Residual Richardson Extrapolation Applied to 2D Incompressible Navier-Stokes Solver on Unstructured Meshes

Clarence Burg[1]

December 28, 2010

Abstract

Richardson extrapolation is a basic numerical technique for improving the order of accuracy of numerical approximations that involve a discretization size h. The application of this technique to the numerical solution of hyperbolic partial differential equations is presented in the context of a node-centered, edge-based unstructured finite volume solver of the incompressible Navier-Stokes equations. This technique views the original solver as a black box and only requires the h-refinement of an original mesh, the calculation of the residual on the coarse and refined mesh, interpolation from the coarse to the fine mesh and the application of Richardson extrapolation to the residuals on the refined mesh. Thus, the residual form of Richardson extrapolation can be readily applied to a variety of numerical partial differential equations solvers without much knowledge of the underlying solver. The details and strengths of this method are presented, especially related to the implementation cost of other higher order methods for unstructured finite volume solvers. Rigorous code verification is performed for the original code and the implementation of Richardson extrapolation once and twice using the method of manufactured solutions, both the inviscid and viscous components. Starting with a second order code, the order of accuracy after one application of Richardson extrapolation has increased to third order. The order increased to fourth order after two applications of Richardson extrapolation. A validation case was presented, involving von Karman vortex shedding on a relatively coarse mesh. For the original code, the vortices dissipate quickly but are maintained much further downstream when using Richardson extrapolation.

Keywords: Richardson extrapolation, Finite Volume Method, Hyperbolic Partial Differential Equations, Incompressible Navier-Stokes solvers, Von Karman Vortex Shedding.

Brief Title: Richardson Extrapolation for 2D Viscous Incompressible Solver.

1 Introduction

For unstructured triangular meshes, the standard node-based finite volume solution algorithm applicable to hyperbolic dominated systems of partial differential equations is at best second order accurate on perfect equilateral triangle meshes. In this algorithm, the conserved quantities are stored at the nodes, and these values are extrapolated to the faces of the finite volume. The governing equations are integrated over the finite volume and converted to a flux across the faces via the divergence theorem. These boundaries integrals are integrated using the one-point midpoint rule, which results in at best second order accuracy. Other numerical approximations complicate the analysis, including the approximation of the gradient via the least-squares algorithm and the use of a linear extrapolation from the node to the face. On non-ideal meshes, these approximations

[1] Assistant Professor of Mathematics, University of Central Arkansas, Conway, AR, 72035

result in a numerical scheme that is second-order accurate in the average sense, if the mesh quality does not degrade substantially.

Several methods have been proposed to improve the order of accuracy of the finite volume method applied to unstructured meshes. These methods include the discontinuous Galerkin method, the spectral difference method, higher order ENO and WENO schemes, and least squares-based schemes which create a higher order approximation of the variables within the finite volume. The discontinuous Galerkin method[1] is a combination of the finite volume method and the finite element method, applicable to hyperbolic dominated flows where shocks and other types of discontinuities may exist. In this method, the solution is specified as the summation of weighted piecewise discontinuous functions, and the flux across the points of discontinuity are solved via the approximate Riemann problem. By designing the functions appropriately, higher order approximations can be obtained per element. The spectral difference method and related methods[2, 4, 3, 5] approximate the variables at several locations within a element, use this information to form a higher order approximation within the element and approximate the flux across the boundary of the element at multiple locations consistent with the degree of the approximation. The calculation of the storage locations within the element and the flux calculation points on the boundary is mathematically quite satisfying and is analogous to the calculations required in higher order accurate finite element methods. In the essentially non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) schemes[6], the values of the variables at neighboring nodes are used to create a higher order approximation to the variable within the finite volume. The stencils used in these calculations are chosen in order to minimize oscillations and improve numerical stability, with these calculations being weighted when there are multiple stencils. By including sufficiently large number of nodes, higher order representations of the variables can be obtained. The repeated use of least squares can also be used to create a higher order representation of the variables. In both ENO/WENO and the least squares methods, multiple flux calculations must be performed at the boundary of the finite volume in order to maintain higher order accuracy in the numerical quadrature step.

The discontinuous Galerkin method and the spectral difference/spectral volume method are both element-based methods; whereas the ENO/WENO and related schemes are node-based finite volume methods. In order to implement these methods within existing standard second order finite volume codes, such as Mississippi State University's U^{2}NCLE[7, 8] and NASA Langley's FUN3D[9], the majority of the solution algorithm must be discarded, since these methods are not extensions of the existing methodologies. Furthermore, the computational cost scales poorly relative to these second order methods, at least when comparing the second order versions of these methods. The computational expense grows substantially for higher order implementations, but the overall computational cost may decrease when the goal is to achieve the same level of numerical error in a solution. The number of solution points and flux points per order for the higher order discontinuous Galerkin and spectral difference methods are provided in Table 1. For the standard second order node-based finite volume scheme on a triangle, there are one solution point and 3 flux points; whereas for these schemes, there are 3 solutions points and 6 flux points.

For the ENO/WENO schemes and schemes based on repeated application of least squares to reconstruct the solution, the information used to reconstruction the solution locally is more and more

Order	Interior Poly.	Soln. Points	Boundary Poly.	Flux Points
1	A	1	A	1×3
2	$A + Bx + Cy$	3	$A + b\xi$	2×3
3	$A + Bx + Cy + Dx^2 + Exy + Fy^2$	6	$A + b\xi + c\xi^2$	3×3
4	Fourth Degree Polynomial	10	$A + b\xi + c\xi^2 + d\xi^3$	4×3

Table 1: Computational Cost of Discontinuous Galerkin and Spectral Difference Methods

distant, as more nodes are included to build the least squares reconstruction. Thus, this information may not be as reliable, especially in regions of higher variation in the variables. Furthermore, for the ENO/WENO schemes, the level of iteration convergence is often limited as the process used to determine the non-oscillatory stencils may result in different stencils from one iteration to the next. This switching from one stencil to another prevents the full convergence of the solution.

In this paper, Richardson extrapolation is presented as an alternative to these other higher order methods, because it can be built on top of existing node-based finite volume solvers. In particular, Richardson extrapolation is applied to the numerical solution of the two-dimensional incompressible Navier-Stokes equations on unstructured triangular meshes. Richardson extrapolation[10, 11] was developed approximately 100 years ago as a means for combining numerical solutions using different discretization sizes in order to remove the leading order error term, resulting in a higher order accurate result. Roache[12, 13] demonstrated that this approach could be applied to the numerical solution of partial differential equations, when he applied the technique to the solution of the Poisson equation on a sequence of uniform rectangular meshes, obtaining a higher order result on the coarser mesh. Later, Roache[14] developed the completed Richardson extrapolation method where he interpolated the solution on the coarse mesh onto the fine mesh and applied Richardson extrapolation on the fine mesh, obtaining a higher order accurate solution on the fine mesh, assuming that the interpolation process was also higher order. This process worked well for the parabolic equation, since the solution was sufficiently smooth. This approach has been applied to the solution of systems of hyperbolic equations, with mixed results; however, Burg[15] developed a more robust and widely applicable methodology where Richardson extrapolation was applied to the discretized governing equations (i.e., the residual) rather than to the solution, at each step of the process. Richards[16] and Sun and Zhang[17] obtained similar results by applying Richardson extrapolation to the solution at each time step, rather than to the converged steady-state solution.

The principle advantage that Richardson extrapolation has over other higher order methodologies is that it can be applied to existing state-of-the-art CFD codes, with little modification to the original underlying solution algorithm, assuming that the original implementation is properly verified to agree with the theoretical order of accuracy. This algorithm relies on the discretized equations or residual produced by the existing CFD code, interpolates the residual from the coarse mesh onto the fine mesh, applies Richardson extrapolation to the interpolated coarse mesh residual and the fine mesh residual to achieve a higher order residual on the fine mesh and uses the existing implicit solver with the higher order residual to update the variables. Thus, the changes to the CFD code are limited to the creation of the fine mesh via h-refinement of the coarse mesh, the interpolation operator and the extrapolation operator. This Richardson extrapolation procedure however is limited to fourth order accuracy because of the limitations of the fourth order interpolation operator. Computationally, the cost of the Richardson extrapolation solution involves the same cost of the solution on the fine mesh, with the additional costs of calculating the coarse mesh residual and performing the Richardson extrapolation step. For 2D triangular meshes, this additional cost is less than 25% of the cost of the original solution on the fine mesh.

This paper consists of a section detailing the standard node-based finite volume method, the process of Richardson extrapolation applied to the residual and a set of verification and validation cases. The original and the Richardson extrapolation codes are verified via the method of manufactured solution on a sequence of equilateral triangular meshes and for an exact solution involving circular flow.

2 Node-Based Finite Volume Method

The node-based finite volume method applied to triangular meshes is based on Barth's work from the early 1990's[18, 19]. In general, a hyperbolic dominated system of partial differential equations can be written as

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

where $\vec{Q}$ is the vector of dependent variables, $\vec{F}(\vec{Q})$ is the flux vector and represents the convective portion of the equations, $\vec{G}(\vec{Q}), \nabla \vec{Q})$ is the diffusion vector associated with viscosity and $S(\vec{Q})$ is the vector of source terms. For hyperbolic dominated flows, the influence of the diffusion vector is small relative to the flux vector. The incompressible Reynolds-averaged Navier-Stokes equation in pseudo-compressibility form can be stated as

$$\begin{aligned}
\frac{\partial P}{\partial t} + \nabla \cdot (\beta u, \beta v) &= 0 \\
\frac{\partial u}{\partial t} + \nabla \cdot \left(u^2 + P, uv\right) &= \frac{1}{Re} \nabla \cdot \left(2\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\frac{\partial v}{\partial t} + \nabla \cdot \left(uv, uv^2 + P\right) &= \frac{1}{Re} \nabla \cdot \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, 2\frac{\partial v}{\partial y}\right)
\end{aligned} \tag{2}$$

where Re is the Reynolds number and β is the Chorin's artificial compressibility parameter[20], typically set to 15. At steady-state, the influence of β is negligible. These equations have been non-dimensionalized and assume that the flow is laminar. Thus, the dependent variables, the flux

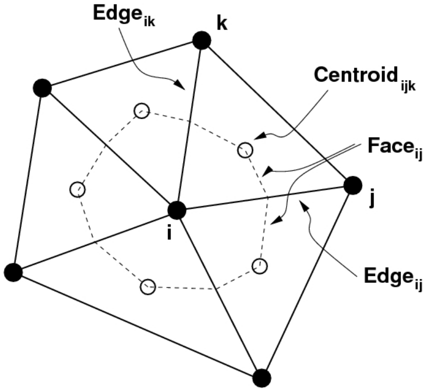


Figure 1: Node-Centered Edge-Based Finite Volume

vector and the diffusion vector can be written as

$$\vec{Q} = \begin{bmatrix} P \\ u \\ v \end{bmatrix}$$
$$\vec{F}(\vec{Q}) = \begin{bmatrix} \beta u, \beta v \\ u^2 + P, uv \\ uv, v^2 + P \end{bmatrix} \tag{3}$$
$$\vec{G}(\vec{Q}, \nabla\vec{Q}) = \frac{1}{Re} \begin{bmatrix} 0, 0 \\ 2\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, 2\frac{\partial v}{\partial y} \end{bmatrix}$$

The finite volume Ω, showing in Figure 1 is the dual of the set of triangular elements, connecting the midpoints of each edge to the centroid of each element. For each internal edge in the two-dimensional mesh, there are two edges of the control volume for the two nodes at either end of the edge. Thus, an edge-based solution algorithm is developed to calculate the contribution to the discretized equations for each edge for the control volumes associated with the edge.

The governing equations are integrated about the finite volume, or

$$\iint_\Omega \left(\frac{\partial \vec{Q}}{\partial t} + \nabla \cdot \vec{F}(\vec{Q}) + \nabla \cdot \vec{G}(\vec{Q}, \nabla\vec{Q}) \right) dA = \iint_\Omega S(\vec{Q}) dA \tag{4}$$

The terms containing the flux and diffusion vectors are transformed to surface integrals via the divergence theorem, so that the governing equations become

$$\iint_\Omega \frac{\partial \vec{Q}}{\partial t} dA + \int_{\partial\Omega} \vec{F}(\vec{Q}) \cdot \hat{n} dS + \int_{\partial\Omega} \vec{G}(\vec{Q}, \nabla\vec{Q}) \cdot \hat{n} dS = \iint_\Omega S(\vec{Q}) dA \tag{5}$$

At this point, numerical approximations are made via one-point quadrature of the area and boundary integrals. The area integrals are approximated by the value at node i and the boundary integrals are approximated via the values at the midpoints of the edges connected to node i, which will be denoted as e_{ij}, so that the standard approximation becomes

$$\frac{\partial \vec{Q}_i}{\partial t}A_i + \sum_{j=0}^{n_i}\vec{F}(\vec{Q}_{ij})\cdot \hat{n}_{ij}l_{ij} + \sum_{j=0}^{n_i}\vec{G}(\vec{Q}_{ij},\nabla\vec{Q}_{ij})\cdot \hat{n}_{ij}l_{ij} - S(\vec{Q}_i)A_i = 0 \tag{6}$$

where $\vec{Q}_i$ is the value of the dependent variables at node i, A_i is the area of the control volume, Q_{ij} is the value of the dependent variables at the midpoint of edge e_{ij}, n_i is the number of edges associated with node i, $\hat{n}_{ij}$ is the normal vector associated with the portion of the control volume boundary along edge e_{ij} and l_{ij} is the length of that portion of the control volume boundary.

For an ideal triangular mesh, consisting of equilateral triangles, the use of one-point quadrature to estimate the area and surface integrals is second order accurate, since the midpoint quadrature rule is second order. However, for non-ideal meshes, the formal order of accuracy degrades to first order, although the influence of the first order term is scaled by the quality of the mesh. In practice, for a typical triangular mesh that is not created in a bias-inducing manner, the observed order of accuracy is second order, and heuristic arguments have been made that the theoretical order of accuracy, especially for steady-state flows with no source term, is still second order, due to cancellations of the first order truncation term over the surface of the control volume. However, for poor-quality triangular meshes, or for triangular meshes created by splitting quadrilateral meshes in a biased manner, the observed and theoretical order of accuracy is degraded to first order, due to the first order error in the one-point quadrature rules.

Further complicating the order of accuracy analysis for the node-based finite volume implementation is the calculation of the dependent variables at the edge midpoints. Since the dependent variables are stored at the nodes, their value at the midpoints are determined via extrapolation, using the nodal values and the gradient values. Since the gradient is not directly calculated, it is estimated via a least-squares approximation. Barth's original second order extrapolation to the midpoint of edge e_{ij} is

$$\vec{Q}_{ij} = \vec{Q}_i + \frac{1}{2}\nabla\vec{Q}\cdot\vec{r}_{ij} \tag{7}$$

where $\vec{r}_{ij}$ is the vector pointing from node i to node j. Burg[21] developed a third order extrapolation technique analogous to Van Leer's MUSCL approach[22] which involves the use of the value at node j, via

$$\vec{Q}_{ij} = \vec{Q}_i + \frac{\chi}{2}(\vec{Q}_j - \vec{Q}_i) + \frac{1-\chi}{2}\nabla\vec{Q}\cdot\vec{r}_{ij} \tag{8}$$

where χ is a parameter from 0 to 1, that mimics Van Leer's MUSCL parameter. For $\chi = 0$, the methodology reverts to Barth's extrapolation. Herein, χ is set to $\frac{1}{2}$. This approximation provides the value of the dependent variables on one side of the flux boundary. The value on the other side is determined via an analogous approximation from node j. These values are used to determine the flux across the boundary via an approximation Riemann solver. For this code, Roe-averaged variables are used within the flux difference splitting algorithm. This algorithm does not have a direct impact on the order of accuracy of the scheme.

A least-squares approximation to the gradient at node i is calculated by determining the best fit linear approximation to the value at node i and the values at the neighboring nodes. This approximation is second order accurate on equilateral meshes. The impact of the order of accuracy of the least-squares gradient approximation on the overall solution order of accuracy is hard to estimate.

The gradient of the velocity must be calculated at the boundary of each finite volume. This estimation is performed at the midpoint of each edge using the following formula

$$\nabla \vec{Q}_{ij} = \overline{\nabla \vec{Q}} + \left[\vec{Q}_j - \vec{Q}_i - \overline{\nabla \vec{Q}} \cdot \vec{r}_{ij} \right] \frac{\vec{r}_{ij}}{|\vec{r}_{ij}|^2} \tag{9}$$

where $\overline{\nabla \vec{Q}} = \frac{\nabla \vec{Q}_i + \nabla \vec{Q}_j}{2}$. This formula uses the average value of the gradient at the two nodes associated with the edge and replaces the component of the gradient in the edge direction with the directional derivative in the direction of the edge estimated via a finite difference. A thorough and detailed presentation of this numerical approach can be found in the work of Hyams [23]

3 Residual Form of Richardson Extrapolation

3.1 Introduction to Richardson Extrapolation

Richardson extrapolation is a method that combines two numerical approximations to delete the leading order error, assuming that the numerical approximation can be expressed in terms of a discretization size h. In this case, the numerical approximation $N(h)$ can be written as

$$N(h) = N(0) + Ah^p + O(h^q) \tag{10}$$

where $N(0)$ is the exact result and p is the order of accuracy of the numerical approximation. By calculating both $N(h)$ and $N(rh)$ where r is the refinement ratio between the two different discretization sizes, the leading order error term Ah^p can be removed by combining these approximations as follows:

$$\begin{aligned} r^p N(h) &= r^p N(0) + r^p A h^p + O(h^q) \\ N(rh) &= N(0) + A(rh)^p + O(h^q) \end{aligned} \tag{11}$$

By subtracting and dividing by $r^p - 1$, the Richardson extrapolation formula becomes

$$\frac{r^p N(h) - N(rh)}{r^p - 1} = N(0) + O(h^q) \tag{12}$$

increasing the order of accuracy from p to q. Richardson extrapolation is often introduced within the context of numerical differentiation and forms the basis for the Romberg algorithm within numerical quadrature. It can be applied within numerical differential equations, but the resulting formula are not generally as computationally efficient as Runge-Kutta methods.

3.2 Application to Numerical Partial Differential Equations

Roache successfully applied Richardson extrapolation to the numerical solution of the Poisson equation on a sequence of rectilinear grids, first achieving higher order accuracy on the coarser

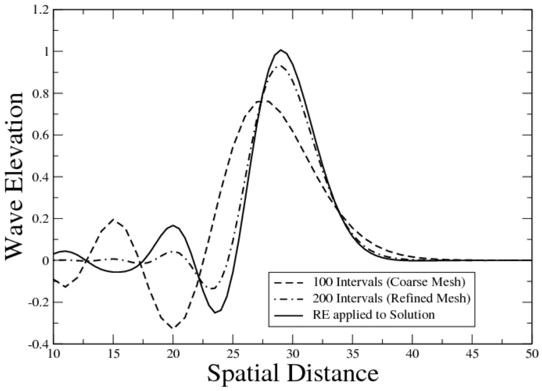


Figure 2: Richardson Extrapolation Applied to Solution

mesh. Later, he achieved higher order accuracy on the finer mesh, by interpolating the coarse mesh solution onto the fine mesh and applying Richardson extrapolation using these values on the fine mesh. Several other efforts to apply Richardson extrapolation to the numerical solutions of partial differential equations have encountered a variety of difficulties, with the general consensus that Richardson extrapolation does not apply to numerical PDEs.

Burg[15] identified the principle challenge for the application of Richardson extrapolation to partial differential equations, which is the impact of dispersion or phase error within the solution. The location of certain features within the solution will be dependent on the spatial resolution of the underlying mesh. As the mesh is refined, the location of the features will shift. Thus, at a particular location within the physical domain, the value of the solution on a sequence of meshes may vary due to the phase error rather than just the numerical resolution, and the application of Richardson extrapolation to the solution near these locations results in poorer numerical approximations. An example of these behavior is shown in Figure 2, which shows the solution to the 1D first-order wave equation with the solution moving from left to right. The equation is solved via the implicit backward-time central space finite difference scheme. The figure shows the solution using 100 and 200 intervals, with significant dispersion behind and to the left of the main wave. By applying Richardson extrapolation to the solution, the solution at the peak of the main wave is improved and retains the original wave height of 1 much better, but behind the main wave, the influence of the dispersion in the original solutions caused a worsening of the solution in the refined mesh. However, by applying Richardson extrapolation to the residual (i.e., the discretization governing equations) at each time step, the impact of dispersion is dramatically reduced, as is shown in Figure 3.

The key to the successful implementation of Richardson extrapolation to the numerical solution of partial differential equations is its application to the residual at each time step, rather than to the final solution. The methods developed by Richards[16] and Sun and Zhang[17] were quite similar except that their methods applied Richardson extrapolation to the solution at each time step. Their methods required the solution to be determined on both the coarse and the fine mesh; whereas, this method only requires the solution on the fine mesh. The process of applying Richardson Extrapolation to the residual is detailed in the next subsection.

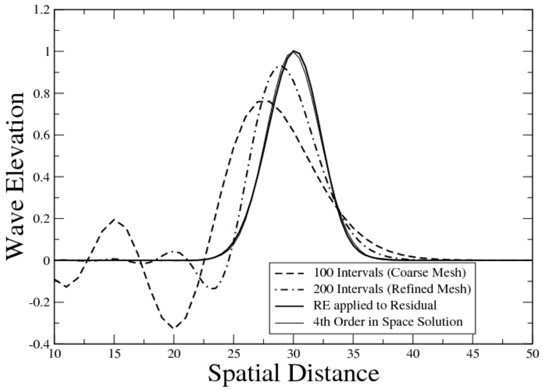


Figure 3: Richardson Extrapolation Applied to Residual

3.3 Numerical Approach

Finite difference, finite volume and finite element discretization methods for partial differential equations, in implicit form, result in a nonlinear system of coupled algebraic equations that must be solved either at each time step or for the steady-state solution. This system can be written as

$$\vec{R}(\vec{Q}^{n+1}, \vec{Q}^{n}, \cdots, \vec{Q}^{n-k}, \chi, \vec{t}) = 0 \tag{13}$$

where $\vec{R}$ is referred to as the residual vector, $\vec{Q}^{n+1}$ is the solution to be determined for time level t_{n+1}, $\vec{Q}^{n}, \cdots, \vec{Q}^{n-k}$ are the known solutions at previous time levels, χ are the spatial locations, and $\vec{t}$ are the temporal locations. This formulation assumes a fixed mesh. The solution to this system of equations can be obtained via iterative methods, such as the Newton-Raphson method.

The residual vector generated via the finite volume or the finite element approach will typically include an extra factor of the area of the control volume or control element due to the integration involved in generating the discretized equation. The finite difference scheme typically involves a division using the appropriate length scales. Typically, the residual for finite difference approximation directly corresponds to the differential operator, while the residual for the finite volume and finite element approximations include an extra factor of the area over which the integration occurs. On a uniform mesh, the resulting residuals vary smoothly for any of these approaches since the area is constant; however, on a nonuniform mesh, the residuals for the finite difference approximation will vary smoothly, while the residuals divided by the area for the finite volume or finite element approximations will vary smoothly. This observation is critical for the effectiveness of the Richardson extrapolation algorithm for nonuniform meshes.

Richardson extrapolation applied to the residual for the solution of partial differential equations involves the communication of certain information between a coarse mesh and a refined mesh. Consider two meshes, χ_c and χ_r where χ_r is a refined mesh consisting of all of the points in the coarse mesh χ_c with spacing that is half of the spacing in χ_c. General χ_r will be obtained via h-refinement of χ_c as shown in Figure 4. A solution to the partial differential equation at the required time level is determined on both meshes, with Q_c and Q_r representing the solutions on χ_c and χ_r, respectively. From these solutions, the value of the discretized equations is obtained for both meshes, or the coarse mesh residual $\vec{R}(\vec{Q}_c, \chi_c, t)$ and the fine mesh residual $\vec{R}(\vec{Q}_r, \chi_r, t)$. As stated above, these residuals must be scaled by the local areas, in order to obtain averaged values over the control volumes. At this point, the residual information on the coarse mesh only corresponds to the residual information on the refined mesh at the points in common between the two meshes. Hence, the residual information on the coarse mesh must be interpolated to the new points created in the fine mesh. This interpolation operator has the form

$$\vec{R}_i^{\chi_r} = I^{\alpha}(\vec{R}(\vec{Q}_c, \chi_c, t), \chi_c, \chi_r) \tag{14}$$

The operator $I^{\alpha}(\vec{R}(\vec{Q}_c, \chi_c, t), \chi_c, \chi_r)$ is an interpolation operator of order α of the residual $\vec{R}(\vec{Q}_c, \chi_c, t)$ on the coarse mesh to the refined mesh, and $\vec{R}_i^{\chi_r}$ is the interpolated residual on the fine mesh. Using this interpolated residual, a higher order residual on the refined mesh can be determined via

$$\vec{R}_{RE}^{\chi_r} = RE(\vec{R}(\vec{Q}_r, \chi_r, t), \vec{R}_i^{\chi_r}, p) \tag{15}$$

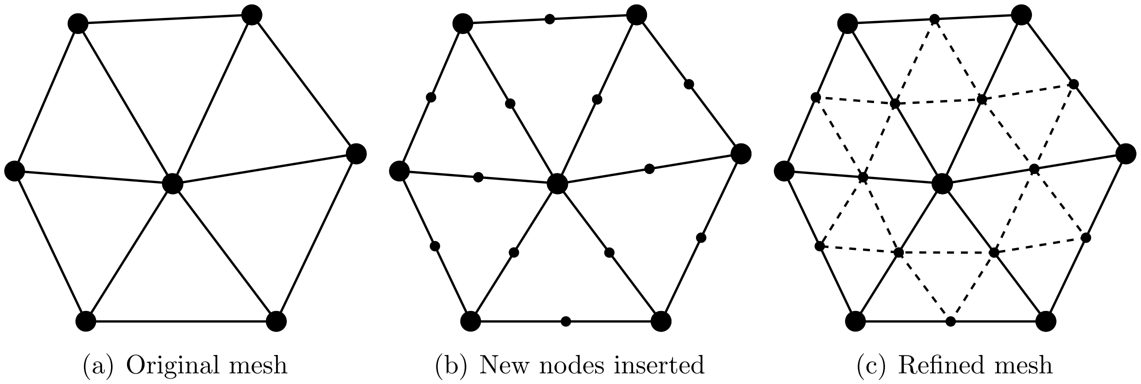
For this approach to work, the interpolation must be sufficiently accurate so that the error introduced into the interpolated residual is of higher degree than the order of accuracy of the numerical method.

Finally, the update to the solution on the fine mesh is obtained using the higher order accurate residual, using the same subroutines using by the original code. These updates are applied to the fine mesh solution $\vec{Q}_r$. The coarse mesh solution is obtained from the fine mesh solution by direct injection or copying of the values at the common points to the coarse mesh.

Since it views the numerical solver as a black box, this Richardson extrapolation-based algorithm reuses the residual calculations within a computational tool, the existing data structures are unchanged, and the same matrix solution algorithm is used. Hence, to achieve the higher order result, only the residual on the coarse and refined meshes must be calculated.

In summary, the Richardson extrapolation based algorithm applied to the residual involves the following components:

1. Calculate the residual for the coarse mesh $\vec{R}(\vec{Q}_c, \chi_c, t)$
2. Calculate the residual for the refined mesh $\vec{R}(\vec{Q}_r, \chi_r, t)$
3. Divide through by length scale factor, if necessary.
4. Interpolate the coarse residual onto the refined mesh.
5. Apply Richardson extrapolation on the interpolated coarse mesh residual and the refined mesh residual.
6. Multiply through by length scale factor, if necessary.
7. Solve for the update to the solution, either explicitly or implicitly.
8. Add these updates to the current approximation to the new solution on the refined mesh.
9. Restrict the new refined mesh solution to the coarse mesh via direct copying for the common nodes.


(a) Original mesh

(b) New nodes inserted

(c) Refined mesh

Figure 4: h-refinement of an Unstructured Triangular Mesh.

4 Test Cases

4.1 Verification Case

The order of accuracy of the original finite volume solver is calculated via the use of the method of manufactured solutions[24, 25, 26]. Since exact solutions do not exist, in general, for the viscous incompressible Navier-Stokes equations, except under simplifying assumptions, another process is needed in order to compare the numerical solution with an exact solution. In the method of manufactured solutions, a known exact solution, such as a exponential or a trigonometric function, is selected to be the solution of a modified system of governing equations. The exact solution is processed through the original governing equations to yield a source term that is added to the governing equations and implemented within the flow solver using the same numerical approach (i.e., the finite volume method). Thus, the numerical solution obtained by the altered code should converge to the non-trivial manufactured solution.

For this system of equations, exponential functions are used for the pressure and velocity components. Since these are manufactured solutions, they have no basis in physical reality. They are defined as

$$\begin{aligned} P(x,y) &= e^{Ax+By} \\ u(x,y) &= Ce^{Dx+Ey} \\ v(x,y) &= Fe^{Gx+Hy} \end{aligned} \tag{16}$$

The coefficients were set as $A = 0.4$, $B = 0.33$, $C = 2.0$, $D = 0.5$, $E = 1.0$, $F = 3.0$, $G = 0.4$ and $H = 0.6$. The

Grid Level	Nodes	Triangles	Edge Length
0	3	1	1
1	6	4	0.5
2	15	16	0.25
3	45	64	0.125
4	153	256	0.0625
5	561	1024	0.03125
6	2145	4096	0.015625
7	8385	16384	0.0078125
8	33153	65536	0.00390625
9	131841	262144	0.001953125

Table 2: Sequence of H-Refined Equilateral Triangular Meshes

Thus, the source terms added to the governing equations at steady state are

$$\begin{aligned}
\text{Mass Equation} &= CDe^{Dx+Ey} + FHe^{Gx+Hy} \\
\text{x-Momentum Equation} &= C^2D^2e^{2Dx+2Ey} + Ae^{Ax+By} + CF(E+H)e^{(D+G)x+(E+H)y} \\
&\quad - \frac{1}{Re}\left(C(2D^2+DE)e^{Dx+Ey} + FGHe^{Gx+Hy}\right) \qquad (17) \\
\text{y-Momentum Equation} &= CF(D+G)e^{(D+G)x+(E+H)y} + F^2H^2e^{2Gx+2Hy} + Be^{Ax+By} \\
&\quad - \frac{1}{Re}\left(CDEe^{Dx+Ey} + F(GH+2H)e^{Gx+Hy}\right)
\end{aligned}$$

The source terms associated with the manufactured solution were evaluated via the application of the divergence theorem and integrated as integrals over the boundary of each control volume, consisting of components from the edge midpoint to the centroid as well as the boundary edges. The results were calculated via Mathematica and copied into the code, after outputting the equations using Mathematica's built-in function 'CFORM', which converts a mathematical expression into appropriate C code. A Dirichlet boundary condition was imposed, where the nodes on the boundary of the domain were specified to the value of the manufactured solution at each boundary nodal location.

A sequence of meshes were generated from an equilateral triangle, via h-refinement. The sides of equilateral triangle were length 1 with the base from $x = 0$ to $x = 1$. Grid 0 consisted of three nodes and one triangle. Grid 1 was an h-refinement of Grid 0 and consisted of 6 nodes and 4 triangles. The details for the other meshes are provided in Table 2

The converged solution was calculated for Grids 3-9, and the error was calculated via an L2-norm defined as the following

$$L2_{norm}(q) = \left(\sum_{i=0}^{N}(q_i - q_{exact})^2 Area_i\right)^{1/2} \qquad (18)$$

where N is the number of nodes in the mesh, $Area_i$ is the area of the control volume associated with node i and q is the variable being measured.

Grid Level	Pressures	Order	u-Velocity	Order	v-Velocity	Order
3	7.439123e-05	NA	2.849665e-05	NA	2.061198e-05	NA
4	2.398805e-05	1.633	9.530386e-06	1.580	6.882741e-06	1.582
5	6.588238e-06	1.864	2.749354e-06	1.793	2.023570e-06	1.766
6	1.737056e-06	1.923	7.403680e-07	1.893	5.610281e-07	1.851
7	4.471277e-07	1.958	1.929088e-07	1.940	1.480183e-07	1.922

Table 3: Error for Original Euler Code

Grid Levels	Pressures	Order	u-Velocity	Order	v-Velocity	Order
3-4	2.251627e-06	NA	1.799794e-06	NA	9.228971e-07	NA
4-5	2.572796e-07	3.130	2.251402e-07	2.999	1.155476e-07	2.998
5-6	3.273409e-08	2.974	2.839377e-08	2.987	1.483388e-08	2.962
6-7	4.223826e-09	2.954	3.590015e-09	2.984	1.911313e-09	2.956

Table 4: Error for Euler Code Richardson Extrapolation Applied Once

Richardson extrapolation was applied once to a pair of meshes and was applied multiple times to a triplet of meshes, increasing the order of accuracy either one or two orders. The results for the inviscid Euler equations for the original code, the application of Richardson Extrapolation once and the application of Richardson extrapolation twice are shown in Tables 3,4, and 5. The original code converges slowly towards second order, while the Richardson extrapolation code converges at a third order rate and the twice applied Richardson extrapolation code converges at a fourth order rate, which is the limit of this approach due to the limitation of the interpolation operator.
The viscous component was also verified using the method of manufactured solutions, using the same application of the divergence theorem and integration along the boundary of the control volume. For this verification, the Reynolds number was set to 1, so that the influence of the viscous component would be measurable. However, since this changed the strongly hyperbolic nature of the solver, into a mixed hyperbolic/parabolic nature, the stability of the algorithm degraded, so that the CFL number was limited to 1. The results shown in Tables 6, 7, and 8 include both the convective and diffusive portions of the solver, at steady-state.

4.2 Von Karman Vortex Shedding

In order to demonstrate the strength of this higher order methodology, the vortices arising from laminar viscous flow around a circular cylinder in two-dimensions was calculated, at a Reynolds

Grid Levels	Pressures	Order	u-Velocity	Order	v-Velocity	Order
3-4-5	1.308142e-07	NA	3.059629e-08	NA	2.751116e-08	NA
4-5-6	6.584450e-09	4.312	1.596662e-09	4.260	1.430188e-09	4.266
5-6-7	3.737264e-10	4.139	7.788319e-11	4.358	8.538590e-11	4.066

Table 5: Error for Euler Code Richardson Extrapolation Applied Twice

Grid Level	Pressures	Order	u-Velocity	Order	v-Velocity	Order
3	2.218107e-04	NA	8.392735e-05	NA	5.464449e-05	NA
4	6.792442e-05	1.707	2.581843e-05	1.701	1.753163e-05	1.640
5	1.825461e-05	1.896	7.194747e-06	1.843	5.104589e-06	1.780
6	4.757064e-06	1.940	1.906519e-06	1.916	1.400368e-06	1.866
7	1.217187e-06	1.966	4.931299e-07	1.951	3.677573e-07	1.929

Table 6: Error for Original Viscous Navier-Stokes Code

Grid Levels	Pressures	Order	u-Velocity	Order	v-Velocity	Order
3-4	3.504839e-06	NA	2.346378e-06	NA	1.257590e-06	NA
4-5	2.947265e-07	3.572	2.524236e-07	3.217	1.336017e-07	3.235
5-6	3.346730e-08	3.139	2.974191e-08	3.085	1.574648e-08	2.998

Table 7: Error for Viscous Code Richardson Extrapolation Applied Once

number of 75. A coarse unstructured mesh was built for a rectangular domain with dimensions $[-20, 40]$ by $[0, 30]$. A circle of radius 0.5 was placed at $(-10, 15)$ with the flow moving from left to right. Hence, there was sufficient computational domain downstream of the circle for the vortices to form and to oscillate. The mesh was refined once, and a second-order accurate solution on the refined mesh using the original algorithm was calculated, along with the higher order accurate solution using Richardson extrapolation. Hence, both solutions were on the same mesh, which consisted of 20,856 nodes and 41,224 triangles and is shown in Figure 5. The mesh is relatively coarse downstream of the circle.

The solution using the original second order code is shown in Figure 6 and the solution using Richardson extrapolation on the same mesh is shown in Figure 7. The background color is based the x-velocity component and the contour plots are based on the pressure variable. Near the circle, the results are similar for the two approaches; however, the vortices quickly dissipate for the original code but dissipate much less quickly for the Richardson extrapolation code, indicating that the numerical diffusion is significantly decreased when using Richardson extrapolation.

5 Conclusion

In this paper, the method of Richardson extrapolation was applied to the incompressible Navier-Stokes equations, solved on triangular unstructured meshes in two-dimensions. The underlying

Grid Levels	Pressures	Order	u-Velocity	Order	v-Velocity	Order
3-4-5	3.427979e-07	NA	9.488700e-08	NA	8.127911e-08	NA
4-5-6	1.589901e-08	4.430	4.605625e-09	4.365	4.023801e-09	4.336
5-6-7	7.309113e-10	4.443	2.162624e-10	4.413	1.987895e-10	4.339

Table 8: Error for Viscous Code Richardson Extrapolation Applied Twice

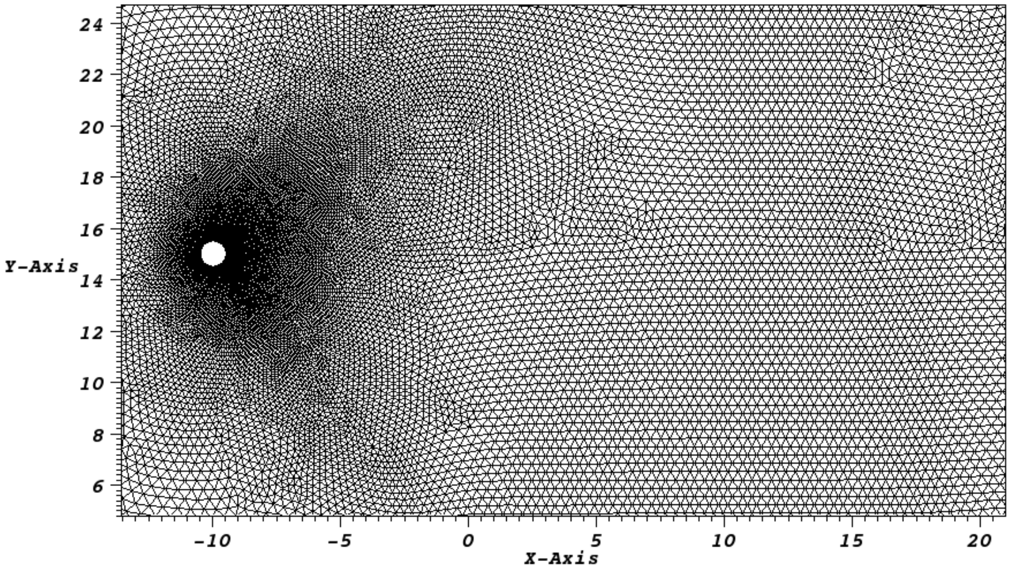


Figure 5: Mesh For VonKarman Vortex Shedding

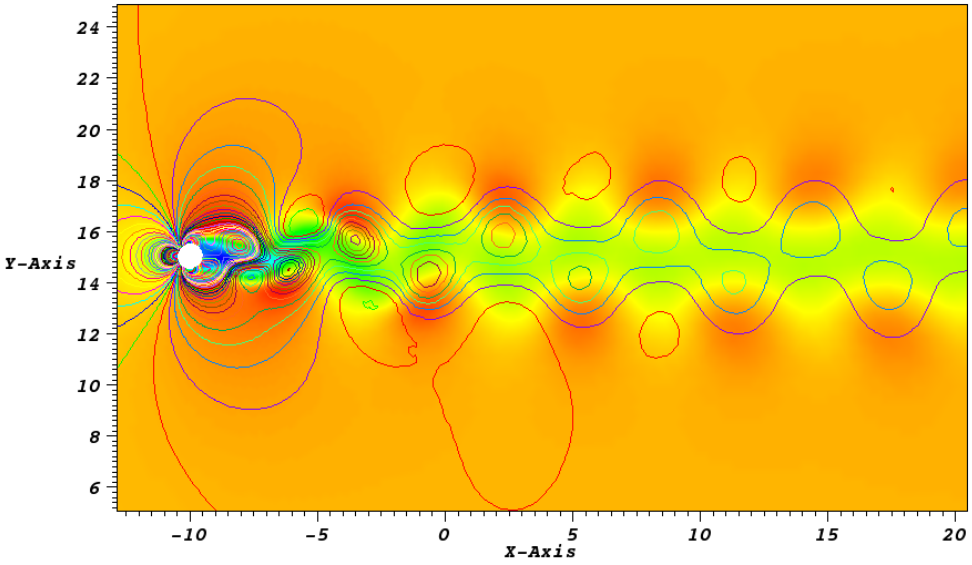


Figure 6: VonKarman Vortex Shedding using Original Code

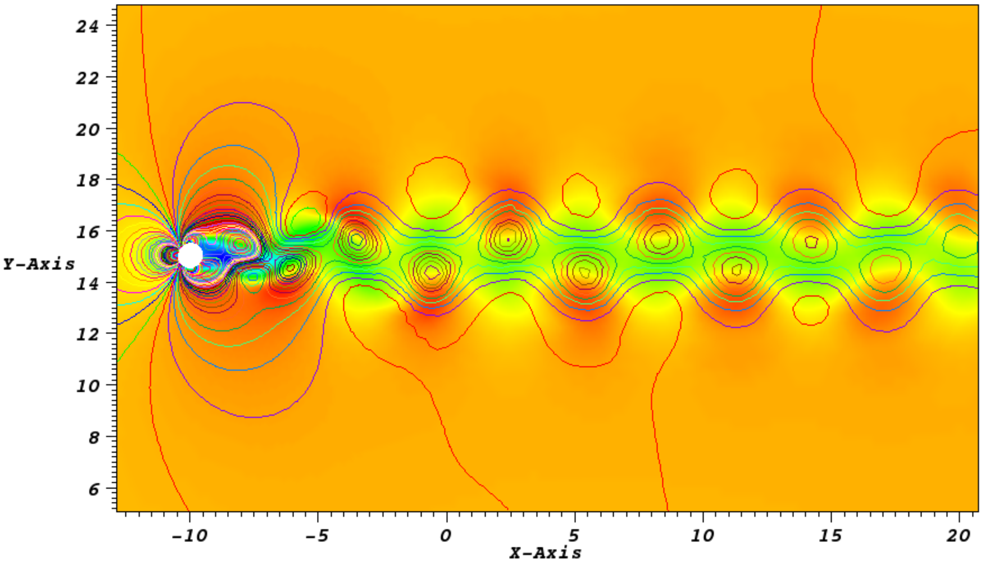


Figure 7: VonKarman Vortex Shedding using Richardson Extrapolation

black-box partial differential equations solver was a node-centered, edge-based finite volume solver, using Roe-averaged variables and the flux differencing method for solving the approximate Riemann problem. Unlike most higher order methods for solving compressible and incompressible Navier-Stokes equations, this methodology reuses existing codes in the construction of the discretized equations and views the residual calculation as a "black-box" only needing to know the order of accuracy of the underlying solver. Other methods typically involve sufficiently different data structures and calculations than existing codes, so that the majority of the subroutines developed for the existing codes must be discarded when going to third order accuracy and above.

There is a common erroneous belief that Richardson extrapolation can not be applied to the solution of hyperbolic systems of equations. Burg, Richards, and Sun and Zhang have demonstrated via similar techniques that this belief is incorrect, as long as Richardson extrapolation is applied at each time level, rather than at the end of the simulation. A brief discussion about the principle cause for the failure of Richardson extrapolation when applied at the end of the simulation was provided. The algorithm for applying Richardson extrapolation to the residual (i.e., the numerical results from the discretized equations) was presented, showing that the algorithm is only slightly more expensive than the solution on the refined mesh. In this algorithm, the mesh must be refined via h-refinement, and subroutines dealing with interpolation from the coarse mesh onto the fine mesh and for performing Richardson extrapolation on the fine mesh are presented.

One detailed verification case was presented, involving the method of manufactured solutions. In this example, the governing equations were altered via a source term so that the steady-state solution would be a non-trivial known function. Both the inviscid and viscous portions of the Navier-Stokes equations were tested via this manufactured solution. Results for the original code, showing its second order nature, along with results from one and two applications of Richardson extrapolation, showing third and fourth order accuracy, respectively, were presented.

Results from simulations of von Karman vortex shedding behind a circular cylinder is two-dimensions clearly showed that the Richardson extrapolation approach significantly decreased the numerical diffusion, since the vortices remained much stronger further downstream of the circle than for the original code.

Because this methodology treats the underlying solution algorithm as a black-box, Richardson extrapolation can be applied quickly and easily to a wide variety of solvers, to increase the order of accuracy of the numerical results. This application can be performed without altering the majority of the well-tested algorithms developed for these types of codes.

6 Acknowledgments

The author wants to acknowledge the work of Taylor Erwin, a former Master's student within the University of Central Arkansas's Applied Mathematics program. Mr. Erwin developed many of the subroutines necessary for the application of Richardson extrapolation to unstructured triangular solvers, including the h-refinement subroutine and the interpolation subroutine, which are detailed in his Master's thesis[27]. The majority of the numerical calculations performed in the development of this paper were performed on the University of Central Arkansas's Callisto computing cluster[28].

References

[1] B. van Leer, and S. Nomura, "Discontinuous Galerkin for Diffusion", 17th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2005-5108, Toronto (2005).

[2] , Y. Liu, M. Vinokur, and Z. Wang, "Multi-Dimensional Spectral Difference Method for Unstructured Grids", 43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2005-0320, Reno (2005).

[3] Z. J. Wang , and Y. Liu, "The spectral difference method for the 2D Euler equations on unstructured grids", 17th AIAA Computational Fluid Dynamics Conf., AIAA Paper 2005-5112, Toronto (2005).

[4] G. May, and A. Jameson, "High-Order Accurate Methods for High-Speed Flow", 17th AIAA Computational Fluid Dynamics Conf., AIAA Paper 2005-5251, Toronto (2005).

[5] Z. J. Wang, "High-Order Spectral Volume Method for Benchmark Aeroacoustic Problems", 41st AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2003-0880, Reno (2003).

[6] W. R. Wolf, and J. L. F. Azevedo, "High-order unstructured essentially nonoscillatory and weighted essentially nonoscillatory schemes for aerodynamic flows", AIAA J., 44-10 pp. 2295-2310 (2006).

[7] http://www.simcenter.msstate.edu/simcenter/docs/uss_u2ncle/

[8] D. G. Hyams, K. Sreenivas, C. Sheng, W. R. Briley, D. L. Marcum, and D. L. Whitfield, "An investigation of parallel implicit solution algorithms for incompressible flows on multi-element unstructured topologies", 38th AIAA Aerospace Sciences Meeting, AIAA Paper 2000-0271, Reno, Nevada, (2000).

[9] http://www.fun3d.com

[10] L. F. Richardson, "The Approximate Arithmetical Solution by Finite Differences of Physical Problems including Differential Equations, with an application to the Stresses in a Masonry Dam", Philo Trans of Royal Soc of London, Series A 210:307-357 (1911).

[11] L. F. Richardson and J. A. Gaunt, "The Deferred Approach to the Limit", Phil Trans of Royal Soc of London, Series A 226:299-349 (1927).

[12] P. J. Roache, "Chapter 16: Semidirect High Order Accuracy Solutions in Non-Orthogonal Grids by Richardson Extrapolation", *Fundamentals of Computational Fluid Dynamics*, Hermosa Publisher, 467-477, (1985).

[13] P. J. Roache, "Scaling of High Reynolds Number Weakly Separated Channel Flows", *Proceedings: Symposium on Numerical and Physical Aspects of Aerodynamic Flows*, T. Cebeci (Editor), California State University at Long Beach, Jan., 1981, 87-98 (1981).

[14] P. J. Roache and P. M. Knupp, "Completed Richardson Extrapolation", Comm. in Num. Methods in Engineering, 9, 365-374 (1994).

[15] C. Burg, and T. Erwin, "Application of Richardson Extrapolation to the Numerical Solution of Partial Differential Equations", Numer Methods Partial Differential Eq, 25, 810-832 (2009).

[16] S. A. Richards, "Completed Richardson Extrapolation in Space and Time", Comm. in Num. Methods in Engineering, 13, 573-582 (1997).

[17] H. Sun and J. Zhang, "A High-Order Finite Difference Discretization Strategy Based on Extrapolation for Convection Diffusion Equations", Num Methods for Partial Differential Eq, 20-1: 18-32, (2004).

[18] T. J. Barth, and P. O. Frederickson, "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction", 28^{th} Aerospace Sciences Meeting, AIAA Paper 1990-0013, Reno (1990).

[19] T. J. Barth, and S. W. Linton, "An Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation", 33^{rd} Aerospace Sciences Meeting and Exhibit, AIAA Paper 1995-0221, Reno (1995).

[20] A. J. Chorin, "A Numerical Method for Solving Incompressible Viscous Flow Problems", J. Comp Physics, 2, 12-26 (1967).

[21] C. O. E. Burg, (2005), "Higher Order Variable Extrapolation for Unstructured Finite Volume RANS Flow Solvers", 17th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2005-4999, Toronto (2005).

[22] B. van Leer, "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method", J. Comp. Physics, 32, 101-136 (1979).

[23] D. G. Hyams, "An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Unstructured Topologies", Ph.D. Dissertation, Mississippi State University, (2000).

[24] S. Steinberg, and P. J. Roache, "Symbolic Manipulation and Computational Fluid Dynamics", *AIAA Journal*, 22-10: 1390-1394 (1984).

[25] W. L. OberkampfMMS, F. G. Blottner, and D. P. Aeschliman, "Methodology for Computational Fluid Dynamics: Code Verification/Validation", 26^{th} AIAA Fluid Dynamics Conference, AIAA Paper 95-2226, San Diego (1995).

[26] K. Salari, and P. Knupp, "Code Verification by the Method of Manufactured Solutions", Tech. Rep. SAND2000-1444, Sandia National Laboratories, Albuquerque, NM, (2000).

[27] T. Erwin, "Application of Richardson Extrapolation to the Numerical Solution of Partial Differential Equations", Master's Thesis, University of Central Arkansas, (2008).

[28] http://www.uca.edu/callisto