

A Survey of the Isentropic Euler Vortex Problem using High-Order Methods

Seth C. Spiegel, * H. T. Huynh, † and James R. DeBonis †

NASA Glenn Research Center, Cleveland, OH, 44135, USA

The flux reconstruction (FR) method offers a simple, efficient, and easy to implement method, and it has been shown to equate to a differential approach to discontinuous Galerkin (DG) methods. The FR method is also accurate to an arbitrary order and the isentropic Euler vortex problem is used here to empirically verify this claim. This problem is widely used in computational fluid dynamics (CFD) to verify the accuracy of a given numerical method due to its simplicity and known exact solution at any given time. While verifying our FR solver, multiple obstacles emerged that prevented us from achieving the expected order of accuracy over short and long amounts of simulation time. It was found that these complications stemmed from a few overlooked details in the original problem definition combined with the FR and DG methods achieving high-accuracy with minimal dissipation. This paper is intended to consolidate the many versions of the vortex problem found in literature and to highlight some of the consequences if these overlooked details remain neglected.

I. Introduction

Accurate and efficient numerical methods for the simulation of complex aerodynamic flows are desired to help reduce the design and development costs of aerodynamic vehicles and aeropropulsion systems. The primary challenge lies in the prediction of the turbulent motion within the flow and its effect on the overall motion of the fluid. Advanced methods such as large-eddy simulation (LES) and direct numerical simulation (DNS) offer great advancements in predicting turbulence, but these methods require significant improvements in accuracy and efficiency in order to become widely used. Focusing on either the accuracy or efficiency aspect of the underlying numerical algorithms, when developing a computational fluid dynamics (CFD) code, usually demands concessions from the other parameter.

Promising results have been found applying LES in the realm of aeropropulsion, but the spatial and temporal resolution needed for LES requires high-order (higher than second) numerical methods historically only found in structured grid methods with large stencils. These methods have numerous drawbacks: difficulty in grid generation, high sensitivity to grid quality, complex boundary conditions and poor scalability for parallel computing. These drawbacks severely limit the ability of these codes to compute complex aeropropulsion configurations, e.g., noise suppressing nozzles.¹

In an attempt to alleviate the difficulties of structured grid generation for complex geometries, researchers have applied LES methods to unstructured grids.^{2,3} However, the limited second-order accuracy of current unstructured methods requires an exorbitant number of grid points and offers little incentive over the traditional structured methods.^{4,5} The flux reconstruction (FR) method⁶⁻⁸ provides a simple, efficient, and easy to implement method for solving the Navier-Stokes equations on unstructured grids and is accurate to an arbitrary order. By employing the high-order FR method with LES to unstructured grids, limitations related to structured grid generation, grid resolution, and computational efficiency can be overcome.⁹⁻¹²

*NASA Postdoctoral Fellow, Inlet and Nozzle Branch, 21000 Brookpark Road, and AIAA Member.

†Aerospace Engineer, Inlet and Nozzle Branch, 21000 Brookpark Road, and AIAA Associate Fellow.

II. Numerical Methods

II.A. Spatial Discretization

We briefly review the FR method below. Consider the 1D conservation law

$$u_t + f_x = 0 \quad (1)$$

where the subscripts t and x denote partial differentiation with respect to time and space, respectively. Let the computational domain Ω be divided into a arbitrary number of nonoverlapping cells. Each cell is transformed from physical space within the global domain to a common reference space/element, $\mathbf{I} = [-1, 1]$, using the mapping $r = \chi(x)$, where r is the local coordinate variable in the reference space. An inverse of this one-to-one mapping is assumed to exist, and it can be used to transform from reference to physical space using $x = \chi^{-1}(r)$. The details of such affine mappings are beyond the scope of this paper; the interested reader can find additional information in many texts on spectral-/finite-element type methods, including but not limited to Ref. 13–16. The following review focuses on using the FR method to solve the transformed version of equation (1) within each reference element.

At a given time t , let the exact solution to the transformed conservation law within the reference element be given by $u_E(r, t)$ which is not guaranteed to be a polynomial or even a function that is smooth and/or continuous. Let $u(r, t)$ be a polynomial of degree P and approximate $u_E(r, t)$ within the reference element. This approximate *solution polynomial*, $u(r, t)$, is local to each element, and may or may not be continuous across cell interfaces. With the dependence on time being understood, we will drop the variable t from $u_E(r, t)$ and $u(r, t)$ to simplify notation.

Let N_P refer to the minimum number of *solution points* needed to define the solution polynomial within the reference element, and let ξ denote the set of solution points, $\xi_n, n = 1, 2, \dots, N_P$. Using tensor products to extend this to higher dimensions, the \mathcal{N} -D conservation law requires $N_P = (P+1)^{\mathcal{N}}$ solution points within each computational cell.

The Lagrange polynomial is defined as the interpolating polynomial through a set of nodal points, for example ξ , that takes the value 1 at ξ_n and zero at all other points $\xi_{i \neq n}$:

$$\ell_n(r) = \prod_{\substack{i=1 \\ i \neq n}}^{N_P} \frac{r - \xi_i}{\xi_n - \xi_i} \quad (2)$$

Note that the Lagrange polynomial in equation (2) is a polynomial of degree $P = N_P - 1$ and the set of Lagrange polynomials for all solution points in ξ is a set of N_P polynomials each of degree P . When extending to higher dimensions using tensor products, each Lagrange polynomial is of degree $2P$ for 2D and of degree $3P$ for 3D.

A *nodal coefficient* of the solution, denoted by \tilde{u}_n , is simply the value of the approximate solution polynomial evaluated at the solution point, ξ_n , i.e., $\tilde{u}_n = u(\xi_n)$. Let $\tilde{\mathbf{u}}_P$ denote a vector containing the set of N_P nodal coefficients for the solution at each of the solution points. Assuming that all of $\tilde{\mathbf{u}}_P$ is known, we can use the Kronecker delta property of equation (2) to construct the approximate solution polynomial as

$$u(r) = \sum_{n=1}^{N_P} \tilde{u}_n \ell_n(r) \quad (3)$$

The above relation is why the set of Lagrange polynomials can also be referred to as a *nodal basis* set. The tilde accent is used in this section to identify the nodal coefficients for a polynomial function.

The derivative of the solution polynomial is a linear combination of the derivatives of the polynomial basis functions

$$\frac{d}{dr} u(r) = u_r(r) = \sum_{n=1}^{N_P} \tilde{u}_n \left[\frac{d}{dr} \ell_n(r) \right] \quad (4)$$

Using the previous equation, we can define the derivative operator, \mathbf{D}_r , as the $N_P \times N_P$ matrix with elements

$$D_r[i, j] = \left[\frac{d}{dr} \ell_j(r) \right] \Big|_{\xi_i} \quad (5)$$

that transforms the nodal coefficients, $\tilde{\mathbf{u}}_P$, into the derivative nodal coefficients, $\tilde{\mathbf{u}}_r$. Note that this derivative is exact only when the function being differentiated is a polynomial of degree P or less.

If we use the nodal solution to evaluate the flux function, f , at each solution point, then interpolate these N_P values by a polynomial of degree P , we get what is called the *discontinuous flux function*. It only uses information inside a given cell and does not account for interaction between it and the data in the neighboring cells. Using the superscript D for discontinuous, the discontinuous flux polynomial, $f^D(r)$, is given by

$$f^D(r) = \sum_{n=1}^{N_P} \tilde{f}_n \ell_n(r) \quad (6)$$

where the nodal coefficients of the discontinuous flux, \tilde{f}_n , are evaluated at the solution points using the nodal coefficients of the solution, i.e., $\tilde{f}_n = f(\tilde{u}_n)$. This collocation projection results in $f^D(r)$ being a polynomial of degree P , which only matches the true flux function if the conservation law in equation (1) is the linear advection equation. For a nonlinear conservation law such as those found in the Euler equations or Navier-Stokes equations, the true flux function evaluated using the solution polynomial is of much higher degree than P and possibly even irrational, and $f^D(r)$ above may not provide a good approximation. We will explore other ways of approximating the true flux function later.

With $f^D(r)$ given by equation (6), we can compute the derivative of the approximated discontinuous flux using the derivative operator from equation (5). Denoting $\tilde{\mathbf{f}}$ as the column vector containing the coefficients \tilde{f}_n , $n = 1, 2, \dots, N_P$, the derivative $\frac{d}{dr} [f^D(r)]$ can be approximated using the matrix-vector product

$$\tilde{\mathbf{f}}_r = \mathbf{D}_r \tilde{\mathbf{f}} \quad (7)$$

where, $\tilde{\mathbf{f}}_r$ is the column vector containing the values of the approximate flux derivative at each solution point. Note that the collocation projection of the discontinuous flux in equation (6) is of degree P . Consequently, its derivative is of degree $P - 1$.

If we employ the above f_r^D to evaluate f_x for the conservation law, we obtain erroneous solutions since such a derivative does not include the interaction of the data between adjacent cells. Therefore, we seek to construct a so-called *continuous flux function*, denoted by $F(r)$, which accounts for this interaction between adjacent cells and approximates the discontinuous flux function in some sense, to which we can then calculate its derivative. The continuous flux function will be obtained by adding a correction to f^D , the discontinuous one.

To this end, at each (cell) interface, let u_L and u_R be the values taken on by the solution polynomials to its left and right, respectively. An upwind flux value common for the two adjacent cells, denoted by f^* , can then be defined via the wind direction. Here, for the case of the Euler equations, we use Roe's flux¹⁷ with an entropy fix.¹⁸

For the cell e , let f_{LB}^* denote the upwind flux at the 'left boundary' of the cell and f_{RB}^* denote the upwind flux at the 'right boundary' of the cell. We can then construct the continuous flux function, $F(r)$, as a polynomial of degree $P + 1$ by requiring that it takes on the upwind flux values at the two cell interfaces, and it approximates f^D via $P - 1$ conditions. Instead of defining F , we define $F - f^D$, which approximates the zero function. At the two interfaces $r = \pm 1$, the function $F(r) - f^D(r)$ takes on the following values, which are called the flux jumps:

$$F(-1) - f^D(-1) = f_{LB}^* - f^D(-1) \quad \text{and} \quad F(1) - f^D(1) = f_{RB}^* - f^D(1) \quad (8)$$

Therefore, $F(r) - f^D(r)$ has prescribed left and right interface values, is of degree $P + 1$, and approximates zero.

We now separate the prescription of the jump at the left interface from that of the right. This separation plays a critical role in the new approach. Again, using the local coordinate, let g_{LB} be the *correction function* for the left interface of the reference element defined by

$$g_{LB}(-1) = 1, \quad g_{LB}(1) = 0 \quad (9)$$

and g_{LB} is a polynomial of degree $P + 1$ approximating the zero function in some sense. We always choose g_{RB} by reflection: $g_{RB}(r) = g_{LB}(-r)$. As a result,

$$g_{RB}(-1) = 0, \quad g_{RB}(1) = 1 \quad (10)$$

Consider the left interface $r = -1$. The polynomial

$$f^D(r) + [f_{\text{LB}}^* - f^D(-1)] g_{\text{LB}}(r) \quad (11)$$

provides a correction at the left interface for $f^D(r)$ by changing the flux value at this interface from $f^D(-1)$ to f_{LB}^* , while leaving the value at the right interface unchanged, namely, $f^D(1)$. Next, the polynomial

$$F(r) = f^D(r) + [f_{\text{LB}}^* - f^D(-1)] g_{\text{LB}}(r) + [f_{\text{RB}}^* - f^D(1)] g_{\text{RB}}(r) \quad (12)$$

provides corrections to both interfaces; using equation (9) and equation (10), one can easily verify that $F(-1) = f_{\text{LB}}^*$ and $F(1) = f_{\text{RB}}^*$. Thus, the above $F(r)$ is of degree $P + 1$, takes on the upwind flux values at the two interfaces, and approximates $f^D(r)$ in the same sense that g_{LB} and g_{RB} approximate the zero function.

The derivative of $F(r)$ at the solution point ξ_n is evaluated as

$$\left. \frac{\partial F(r)}{\partial r} \right|_{\xi_n} = F_r(\xi_n) = \tilde{f}_{r,n} + [f_{\text{LB}}^* - f^D(-1)] g'_{\text{LB}}(\xi_n) + [f_{\text{RB}}^* - f^D(1)] g'_{\text{RB}}(\xi_n) \quad (13)$$

After using the inverse mapping, χ^{-1} , from reference space to physical space, F_r can be employed to approximate f_x at each solution point within the element. Finally, the solution to equation (1) can then be advanced in time via an explicit Runge-Kutta method.

In order to complete this brief review of the FR method, we must define the correction function g_{LB} ; the definition of g_{RB} follows by reflection as discussed previously. With L_{P+1} denoting the (standard) Legendre polynomial of degree $P + 1$, the right Radau polynomial of degree $P + 1$ ($P \geq 0$) is defined by

$$R_{R,P+1} = \frac{(-1)^{P+1}}{2} (L_{P+1} - L_P) \quad (14)$$

A family of correction functions for the left boundary that results in stable schemes is given by

$$g_{\text{LB}} = (\alpha) R_{R,P+1} + (1 - \alpha) R_{R,P} \quad \text{with} \quad 0 \leq \alpha \leq 1 \quad (15)$$

Note that $\alpha = 1$ results in the discontinuous Galerkin (DG) method, and $\alpha = \frac{P+1}{2P+1}$ results in the modified Spectral Difference (SD) method that is stable.

II.B. Temporal Discretization

The results presented in this paper were integrated in time using the 3-stage strong stability preserving (SSP) Runge-Kutta method,¹⁹ which is 3rd-order accurate in time. All of the results used a constant (global) time step to advance the solution in time. The size of the time step for each simulation was chosen to be sufficiently small meaning further reduction in its size would not produce significant changes in the solution error.

III. Governing Equations

The Euler equations describe the motion of an inviscid fluid such as the isentropic vortex that is the topic of this paper. The 2D Euler equations for a calorically perfect, compressible fluid can be written in differential form as

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho v_x \\ \rho v_x^2 \\ \rho v_x v_y \\ v_x (\rho E + p) \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 \\ v_y (\rho E + p) \end{bmatrix} = 0 \quad (16)$$

where ρ , v_i , p , and E respectively refer to the density, velocity in coordinate direction i , pressure, and the total energy per unit mass of the system. Thermodynamic closure is found through the total energy equation

$$\rho E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho (v_x^2 + v_y^2) \quad (17)$$

where γ is the ratio of specific heats for the fluid in question. Additionally, the ideal gas law

$$p = \rho R_{gas} T \quad (18)$$

is utilized to relate the temperature, T , to the pressure and density where R_{gas} is the specific gas constant of the fluid. The speed of sound, a , is computed using the relation

$$a = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\gamma R_{gas} T} \quad (19)$$

In the present work, the fluid is assumed to be calorically perfect air where $\gamma = 1.4$ and the dimensionalized specific gas constant is $R_{gas} = 287.15 \frac{\text{J}}{\text{kg}\cdot\text{K}}$.

IV. Isentropic Euler Vortex Problem

IV.A. Problem Definition

The isentropic Euler vortex problem is commonly used^{9,14,20-30} for testing the order of accuracy of a numerical method because it is easy to implement and the exact solution is known at all times. This problem is ideal for high-order methods because their theoretical accuracy should allow them to propagate the vortex with minimal entropy production for an indefinite amount of time. Being able to sustain vortical flow structures without the numerical method contributing unwanted numerical dissipation is crucial for advanced turbulence methods like LES. All of these reasons are why this is becoming a popular test case for high-order methods that are designed to produce minimal numerical dissipation.

There are numerous variations in the definition of the vortex problem throughout literature. To the authors' knowledge, the first version of this problem was used by Shu²⁰ to compare high-order weighted essentially non-oscillatory (WENO) finite-volume (FV) methods with a traditional second-order FV method. The definition for the vortex problem that is presented here modifies this first version in an attempt to unify many of the variations that exist.

IV.A.1. Analytic Solution

The initial conditions for the vortex problem are found by superimposing perturbations in velocity and temperature onto a uniform mean flow. These perturbations are defined such that the entropy, $S = p/\rho^\gamma$, is constant throughout the entire grid domain, i.e., $\delta S = 0$. By compounding various parameters, these perturbations are ultimately based upon a Gaussian function of the form

$$\Omega = \beta e^f \quad (20)$$

where β is the maximum strength of the perturbation and the function f determines the strength of the perturbation at a particular grid coordinate through the relation

$$f(x, y) = -\frac{1}{2\sigma^2} \left[\left(\frac{x}{R}\right)^2 + \left(\frac{y}{R}\right)^2 \right] \quad (21)$$

The parameter σ is the familiar standard deviation term found in a general Gaussian function that is used to control the spread of the distribution, whereas R is a characteristic length scale for the grid. These two parameters, σ and R , could have been combined into a single parameter to further simplify the problem definition, but they have been kept separate in order to unify dimensional and nondimensional versions of the vortex problem. Keeping these separate will also prove to be useful later when looking at what effects the spread of the perturbation can have on the results of a simulation.

Using the Gaussian function in equation (20), the perturbations in velocity and temperature are given by

$$\begin{aligned} \delta v_x &= -\frac{y}{R} \Omega \\ \delta v_y &= +\frac{x}{R} \Omega \\ \delta T &= -\frac{(\gamma - 1)}{2} \Omega^2 \end{aligned} \quad (22)$$

Table 1: Parameters defining the isentropic Euler vortex problem. Quantities for these parameters are given for a few versions of this problem that can be found in literature. All values without units are nondimensional.

	α	M_∞	ρ_∞	p_∞	T_∞	R_{gas}	R	σ	β	L
Shu ²⁰	45°	$\sqrt{\frac{2}{\gamma}}$	1	1	1	–	1	1	$M_\infty \frac{5\sqrt{2}}{4\pi} e^{\frac{1}{2}}$	5
Vincent ²⁴	90°	0.4	1	$\frac{1}{\gamma M_\infty^2}$	1	–	$\frac{3}{2}$	1	$M_\infty \frac{27}{4\pi} e^{\frac{2}{9}}$	20
Hesthaven and Warburton ¹⁴	0°	$\sqrt{\frac{1}{\gamma}}$	1	1	1	–	$\sqrt{\frac{1}{2}}$	1	$M_\infty \frac{5}{2\pi} e^1$	5
High-Order Workshop (Slow) ²⁵	0°	0.05	–	100 kPa	300 K	287.15 $\frac{J}{kg \cdot K}$	0.005	1	$\frac{1}{50}$	0.05
High-Order Workshop (Fast) ²⁵	0°	0.5	–	100 kPa	300 K	287.15 $\frac{J}{kg \cdot K}$	0.005	1	$\frac{1}{5}$	0.05
Stationary Vortex	0°	0	1	1	1	–	1	1	$\frac{5}{2\pi\sqrt{\gamma}} e^{\frac{1}{2}}$	10

These perturbations, along with the isentropic relations between density, pressure, and temperature, are used to define the initial flow conditions of the primitive variables as

$$\begin{aligned}
 \tilde{\rho}_0 &= (1 + \delta T)^{\frac{1}{\gamma-1}} \\
 \tilde{v}_{x,0} &= M_\infty \cos \alpha + \delta v_x \\
 \tilde{v}_{y,0} &= M_\infty \sin \alpha + \delta v_y \\
 \tilde{p}_0 &= \frac{1}{\gamma} (1 + \delta T)^{\frac{\gamma}{\gamma-1}}
 \end{aligned} \tag{23}$$

Above, M represents the Mach number with the subscript ∞ indicating a mean flow quantity, the parameter α is the angle of attack for the mean flow which determines the direction of propagation for the vortex, and the subscript 0 denotes a quantity given at $t = 0$. The tilde accents on the primitive variables indicate that each is a nondimensional quantity where ρ_∞ , a_∞ , and T_∞ have been used as the characteristic density, velocity, and temperature scales, respectively.

To further simplify the problem, an assumption has been made in equations (21) and (22) that the center of the vortex lies at the coordinate origin. This assumption will continue for the remainder of this work where the computational domain is defined to be a square that is centered about the origin. Some versions of this problem use a grid domain where the coordinate origin is located on one of the grid boundaries invalidating this assumption. A simple fix can be found by simply replacing the coordinate variables x and y with \bar{x} and \bar{y} where

$$\begin{aligned}
 \bar{x} &= x - x_c \\
 \bar{y} &= y - y_c
 \end{aligned} \tag{24}$$

and the coordinate pair (x_c, y_c) gives the location of the vortex center.

Nine total parameters must be provided in order to use the above definition for the vortex problem. Five of these parameters are standard requirements to define any general inviscid flow: α , M_∞ , and any three of ρ_∞ , p_∞ , T_∞ , and R_{gas} . Three of the remaining four parameters required, R , σ , and β , control the shape of the Gaussian function that is perturbing the mean flow. Finally, the parameter L is equal to half the length of the computational domain and will be discussed further in the following sections. Given the proper set of these parameters to use with equations (20) to (23), many of the variations to the isentropic Euler vortex problem can be recovered. A few examples of these variations are shown in table 1 which illustrates the range in differences between some of the formulations.

IV.A.2. Computational Domain

The last remaining item needed in order to complete the problem is the definition of the computational domain. Two items pertaining to the computational domain have been mentioned thus far: the computational domain is square and the center of the vortex is located at the coordinate origin. This implies that the boundaries of the grid are located at plus/minus some length L in both the x and y directions. A general guideline for determining the value for L will be discussed in section IV.B.1.

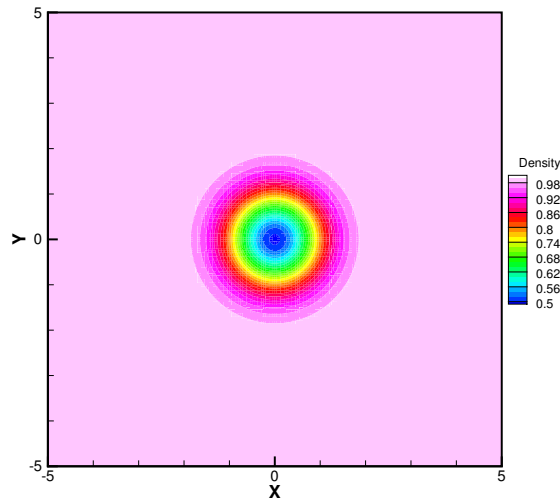


Figure 1: Contours of the initial density profile for the Shu version of the vortex problem.

Finally, we need to define the boundary conditions to completely define the computational domain. The boundary conditions used vary greatly between the different versions of this problem. Examples include but are not limited to: periodic in all directions, periodic in one direction and imposed free-stream conditions in the other direction, imposed analytical solution on the boundaries, and characteristic boundary conditions. Periodic boundary conditions are by far the most popular and are required in order to model the vortex for any significant amount of time. Analytical boundary conditions will not be explored in this work because they prevent a true measure of the error generated by a numerical method. Each time the vortex passes across a grid boundary, the vortex profile is essentially reset to the initial conditions removing all accumulated error.

To illustrate the end product of this problem definition, a contour plot showing initial density profile for the Shu vortex is found in figure 1. The original version of this problem used a computation domain of $L = 5$ and all boundaries were set to periodic conditions. Each simulation was run for a total of 10 time periods, where a period refers to the amount of time it takes for the center of the vortex to propagate completely through the computational domain and return back to its initial location.

IV.B. Propagating Vortex

IV.B.1. Verification of Numerical Accuracy

In literature, the primary use of this problem is to verify the numerical accuracy of a method using grid refinement (h -refinement) and this is where we will begin as well. This problem is ideal for evaluating the numerical accuracy of a method because the exact solution to the problem is just the translation of the initial conditions based on propagation speed and simulation time. Therefore, at any given time, we can easily compute the error between the simulation result and the translated initial conditions.

We started by running the Shu vortex problem on Cartesian grids containing cell counts of 20^2 , 40^2 , 60^2 , 80^2 , and 100^2 . For each grid that was used, we ran simulations for solution polynomials of orders one ($\mathcal{P}1$) through six ($\mathcal{P}6$) amounting to a total of 30 individual simulations. Each simulation was run through one period and a log-log plot of the L^2 error versus length scale was created. The length scale, h , is defined as

$$h = \frac{1}{\sqrt{N_{\text{DOF}}}} \quad (25)$$

where N_{DOF} is the total number of solution points (degrees-of-freedom) used for a simulation. Figure 2 shows two of these plots; the left plot shows the density error versus length scale whereas the right plot shows the x -velocity error versus length scale.

The density error converges as expected except for the $\mathcal{P}6$ solution on the finer grid levels. However, something is clearly limiting the lower bound of the velocity error for the $\mathcal{P}3$ – $\mathcal{P}6$ solutions. After further testing, it was determined that the exponential function within the velocity perturbations (equations (20)

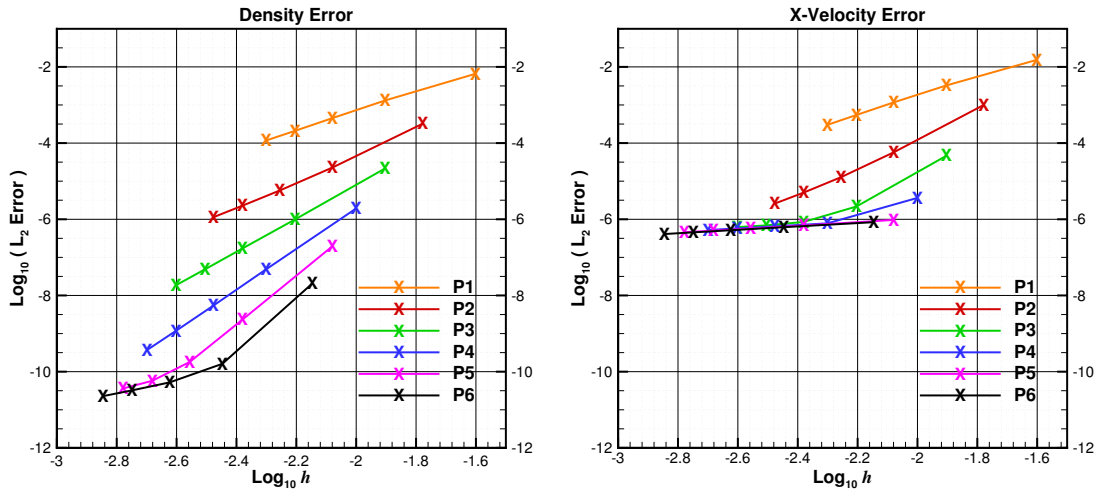
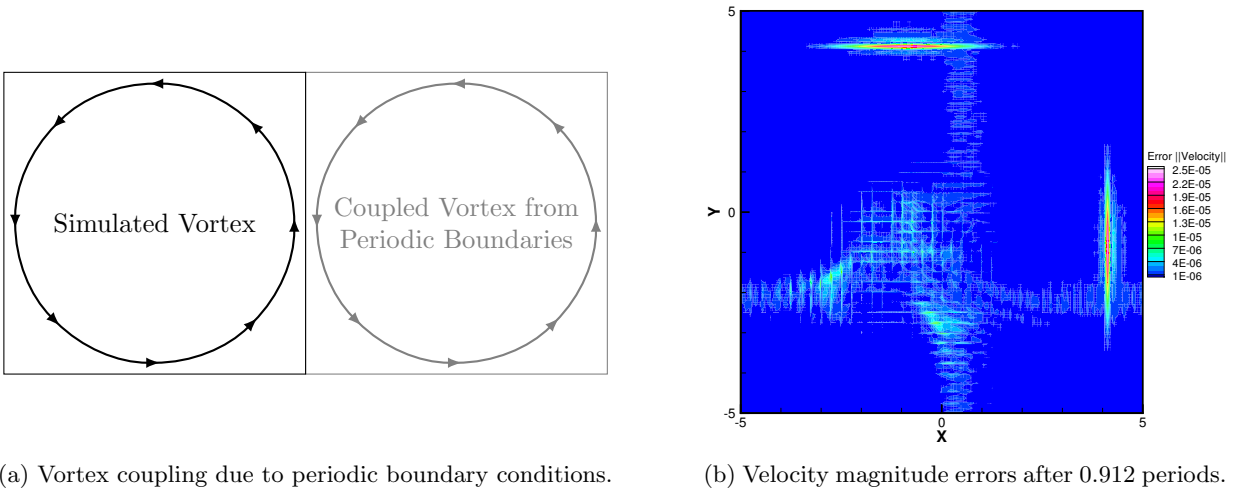


Figure 2: L^2 errors for the Shu vortex problem after one period using a domain of size $[-5, 5] \times [-5, 5]$.



(a) Vortex coupling due to periodic boundary conditions.

(b) Velocity magnitude errors after 0.912 periods.

Figure 3: Errors due to the artificial shear layers that are created because the grid domain is not large enough for the perturbations to reach zero at the domain boundaries.

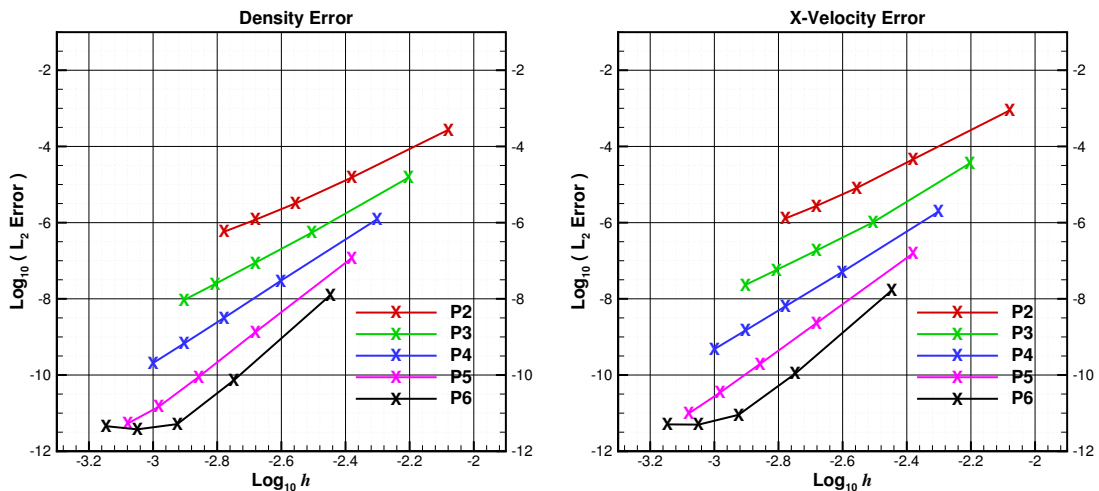


Figure 4: L^2 errors for the Shu vortex problem after one period using a domain of size $[-10, 10] \times [-10, 10]$.

to (22)) is not zero at the domain boundaries. This fact combined with the periodic boundaries creates two very small shear layers, one in each coordinate direction. An illustration of how these shear layers are artificially created by the periodic boundaries is shown in figure 3. The errors associated with these shear layers are orders of magnitude larger than the error from the numerical method resulting in the error limit seen in the right plot of figure 2.

The fix to this problem is to make sure that the velocity perturbations are essentially zero at the grid boundaries. From the problem definition given in section IV.A.1, the three parameters L , R , and σ must satisfy the relation

$$\varepsilon \geq \frac{L}{R} \exp \left[\frac{-L^2}{2R^2\sigma^2} \right] \quad (26)$$

where ε is some small number very near zero. Since all of our computations are performed using double precision (64-bit) real numbers, we chose ε to be 64-bit machine epsilon, $\varepsilon_{64} = 2^{-52}$; this is the smallest positive number capable of being represented by a 64-bit real number. Using ε_{64} and keeping the values for R and σ constant, solving equation (26) for L indicates that L must be greater than 8.8. If machine epsilon for quadruple precision is used, this increases even further to 12.7

In order to eliminate the artificial shear layers, L was increased from 5 to 10 and the previous simulations were redone. Since the length of the domain in each direction was doubled, we also doubled the number of cells in each direction to maintain the same grid cell sizes. This resulted in the cell counts for the five grid levels becoming 40^2 , 80^2 , 120^2 , 160^2 , and 200^2 . The solution errors after one period using the new grids are shown in figure 4. This shows that the size of the grid domain needs to be increased to perform an accurate h -refinement test using this problem as originally defined in Ref. 20.

IV.B.2. Long Term Stability

The motivation for this section originated from trying to replicate the results found in Ref. 23 regarding the stability of various nodal quadratures within triangular elements. Their work used a $\mathcal{P}4$ solution polynomial and the same problem definition as Shu except that they increased the size of their computational domain to $L = 10$. The grid used for their study was an unstructured grid with 800 triangular cells formed by diagonalizing a regular Cartesian grid with 20^2 cells. They found that using the α -optimized nodal points¹⁴ based on the 1D Legendre-Gauss-Lobatto points produced large aliasing instabilities, eventually causing their simulation to blow up at $t = 360$. A second simulation was performed using a nodal set from Ref. 31 that provides a better quadrature rule within the reference triangle. The resulting “simulation was not afflicted by aliasing instabilities and remained stable until it was stopped at $t = 4000$ ”.²³

To best replicate these results, we made sure that the same problem definition, computational grid, and numerical methods identified within their work were used. We found that using the α -optimized points caused our code to blow up at $t = 359.38$, reproducing the results of their first simulation. Figure 5 shows the density contours at $t = 358$ and illustrates the instabilities within the vortex just before the simulation finally blows up. This figure is similar to figure 12 in Ref. 23.

Instead of using the same quadrature rule for the second simulation, barycentric coordinates were used to map the 1D Legendre-Gauss-Lobatto nodes to the reference triangle. Surprisingly, this very simple mapping was significantly more stable than the α -optimized points for this problem with the simulation eventually blowing up just before $t = 900$ or 45 periods. When examining the results for this simulation, we found an odd trend in that the error seemed to increase exponentially through approximately 15 periods before slowing to a linear accumulation for the remainder of the simulation. These trends are shown along with the error at each period in figure 6.

These trends in the error accumulation contradict all previous experience with numerical methods where the error accumulates linearly before it starts to diverge at an exponential rate. In trying to understand this phenomenon, we switched to the use of quadrilateral cells over triangles. This eliminated any uncertainty in the stability of the quadrature rules for triangles replacing it with proven quadrature rules based on a tensor-product formulation of the standard 1D Legendre-Gauss or Legendre-Gauss-Lobatto quadratures. Another concern was that this grid was too coarse for the version of the problem being used because the majority of the vortex perturbations were contained within only a 6^2 section of grid cells. We therefore chose the 40^2 , 80^2 , and 160^2 sized grids with $\mathcal{P}2$ – $\mathcal{P}4$ solutions to see what effects h/p -refinement has on the long term stability for this problem. The log of the L^2 error in density versus the number of periods transversed is shown for each of these cases in figure 7.

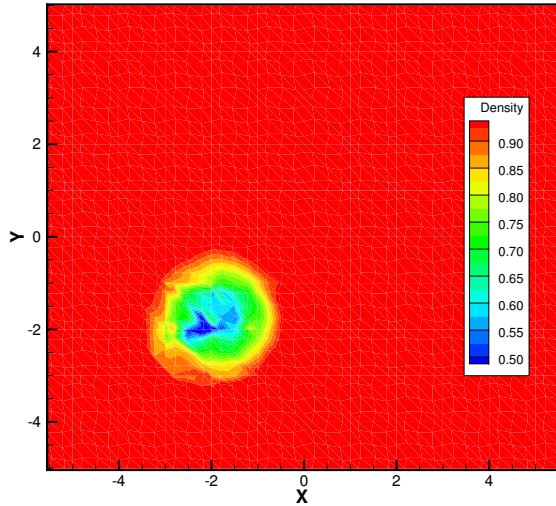


Figure 5: Contours of density at $t = 358$ using the α -optimized nodal points. This figure can be used to compare the presented results to those found in Ref. 23 since this simulation attempted to verify our code by replicating their results.

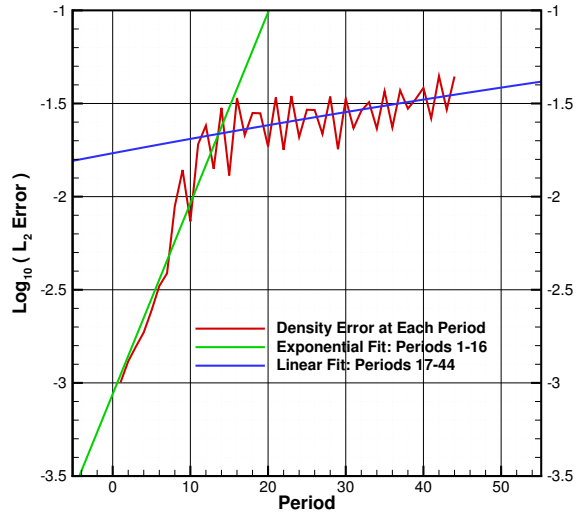


Figure 6: L^2 norm of density error at each simulation period with overlaying trend lines. The green trend line identifies the initial exponential accumulation of error whereas the blue trend line shows the subsequent linear accumulation before the simulation eventually blows up.

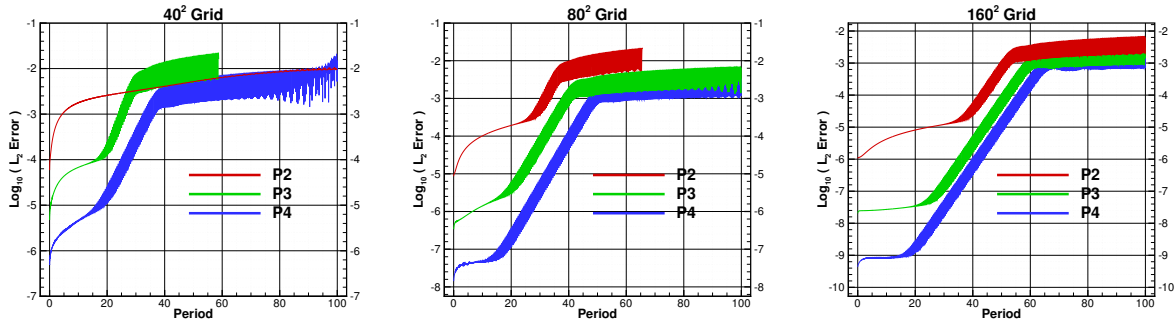


Figure 7: L^2 norm of errors for the Shu vortex problem through 100 periods.

Each of these plots can be broken down into three distinct regions. The first region, beginning at $t = 0$, sees an exponential increase in error and generally lasts for the first three to five periods. When the solution is initialized, the analytical solution is evaluated at each solution point within the grid to provide the exact initial conditions. This initialization is essentially a collocated projection of the analytical solution onto the solution polynomial space. This exponential increase in error over the first few periods is caused by the error associated with projecting the initial conditions.

After a few periods, the second region begins with the error leveling out and it then starts to increase at a linear rate. This region represents the error accumulation of the numerical method once the initial collocation projection has relaxed to its best approximation of the exact solution in the solution polynomial space. Ideally, we want whatever method we are using to sustain this linear accumulation of error indefinitely.

Unfortunately, all of these simulations contain a third region that begins after approximately 20 periods and consists of another exponential accumulation of error. This time the error accumulates until it starts destabilizing the vortex along with the simulation itself. What before seemed to be a contradiction of past experience with numerical methods was actually what we should be expecting with a diverging solution. However, the error levels off again once it reaches the range 0.01–0.001, almost as if the simulation has re-stabilized itself. Clearly there is some kind of instability present within all of these simulations that is causing them to diverge.

It is nearly impossible to understand what is causing these instabilities just by looking at how the total

error within the system evolves over time. We can achieve a greater understanding of this phenomenon by looking at how the solution itself is evolving. Figure 8 shows snap shots of the density contours at different points in time for a simulation using a $\mathcal{P}3$ solution on a grid with 160^2 cells. Below the time lapse is a plot of the L^2 norm of the density error with green lines indicating the times for each of the snap shots. After 20 periods, the error begins to coalesce in the outer regions of the domain which coincides with the start of the third region where the error accumulates at an exponential rate. Now we can see that this accumulation is simply the error continuing to coalesce in the outer regions. After approximately 60 periods, the error outside the vortex appears to reach a point of maximum saturation and the total density error levels off. This error that has accumulated in the outer regions rotates around the vortex slowly deforming it more and more as the simulation continues over the next 90–100 periods. Around 150 periods, the strain on the vortex caused by the surrounding error finally reaches a critical point, and it quickly gets torn apart. This point of “collapse” is generally where a simulation will blow up due to a negative density or pressure. However, sometimes the simulation will continue on for an indefinite amount of time with the remnants of the vortex aimlessly floating around within the computational domain.

We tried various strategies to eliminate this instability such as reducing the time step size, increasing the computational domain to $L = 20$ as well as $L = 40$, and even running the simulation using quadruple (128-bit) precision. We also tried changing the explicit Runge-Kutta method used for time integration, specifically the 5-stage/4th-order SSP³² and 5-stage/4th-order low-storage³³ variations. Furthermore, we also tried changing the upwinding method used to compute the common interface fluxes, specifically trying the HLLC (Harten, Lax, and van Leer Riemann-solver with restored contact surface),³⁴ AUSM+ (improved Advection Upstream Splitting Method),³⁵ and LDFSS (Low-Diffusion Flux-Splitting Scheme)³⁶ upwinding methods.

After all of these attempts failed, we wanted to determine if these instabilities were due to the nonlinearity of the Euler equations. By altering the problem definition so that the velocity perturbations are zero and the pressure is held constant, the momentum and energy equations reduce down to the continuity equation or more simply the linear advection equation. These changes result in the simulation of a density/entropy wave in the form of a Gaussian distribution that is convected at a constant velocity. Using this linear advection problem, we were able to run this problem successfully for 800 periods using a $\mathcal{P}3$ solution, Lobatto points, and a 20^2 grid with $L = 5$. The error in figure 9 shows that the error accumulation is linear for all 800 periods with no signs of it diverging at any time soon afterwards.

These results seemed to indicate that we were experiencing the infamous aliasing instabilities that are inherent to DG methods when applied to nonlinear equations. Common methods for suppressing aliasing error are filtering the higher-order modes of the solution polynomial^{14,37} and increasing the quadrature within each cell which is sometimes referred to as over-integration.^{37–39} Application of an exponential filter¹⁴ to this problem failed at preventing the error from accumulating at an exponential rate regardless of the parameters used for the filter function. A more extensive study using over-integration is found in Ref. 40. In that study, we found that over-integrating was successful in eliminating aliasing errors caused by using an insufficient quadrature to integrate the nonlinear flux functions. However, over-integration was not successful in preventing the vortex problem from diverging for the simulations shown in figure 7. In fact, the simulation shown in figure 8 was over-integrated to $2\mathcal{P}$ yet still succumbed to this instability.

One final sanity check was done by independently verifying our results using the finite-difference code WRLES⁴¹ (Wave Resolving Large-Eddy Simulation). WRLES uses dispersion-relation-preserving (DRP) schemes to solve the Favre filtered Navier-Stokes equations, and is accurate up to 12th-order in space. The solution found by WRLES for this version of the vortex problem was very similar to the FR results and exhibited the same trends in the accumulation of error.

Both codes finding similar results indicates that this nonlinear stability is independent of the numerical method being used and is likely caused by the definition of the problem itself. Most of the versions of this problem use either periodic or analytic boundary conditions in order to allow the vortex to propagate indefinitely. However, Vincent et al. made a subtle new change to the boundary conditions of the problem in Ref. 24 without any reasoning for the change. As others had done before, the free-stream velocity was changed so that the vortex propagated in only one direction, in this case the y -direction only, i.e. $v_{x,\infty} = 0$. The new twist was that the constant x boundaries were set to free-stream conditions instead of leaving them periodic. With this slight change, they were able to successfully perform an h -refinement study using the vortex problem and showed super accuracy through 45 periods without any instabilities.

We made a similar change to our problem by propagating the vortex in only the x -direction, i.e. $v_{y,\infty} = 0$,

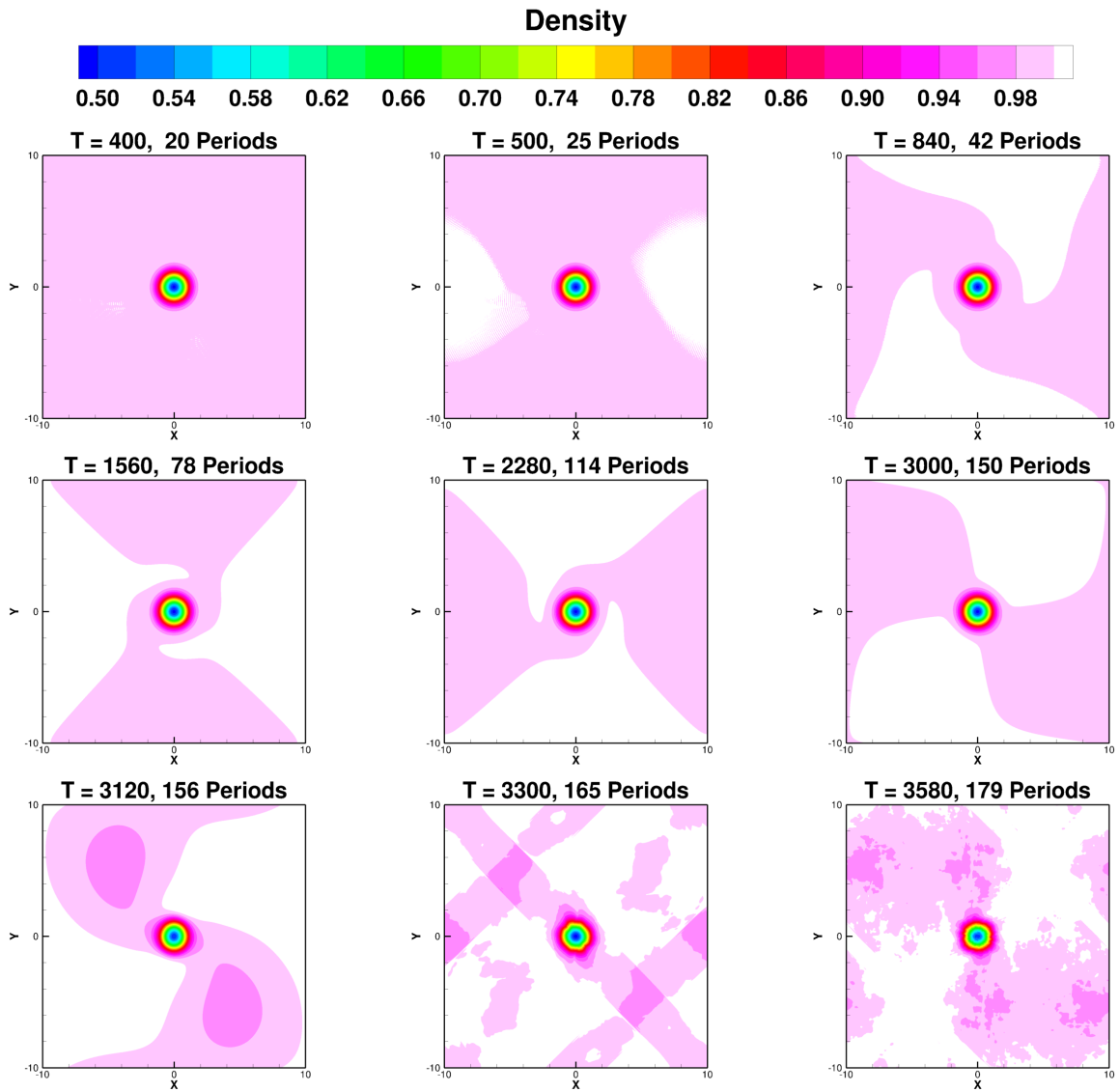
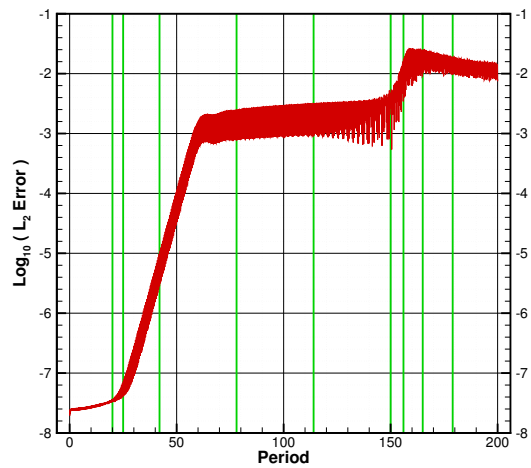


Figure 8: Time series showing how the density solution evolves over time when all domain boundaries are periodic. The plot to the right shows the L^2 norm of the density error versus period. The embedded green lines correspond to the times at which each of the above contour snapshots were taken.



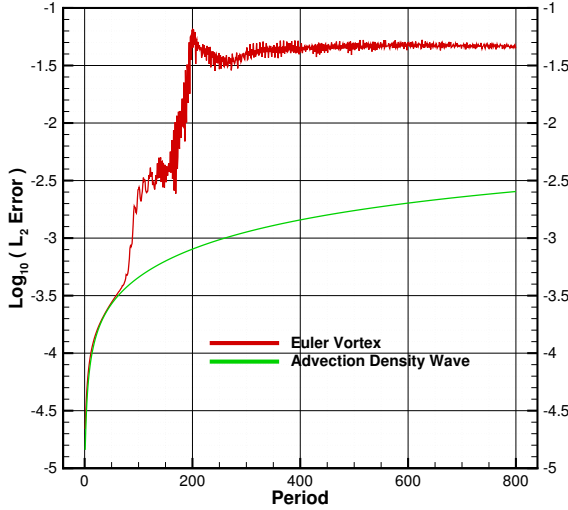


Figure 9: L^2 norm of density error comparing the accumulation of the error over the number of periods between the linear advection equation and the Euler vortex problem.

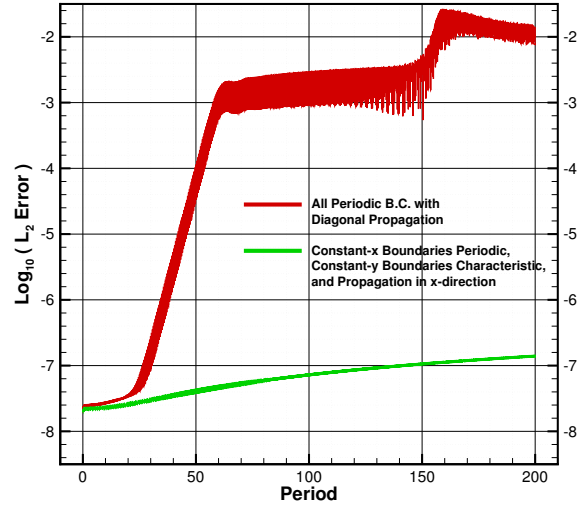


Figure 10: Differences in density errors over 200 periods due to boundary conditions.

and changing the constant y boundaries to characteristic boundary conditions. A comparison between the version using all periodic boundaries and the version using a mixture of periodic and characteristic boundary conditions is shown in figure 10. The simulation using a mixture of boundary conditions remains stable through 200 periods, whereas the simulation with full periodic boundary conditions starts experiencing instabilities after only 20 periods. Clearly we have determined that the nonlinear instability we are encountering is related to the periodicity of the boundary conditions.

IV.C. Stationary Vortex

We have successfully determined that the source of the nonlinear instability we have been encountering is the periodicity of the boundary conditions. A subtle consequence of using periodic boundaries is that the problem we are solving is no longer the exact problem that was intended by the original definition. As we showed in figure 3a, the periodic boundaries actually create a coupled vortex across each of the domain boundaries. So instead of simulating a single, uncoupled, propagating vortex as intended by the original problem definition, we are actually simulating an infinite array of vortices with centers that are separated by a distance of $2 \times L$. Even if the computational domain is made very large, eventually there will be interaction between the coupled vortices due to the nonlinearity of the Euler equations.

To further understand the effects the boundary conditions have on this problem, we will change the problem so the velocity of the mean flow is zero. This creates a stationary vortex in that the center of the vortex never moves from its initial location throughout the simulation. This removes all dependence on periodic boundaries since the vortex is no longer required to propagate across the domain boundary. This is also the only way to simulate a single, decoupled vortex for a large length of time without having to use moving/deforming mesh methods or an excessively large domain.

Because the vortex is no longer propagating, a small complication arises in that we no longer have a definition for the amount of time in one period. In order to keep comparisons simple, we will define one period for the stationary vortex as the same amount of nondimensional time required for the propagating vortex to complete one period.

In the following two subsections, we present results for this stationary version of the problem using two different sets of boundary conditions. The first case is the same as the original propagating version with all periodic boundaries and the second case sets all boundaries to characteristic boundary conditions.

IV.C.1. Periodic Boundary Conditions

The first test case presented for the periodic stationary vortex is a $\mathcal{P}3$ solution on a grid containing 160^2 cells. Additionally, a $\mathcal{P}6$ quadrature was used to over-integrated the solution in an effort to prevent aliasing instabilities. A time lapse showing the density error propagating throughout the computation domain is given in figure 11. An interesting observation is that the density error emanates as waves outward from the center of vortex. For this particular problem, these error waves reach the boundary around $t = 6$. After this point in time, the periodic boundaries cause the waves to be recirculated back into the domain from the opposite boundary. Because error is continuously being created within the vortex interior, the recirculating error waves begin interacting with each other. These errors start building over time and eventually it reaches a large enough magnitude that causes the vortex to become unstable, resulting in the solution diverging.

A parametric study was performed using $\mathcal{P}3$ – $\mathcal{P}6$ solutions on grids containing 40^2 , 80^2 , and 160^2 cells in order to get a sense of how the periodic stationary vortex is affected by h/p -refinement. The error for each of these cases through 100 periods is shown in figure 12. We see the same results as before with the error showing the same three regions described previously in section IV.B.2. A very noteworthy fact of figure 12 is that the third regions of nearly all cases are now aligned. Specifically, only two of the twelve are not aligned: the $\mathcal{P}3$ solution on the 40^2 grid and the $\mathcal{P}6$ solution on the 160^2 grid. For the $\mathcal{P}3$ – 40^2 solution, the error due to the grid resolution and numerical method is large enough that the vortex becomes unstable before it is able to reach the area where the error accumulates at an exponential rate. The $\mathcal{P}6$ solution has the exact opposite problem. The combination of the high-order polynomial solution combined with the higher fidelity grid results in a larger sensitivity to machine precision type errors. Running the $\mathcal{P}6$ – 160^2 case with decreasing values for the size of the time step created large horizontal shifts in both directions for the start of the third region. Essentially these should both be treated as anomalies and separated from the other aligned cases.

Added to figure 12 is a diagonal cyan line that was created from an exponential fit of the data in the third region. The equation defining this line is

$$y = (2.22 \times 10^{-16}) e^{0.3x} \quad (27)$$

The remarkable part of this curve fit is the value of the coefficient in front of the exponential term, which is the same as machine epsilon for a 64-bit floating point number. This seems to indicate that this exponential error accumulation is due to machine rounding errors related to the floating point precision. It appears that this machine error accumulates behind the scenes, masked by the discretization error of the problem.

IV.C.2. Characteristic Boundary Conditions

Because the vortex is no longer propagating, there is no longer a dependence on the domain boundaries being periodic. This allows us to set all of the boundaries to characteristic boundary conditions allowing for information to flow in/out of the domain as needed. The initial test case used in section IV.C.1 that produced figure 11 was performed a second time with the only change being that all boundary conditions are now characteristic boundaries. An identical time lapse to that in figure 11 but with all characteristic boundary conditions is shown in figure 13. Here, the waves of error that emanate from the vortex leave the domain and do not reenter and interact.

Now that we have removed this instability, we can perform an h -refinement study using the vortex problem over a large number of periods. To complete this study, we used $\mathcal{P}2$ – $\mathcal{P}4$ order solutions on grids containing 40^2 , 80^2 , 120^2 , 160^2 , and 200^2 cells. The density error versus period time for all but one of these cases is shown in figure 14. Unfortunately, we were unable to prevent the $\mathcal{P}2$ simulation on the 40^2 grid from diverging due to insufficient resolution, thus those results have been omitted.

In order to see how the order of accuracy develops over time, a linear fit was computed using the density error from each of the grid levels for each point in time. The results of this linear fit using all of the simulations in figure 14 gives what is shown in figure 15a. If we examine the order after a single period, the order of accuracy appears to be $\mathcal{P} + 1$ as expected with values of 3.3, 4.8, and 5.5 for the $\mathcal{P}2$, $\mathcal{P}3$, and $\mathcal{P}4$ solutions, respectively. However, if we were to do the same thing after five periods, the respective values change to 4.0, 5.8, and 5.5 indicating that the $\mathcal{P}3$ solution is of higher order than the $\mathcal{P}4$ solution. Jumping all the way to 100 periods, it seems that the $\mathcal{P}2$ solution is 6th order accurate and the $\mathcal{P}3$ solution is nearly 8th order accurate.

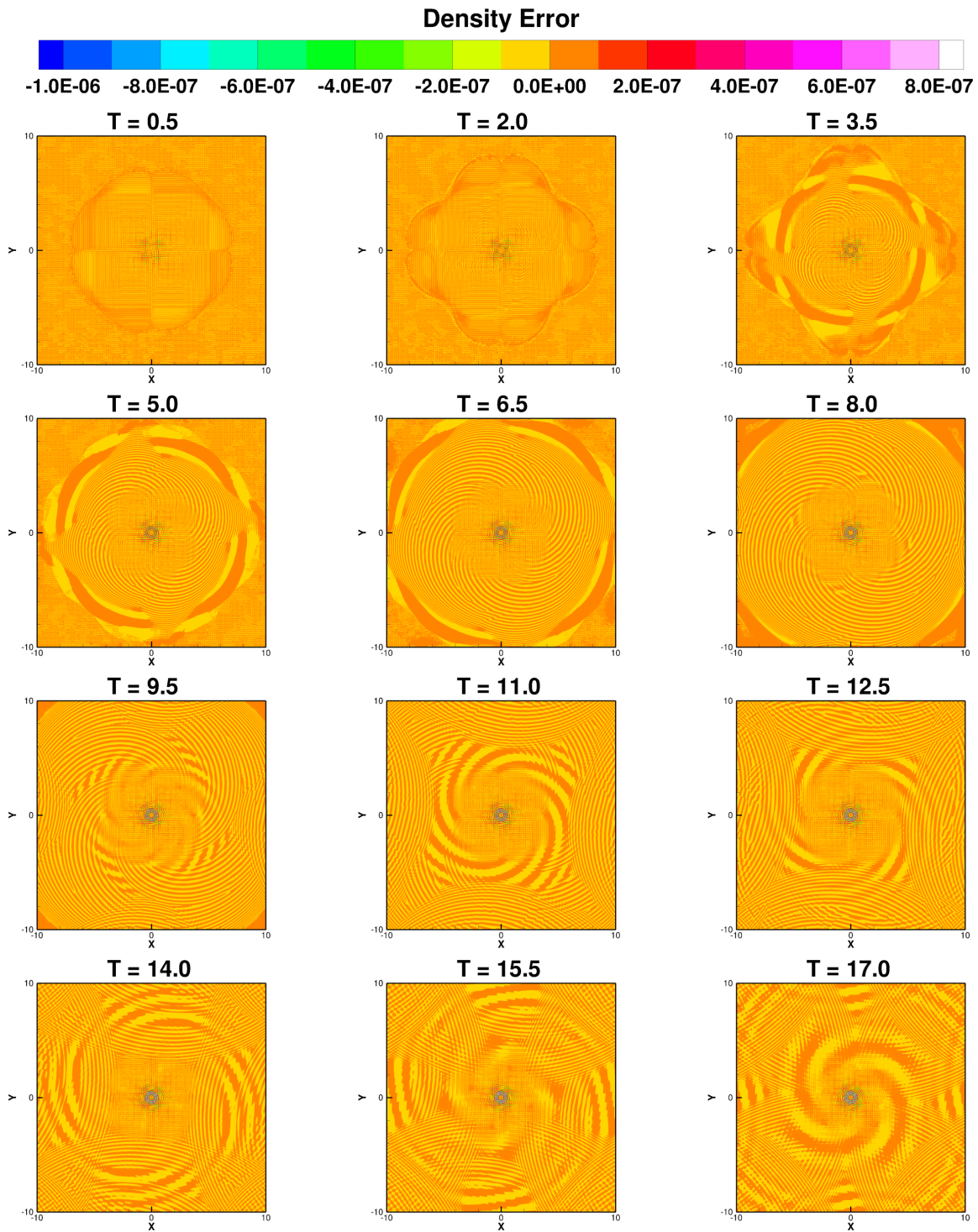


Figure 11: Time series showing how the density error recirculates through the domain because of the periodic boundary conditions.

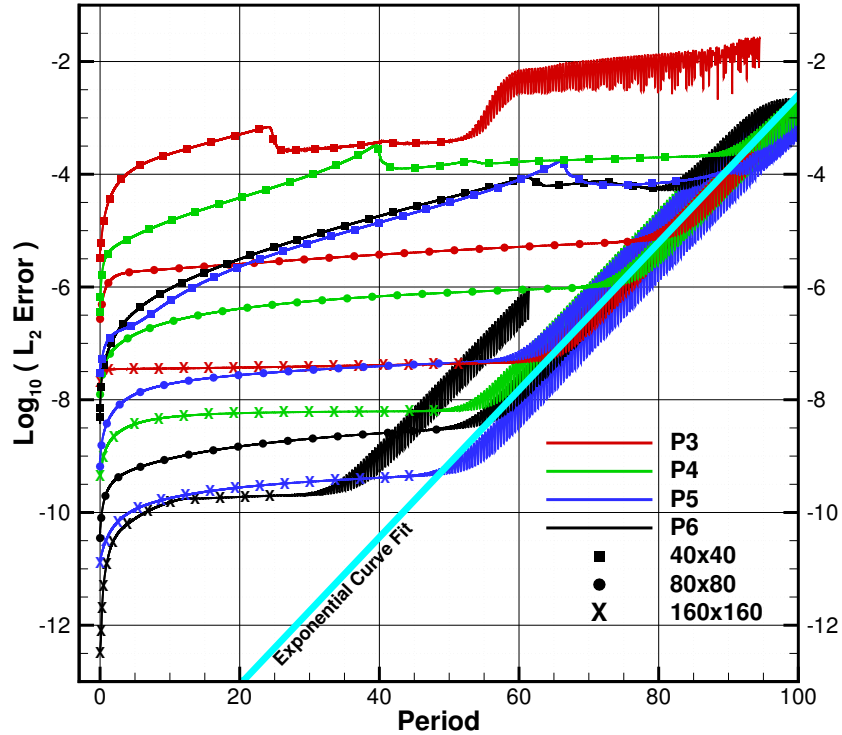


Figure 12: Density errors for the stationary vortex with all periodic boundaries.

The issue is that some of the lower resolution simulations used for this h -refinement study suffered from “non-fatal” instabilities. This simply means an instability of some kind seems to be present without causing the simulation to blow up. Looking at figure 14, non-fatal instabilities can be found in the following simulations: $\mathcal{P}2$ - 80^2 , $\mathcal{P}3$ - 40^2 , and $\mathcal{P}4$ - 40^2 . These instabilities cause the linear fit computation to report an incorrect order of accuracy.

The point of presenting these results is to make the reader aware that care needs to be taken when performing an h -refinement study using the vortex problem. In most literary articles that use the isentropic Euler vortex problem for this purpose^{9, 14, 21–23, 25–27, 29, 30} is computed using the solution error at some arbitrary point in time on a series of grid resolutions. Here we have shown that this can be extremely error prone with us incorrectly showing a 3rd order method with 8th order accuracy. Our incorrect results showed the method in a positive light but this issue could very easily present a negative result with an order of accuracy that is less than expected. A common source of negative results is when a simulation is so overresolved that it encounters machine precision limits. This reasoning is exactly why the present study was capped at $\mathcal{P}4$ solutions due to $\mathcal{P}5$ and $\mathcal{P}6$ simulations incorrectly reporting poor performance.

By prescreening the results of all our simulations, we can selectively remove from our h -refinement study those cases that show any indication there was either too little or too much grid resolution. Since one or more simulations on the 40^2 and 80^2 showed signs of instabilities, we removed all the simulations on those grids from our h -refinement study leaving only the simulations run on the 120^2 , 160^2 , and 200^2 grids. This was done to keep the grid levels identical for all polynomial orders. Recomputing the linear fits with this selectively chosen set of simulations produces the order of accuracy versus time plot shown in figure 15b.

These results are much cleaner than when we blindly used all of our simulations and closer to what is expected. Initially, the accuracy for each polynomial order is $\mathcal{P} + 1$ which is a result commonly found for DG type methods used with nonlinear equations. As time continues, the accuracy increases for all polynomial orders reaching an approximate accuracy of $\mathcal{P} + 3$ through 100 periods. This clearly verifies that the FR method achieves some degree of super accuracy, that is accuracy greater than $\mathcal{P} + 1$, for this problem.

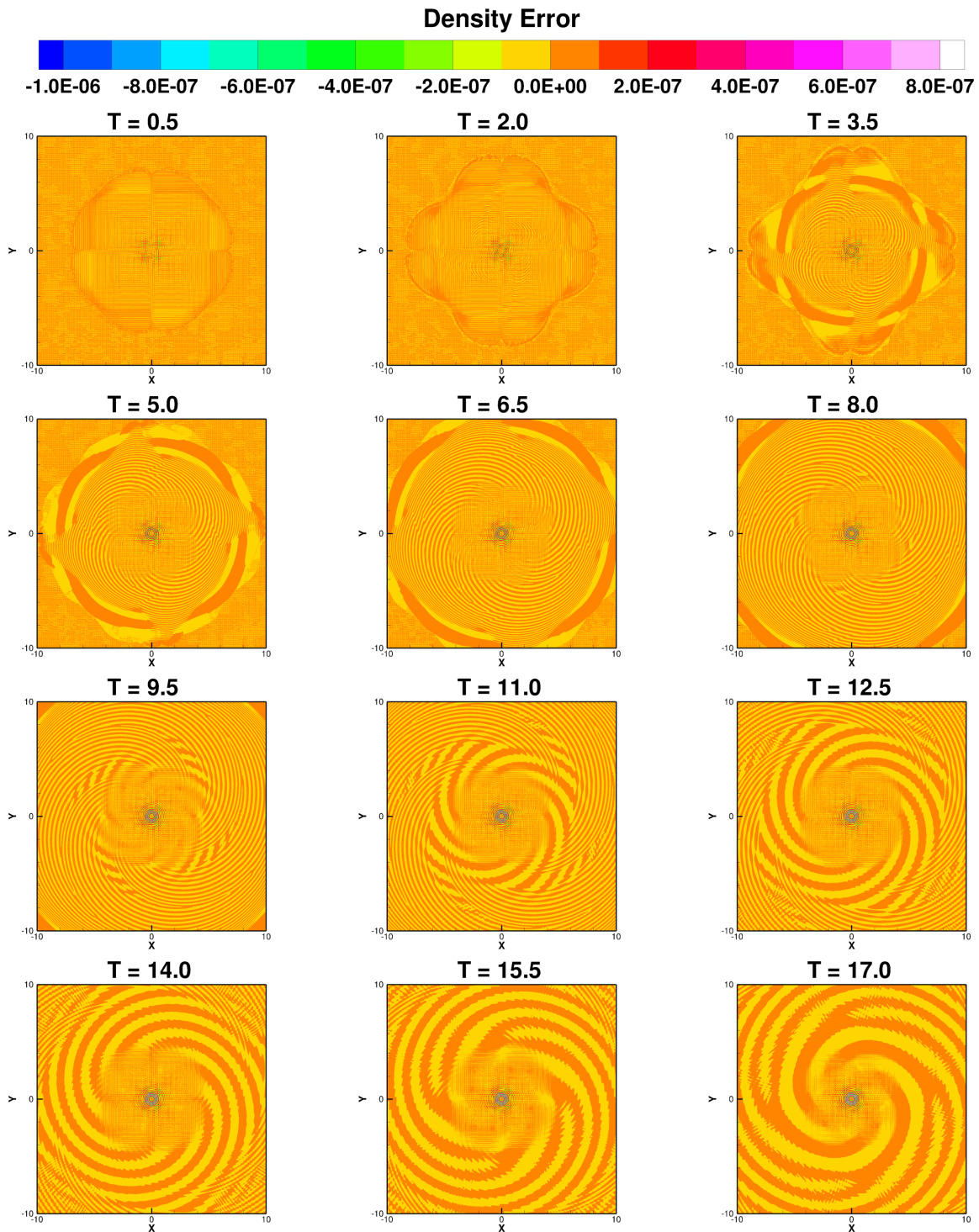


Figure 13: Time series showing how the density error emanates from the interior of the vortex. All boundaries use characteristic boundary conditions to prevent errors from accumulating within the computational domain.

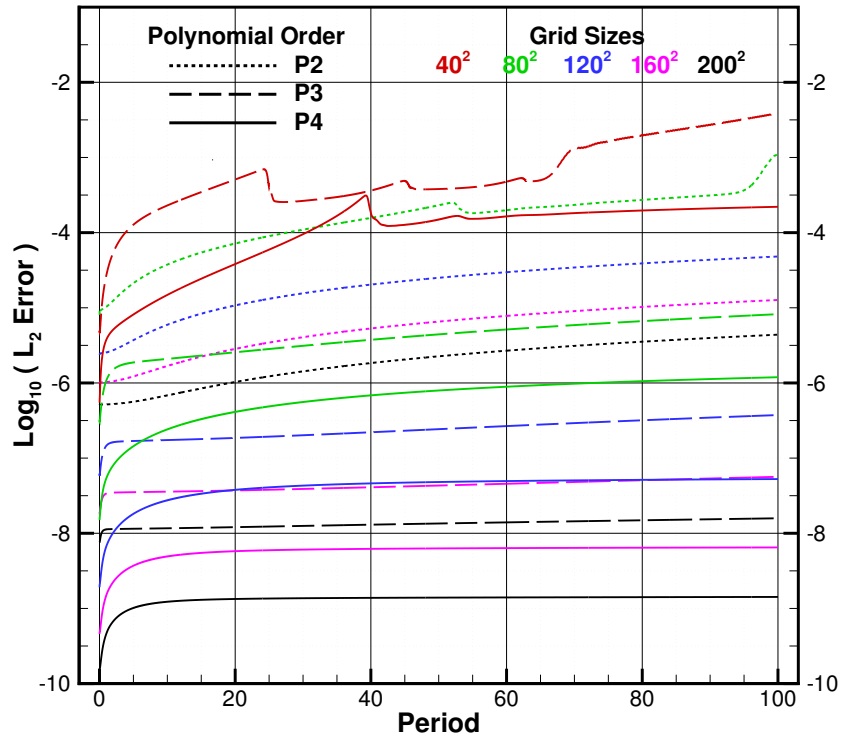
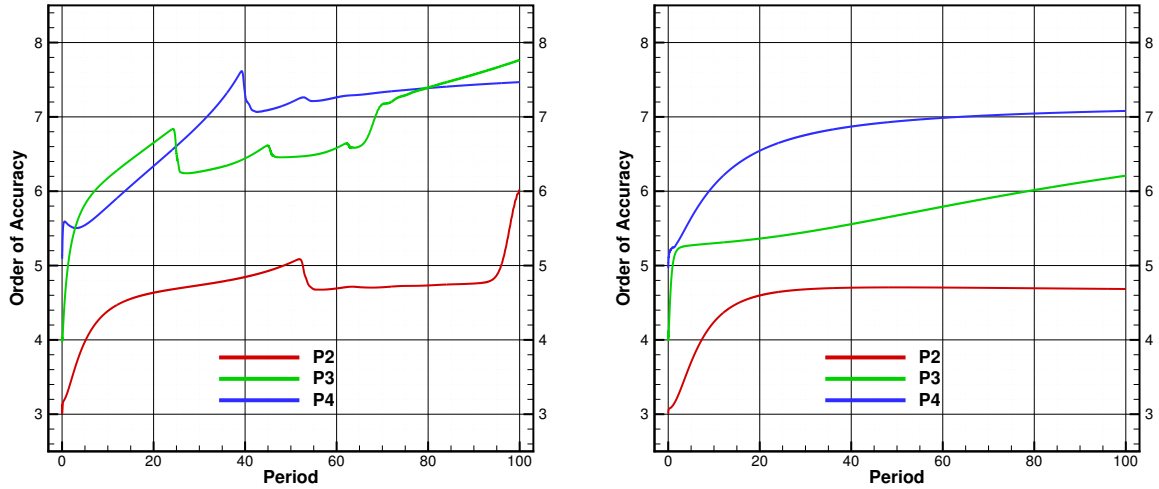


Figure 14: Density error versus period for all the stationary vortex simulations used to perform an h -refinement study with the FR method.

IV.D. Analysis and Recommendations

Based on our experiences with the isentropic Euler vortex problem, the following is a list of general guidelines and recommendations to keep in mind with its use.

1. If using periodic boundary conditions, make sure that the size of the computational domain is large enough so that the velocity perturbations are zero at the boundaries. Otherwise, the opposing tangential velocities across the periodic boundary create an artificial shear layer that does not exist in the exact analytical solution. The error associated with this shear layer masks the error of the discretization producing an incorrect total solution error that is larger than expected.
2. When using the vortex problem to perform an h -refinement study, make sure that all of the simulations used in the study are not under- or overresolved. Both of these situations can cause the computed order of accuracy to be incorrect.
3. Because the error can change rapidly between multiple simulations, it is also recommended to perform an h -refinement analysis at several points in time to make sure that the computed order of accuracy is itself accurate. An increased sampling of the accuracy is the only way to ensure that the results are correct.
4. If propagating the vortex for more than a single period, the numerical error must have a means of escaping the computational domain in order to get the best results. This is especially critical when using high-order methods that produce minimal numerical dissipation. When the vortex is propagating, it takes less than one period for error to recirculate completely and impinge on the vortex from the opposite direction. We found that setting the boundaries in one direction to some sort of outflow boundary condition is sufficient to prevent error from accumulating.
5. If any boundaries are periodic, the simulation is not of a single vortex but of an infinite array of coupled vortices that interact due to the nonlinearity of the Euler equations. The only way to truly decouple the vortex is to keep the vortex stationary by changing the mean flow velocity to zero and using an inflow/outflow condition on all domain boundaries. These changes require extra care that sufficient grid resolution is used due to additional numerical instabilities related to low-speed flows. Provided this extra care is taken, the stationary, decoupled vortex can provide additional insight into the error



(a) Including simulations with signs of instabilities.

(b) Selectively using only stable simulations.

Figure 15: The development of the order of accuracy over time for the stationary vortex using all characteristic boundaries.

generated by a numerical method.

Upon further examination, the nonlinear instability previously encountered for the isentropic Euler vortex problem seems to be the confluence of several factors. The first factor is that, throughout the entirety of this work, we have been using the correction function that recovers the nodal DG method which is known for producing very little numerical dissipation. The second factor is that there is zero physical dissipation in the governing Euler equations. The third and final factor is that there is no way for energy/information to truly exit the domain if all the boundaries of the computational domain are periodic. Combining all of these factors results in a system with little to no ability to dissipate the error which has no means of escaping. The error that is produced throughout the simulation, whether it is from the truncation error of the numerical method or from something as small as machine roundoff error, will continuously build upon itself until it eventually becomes large enough to affect the system in some capacity.

If we were instead using a 2nd-order method, say a finite-volume MUSCL (Monotonic Upstream-Centered Scheme for Conservation Laws) scheme, this problem would not exist because the numerical method provides sufficient dissipation to suppress the error from accumulating. However, without a very high grid resolution it is unlikely that a 2nd-order finite-volume method will be able to propagate the vortex for very long before the vortex is dissipated completely, leaving only the underlying mean flow to remain.²⁰

As this problem has become increasingly popular for verifying high-order methods, various literary works have reported instabilities very similar to what has been shown in the current work.^{23,27} When encountered within the family of FR methods, these instabilities have generally been explained as being caused by polynomial aliasing.^{23,27} Whereas numerical methods utilizing polynomial basis functions have been known to encounter aliasing instabilities,^{13,14,23,27,37–39} the results in the current work, along with those in Ref. 40, indicate that these instabilities with the vortex problem are more likely related to the problem being ill-posed. This seems to be a case of mistaken identity simply because aliasing errors are not completely understood and present an open problem for these methods.

Finally, we should reevaluate our results within the context of the true objective for this test case. The primary purpose of this canonical problem is to demonstrate the ability of a numerical method to sustain vortical structures inherent to turbulent flows without contributing unwanted numerical dissipation. Because the final goal is simulating turbulent flows, the issue of insufficient dissipation solving the Euler equations is mostly irrelevant due to the physical dissipation within the Navier-Stokes equations. Additionally, the computational domain for nearly all real-world problems contains some form of boundary condition allowing for outflow from the domain. The previous results have clearly demonstrated the primary purpose of this problem with our ability to indefinitely propagate the vortex provided that there is a way for error to leave the system.

V. Conclusions

The isentropic Euler vortex problem is a simple, canonical test case commonly used to test the accuracy of an inviscid CFD code. Throughout the many variations of this problem that exist in literature, several assumptions that are commonly made in the problem definition have been found to cause numerical instabilities, especially when using high-order methods. A list of general guidelines and recommendations has been provided to help prevent these problems for the unknowing researcher looking add this problem to their CFD toolbox.

After following these recommendations ourselves, we used the isentropic Euler vortex problem to test the accuracy of our high-order code that is based on the FR method. We were able to successfully verify its accuracy and found $\mathcal{P} + 1$ order accuracy through 1 period and super-accuracy, i.e., greater than $\mathcal{P} + 1$, through 100 periods. By solving this problem using the FR method, we have shown that vortical flow structures can be accurately predicted with minimal unwanted numerical dissipation introduced into the system.

References

- ¹DeBonis, J., “Progress Towards Large-Eddy Simulations for Prediction of Realistic Nozzle Systems,” *AIAA Journal of Propulsion and Power*, Vol. 23, No. 5, 2007, pp. 971–980.
- ²Forsythe, J., Squires, K., Wurtzler, K., and Spalart, P., “Detached-Eddy Simulation of Fighter Aircraft at High-Alpha,” *Journal of Aircraft*, Vol. 41, No. 2, 2004, pp. 193–200.
- ³Mavriplis, D., Pelaez, J., and Kandil, O., “Large Eddy and Detached Eddy Simulations Using an Unstructured Multigrid Solver,” *DNS/LES - Progress and Challenges. Proceedings of the Third AFOSR International Conference on DNS/LES*, Columbus, OH, 2001.
- ⁴Moreau, S., Christophe, J., and Roger, M., “LES of the Trailing-Edge Flow and Noise of a NACA0012 Airfoil Near Stall,” *Proceedings of the Summer Program 2008, Center for Turbulence Research*, Stanford University/NASA Ames, 2008.
- ⁵Kang, S., Iaccarino, G., Ham, F., and Moin, P., “Prediction of Wall-Pressure Fluctuation in Turbulent Flows with an Immersed Boundary Method,” *Journal of Computational Physics*, Vol. 228, No. 18, 2009, pp. 6753–6772.
- ⁶Huynh, H., “A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods,” AIAA Paper 2007-4079, Jun 2007.
- ⁷Huynh, H., “A Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin for Diffusion,” AIAA Paper 2009-403, Jan 2009.
- ⁸Huynh, H., “High-Order Methods Including Discontinuous Galerkin by Reconstructions on Triangular Meshes,” AIAA Paper 2011-44, Jan 2011.
- ⁹Vermeire, B. C., Cagnone, J.-S., and Nadarajah, S., “ILES Using the Correction Procedure via Reconstruction Scheme,” AIAA Paper 2013-1001, Jan 2013.
- ¹⁰Vermeire, B. C., Nadarajah, S., and Tucker, P. G., “Canonical Test Cases for High-Order Unstructured Implicit Large Eddy Simulation,” AIAA Paper 2014-0935, Jan 2014.
- ¹¹Skarolek, V. and Miyaji, K., “Transitional Flow over a SD7003 Wing Using Flux Reconstruction Scheme,” AIAA Paper 2014-0250, Jan 2014.
- ¹²Haga, T., Tsutsumi, S., Kawai, S., and Takaki, R., “Large-Eddy Simulation of a Supersonic Jet Using High-Order Flux Reconstruction Scheme,” AIAA Paper 2015-0831, Jan 2015.
- ¹³Karniadakis, G. and Sherwin, S., *Spectral/hp Element Methods for Computational Fluid Dynamics*, Oxford University Press, 2013.
- ¹⁴Hesthaven, J. S. and Warburton, T., *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Vol. 54 of *Texts in Applied Mathematics*, Springer New York, 2008.
- ¹⁵Kopriva, D. A., “Metric Identities and the Discontinuous Spectral Element Method on Curvilinear Meshes,” *Journal of Scientific Computing*, Vol. 26, No. 3, 2006, pp. 301–327.
- ¹⁶Kopriva, D. A., *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*, Springer Science & Business Media, 2009.
- ¹⁷Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.
- ¹⁸Huynh, H. T., “Accurate Upwind Methods for the Euler Equations,” *SIAM Journal on Numerical Analysis*, Vol. 32, No. 5, 1995, pp. 1565–1619.
- ¹⁹Gottlieb, S. and Shu, C.-W., “Total Variation Diminishing Runge-Kutta Schemes,” *Mathematics of Computation*, Vol. 67, No. 221, 1998, pp. 73–85.
- ²⁰Shu, C.-W., “Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws,” *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, edited by A. Quarteroni, Vol. 1697 of *Lecture Notes in Mathematics*, Springer Berlin Heidelberg, 1998, pp. 325–432.
- ²¹Wang, Z. J., Liu, Y., May, G., and Jameson, A., “Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations,” *Journal of Scientific Computing*, Vol. 32, No. 1, 2007, pp. 45–71.

- ²²Wang, Z. and Gao, H., “A Unifying Lifting Collocation Penalty Formulation Including the Discontinuous Galerkin, Spectral Volume/Difference Methods for Conservation Laws on Mixed Grids,” *Journal of Computational Physics*, Vol. 228, No. 21, 2009, pp. 8161–8186.
- ²³Castonguay, P., Vincent, P., and Jameson, A., “Application of High-Order Energy Stable Flux Reconstruction Schemes to the Euler Equations,” AIAA Paper 2013-686, Jan 2011.
- ²⁴Vincent, P., Castonguay, P., and Jameson, A., “Insights from von Neumann Analysis of High-Order Flux Reconstruction Schemes,” *Journal of Computational Physics*, Vol. 230, No. 22, 2011, pp. 8134–8154.
- ²⁵Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., “High-order CFD Methods: Current Status and Perspective,” *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845.
- ²⁶Gao, H. and Wang, Z., “A Conservative Correction Procedure via Reconstruction Formulation with the Chain-Rule Divergence Evaluation,” *J. Comput. Physics*, Vol. 232, No. 1, 2013, pp. 7–13.
- ²⁷Williams, D. M. and Jameson, A., “Nodal Points and the Nonlinear Stability of High-Order Methods for Unsteady Flow Problems on Tetrahedral Meshes,” AIAA Paper 2013-2830, Jun 2013.
- ²⁸Witherden, F. D., Farrington, A. M., and Vincent, P. E., “PyFR: An Open Source Framework for Solving Advection-Diffusion Type Problems on Streaming Architectures Using the Flux Reconstruction Approach,” *Computer Physics Communications*, Vol. 185, No. 11, 2014, pp. 3028–3040.
- ²⁹De Grazia, D., Mengaldo, G., Moxey, D., Vincent, P. E., and Sherwin, S. J., “Connections Between the Discontinuous Galerkin Method and High-Order Flux Reconstruction Schemes,” *International Journal for Numerical Methods in Fluids*, Vol. 75, No. 12, 2014, pp. 860–877.
- ³⁰Yu, M., Wang, Z., and Liu, Y., “On the Accuracy and Efficiency of Discontinuous Galerkin, Spectral Difference and Correction Procedure via Reconstruction Methods,” *Journal of Computational Physics*, Vol. 259, 2014, pp. 70–95.
- ³¹Taylor, M. A., Wingate, B. A., and Bos, L. P., “Several New Quadrature Formulas for Polynomial Integration in the Triangle,” *arXiv:math/0501496v2*, 2007.
- ³²Spiteri, R. J. and Ruuth, S. J., “A New Class of Optimal High-Order Strong-Stability-Preserving Time Discretization Methods,” *SIAM Journal on Numerical Analysis*, Vol. 40, No. 2, 2002, pp. 469–491.
- ³³Carpenter, M. H. and Kennedy, C. A., “Fourth-Order 2N-Storage Runge-Kutta Schemes,” NASA TM-1994-109112, 1994.
- ³⁴Toro, E. F., Spruce, M., and Speares, W., “Restoration of the Contact Surface in the HLL-Riemann Solver,” *Shock waves*, Vol. 4, No. 1, 1994, pp. 25–34.
- ³⁵Liou, M.-S., “A Sequel to AUSM: AUSM+,” *Journal of Computational Physics*, Vol. 129, No. 2, 1996, pp. 364–382.
- ³⁶Edwards, J. R., “A Low-Diffusion Flux-Splitting Scheme for Navier-Stokes Calculations,” *Computers & Fluids*, Vol. 26, No. 6, 1997, pp. 635–659.
- ³⁷Gassner, G. J. and Beck, A. D., “On the Accuracy of High-Order Discretizations for Underresolved Turbulence Simulations,” *Theoretical and Computational Fluid Dynamics*, Vol. 27, No. 3-4, 2013, pp. 221–237.
- ³⁸Kirby, R. M. and Karniadakis, G. E., “De-Aliasing on Non-Uniform Grids: Algorithms and Applications,” *Journal of Computational Physics*, Vol. 191, No. 1, 2003, pp. 249–264.
- ³⁹Kirby, R. M. and Sherwin, S. J., “Aliasing Errors Due to Quadratic Nonlinearities on Triangular Spectral/hp Element Discretisations,” *Journal of Engineering Mathematics*, Vol. 56, No. 3, 2006, pp. 273–288.
- ⁴⁰Spiegel, S., Huynh, H. T., and DeBonis, J. R., “De-Aliasing through Over-Integration Applied to the Flux Reconstruction and Discontinuous Galerkin Methods,” AIAA Paper 2015-TBD, Jun 2015.
- ⁴¹DeBonis, J. R., “A High-Resolution Capability for Large-Eddy Simulation of Jet Flows,” AIAA Paper 2010-5023, Jun 2010.