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

AIAA 2009-3653

Application of Richardson Extrapolation to the Numerical Solution of Partial Differential Equations

Clarence O. E. Burg* and Taylor Erwin†
University of Central Arkansas, Conway, AR, 72035

Richardson extrapolation is a methodology for improving the order of accuracy of numerical solutions that involve the use of a discretization size h. By combining the results from numerical solutions using a sequence of related discretization sizes, the leading order error terms can be methodically removed, resulting in higher order accurate results. Richardson extrapolation is commonly used within the numerical approximation of partial differential equations to improve certain predictive quantities such as the drag or lift of an airfoil, once these quantities are calculated on a sequence of meshes, but it is not widely used to determine the numerical solution of partial differential equations. Within this paper, Richardson extrapolation is applied directly to the solution algorithm used within existing numerical solvers of partial differential equations to increase the order of accuracy of the numerical result without referring to the details of the methodology or its implementation within the numerical code. Only the order of accuracy of the existing solver and certain interpolations required to pass information between the mesh levels are needed to improve the order of accuracy and the overall solution accuracy. With the proposed methodology, Richardson extrapolation is used to increase the order of accuracy of numerical solutions of Laplace's equation, the wave equation, the shallow water equations, and the compressible Euler equations in two-dimensions.

I. Introduction

I.A. Introduction to Richardson Extrapolation

Richardson extrapolation is a methodology for increasing the order of accuracy of numerical solutions involving a discretization size h. In most introductory numerical analysis textbooks,[1] this methodology is taught in the chapter on numerical differentiation as a way to increase the order of accuracy of stencils for approximating various derivatives. This methodology also forms the basis for the Romberg algorithm used within numerical integration, which relates the trapezoid rule, Simpson's rule and Boole's rule for approximating the value of a definite integral. It can also be used within numerical solutions to ordinary differential equations, although it is not competitive in terms of computational costs with other higher-order accurate methods, such as the Runge-Kutta method, and hence it is not taught widely. Within various applications of numerical solutions to partial differential equations, it has been used to increase the accuracy of various integral outputs such as the calculation of the lift or drag coefficient for flow past an airfoil.

Richardson extrapolation relies on an observation about the shape of the error terms in a numerical approximation. Assume $N(h)$ is a numerical approximation of order p to an exact result $N(0)$. The algorithm is consistent when the exact result is obtained as h goes to 0. Using these assumptions, the numerical approximation can be expanded via Taylor series as

$$N(h) = N(0) + Ah^p + O(h^{p+1}) \tag{1}$$

where Ah^p is the leading order error term. Once the order of accuracy is determined via analysis of the numerical scheme and verified via analysis of the numerical results, multiple numerical approximations can

*Assistant Professor, Department of Mathematics
†Graduate Student, Department of Mathematics

be combined to remove the leading order error term. Consider the Taylor series expansions of a numerical approximation using two discretization sizes h and rh, or

$$\begin{aligned} N(h) &= N(0) + Ah^p + O(h^{p+1}) \\ N(rh) &= N(0) + Ar^p h^p + O(h^{p+1}) \end{aligned} \tag{2}$$

By multiplying $N(h)$ by r^p and subtracting off $N(rh)$, the leading order error term is removed. Thus, given a refinement ratio r, an order of accuracy p, and numerical approximations $N(h)$ and $N(rh)$, the Richardson extrapolation formula for improving the order of accuracy to at least $p+1$ is

$$N_{\mathrm{RE}}(h) = \frac{r^p N(h) - N(rh)}{r^p - 1} = N(0) + O(h^{p+1}) \tag{3}$$

If the original numerical scheme does not have an error term of the form h^{p+1}, then the order of accuracy of the new numerical scheme is based on the next lowest error term.

Richardson extrapolation can be illustrated via an example involving numerical differentiation. Consider the 2^{nd} order central difference approximation to the first derivative, with discretization size h, and the corresponding expansion

$$\begin{aligned} f'(x) \approx \mathrm{CD}(h) &= \frac{f(x-h) - f(x+h)}{2h} \\ &= f'(x) + f'''(x)\frac{h^2}{6} + f^{iv}\frac{h^4}{24} + O(h^6) \end{aligned} \tag{4}$$

Doubling the discretization size to $2h$ yields a related central difference approximation to the first derivative

$$\begin{aligned} f'(x) \approx \mathrm{CD}(2h) &= \frac{f(x-2h) - f(x+2h)}{4h} \\ &= f'(x) + f'''(x)\frac{(2h)^2}{6} + f^{iv}\frac{(2h)^4}{24} + O(h^6) \end{aligned} \tag{5}$$

By applying the Richardson extrapolation formula in Eq. (3) with $r = 2$, a 4^{th} order central difference approximation to the first a derivative is obtained. The new central difference formula and the corresponding expansion are given by

$$\begin{aligned} f'(x) \approx \mathrm{CD}_{\mathrm{RE}}(h) &= \frac{4\,\mathrm{CD}(h) - \mathrm{CD}(2h)}{3} \\ &= \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} \\ &= f'(x) - f^{iv}(x)\frac{h^4}{6} + O(h^6) \end{aligned} \tag{6}$$

Because this central difference stencil has no odd-ordered error terms, Richardson extrapolation increases the order of accuracy of this approximation by two, from second order to fourth order.

II. Issues with Richardson Extrapolation

Richardson extrapolation has been used to increase the order of accuracy of numerical approximations to the derivative of a function and to the integral of a function over a bounded domain. It can also be applied to the numerical solution of ordinary differential equations, although the computational cost is greater than certain specialized methods such as the Runge-Kutta method. Unfortunately, the application of Richardson extrapolation to partial differential equations is subject to certain issues that must be addressed for the method to achieve higher order accuracy. These issues include the effects of dispersion, which can cause variation in the locations of physical features of the solution based on mesh resolution, improper implementation of the algorithm so that the original theoretical order of accuracy is not achieved, and poor mesh refinement resulting in sequences of meshes that are not geometrically similar.

The effects of these issues are evident in the results cited by Baker[2] concerning the AIAA Second Drag Prediction Workshop. Each submission to the workshop contained a set of lift and drag coefficients found on a coarse, medium, and fine mesh. From these results, an order of accuracy could be determined and an improved prediction could ideally be made using this order of accuracy and Richardson extrapolation. The calculated orders of accuracy, however, varied from 0.09 to 23.7 along with some negative and some complex orders of accuracy. Since the observed order of accuracy for the vast majority of these submissions was unreasonable, the results from the application of Richardson extrapolation were similarly dubious. The primary reasons for these problems were the use of unverified computational tools, poor mesh refinement strategies, and the choice of a test case which contained certain physical features that could be affected by numerical dispersion. The following sections address these key issues further.

II.A. Dispersion

For a numerical scheme that permits dispersion, waves of different frequencies travel at different speeds. If the numerical wave speeds differ from the wave speed within the exact solution, then non-physical oscillations may develop within a simulation. Solutions of partial differential equations that contain discontinuities also experience oscillations that are directly related to dispersion. These numerical wave speeds are affected by the mesh resolution. As such, the location of these oscillations and of physical features within a simulation, such as a standing wave or a shock, may change due solely to the mesh resolution.

Figure 1 shows the solution from a first-order in time and second-order in space numerical scheme for the 1-D wave equation at three different mesh resolutions. The solution should exhibit a single standing wave center about $x = 30$ with an amplitude of 1. Significant oscillations are clearly visible to the left of the main crest in all solutions save the one on the most refined mesh. In addition, the location of the oscillations and of the main crest vary between the three meshes. figure 2 shows the result of applying Richardson extrapolation to the solutions on the two coarsest meshes. The application of Richardson extrapolation has clearly failed in this case.

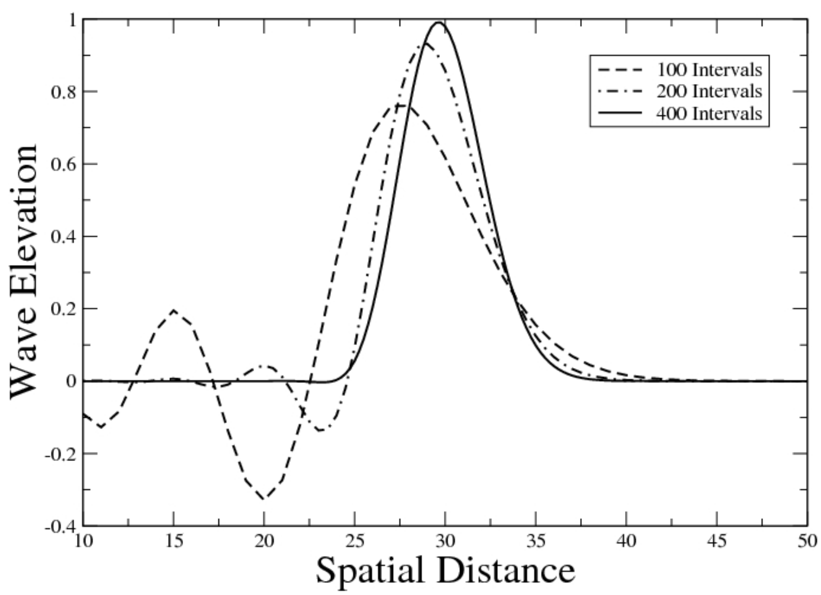


Figure 1. Dispersion in a numerical solution to the 1-D wave equation.

Another example of this phenomenon is shown in figure 3, which shows the steady-state water waves generated by a submerged hydrofoil.[3] This figure shows the distance of the standing wave from the leading edge for a two dimensional solution found on an unstructured triangular mesh. The frequency of these waves is clearly influenced by the mesh resolution. The waves are increasingly out of phase after the first wave crest as the mesh is coarsened. Hence, Richardson extrapolation applied to these solutions will achieve a meaningful result only for the first couple of wave crests.

For numerical schemes and partial differential equations that do not suffer from dispersion, such as a central difference approximation to the heat equation, Richardson extrapolation can be successfully applied to the numerical solution, assuming that the mesh is properly refined and that the numerical scheme is properly implemented. This observation is demonstrated in previous work by Burg and Erwin[4] and is the

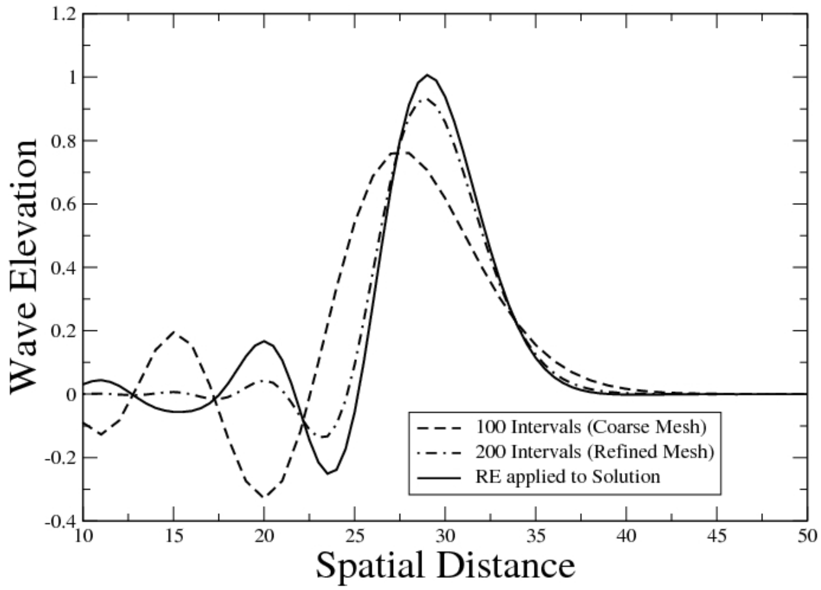


Figure 2. Numerical solution to the 1-D wave equation with Richardson extrapolation applied to the solution.

basis for its successful implementation by Roache on a Poisson problem.[5] Also, if the mesh is sufficiently resolved so that dispersion is unobserved, as with the most refined mesh in figure 1, direct application of Richardson application to the solution may be feasible. Roache and Knupp experienced this behavior in applying Richardson extrapolation to solutions of an advection-diffusion problem.[6] Their solutions only exhibited dispersion for the coarse meshes, resulting in a reduction of the observed order of accuracy. On the refined meshes, where dispersion was not observed, the expected order of accuracy was indeed achieved.

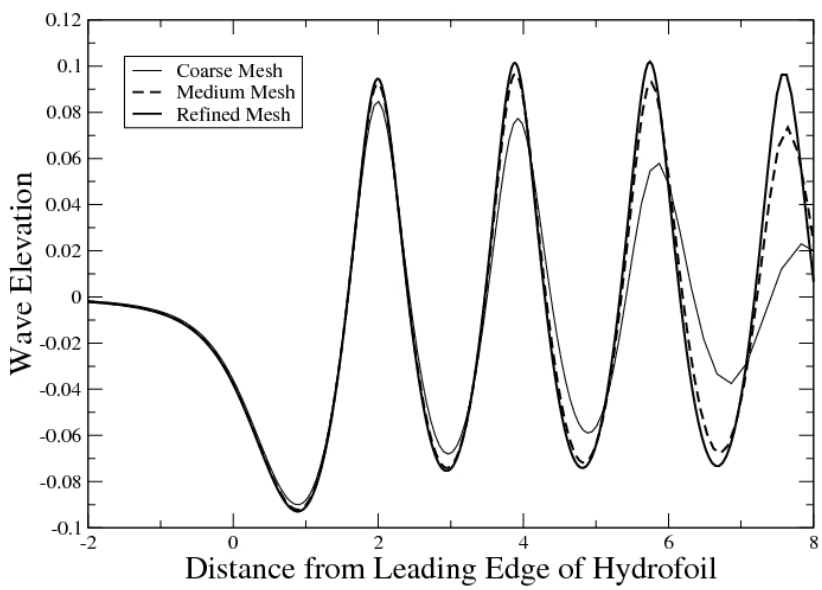


Figure 3. Dispersion in water waves past a submerged hydrofoil.

Using the 1-D counterpart to the approach outlined in section III below, Burg and Erwin[4] were able to increase the order of accuracy of the 1-D wave equation at all mesh resolutions. As shown in figure 4, dispersion is no longer an issue when Richardson extrapolation is applied to the residual at each iteration of the solver rather than to the solution.

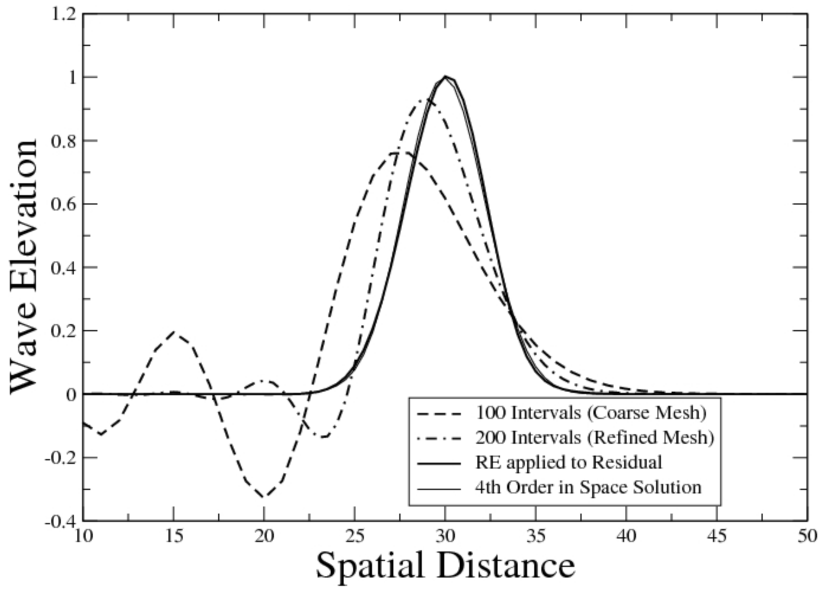


Figure 4. Numerical solution to the 1-D wave equation with Richardson extrapolation applied to the residual.

II.B. Proper Implementation

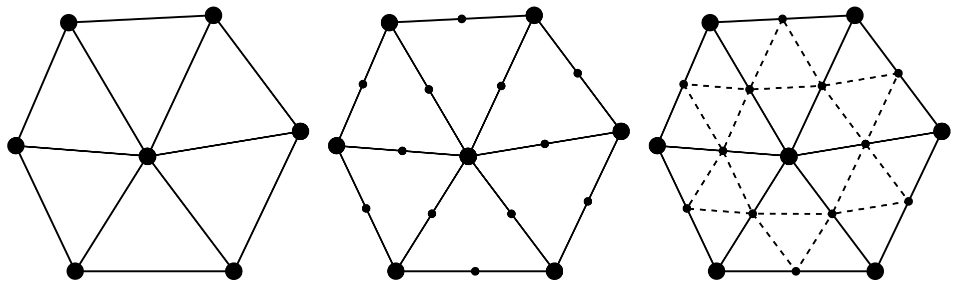
As indicated in section I.A, successful application of Richardson extrapolation to any numerical approximation relies on certain knowledge regarding the approximation. In particular, the original order of accuracy must be known in advance so that the proper Richardson extrapolation formula may be applied. Codes that solve partial differential equations are quite complex; and while the theoretical order of accuracy may be well known and well understood, the code may not achieve this theoretical order of accuracy in practice. For Richardson extrapolation to be applied successfully, the numerical results must demonstrate this order of accuracy. As such, rigorous code verification is critical for success. Unfortunately, rigorous code verification is time consuming, and many advanced computational codes have not undergone this demanding procedure.

Ideally, code verification would be performed through comparison of any numerical solution found for a given partial differential equation or system of equations to a known exact solution. Unfortunately, exact solutions, or at least non-trivial exact solutions, are not generally known for the equations or systems of equations being solved. When this is the case, another approach to code verification must be utilized. In this research, when non-trivial exact solutions are not known, code verification is performed via the method of manufactured solutions.[7,8] As the name implies, the method of manufactured solutions involves "manufacturing" a non-trivial exact solution to the governing partial differential equations in question. A source term is then introduced into the governing equations that drives the solution to the chosen exact solution.

II.C. Mesh Refinement

In addition to the requirement that the order of accuracy of a solver be verified, Richardson extrapolation also depends on a uniform refinement of the discretization size. When applying Richardson extrapolation to a relatively simple approximation such as that of a derivative at a given location, an integral over a bounded region, or a finite difference stencil as presented in section I.A, the discretization size can be easily refined within the scheme. When applying Richardson extrapolation to partial differential equations in one dimension, uniform refinement remains relatively simple. The computational domain, or mesh, in one dimension consists of a straight line subdivided into appropriate intervals. A typical mesh refinement consists of dividing each subinterval of the original mesh in half. The result, obviously, is a new mesh with twice as many intervals as the original, a refinement ratio of 2. With two dimensions, refinement of a structured mesh remains essentially the same, but extended to an extra dimension. The same refinement ratio of 2 increases the number of elements by a factor of 4.

The issue of refinement becomes significantly more difficult, however, when dealing with unstructured meshes in two dimensions. For a sequence of meshes to be appropriate for Richardson extrapolation, the meshes must be geometrically similar. Hence, regions where quadrilaterals are used in a coarse mesh must be gridded using similar quadrilaterals in each successively refined mesh. Similarly, regions where triangles are used must continue to use similar triangles. In the present work, which deals with unstructured triangular meshes, h-refinement is used to ensure similarity among a sequence of meshes. The process of h-refinement is similar to the process described above for refinement in one dimension. To h-refine an unstructured triangular mesh, new nodes are inserted at the midpoint of each edge in the original mesh. Then those new nodes are connected by new edges, subdividing each triangular element into four similar triangles. Figure 5 illustrates this process on a small unstructured mesh of six triangular elements.



(a) Original mesh (b) New nodes inserted (c) Refined mesh

Figure 5. h-refinement of an unstructured triangular mesh.

II.D. Additional Remarks

Because of these issues, Richardson extrapolation when applied directly to the numerical solution of a partial differential equation often fails to increase the order of accuracy of the numerical result. This failure does not imply that Richardson extrapolation can not or should not be applied to this class of numerical simulations; it merely indicates that care must be used in order to achieve the desired result. Two of the issues discussed have already been addressed. Improper implementation can be addressed through rigorous code verification, which is carried out in this research through comparison to exact solutions when available and through the method of manufactured solutions otherwise. In regard to mesh refinement, h-refinement is utilized herein to ensure geometric similarly among a sequence of meshes. Finally, the issue of dispersion is addressed through the approach described in the following section.

III. Numerical Approach

In the proposed method, Richardson extrapolation is applied to the numerical calculation of the discretized equations on a sequence of meshes rather than to the solution on these meshes. The discretized governing equations can be written as

$$\vec{R}\left(\vec{Q}^{n+1}, \vec{Q}^{n}, \ldots, \vec{Q}^{n-k}, \chi, \vec{t}\right) = 0 \tag{7}$$

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 equation must be solved for the value at the new time level. For an explicit scheme, the unknown Q^{n+1} can be determined explicitly, or

$$\vec{Q}^{n+1} = F\left(\vec{Q}^{n}, \ldots, \vec{Q}^{n-k}, \chi, \vec{t}\right) \tag{8}$$

whereas for an implicit scheme, the unknown Q^{n+1} can be determined via Newton's method

$$\frac{\partial \vec{R}}{\partial \vec{Q}^{n+1}} \Delta Q^{n+1,m+1} = -\vec{R}\left(\vec{Q}^{n+1,m}, \vec{Q}^{n}, \ldots, \vec{Q}^{n-k}, \chi, \vec{t}\right) \tag{9}$$

with $\vec{Q}^{n+1,m+1} = \vec{Q}^{n+1,m} + \Delta\vec{Q}^{n+1,m+1}$ and $\vec{Q}^{n+1,0} = \vec{Q}^{n}$.

Burg and Erwin successfully applied Richardson extrapolation to the residual within one-dimensional partial differential equation solvers.[4] Herein, the method is extended to two-dimensional partial differential equations. 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. On a uniform mesh, the resulting residuals vary smoothly since the area is constant; however, on a nonuniform mesh, the residuals must be divided by the area to vary smoothly. This observation is critical for the effectiveness of the Richardson extrapolation algorithm for nonuniform meshes.

On unstructured triangular meshes, a sequence of meshes is obtained via h-refinement as discussed in section II.C. Appropriate information must be maintained in order to communicate information between meshes. Richardson extrapolation can be applied immediately to the nodes that are common to all meshes in the sequence, while interpolation must be performed to pass the information to the new points in the refined meshes.

For node-based finite volume methods and most finite element methods, the variables Q and their gradients ∇Q are stored at the nodes. h-refinement subdivides each edge by placing a point at the midpoint of each edge $\vec{x}_{ij}$. Using the variable and the gradient at the two endpoints at x_i and x_j, a fourth-order accurate interpolation can be achieved at the midpoint via

$$Q_{ij} = \frac{1}{2}\left(Q_i + Q_j\right) + \frac{1}{8}\left(\nabla Q_i \cdot \vec{r}_{ij} - \nabla Q_j \cdot \vec{r}_{ij}\right) \tag{10}$$

where $\vec{r}_{ij} = x_j - x_i$.

The gradient can be estimated to second-order accuracy via Green's Theorem or via a least-squares approach. More complicated stencils have been used to estimate higher derivatives for use within higher order methods, but these stencils are not as compact as those typically used to estimate the gradient.

The interpolation stencil given in Eq. (10) restricts the order of accuracy of this Richardson extrapolation approach to fourth-order accuracy, since the interpolation stencil must be more accurate than the leading order error term being removed via Richardson extrapolation.

The Richardson extrapolation based algorithm applied to the residual involves the following components at each iteration within the solver:

1. Calculate the residual for the coarse mesh $R_{\text{coarse}}\left(\vec{Q}^{n+1}_{\text{coarse}}, \vec{Q}^{n}_{\text{coarse}}, \ldots, \vec{Q}^{n-k}_{\text{coarse}}, \chi_{\text{coarse}}, \vec{t}\right)$
2. Calculate the residual for the refined mesh $R_{\text{fine}}\left(\vec{Q}^{n+1}_{\text{fine}}, \vec{Q}^{n}_{\text{fine}}, \ldots, \vec{Q}^{n-k}_{\text{fine}}, \chi_{\text{fine}}, \vec{t}\right)$
3. Divide both the coarse and refined mesh residuals through by area.
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 the refined mesh residual through by area.
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.

IV. Numerical Results

All of the numerical results presented here utilize a node-based finite volume method.[9] In addition, all solutions are obtained on unstructured triangular meshes. The error in the solution on a sequence of meshes is calculated via the L_1 norm between the computed solution and the exact solution or

$$L_1(Q^{\text{computed}}, Q^{\text{exact}}) = \frac{\sum_{i=1}^{\text{nodes}} \left| Q_i^{\text{computed}} - Q_i^{\text{exact}} \right| A_i}{\sum_{i=1}^{\text{nodes}} A_i} \tag{11}$$

where A_i represents the area of control volume i. Hence, the error when Q^{computed} is found on a particular mesh χ is given by

$$E(\chi) = L_1(Q^{\text{computed}}, Q^{\text{exact}}) \tag{12}$$

The observed order of accuracy is then calculated based on the error of solutions on two related meshes via

$$p = \frac{\ln\left(\frac{E(\chi_{\text{coarse}})}{E(\chi_{\text{fine}})}\right)}{\ln r} \tag{13}$$

where r is the refinement ratio between respective meshes. In the proposed methodology, a refinement ratio of $r = 2$ is used.

IV.A. Laplace's Equation

The first test case deals with Laplace's equation, an elliptic partial differential equation in two dimensions. The governing equation is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{14}$$

For the present case, Laplace's equation is solved on a rectangular domain $[0,1]\times[0,1]$ Dirichlet boundary conditions are applied and are defined as

$$\begin{aligned} u_{\text{left}}(x,y) &= u(0,y) = \sin(\pi y) \\ u_{\text{right}}(x,y) &= u(1,y) = \sin(2\pi y) \\ u_{\text{top}}(x,y) &= u(x,1) = \sin(2\pi x) \\ u_{\text{bottom}}(x,y) &= u(x,0) = \sin(\pi x) \end{aligned} \tag{15}$$

As such, the exact solution is given by

$$u(x,y) = A(y)\sin(2\pi x) + B(y)\sin(\pi x) + A(x)\sin(2\pi y) + B(x)\sin(\pi y) \tag{16}$$

where

$$\begin{aligned} A(x) &= \frac{\sinh(2\pi x)}{\sinh(2\pi)} \\ B(x) &= \cosh(\pi x) - \frac{\sinh(\pi x)}{\tanh(\pi)} \end{aligned} \tag{17}$$

This exact solution is shown in figure 6. A sequence of h-refined meshes on the triangular domain is shown in figure 7.

The error results for Laplace's equation are shown in table 1, along with the observed order of accuracy. For this problem, the observed order of accuracy is second order, which agrees with the theoretical order. In table 2, the errors in the converged solution for the application of Richardson extrapolation to the residual are shown. As expected, the observed order of accuracy has increased to third order.

Table 1. Observed errors for the solution of Laplace's equation.

Elements	Error	Order of Accuracy
64	1.117162e-02	NA
256	2.733610e-03	2.030959
1024	6.724619e-04	2.023283
4096	1.663856e-04	2.014931

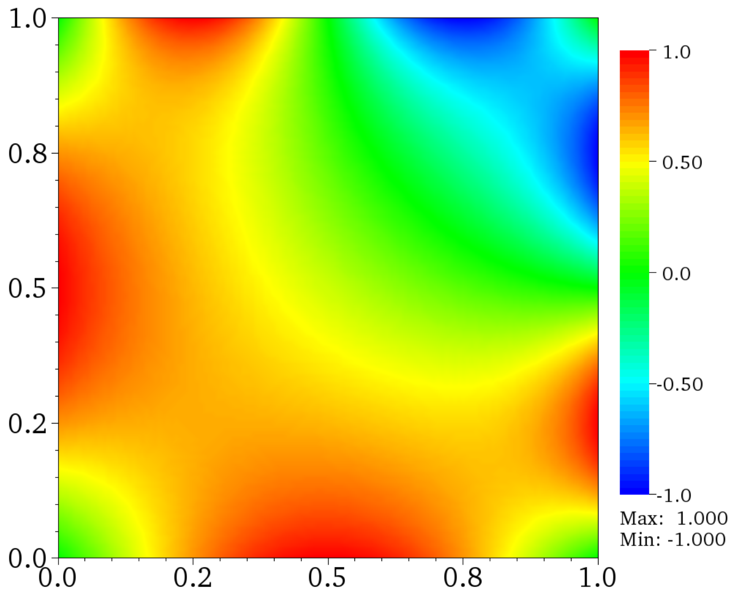
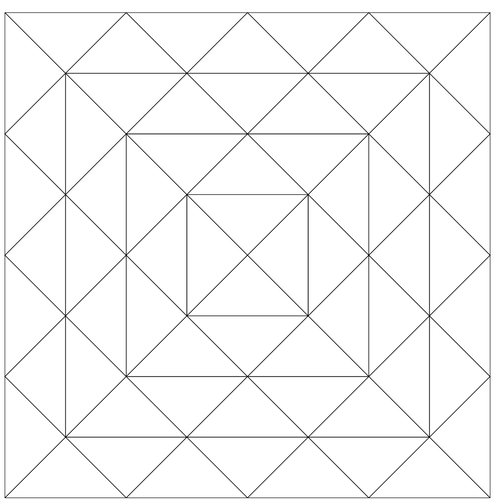
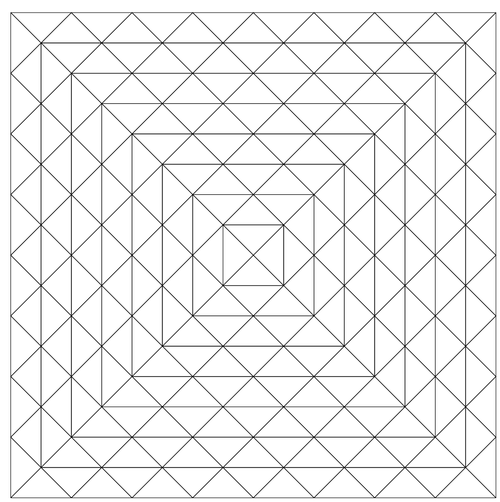


Figure 6. Exact solution for Laplace's equation



(a) 64 elements



(b) 256 elements

Figure 7. Sequence of h-refined meshes for Laplace's equation.

Table 2. Observed errors for Richardson extrapolation applied to the residual of Laplace's equation.

Mesh Pair	Error	Order of Accuracy
32/64	1.014534e-03	NA
64/256	1.553590e-04	2.7071395
256/1024	1.982677e-05	2.9700842
1024/4096	2.535547e-06	2.9670807

IV.B. Wave Equation

A linear coefficient, first-order wave equation in two dimensions is given by

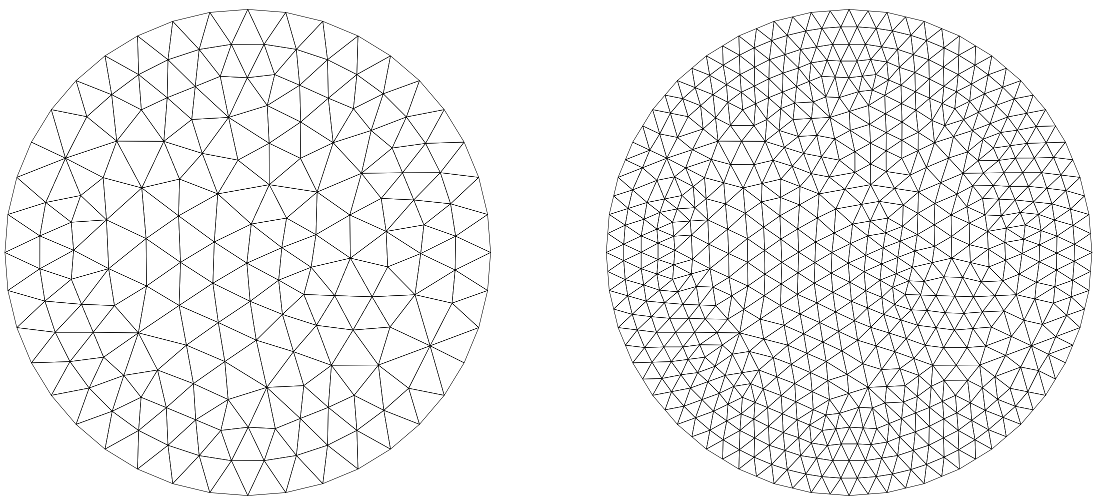
$$\frac{\partial u}{\partial t} - y\frac{\partial u}{\partial x} + x\frac{\partial u}{\partial y} = 0 \tag{18}$$

The choice of coefficients creates a solution that rotates about the origin counter-clockwise with a period of 2π. Thus the exact solution is given by

$$u(x, y, t) = u_0(r\cos(\theta - t), r\sin(\theta - t)) \tag{19}$$

where $\theta = \tan^{-1}(\frac{y}{x})$, $r = \sqrt{x^2 + y^2}$, and $u_0(x, y) = u(0, x, y)$ is the initial condition.

This test case is solved on a circular domain centered about the origin with a radius of 20. A sequence of h-refined meshes on the computational domain is shown in figure 8.



(a) 302 elements

(b) 1208 elements

Figure 8. Sequence of h-refined meshes for the 2D wave equation.

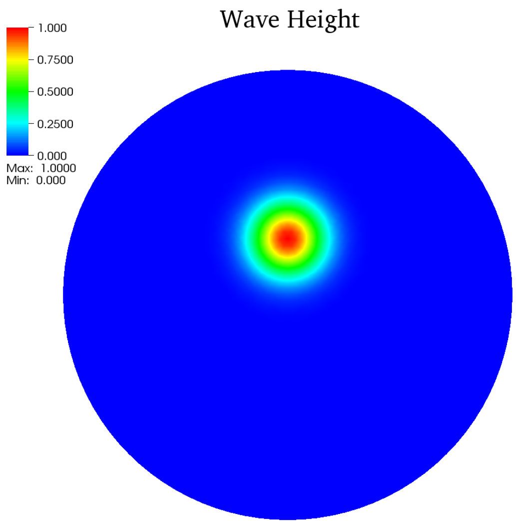
This test case uses a Gaussian initial condition centered at $(0, 5)$ given by

$$u_0(x, y) = e^{\dfrac{-(x^2 + (y-5)^2)}{10}} \tag{20}$$

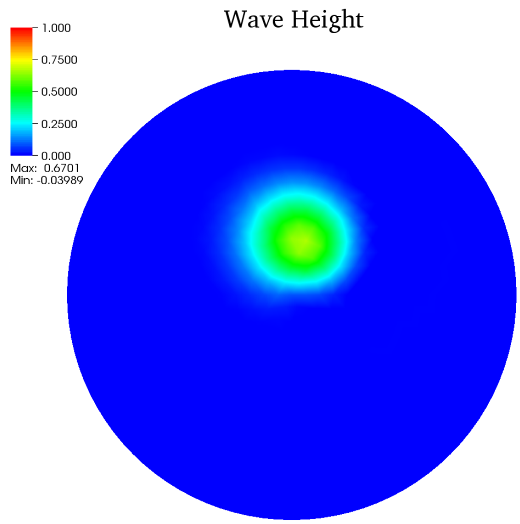
as shown in figure 9(a). The boundary is set to be consistent with the Gaussian function.

In order to evaluate the performance of the code, a time-accurate solution is found at $t = 2\pi$. Due to the nature of the problem, the final solution should exactly match the initial conditions. Figure 9 shows the exact solution at $t = 2\pi$ as well approximate solutions at $t = 2\pi$ on a coarse mesh, a refined mesh, and a refined mesh with Richardson extrapolation applied.

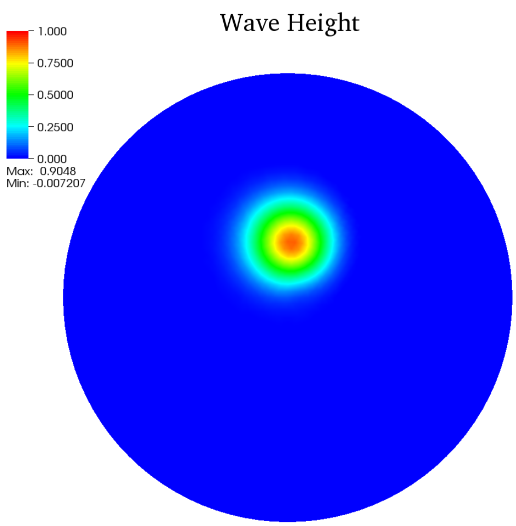
Since the solution at time $t = 2\pi$ should be the same as the initial condition, the improvement achieved can be qualitatively verified by comparing the solutions in figure 9 to the initial condition in figure 9(a). The solution with Richardson extrapolation applied to the residual, shown in figure 9(d), shows less diffusion than either the coarse or refined mesh solutions. There is some dispersive error present in the solution with Richardson extrapolation, however. Table 3 shows the observed order of accuracy of the original solver, while table 4 shows the observed order of accuracy with Richardson extrapolation applied to the residual. As expected, the observed order has been promoted from 2$^{\text{nd}}$ to 3$^{\text{rd}}$.



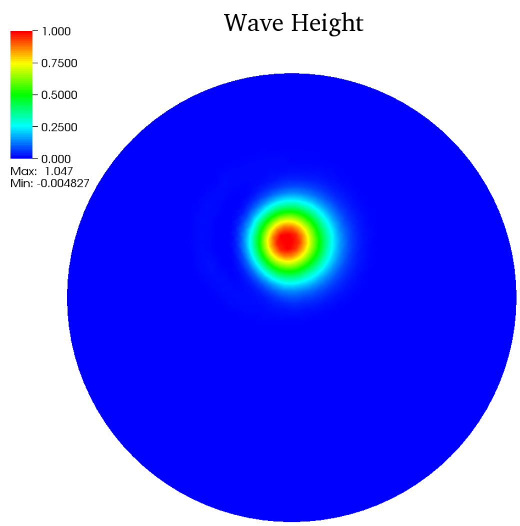
(a) Initial condition and exact solution (4832 elements).



(b) Coarse mesh solution (1208 elements).



(c) Refined mesh solution (4832 elements).



(d) Refined mesh solution with Richardson extrapolation applied to the residual (4832 elements).

Figure 9. Solution at $t = 2\pi$ on a series of meshes for the 2D wave equation.

Table 3. Observed errors for the solution of the wave equation.

Elements	Error	Order of Accuracy
302	4.610231e-02	NA
1208	1.638085e-02	1.492829
4832	3.596115e-03	2.187499

Table 4. Observed errors for Richardson extrapolation applied to the residual of the wave equation.

Mesh Pair	Error	Order of Accuracy
302/1208	2.235483e-02	NA
1208/4832	3.169143e-03	2.818422
4832/19328	3.907454e-04	3.019792

IV.C. Shallow Water Equations

The shallow water equations are a two-dimensional simplification of the incompressible Navier-Stokes equations applied to water flow in open channels.[10] They consist of an equation for the conservation of mass and equations for the conservation of momentum in both dimensions. The depth of flow h represents the mass, and the discharge in the x and y directions, hu and hv respectively, represent the momentum (where u and v are the x and y velocity components). The governing equations are given by

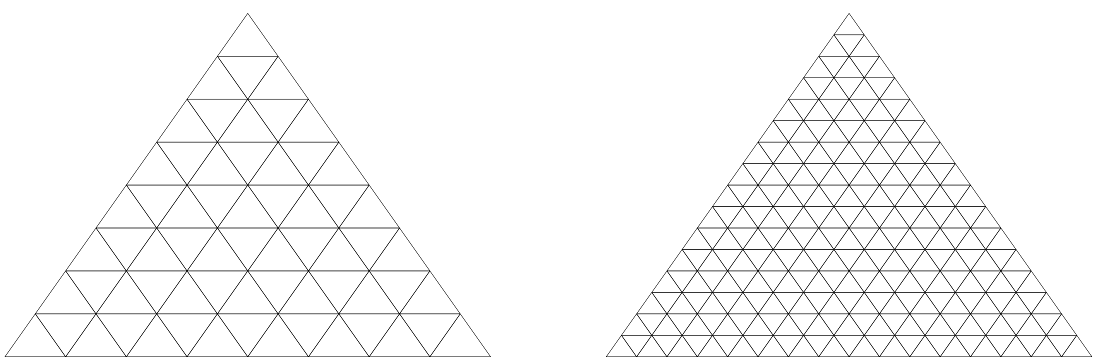
$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} &= 0 \\ \frac{\partial hu}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(huv) &= 0 \\ \frac{\partial hu}{\partial t} + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hu^2 + \frac{1}{2}gh^2\right) &= 0 \end{aligned} \tag{21}$$

IV.C.1. Manufacutured Solution

As exact solutions to these equations do not exist in general, the method of manufactured solutions[7,8] is utilized for code verification purposes. A source term is added so as to drive the solution to a predetermined solution. In particular, the solutions are chosen as

$$\begin{aligned} h(x,y) &= e^{Ax+By} \\ u(x,y) &= Ce^{Dx+Ey} \\ v(x,y) &= Fe^{Gx+Hy} \end{aligned} \tag{22}$$

where A, B, C, D, E, F, G, and H are constants. These manufactured solutions are shown in figure 11. Initial conditions are set the exact solution. Dirichlet boundary conditions are enforced on all boundaries. Code verification for the shallow water code is performed on a triangular domain. A sequence of h-refined meshes on this domain are shown in figure 10.

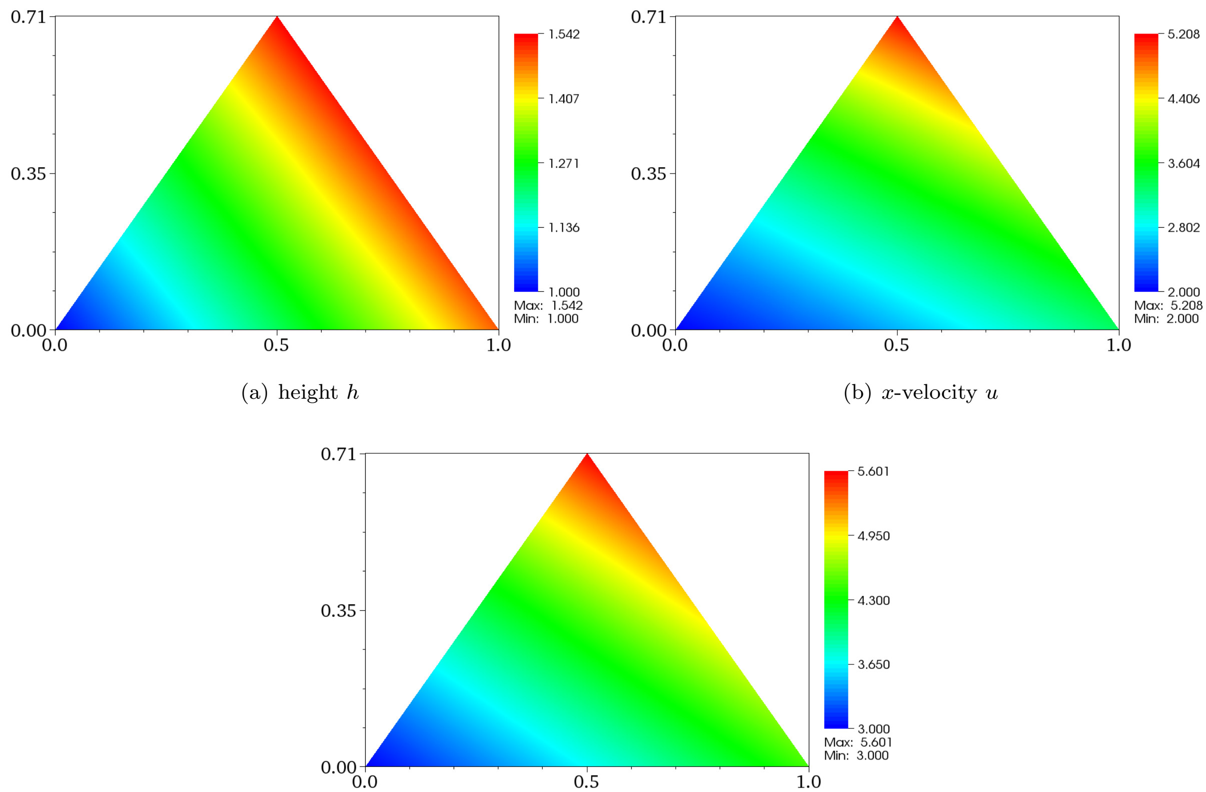


(a) 64 elements

(b) 256 elements

Figure 10. Sequence of h-refined meshes for the method of manufactured solutions of the shallow water equations.

The results for the original shallow water code, in both a first-order and second-order form, are shown in table 5. These results show that the original implementation including the source terms achieves the assumed order of accuracy. In table 6, the errors in the converged solution for the application of Richardson extrapolation to the residual are presented. Since the observed order of accuracy approaches 2 with the first-order code and 3 with the second-order code, the application of Richardson extrapolation to the residual has increased the order of accuracy appropriately.



(a) height h

(b) x-velocity u

(c) y-velocity v

Figure 11. Manufactured solution to the shallow water equations.

Table 5. Observed errors for the solutions of the shallow water equations.

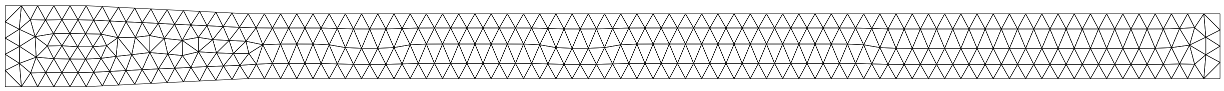
Elements		1st Order Code Error	1st Order Code Order of Accuracy	2nd Order Code Error	2nd Order Code Order of Accuracy
256	h:	1.004811e-03	NA	3.838187e-05	NA
	hu:	1.648477e-02	NA	3.049906e-04	NA
	hv:	1.162203e-02	NA	1.703099e-04	NA
1024	h:	5.380576e-04	0.901092	1.070974e-05	1.841502
	hu:	9.312029e-03	0.823966	9.597013e-05	1.668107
	hv:	6.633756e-03	0.808964	5.728587e-05	1.571911
4096	h:	2.792636e-04	0.946133	2.832796e-06	1.918625
	hu:	4.980990e-03	0.902663	2.720039e-05	1.818958
	hv:	3.621327e-03	0.873307	1.643939e-05	1.801023
16,384	h:	1.428839e-04	0.966784	7.294829e-07	1.957281
	hu:	2.592629e-03	0.942017	7.265597e-06	1.904474
	hv:	1.926644e-03	0.910428	4.347045e-06	1.919050
65,636	h:	7.249758e-05	0.978839	1.852406e-07	1.977474
	hu:	1.332921e-03	0.959824	1.874095e-06	1.954887
	hv:	1.004940e-03	0.938981	1.112884e-06	1.965732
262,144	h:	3.657910e-05	0.986913	4.668948e-08	1.988231
	hu:	6.796087e-04	0.971815	4.755524e-07	1.978518
	hv:	5.162394e-04	0.960997	2.816181e-07	1.982491

Table 6. Observed errors for Richardson extrapolation applied to the residual of the shallow water equations.

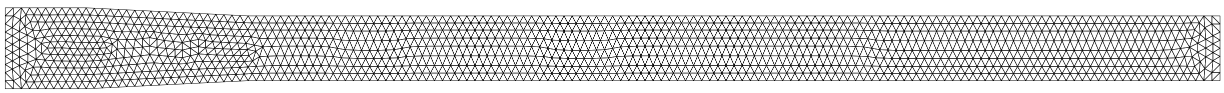
Mesh Pair		1st Order Code Error	1st Order Code Order of Accuracy	2nd Order Code Error	2nd Order Code Order of Accuracy
256/1024	h:	3.461320e-05	NA	3.877790e-07	NA
	hu:	1.585805e-04	NA	1.082680e-05	NA
	hv:	7.641130e-05	NA	6.838467e-06	NA
1024/4096	h:	1.131500e-05	1.613086	5.155147e-08	2.911149
	hu:	5.326408e-05	1.573981	1.658565e-06	2.706599
	hv:	2.654046e-05	1.525593	1.048689e-06	2.705086
4096/16,384	h:	3.213432e-06	1.816050	6.448048e-09	2.999079
	hu:	1.550372e-05	1.780549	2.286307e-07	2.858845
	hv:	7.583350e-06	1.807286	1.450218e-07	2.854245
16,384/65,536	h:	8.551980e-07	1.909785	8.041037e-10	3.003409
	hu:	4.174233e-06	1.893031	2.999300e-08	2.930321
	hv:	1.997324e-06	1.924767	1.919612e-08	2.917383

IV.C.2. Channel Contraction

In order to further evaluate the performance of Richardson extrapolation applied to the residual, a physical test case for the shallow water equations is presented. This test cases consists of a long channel with a single contraction. Figure 12 shows a sequence of h-refined meshes on the computational domain.



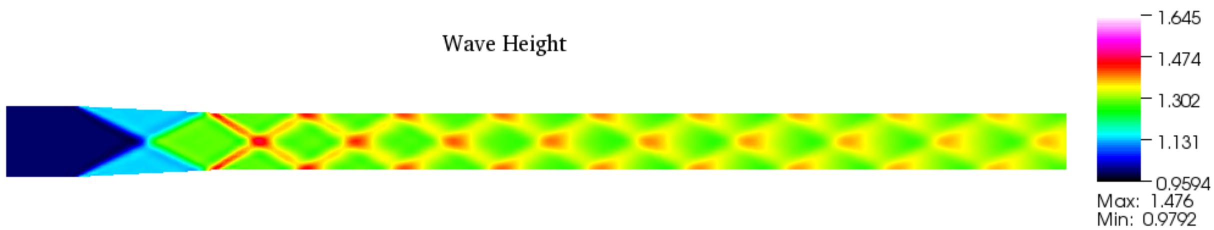
(a) 641 elements



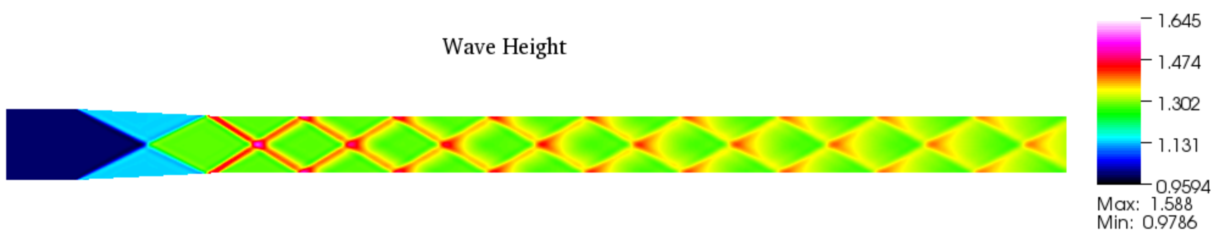
(b) 2564 elements

Figure 12. Sequence of h-refined meshes for the channel contraction case of the shallow water equations.

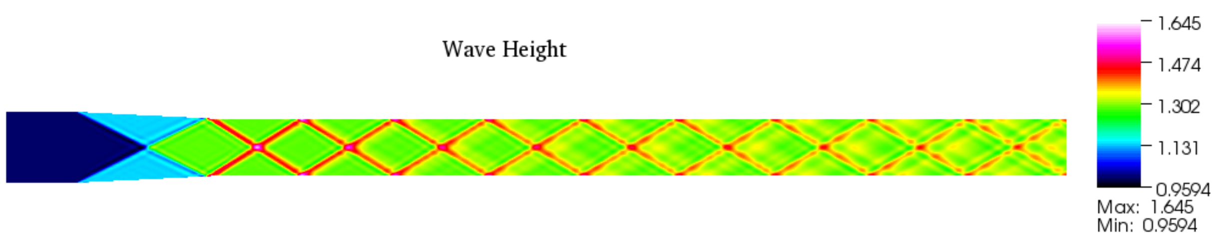
As water flows from left to right through the channel, the contraction creates a series of standing waves that propagate downstream. Figure 13 shows the solution on a coarse and a refined mesh, as well as the solution with Richardson extrapolation applied on the refined mesh.



(a) Coarse mesh solution (10256 elements).



(b) Refined mesh solution (41024 elements).



(c) Refined mesh solution with Richardson extrapolation applied to the residual (41024 elements).

Figure 13. Solution on a series of meshes for the channel contraction case of the shallow water equations.

The solution in figure 13(c) shows much more refined wave structure as the flow moves downstream. This improvement stems from the elimination of the leading order, diffusive error term. However, dispersion is evident in the Richardson extrapolation solution that did not appear in either the coarse or refined solutions.

IV.D. Compressible Euler Equations

The compressible Euler equations are a simplification of the compressible Navier-Stokes equations with viscous and heat conduction terms removed. They consists of an equation for the conservation of mass, equations for the conservation of momentum in all dimensions, and an equation for the conservation of energy. In two-dimensions the governing equations are

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} &= 0 \\
\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}(\rho u^2 + P) + \frac{\partial}{\partial y}(\rho u v) &= 0 \\
\frac{\partial \rho v}{\partial t} + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v^2 + P) &= 0 \\
\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}((E+P)u) + \frac{\partial}{\partial y}((E+P)v) &= 0
\end{aligned} \tag{23}$$

An equation relating energy and pressure closes the system

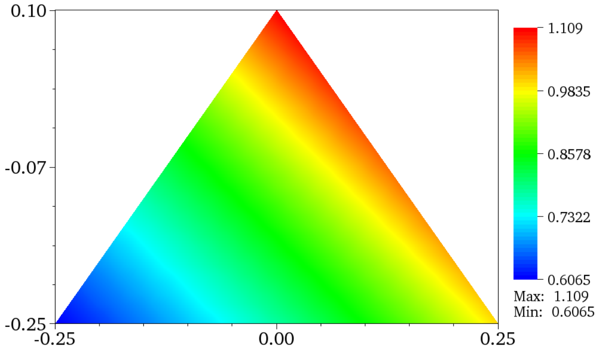
$$E = \frac{P}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2) \tag{24}$$

As exact solutions to these equations do not exist in general, the method of manufactured solutions[7,8] is utilized for code verification purposes. A source term is added so as to drive the solution to a predetermined solution. In particular, the solutions are chosen as

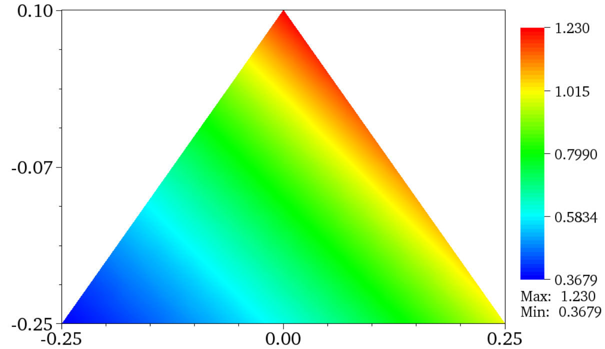
$$\begin{aligned}
\rho(x,y) &= e^x e^y \\
u(x,y) &= e^{2x} e^{2y} \\
v(x,y) &= e^{2x} e^{2y} \\
E(x,y) &= \frac{3}{2} e^{5x} e^{5y} \\
P(x,y) &= \frac{1}{2}(\gamma - 1) e^{5x} e^{5y}
\end{aligned} \tag{25}$$

These manufactured solutions are shown in figure 14. Initial conditions are set to the exact solution. Dirichlet boundary conditions are enforced on all boundaries. Code verification for the compressible Euler equations is performed on a triangular domain. A sequence of h-refined meshes on this domain is shown in figure 15.

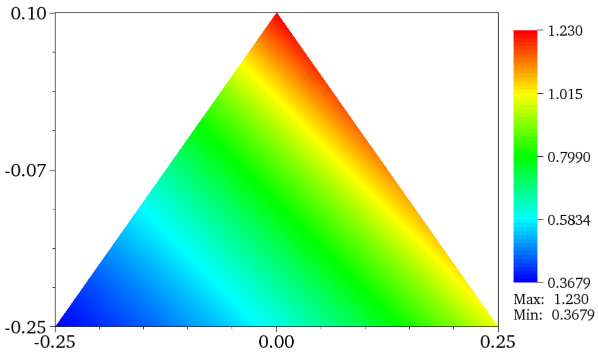
The errors and observed orders of accuracy for the compressible Euler code, in both a first-order and second-order form, are shown in table 7. These results show that the original implementation achieves the assumed order of accuracy. In table 8, the errors and observed orders in the converged solution for the application of Richardson extrapolation to the residual are presented. Since the observed order of accuracy approached 2 with the first-order code and 3 with the second-order code, the application of Richardson extrapolation to the residual has increased the order of accuracy appropriately.



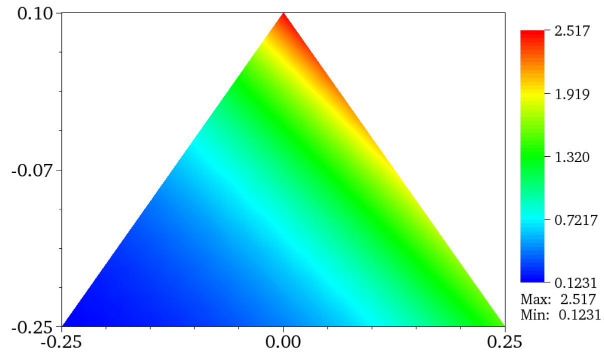
(a) density ρ



(b) x-velocity u

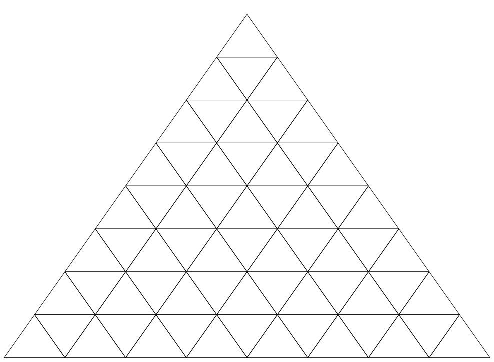


(c) y-velocity v

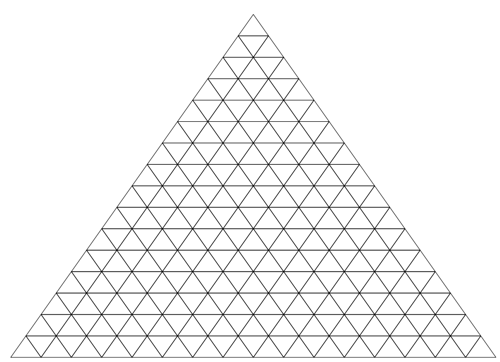


(d) energy E

Figure 14. Manufactured solution to the compressible Euler equations.



(a) 64 elements



(b) 256 elements

Figure 15. Sequence of h-refined meshes for the method of manufactured solutions of the compressible Euler equations.

Table 7. Observed errors for the solutions of the compressible Euler equations.

Elements		1st Order Code Error	1st Order Code Order of Accuracy	2nd Order Code Error	2nd Order Code Order of Accuracy
64	ρ:	6.243661e-02	NA	6.288113e-04	NA
	ρu:	3.561943e-02	NA	5.384710e-04	NA
	ρv:	2.983548e-02	NA	5.272185e-04	NA
	E:	2.340341e-01	NA	1.554669e-03	NA
256	ρ:	4.318788e-02	0.531766	1.704803e-04	1.883022
	ρu:	1.421057e-02	1.325700	9.583576e-05	2.490233
	ρv:	2.138820e-02	0.480214	1.541572e-04	1.773999
	E:	1.332932e-01	0.812116	3.588894e-04	2.114996
1024	ρ:	2.540084e-02	0.765750	5.013527e-05	1.765707
	ρu:	7.985967e-03	0.831425	2.073865e-05	2.208242
	ρv:	1.607354e-02	0.412127	4.390791e-05	1.811850
	E:	7.295189e-02	0.869586	9.017603e-05	1.992723
4096	ρ:	1.465793e-02	0.793195	1.381394e-05	1.859701
	ρu:	4.949478e-03	0.690191	5.148393e-06	2.010128
	ρv:	1.089084e-02	0.561572	1.195170e-05	1.877265
	E:	3.828206e-02	0.930277	2.368187e-05	1.928961
16,384	ρ:	8.297815e-03	0.820878	3.594767e-06	1.942155
	ρu:	2.934734e-03	0.754047	1.323244e-06	1.960043
	ρv:	6.653576e-03	0.710913	3.109137e-06	1.942630
	E:	1.965661e-02	0.961654	6.068572e-06	1.964354
65,636	ρ:	4.648307e-03	0.836026	9.141098e-07	1.975459
	ρu:	1.646155e-03	0.834130	3.326990e-07	1.991790
	ρv:	3.780576e-03	0.815524	7.912707e-07	1.974271
	E:	9.955006e-03	0.981520	1.538011e-06	1.980291

Table 8. Observed errors for Richardson extrapolation applied to the residual of the compressible Euler equations.

Mesh Pair		1st Order Code Error	1st Order Code Order of Accuracy	2nd Order Code Error	2nd Order Code Order of Accuracy
64/256	ρ:	3.690484e-03	NA	1.377267e-04	NA
	ρu:	1.840476e-03	NA	9.524637e-05	NA
	ρv:	3.205471e-03	NA	1.285300e-04	NA
	E:	1.224439e-02	NA	3.146878e-04	NA
256/1024	ρ:	1.977173e-03	0.900371	2.112359e-05	2.704881
	ρu:	7.771399e-04	1.243833	1.294531e-05	2.879235
	ρv:	1.634139e-03	0.972006	2.127177e-05	2.595093
	E:	4.109822e-03	1.574973	5.452998e-05	2.528800
1024/4096	ρ:	6.817397e-04	1.536146	2.614919e-06	3.014017
	ρu:	2.636221e-04	1.559703	1.651407e-06	2.970662
	ρv:	5.600276e-04	1.544961	3.014397e-06	2.818999
	E:	1.178449e-03	1.802187	7.943693e-06	2.779168
4096/16,384	ρ:	1.878626e-04	1.859543	3.319775e-07	2.977609
	ρu:	7.057136e-05	1.901317	2.081320e-07	2.988125
	ρv:	1.559270e-04	1.844627	4.075815e-07	2.886709
	E:	3.140888e-04	1.907645	1.078940e-06	2.880195
16,384/65,536	ρ:	4.800093e-05	1.968544	4.329100e-08	2.938946
	ρu:	1.763741e-05	2.000444	2.642980e-08	2.977261
	ρv:	4.052927e-05	1.943835	5.393261e-08	2.917859
	E:	8.010191e-05	1.971264	1.409335e-07	2.936528

V. Conclusion

A method for extending Richardson extrapolation to general two-dimensional partial differential equation solvers has been developed. Richardson extrapolation has been applied to the discretized result (i.e., the residual) developed within existing numerical partial differential equation solvers. By applying Richardson extrapolation directly to the residual on a sequence of h-refined, unstructured triangular meshes, so that the residual is higher order, the solution maintains the similar higher order accuracy.

Several issues regarding Richardson extrapolation have been addressed. Successful application to partial differential equations that exhibit significant dispersion has been performed via application to the residual rather than the solution. The issue of proper code implementation has been addressed through extensive code verification, either through comparison to the known exact solution, or through a "manufactured" solution. Lastly, Geometrically similar meshes have been created through h-refinement.

A key advantage of this new methodology over other higher order methods involves ease of implementation. The application of Richardson extrapolation treats the existing solver as a "black box," requiring minimal modification of the existing solver algorithms. As such, the greatest motivation behind this research has been the development of a method with the ability to promote existing, mature solvers to higher orders of accuracy while minimizing the amount of work involved.

This new methodology has been verified on four two-dimensional PDE solvers: Laplace's equation and a 2D wave equation through comparison to the known exact solution, and the shallow water equations and compressible Euler equations via the method of manufactured solutions. Additionally, a physical test case for the shallow water equations was studied. The test case involved a long open channel with a contraction near the inlet. While a significant reduction in diffusive error was observed, new dispersive error appeared which could not be explained.

Future work could focus on further investigation of the observed dispersion. The application of Richardson extrapolation developed herein has shown a great deal of success and holds a great deal of promise.

References

[1] Cheney, W. and Kincaid, D., *Numerical Mathematics and Computing*, Thomson, 5th ed., 2004.

[2] Baker, T. J., "Parsing the Results of the Second Drag Prediction Workshop," *23rd AIAA Applied Aerodynamics Conference*, Toronto, Ontario, CA, 2005, AIAA-2005-4731.

[3] Burg, C. O. E., Sreenivas, K., Hyams, D. G., and Mitchell, B., "Unstructured Nonlinear Free Surface Flow Solutions: Validation and Verification," *32nd AIAA Fluid Dynamics Conference*, St. Louis, MI, 2002, AIAA-2002-2977.

[4] Burg, C. O. E. and Erwin, T., "Application of Richardson Extrapolation to the Numerical Solution of Partial Differential Equations," *Numerical Methods for Partial Differential Equations*, Vol. 25, No. 4, July 2009, pp. 810–832.

[5] Roache, P. J., "Scaling of High Reynolds Number Weakly Separated Channel Flows," *Symposium on Numerical and Physical Aspects of Aerodynamic Flows*, 1981.

[6] Roache, P. J. and Knupp, P. M., "Completed Richardson Extrapolation," *Communication in Numerical Methods in Enginerring*, Vol. 9, No. 5, 1994, pp. 365–374.

[7] Murali, V. K., *Code Verification Using the Method of Manufactured Solutions*, Master's thesis, Mississippi State University, Dec. 2002.

[8] Roache, P. J., *Verification and Validation in Computation Science and Engineering*, Hermosa, 1998.

[9] Hyams, D. G., *An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Unstructured Topologies*, Ph.D. thesis, Mississippi State University, May 2003.

[10] Whitfield, D. L., "Numerical Solution of the Shallow Water Equations," Tech. Rep. MSSU-EIRS-ERC-96-4, Mississippi State University, July 1996.