional off-diagonal blocks in the matrix that couple the solution update in those elements associated with the boundary condition.

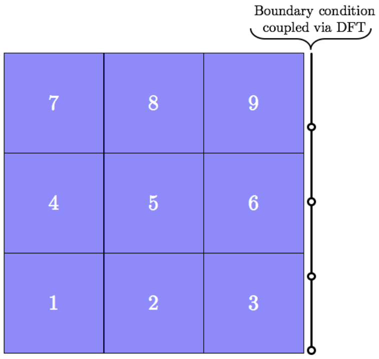


Figure 1. Example 3x3 grid with nonreflecting boundary condition.

III. Results

The results for this study include simulations of a turbine cascade using both truncated and extended domains in order to quantify the effectiveness of the nonreflecting boundary condition. The turbine geometry used was Stator 1 from the Aachen turbine rig geometry at approximately 50% span. Since the present calculations use only the Euler equations, the trailing edge of the original geometry was sharpened. Three simulations were run for this investigation. These are summarized in Table 1.

Configuration	Grid	Outlet boundary condition
1	Reference	Constant pressure
2	Short	Constant pressure
3	Short	Implicit Giles

Table 1. Simulation configurations

Figure 4 shows the computational grids used in this study. The multi-block grids were generated using AutoGrid5$^{\text{TM}}$ and higher-order elements were constructed by agglomerating elements. Block-to-block connectivity and communication occurs via the general Chimera interface, which allows abutting and overlapping elements. Additionally, non-matching periodic boundaries were implemented via the Chimera interface by

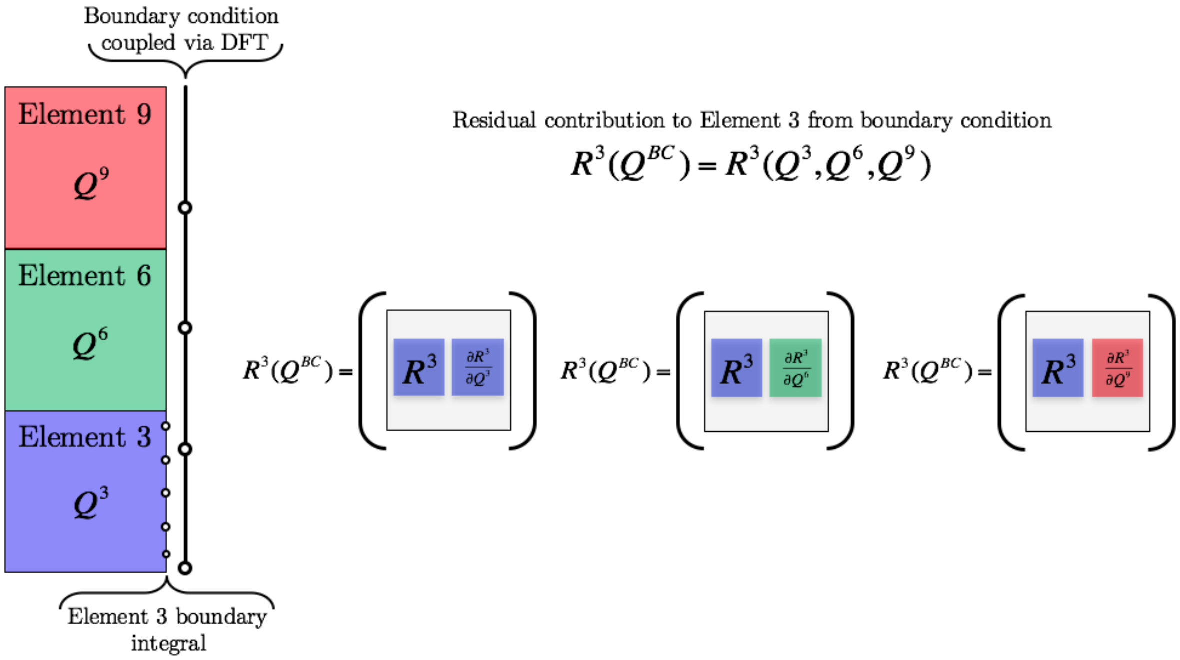
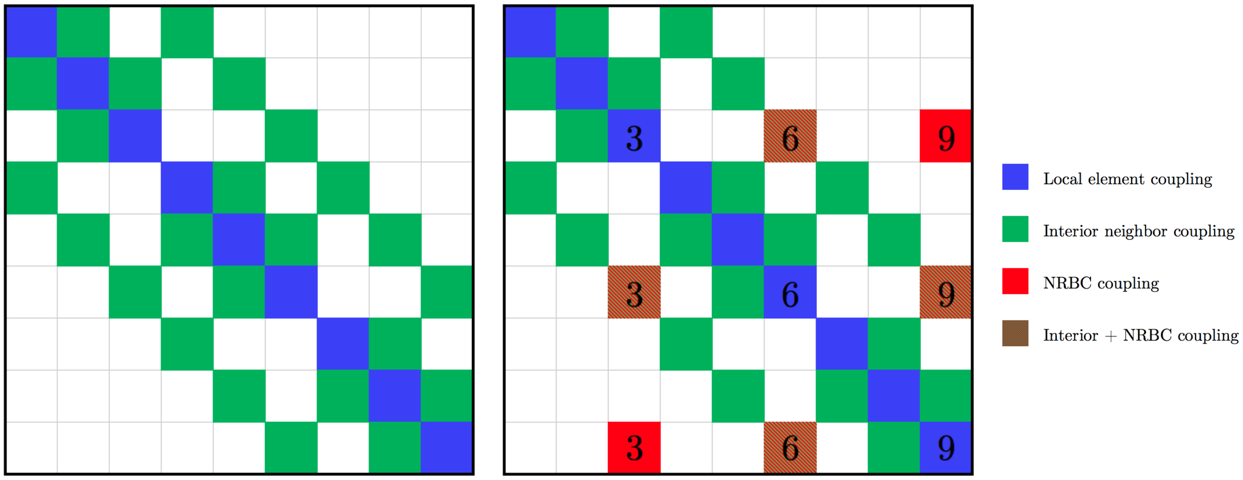

Boundary condition coupled via DFT
Element 9
Q^9
Element 6
Q^6
Element 3
Q^3
Element 3 boundary integral
Residual contribution to Element 3 from boundary condition
$R^3(Q^{BC}) = R^3(Q^3, Q^6, Q^9)$
$R^3(Q^{BC}) =$
R^3
$\frac{\partial R^3}{\partial Q^3}$
$R^3(Q^{BC}) =$
R^3
$\frac{\partial R^3}{\partial Q^6}$
$R^3(Q^{BC}) =$
R^3
$\frac{\partial R^3}{\partial Q^9}$

Figure 2. Linearization of boundary integral with respect to coupled elements using automatic differentiation.

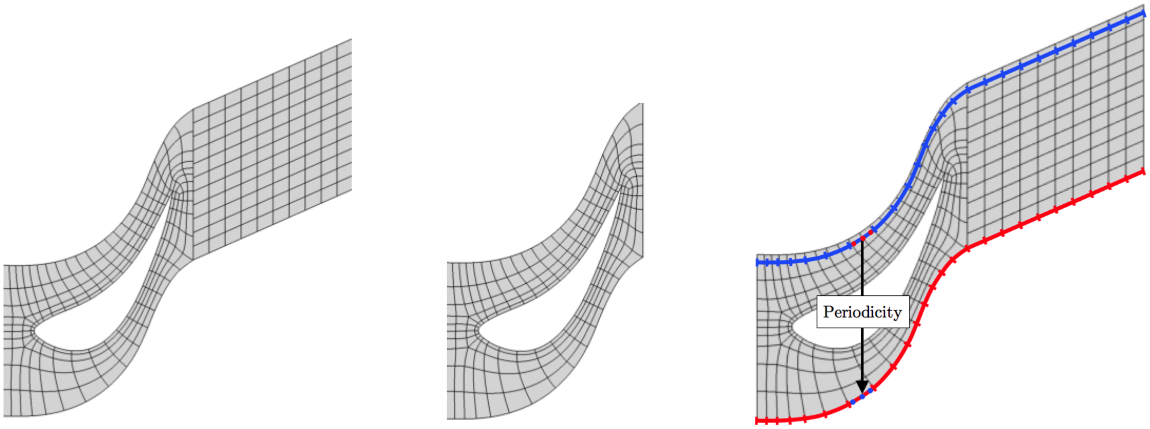


(a) Locally coupled boundary condition

(b) Fully-coupled boundary condition

Figure 3. Matrix structure. Red blocks result of coupling across outlet boundary condition.

simply offsetting the coordinates of the periodic face by the periodicity when searching for Chimera donor elements. This is diagrammed in Figure 4(c).



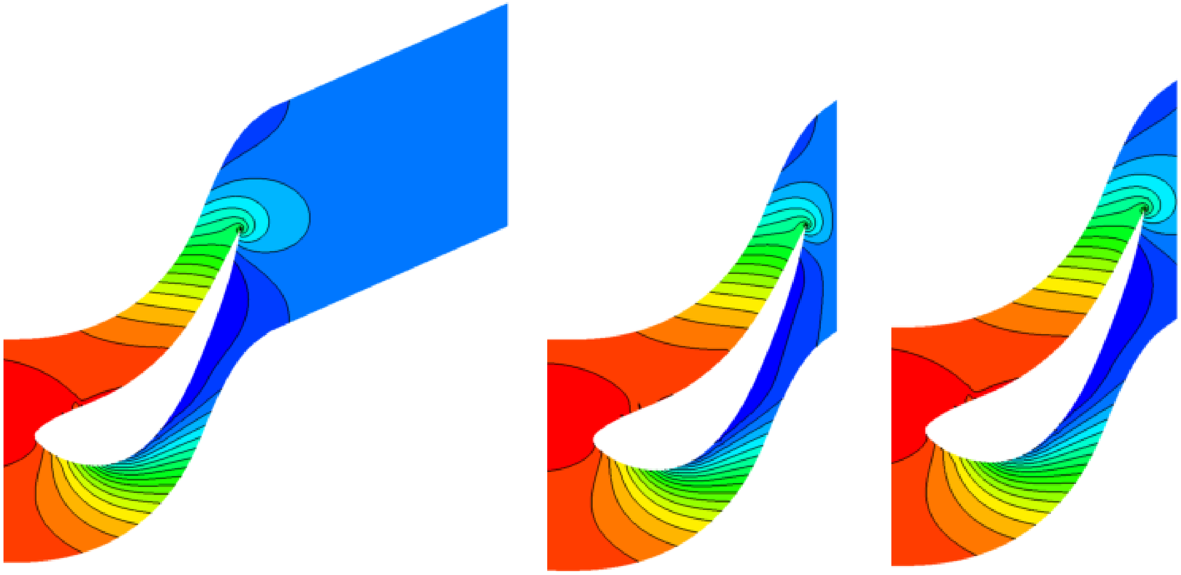
(a) Long domain

(b) Short domain

(c) Chimera-based periodic boundary

Figure 4. Stator cascade grids using quartic elements.

These simulations were run using a pressure ratio of $p/p_o = 0.91$ and $CFL^0 = 8$ was used for the Quasi-Newton solver. Pressure contours of the results are shown in Figure 5. Figure 5(a) shows the constant pressure outlet boundary condition applied to the long domain configuration. This is used as the reference against which the short domain calculations will be compared since the region far away from the trailing edge should be nearly constant and the application of the constant pressure boundary condition is reasonable. Figure 5(b) shows the constant pressure outlet boundary condition applied to the short domain. It is evident from the contours that the constant pressure boundary condition distorts the true pressure field around the trailing edge of the blade, as no contours appear to cross the outlet boundary. This pressure field is noticeably different than the reference calculation. Figure 5(c) shows the implicit Giles nonreflecting boundary condition applied to the short domain configuration. Visual inspection shows a very nice comparison with the reference calculation.



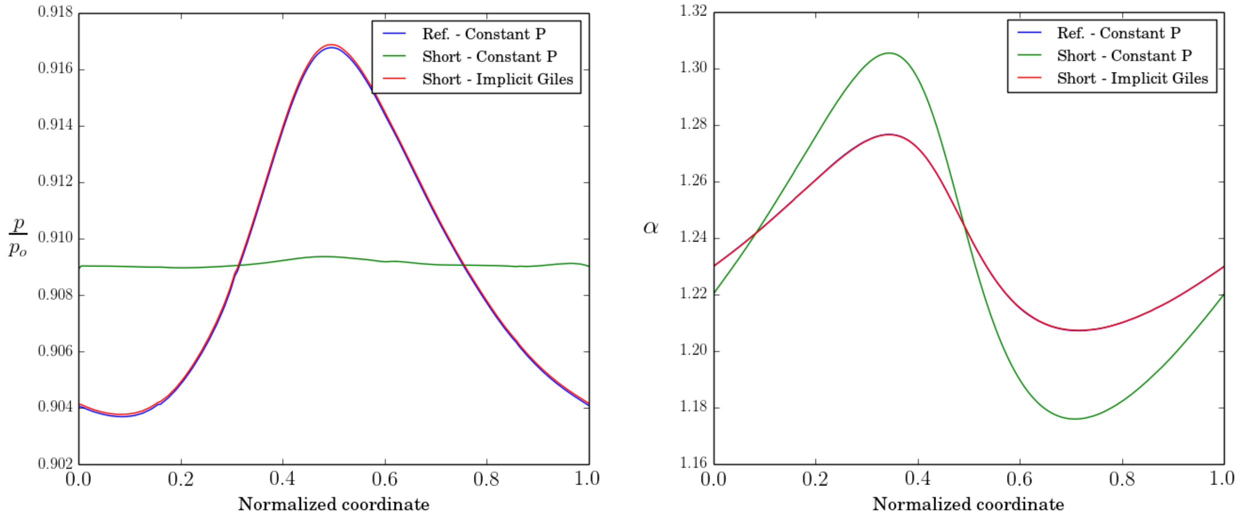
(a) Constant Pressure

(b) Constant Pressure

(c) Implicit Giles

Figure 5. Pressure contours using the *Constant Pressure* and *Giles* outlet boundary conditions.

In order to more directly compare the effectiveness of the boundary conditions, profiles of pressure and flow angles were extracted from all three configurations at the location of the short domain exit plane. These profiles are compared against one another in Figures 6(a) and 6(b) for pressure and flow angles respectively. The flow angle is defined here as $\alpha = atan^{-1}(v/u)$. The comparison of the profile for pressure and flow angles between the implicit Giles boundary condition and the reference calculation is very well-matched. This result is expected, as this technique has been applied many times by other researchers. However the contribution of this work is the implicit formulation of the boundary condition within Newton's method. Figure 7 shows the nonlinear convergence of the solver using the implicit formulation of the Giles nonreflecting boundary condition. Each simulation was initialized from the solution state of the next lowest order of accuracy. Significantly, the calculations converge in seven or fewer Newton iterations up through 4^{th}-order solution accuracy. Not including the full coupling of the Giles boundary condition in the Newton linearization prevented the implementation from converging or even caused it to diverge.



(a) Pressure at short-domain exit plane.

(b) Flow angles at short-domain exit plane.

Figure 6. Comparison of exit plane data between boundary conditions.

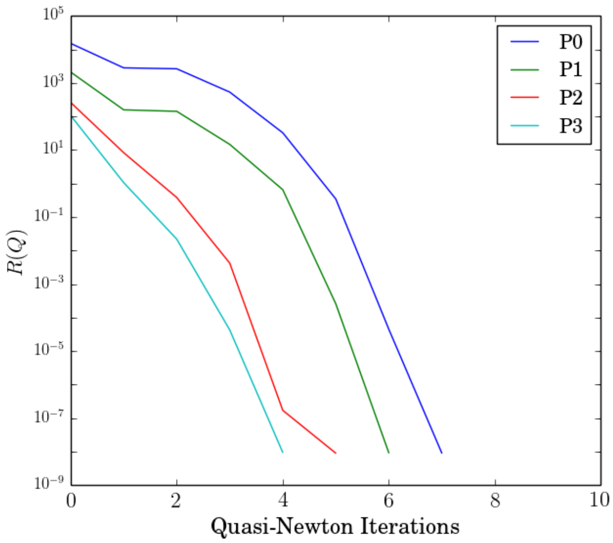


Figure 7. Nonlinear convergence using implicit Giles. $1^{st} - 4^{th}$-order accurate solutions.

IV. Conclusions

This work demonstrated the implementation of an implicit formulation of the two-dimensional, steady version of Giles' nonreflecting boundary condition. This was achieved without introduction of ghost elements or solution states and without a lagged implementation of the boundary condition as had previously been demonstrated. The implicit formulation used in this work was in the context of Newton's method where coupling of all elements associated with the nonreflecting boundary condition was accounted for in the Newton linearization. This was facilitated via an automatic differentiation technique implemented in the solver framework. Results show excellent comparison of exit plane pressure and flow angle profiles between the implicit Giles simulation and a reference calculation using an extended domain for Stator 1 of the Aachen turbine rig geometry. Significantly, simulations using the implicit Giles nonreflecting boundary condition converged in seven or fewer iterations through 4^{th}-order solution accuracy. This demonstrates the value of a fully-implicit formulation, including the boundary conditions for improved convergence of the nonlinear solver.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1610397.

References

[1] Bauer, M., Dierke, J., and Ewert, R., "Application of a Discontinuous Galerkin Method to Discretize Acoustic Perturbation Equations," *AIAA Journal*, Vol. 49, No. 5, May 2011, pp. 898–908.

[2] Kelly, J. F. and Giraldo, F. X., "Continuous and discontinuous Galerkin methods for a scalable three-dimensional non hydrostatic atmospheric model: Limited-area mode," *Journal of Computational Physics*, Vol. 231, 2012, pp. 7988–8008.

[3] Hartmann, R., Held, J., Leicht, T., and Prill, F., "Discontinuous Galerkin methods for computational aerodynamics - 3D adaptive flow simulation with the DLR PADGE code," *Aerospace Science and Technology*, Vol. 14, 2010, pp. 512–519.

[4] Wukie, N. A. and Orkwis, P. D., "A implicit, discontinuous Galerkin Chimera solver using automatic differentiation," AIAA 2016-2054, 2016.

[5] Galbraith, M. D., *A Discontinuous Galerkin Chimera Overset Solver*, Dissertation, University of Cincinnati, 2013.

[6] Orkwis, P. D. and McRae, D. S., "Newtons Method Solver for High-Speed Viscous Separated Flowfields," *AIAA Journal*, Vol. 30, 1992, pp. 78–85.

[7] Galbraith, M. C., Allmaras, S. R., and Darmofal, D. L., "A Verification Driven Process for Rapid Development of CFD Software," AIAA Paper 2015-0818, 2015.

[8] Berenger, J.-P., "A Perfectly Matched Layer for the Absorption of Electromagnetic Waves," *Journal of Computational Physics*, Vol. 114, 1994, pp. 185–200.

[9] Engquist, B. and Majda, A., "Absorbing Boundary Conditions for the Numerical Simulation of Waves," *Mathematics of Computation*, Vol. 31, No. 139, 1977.

[10] Giles, M. B., "Nonreflecting Boundary Conditions for Euler Equation Calculations," *AIAA Journal*, Vol. 28, No. 12, 1990.

[11] Saxer, A. P. and Giles, M. B., "Quasi-Three-Dimensional Nonreflecting Boundary Conditions for Euler Equations Calculations," *Journal of Propulsion and Power*, Vol. 9, No. 2, 1993.

[12] Hall, K. C., Lorence, C. B., and Clark, W. S., "Nonreflecting Boundary Conditions for Linearized Unsteady Aerodynamic Calculations," AIAA Paper 1993-0882, 1993.

[13] Moinier, P., Giles, M. B., and Coupland, J., "Three-Dimensional Nonreflecting Boundary Conditions for Swirling Flow in Turbomachinery," *Journal of Propulsion and Power*, Vol. 23, No. 5, 2007.

[14] Orkwis, P. D. and McRae, D. S., "Newton's method solver for High-Speed Separated Flowfields," *AIAA Journal*, Vol. 30, No. 1, 1992, pp. 78–85.

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

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

[17] Roe, P. L., "The use of the Riemann problem in finite-difference schemes," *Lecture Notes in Physics*, Vol. 141, 1980.

[18] Giles, M., "UNSFLO: A numerical method for the calculation of unsteady flow in turbomachinery," GTL Report 205, 1991.

[19] Galbraith, M. C., Benek, J. A., Orkwis, P. D., and Turner, M. G., "A discontinuous Galerkin Chimera scheme," *Computers and Fluids*, Vol. 98, 2014, pp. 27–53.

[20] Merz, R., Krückels, J., Mayer, J. F., and Stetter, H., "Computation of three-dimensional viscous transonic turbine stage flow including tip clearance effects," ASME 96-GT-76, 1995.