

Mid-Year Report

Discontinuous Galerkin Euler Equation Solver

Friday, December 14, 2012

Andrey Andreyev

Advisor: Dr. James Baeder

Abstract:

The focus of this effort is to produce a two dimensional inviscid, compressible flow solver using the Discontinuous Galerkin Finite Element approach. The Discontinuous Galerkin method seeks to project the exact solution onto a finite polynomial space while allowing for discontinuities at cell interfaces. This allows for the natural discontinuity capture that is required for a compressible flow solver. The appeal of the Discontinuous Galerkin Method is that it handles higher order spatial discretization without the use of larger stencils which is required in Finite Volume implementations.

Initially the goal was to have a two dimensional version of the code implemented (not validated). However a series of setbacks limited the scope of this work to a one dimensional flow solver.

Thus far, a one-dimensional Discontinuous Galerkin Euler Equation solver was implemented using second and third order spatial discretization and a third order Runge-Kutta time-stepping method. A local projection limiter in the characteristic fields is applied to ensure the method is total variation bounded (TVB).^[1,2] The ability of the method to capture discontinuities is tested on a known set of one dimensional problems: Sod's Shock Tube and Osher's Shock-Entropy Wave Interaction problem. It was found that the method captures the discontinuities in Sod's Shock tube fairly well; however the method had problems capturing the highly non-linear behavior in Osher's problem. Correcting this problem is the focus of current efforts.

1. Introduction

The need to solve inviscid, compressible flow governed by the Euler Equations comprises a large set of engineering problems of practical importance. Unfortunately the Euler Equations are a system of non-linear partial differential equations which have few exact solutions. The exact solutions that do exist are based on a series of simplifying assumptions that make the solutions not applicable to engineering problems. The need of robust numerical methods to solve the Euler Equations is of great importance.

Traditionally finite difference and finite volume methods are used in development of compressible flow solvers with finite volume method being the dominant because of its natural conservative properties.^[3] However these methods are difficult to make higher order in space because of the need for a larger stencil size to approximate the gradients numerically. The details of this will be discussed in further detail in the mathematical background section. The appeal of the Discontinuous Galerkin method is that in finite element formulations gradients of the approximate solution are represented as shape functions and thus there is no need to use a larger computational stencil. This has several implications. The first being is that boundary treatment is simpler because there is no need to adjust the computational stencil around domain boundaries. The second implication is that there is less "numerical smearing" around flow discontinuities because the numerical gradients are not calculated based on several values across the discontinuity. This feature is known as locality.^[2]

2. Background

2.0 Mathematical Background

A general form of the Euler Equations are presented in equation (1.1), is presented. The main advantage of the Discontinuous Galerkin Method is its abilities to automatically handle discontinuities in the flow fields and higher order spatial discretization without the use of a larger computational stencil.

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y + \mathbf{H}(\mathbf{U})_z = \mathbf{0}$$

$$\mathbf{U} = [\rho, \rho u, \rho v, \rho w, E]^T \quad \mathbf{F} = [\rho u, \rho u^2 + p, \rho uv, \rho uw, u(E + p)]^T \quad (2.0.1)$$

$$\mathbf{G} = [\rho v, \rho uv, \rho v^2 + p, \rho vw, v(E + p)]^T \quad \mathbf{H} = [\rho w, \rho uw, \rho vw, \rho w^2 + p, w(E + p)]^T$$

The one-dimensional version of the Euler Equations is given in 1.2

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{0}$$

$$\mathbf{U} = [\rho, \rho u, E]^T \quad \mathbf{F} = [\rho u, \rho u^2 + p, u(E + p)]^T \quad (2.0.2)$$

$$\mathbf{U}(x, 0) = \mathbf{U}^0(x)$$

One major characteristic of hyperbolic equations such as the Euler Equations is that it allows for discontinuities in the solution and the ability of numerical methods to handle the discontinuities automatically as part of the solution procedure is a desired property. Figure 1 shows the difference between a conservative and non-conservative method solution in one dimension.[3]

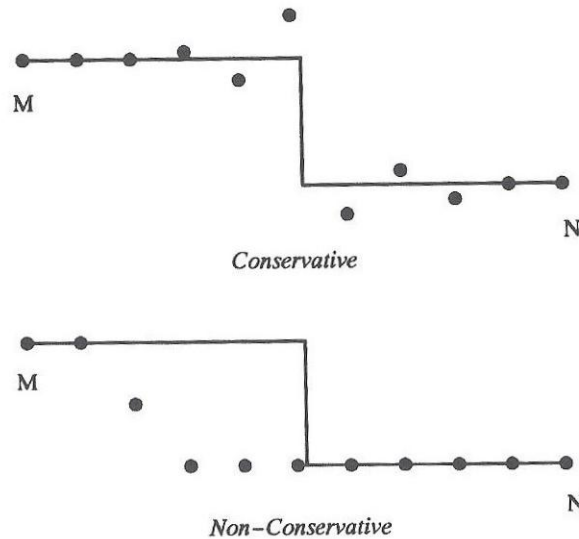


Figure 1 Difference Between conservative and non-conservative methods^[3]

The discontinuities also present a problem when using higher stencils across these discontinuities. Stencils are locations where functions are evaluated. These function evaluations are used to calculate the numerical derivatives present in the Euler Equations. A graphical representation of a numerical stencil is shown in figure 2. ^[3] The stencil presented is for one dimensional problems. The horizontal axis is the spatial variable discretization and the vertical axis is the time.

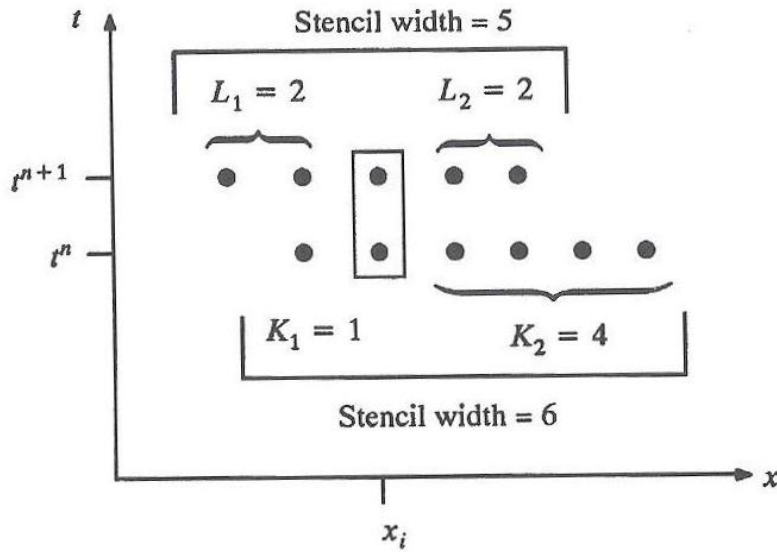


Figure 2 Graphical Representation of a computational stencil

As one can imagine, if there is a discontinuity within the stencil, the result for the derivative will be polluted. One can also see the boundary treatment implications. Imagine a boundary on the right side of the horizontal axis, the derivative approximation near that boundary will be problematic once points right of the boundary are required.

As mentioned earlier finite element implementation avoids the issue of larger stencils by using higher order shape functions to approximate the variation (derivative) of the solution. The reason why finite element methods were not used for Euler Equations solvers is that traditional methods were not conservative because they enforced discontinuities across cell interfaces. It is the Discontinuous Galerkin Method that allows for discontinuities across cell interfaces and thus allows discontinuous solutions.

2.1 Background of Discontinuous Galerkin Method

The development of the method will be shown using the one-dimensional version of the equations. The unknown solution \mathbf{U} is projected onto a finite k^{th} order polynomial function space $\mathbf{V}_h^{[1]}$ and the resulting approximate solution is labeled as $\mathbf{u}^h(x, t) = \sum_{l=1}^k a_l \mathbf{u}_j^{(l)} v_l^{(j)}(x)$. Where $v_l^{(j)}(x)$ are the components of the function space and $\mathbf{u}_j^{(l)}$ are the degrees of freedom of each element. The components of the function space up to third order are ^[2]:

$$v_1^j = 1 \quad v_2^j = x - x_j \quad v_3^j = (x - x_j)^2 - \frac{1}{12} \Delta x_j^2 \quad (2.1.1)$$

The degrees of freedom are given by ^[2]

$$\mathbf{u}_j^{(l)}(t) = \frac{1}{\Delta x^l} \int \mathbf{U}(x, t) v_l^{(j)} dx \quad (2.1.2)$$

$$a_j \frac{\Delta x_j^l}{\int [v_l^{(j)}(x)]^2 dx}$$

Note that in the case of $l = 1$, the first degree of freedom represents the cell average. The second degree of freedom represents the linear variation within the cell and so on ^[2].

To develop the finite element model, a weak formulation is required. To do this, the exact solution is replaced by $\mathbf{u}^h(x, t)$ and multiplied by a weight function. In the case of the Galerkin method, the weight functions are the same as the components of the function space. Applying this procedure and integrating by parts results in ^[1]

$$\frac{d}{dx} \mathbf{u}_j^{(l)}(t) + \Delta_+ \left[v_l^{(j)}(x_{j-1/2}) \mathbf{f}(x_{j-1/2}) \right] - \frac{1}{\Delta x^l} \int \mathbf{f}(\mathbf{u}^h(x, t)) \frac{d}{dx} v_l^{(j)}(x) dx = 0 \quad 2.1.3)$$

Note that the second term in (1.5) involves the flux across the cell interface. It is this flux that allows for the discontinuity in the solution. The choice of the interface flux is somewhat arbitrary and there are quite a lot of choices given in [3]. For this implementation a Roe Flux with entropy fix was used. Roe Flux is given by $f(\mathbf{u}_L, \mathbf{u}_R)$ ^[3] The Roe Flux is an approximation to a Riemann Problem. The Riemann problem will be discussed in more detail later as it will be used as a validation case.

$$\begin{aligned} \rho_{RL} &= \sqrt{\rho_R \rho_L} & u_{RL} &= \frac{\sqrt{\rho_R} u_R + \sqrt{\rho_L} u_L}{\sqrt{\rho_R} + \sqrt{\rho_L}} \\ h_{RL} &= \frac{\sqrt{\rho_R} h_R + \sqrt{\rho_L} h_L}{\sqrt{\rho_R} + \sqrt{\rho_L}} & a_{RL} &= \sqrt{(\gamma - 1) \left(h_{RL} - \frac{1}{2} u_{RL}^2 \right)} \\ \Delta v_1 &= \Delta p - \frac{\Delta p}{a_{RL}^2} & \Delta v_2 &= \Delta u + \frac{\Delta p}{\rho_{RL} a_{RL}} & \Delta v_3 &= \Delta u - \frac{\Delta p}{\rho_{RL} a_{RL}} \\ \mathbf{r}_1 &= \left[1, u_{RL}, \frac{1}{2} u_{RL}^2 \right]^T & \mathbf{r}_2 &= \frac{\rho_{RL}}{2 a_{RL}} \left[1, u_{RL} + a_{RL}, h_{RL} + a_{RL} u_{RL} \right]^T \\ \mathbf{r}_3 &= -\frac{\rho_{RL}}{2 a_{RL}} \left[1, u_{RL} - a_{RL}, h_{RL} - a_{RL} u_{RL} \right]^T \\ \lambda_1 &= u_{RL} & \lambda_2 &= u_{RL} + a_{RL} & \lambda_3 &= u_{RL} - a_{RL} \\ \mathbf{f}(x_{j+1/2}) &= \frac{1}{2} (\mathbf{f}(\mathbf{u}_R) + \mathbf{f}(\mathbf{u}_L)) - \frac{1}{2} \sum \mathbf{r}_i |\lambda_i| \Delta v_i \end{aligned} \quad 2.1.4)$$

The evaluation of $\mathbf{u}_R, \mathbf{u}_L$ at the interface is obtained by evaluating (1.4) at the each interface. After some manipulation

$$\mathbf{u}_L = \mathbf{u}_j^1 + 6\mathbf{u}_j^2 + 30\mathbf{u}_j^3 = \mathbf{u}_j^1 + \tilde{\mathbf{u}}_j \quad \mathbf{u}_R = \mathbf{u}_{j+1}^1 - 6\mathbf{u}_{j+1}^2 + 30\mathbf{u}_{j+1}^3 = \mathbf{u}_{j+1}^1 - \tilde{\mathbf{u}}_{j+1} \quad 1.7$$

where \mathbf{u}_j^l are the degrees of freedom. Note that the above equation is for a third order scheme. To obtain second order scheme, the third term is dropped in each equation.

The spatial discretization is complete. However, a local projection limiter still needs to be applied to stabilize the scheme around discontinuities.^[2] A characteristic limiter is used. The procedure is outlined below.

Compute $\mathbf{u}_{j+1/2} = (\mathbf{u}_j + \mathbf{u}_{j+1})/2$ and calculate the Flux Jacobian $\frac{\partial f}{\partial \mathbf{u}}$ at $\mathbf{u}_{j+1/2}$. Calculate the matrices $\overline{Q}, \overline{Q}^{-1}$ which consist of right eigenvectors as columns and left eigenvectors as rows respectively.^[3]

Calculate $\widetilde{\mathbf{u}}_j, \widetilde{\mathbf{u}}_{j+1}^1, \mathbf{u}_j^1, \mathbf{u}_{j+1}^1, \Delta_- \mathbf{u}_j^1, \Delta_+ \mathbf{u}_j^1, \Delta_+ \mathbf{u}_{j+1}^1$ and multiply each by Q^{-1} to project each of the variables onto the left eigenspace.

$\mathbf{a}^p = Q^{-1} \cdot \mathbf{a}$ where \mathbf{a} represents all of the above variables. The goal is to limit $\widetilde{\mathbf{u}}_j, \widetilde{\mathbf{u}}_{j+1}^1$ in the left eigenspace by applying the minmod limiter. A minmod function

$\text{minmod}(a, b, c)$ returns the lowest value unless there is a sign difference in which case it returns zero.^[1]

$$\widetilde{\mathbf{u}}_j^{\text{mod}(p)} = \text{minmod}(\widetilde{\mathbf{u}}_j^{(p)}, \Delta_+ \mathbf{u}_j^{1(p)}, \Delta_- \mathbf{u}_j^{1(p)})$$

$$\widetilde{\mathbf{u}}_{j+1}^{\text{mod}(p)} = \text{minmod}(\widetilde{\mathbf{u}}_{j+1}^{(p)}, \Delta_+ \mathbf{u}_j^{1(p)}, \Delta_+ \mathbf{u}_{j+1}^{1(p)}) \quad 2.1.5$$

The modified values are now used to project $\mathbf{u}_j^{1(p)}, \mathbf{u}_{j+1}^{1(p)}$ onto the cell interface

$$\mathbf{u}_{j+1/2}^{-(p)} = \mathbf{u}_j^{1(p)} + \widetilde{\mathbf{u}}_j^{\text{mod}(p)} \quad \mathbf{u}_{j+1/2}^{+(p)} = \mathbf{u}_{j+1}^{1(p)} - \widetilde{\mathbf{u}}_{j+1}^{\text{mod}(p)}$$

and project back on to component space by

$$\mathbf{u}_{j+1/2}^\pm = \overline{Q} \mathbf{u}_{j+1/2}^{\pm(p)}$$

This result is used in the Roe Flux function shown earlier.

The only step left is to march in time using the third order Runge-Kutta Technique^[1]

$$\begin{aligned} (\mathbf{u}^h)^{(i)} &= \sum_{l=0}^{i-1} [\alpha_{il} (\mathbf{u}^h)^{(l)} + \beta_{il} \Delta t \mathbf{L}_h((\mathbf{u}^h)^{(l)}, t^n + d_l \Delta t)], \quad i = 1, \dots, r, \\ (\mathbf{u}^h)^{(0)} &= (\mathbf{u}^h)^n, \quad (\mathbf{u}^h)^{(r)} = (\mathbf{u}^h)^{n+1}. \end{aligned} \quad 2.1.6$$

$$\begin{aligned} \alpha_{10} &= \beta_{10} = 1, \quad \alpha_{20} = \frac{3}{4}, \quad \beta_{20} = 0, \\ \alpha_{21} &= \beta_{21} = \frac{1}{4}, \quad \alpha_{30} = \frac{1}{3}, \quad \beta_{30} = \alpha_{31} = \beta_{31} = 0, \\ \alpha_{32} &= \beta_{32} = \frac{2}{3}, \quad d_0 = 0, \quad d_1 = 1, \quad d_2 = \frac{1}{2}; \\ \text{CFL} &= 1, \end{aligned}$$

$$\Delta t = \min \left(\frac{u_j + a_j}{\Delta x_j} \right) \quad 2.1.7)$$

Note that L_h is recalculated and the projection limiter is applied at every sub-time step. This concludes the background of the Discontinuous Galerkin Method.

3. Implementation

This section describes the implementation of the above Discontinuous Galerkin method using pseudo-code. Some definitions are required. The Residual is defined as at each cell

$$\mathbf{Re} = \Delta_+ \left[v_l^{(j)}(x_{j-1/2}) \mathbf{f}(x_{j-1/2}) \right] - \frac{1}{\Delta x_l} \int \mathbf{f}(\mathbf{u}^h(x, t)) \frac{d}{dx} v_l^{(j)}(x) dx \quad 3.1)$$

The first term in the residual is the cell interface flux calculated using 2.1.4 and the second term is the integration performed using Gauss Quadrature. The partial differential equation is now reduced to the following ordinary differential equation at each cell.

$$\frac{d\mathbf{u}_j}{dt} = -\mathbf{Re}_j \quad 3.2)$$

Thus equation 3.2 can be solved the third order Runge-Kutta method given in equation 2.1.6.

The pseudo-code for the implementation is:

```

Calculate the initial degrees of freedom using the initial condition, DOFs are given by equation 2.1.2
time=0
do time=0: final_time
calculate time step dt using equation 2.1.7
  do rk=1:nRungeKuttaSteps (3)
    Apply Flux Limiting
    Calculate the Residual according to equation 3.1
    Update solution to the rkth step
  end do
  time=time+dt
end do

```

4. Results

The code was run using test cases that have known solutions with sharp discontinuities such as a Sod's Shock tube problem. Sod's shock tube problem is a Riemann problem with the following initial condition.

$$x \leq 0 \text{ then } \begin{cases} \rho = 1 \\ u = 0 \\ p = 1 \end{cases} \quad x > 0 \text{ then } \begin{cases} \rho = 0.125 \\ u = 0 \\ p = 0.1 \end{cases} \quad (4.1)$$

The shock tube and the flow details are illustrated in figure 1.

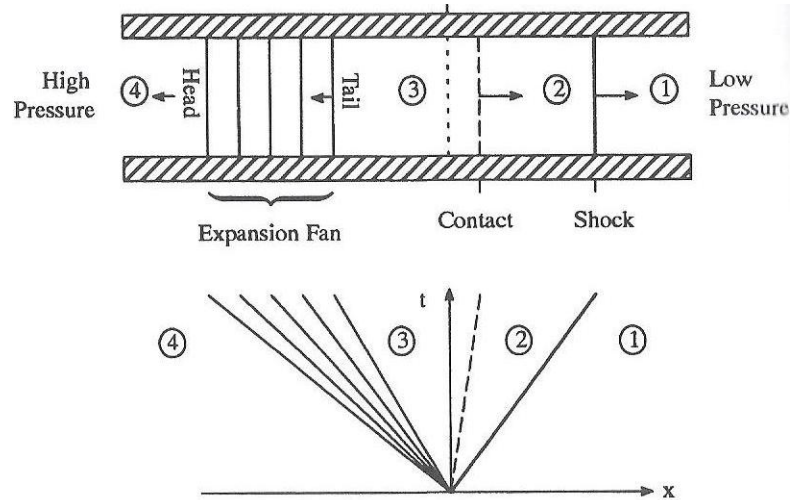


Figure 3. Illustration of a shock tube ^[3]

The shock tube problem's exact solution is given by ^[3]

$$u_2 = u_1 + \frac{a_1}{\gamma} \frac{\frac{p_2}{p_1} - 1}{\sqrt{\frac{\gamma+1}{2\gamma} \left(\frac{p_2}{p_1} - 1\right) + 1}},$$

$$S = u_1 + a_1 \sqrt{\frac{\gamma+1}{2\gamma} \left(\frac{p_2}{p_1} - 1\right) + 1}.$$

(4.2)

$$\frac{p_4}{p_1} = \frac{p_2}{p_1} \left\{ 1 + \frac{\gamma-1}{2a_4} \left[u_4 - u_1 - \frac{a_1}{\gamma} \frac{\frac{p_2}{p_1} - 1}{\sqrt{\frac{\gamma+1}{2\gamma} \left(\frac{p_2}{p_1} - 1\right) + 1}} \right] \right\}^{-2\gamma/(\gamma-1)}$$

Expansion fan

$$u(x, t) = \frac{2}{\gamma+1} \left(\frac{x}{t} + \frac{\gamma-1}{2} u_4 + a_4 \right),$$

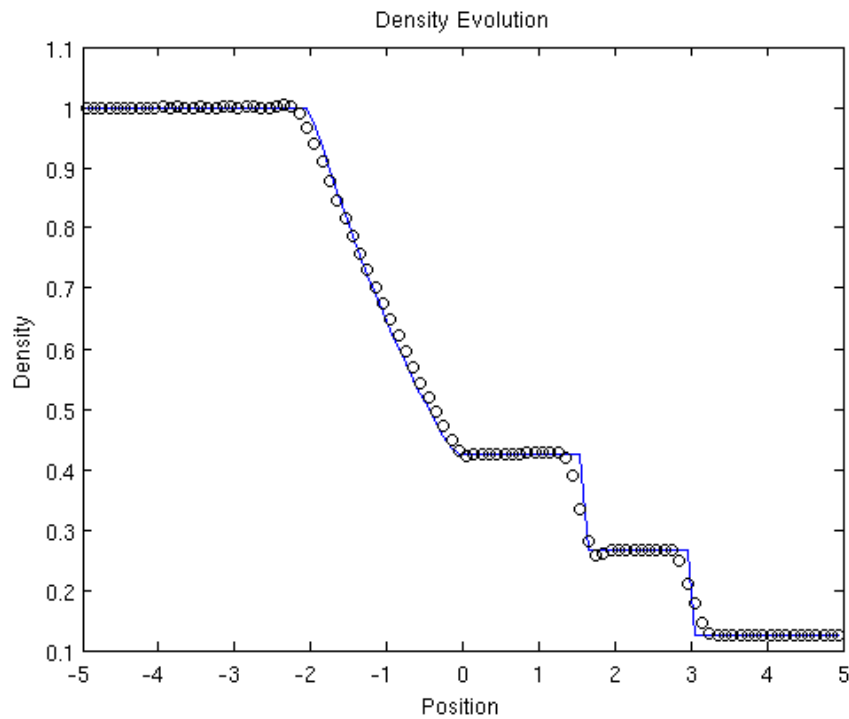
$$a(x, t) = u(x, t) - \frac{x}{t} = \frac{2}{\gamma+1} \left(\frac{x}{t} + \frac{\gamma-1}{2} u_4 + a_4 \right) - \frac{x}{t},$$

$$p = p_4 \left(\frac{a}{a_4} \right)^{2\gamma/(\gamma-1)}.$$

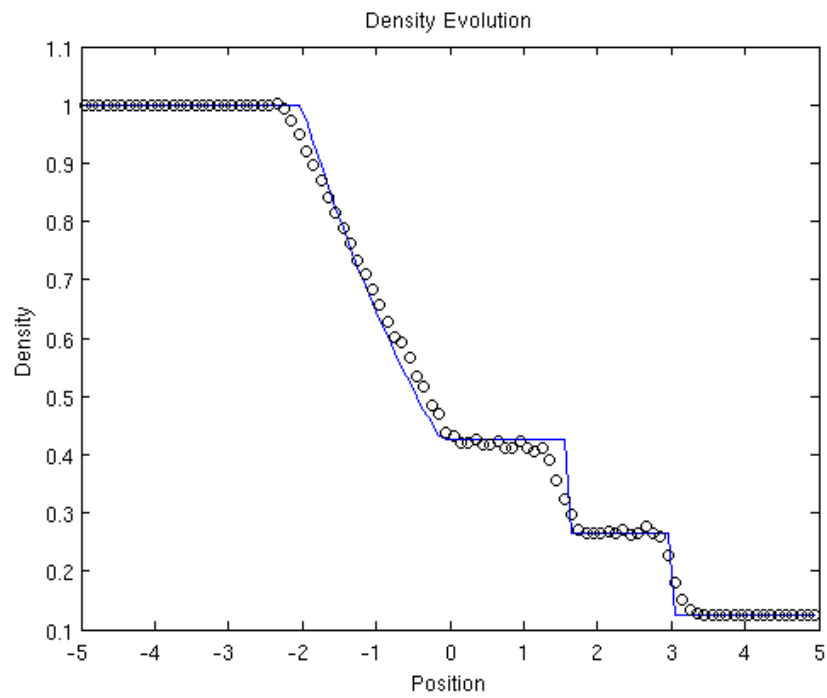
This problem proves to be a good test problem because it contains many of the features one hopes to resolve when using a CFD code such as shocks and entropy waves. Many schemes prove to be oscillatory around flow discontinuities. Thus in order for this scheme to pass the test, it not only has to predict the flow features of the flow, but it also cannot oscillate around shocks and entropy waves.

The results for this problem are given in the figures below for the second and third order methods with the exact solution. The third order method does smear the discontinuities more. That is to be expected since the limiter is activated at more cells. However it is essentially non-oscillatory in both second and third order schemes. However there does seem to be oscillations present which point to the existence of a bug in the third order implementations. The figures only show density because it shows all of the flow features: entropy wave, shock wave, and the expansion fan. All other flow variables are also can be provided upon request.

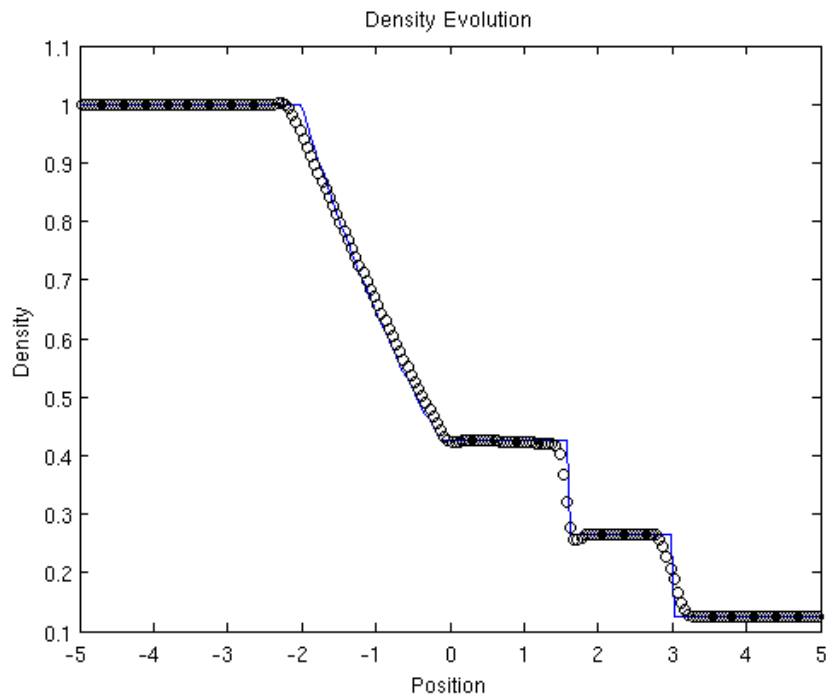
2nd Order 100 Cells



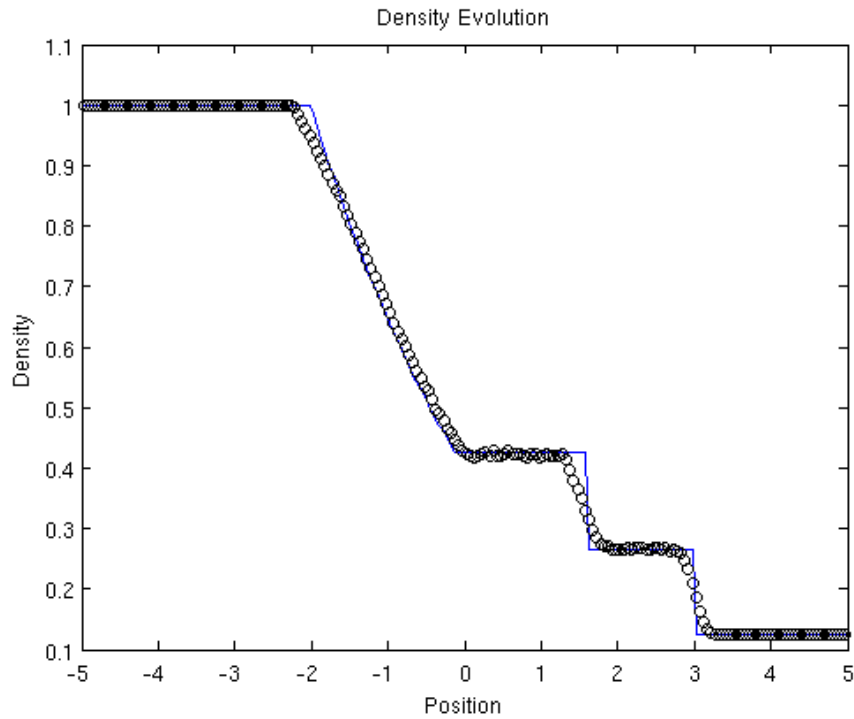
3rd Order 100 Cells



2nd Order 200 Cells



3rd Order 200 Cells



An entropy and shock wave interaction was also used as a test case. This is known as the Osher problem and was taken out of [1]. The initial condition is given by 2.2. The figures for the results are shown below. The method was able to capture the n-wave patterns of the density variation, however the third order method could not capture the non-linear features around the entropy-shock wave collision (circled in figure 2). The limiter believed to be the reason for this issue as it seems the smoothing is too aggressive; relaxing the limiting criteria renders the method unstable however. This issue is currently under investigation.

$$(\rho_L, u_L, p_L) = (3.857143, 2.62369, 10.3333), \quad \text{when } x < -4 \quad (2.1)$$

$$(\rho_R, u_R, p_R) = (1 + 0.2 \sin(5x), 0.1) \quad \text{when } x \geq -4$$

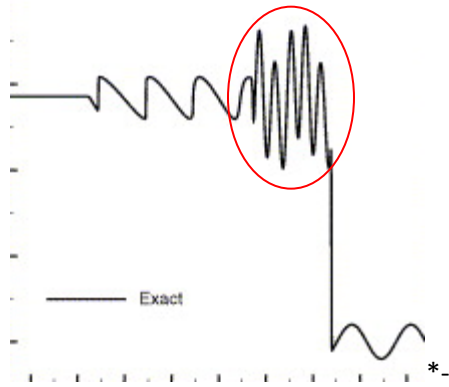
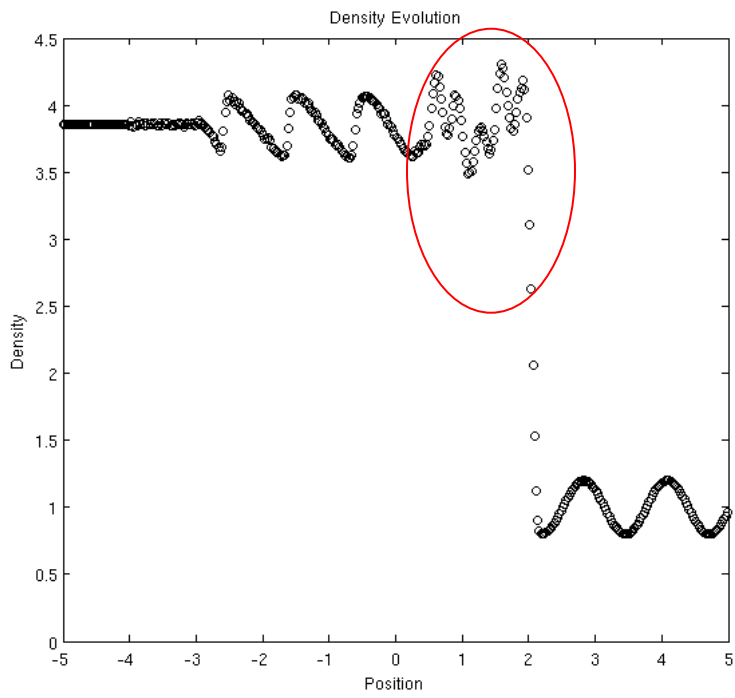
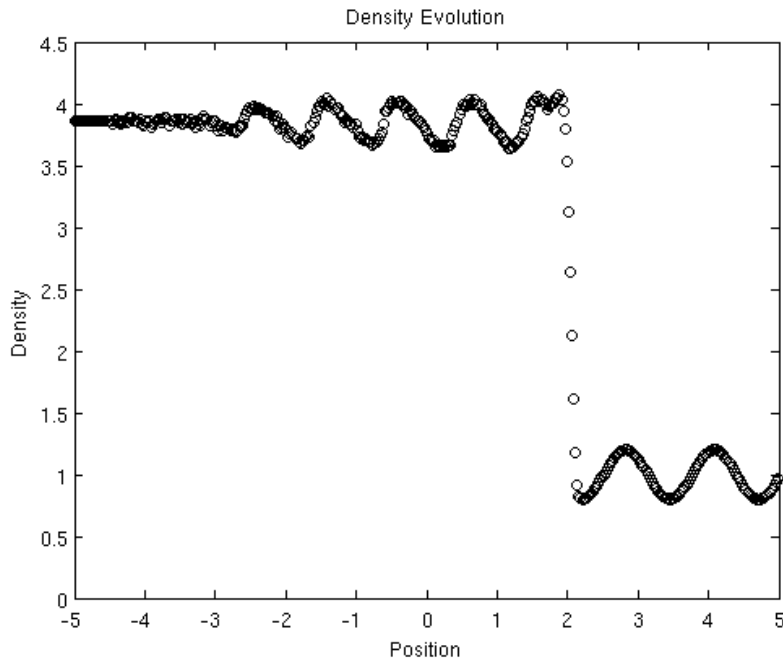


Figure 4. Osher Problem Solution ^[4]

2nd Order Implementation



3rd Order Implementation



The third order method does not seem to perform well under the given conditions.

5. Validation

For validation a grid convergence study was performed on the shock tube problem. This is done by decreasing the cell size systematically and examining if the error reduces by the order the spatial discretization dictates. For example if the cell size is reduced by a factor of two, the error should go be reduced by a factor of two, four, and eight for 1st, 2nd, and 3rd order methods respectively. The error is defined as the time average of the L2 norm of the difference between the exact and numerical solution.

$$E = \frac{1}{\Delta t} \int |u - u_h|_2 dt \approx \frac{1}{\Delta t} \sum |u - u_h|_2 \Delta t \quad (5.1)$$

The shock tube problem turned out to not be a good test because most of the solution is flat and the shock turns on the limiters which are not guaranteed to produce the correct order solution in the region with discontinuities. To counter this, a grid convergence study is being performed currently on an entropy wave convection problem. This problem has a smooth solution requiring no limiting and has a sinusoidal profile eliminating any flat regions. The problem is defined by

$$\mathbf{U}(x, 0) = \begin{cases} \rho_\infty + A \sin(\pi x) \\ u_\infty \\ p_\infty \end{cases} \quad 5.2)$$

$$\mathbf{U}(x, t) = \begin{cases} \rho_\infty + A \sin(\pi(x - u_\infty t)) \\ u_\infty \\ p_\infty \end{cases}$$

The solution is simply the convection of the density wave with the velocity u_∞ .^[5]

This solution does not implement the limiter and thus should give a smooth solution which follows the grid convergence described earlier. This is currently being performed. Initial tests show that the second order method gives good results and a formal grid convergence study for the 2nd order will be provided in an auxiliary report. The third order method is non-convergent without the limiter proving that there is a bug in the 3rd method. This is currently the focus of the work.

Preliminary 2nd order results are shown in figure 4 for clarity only.

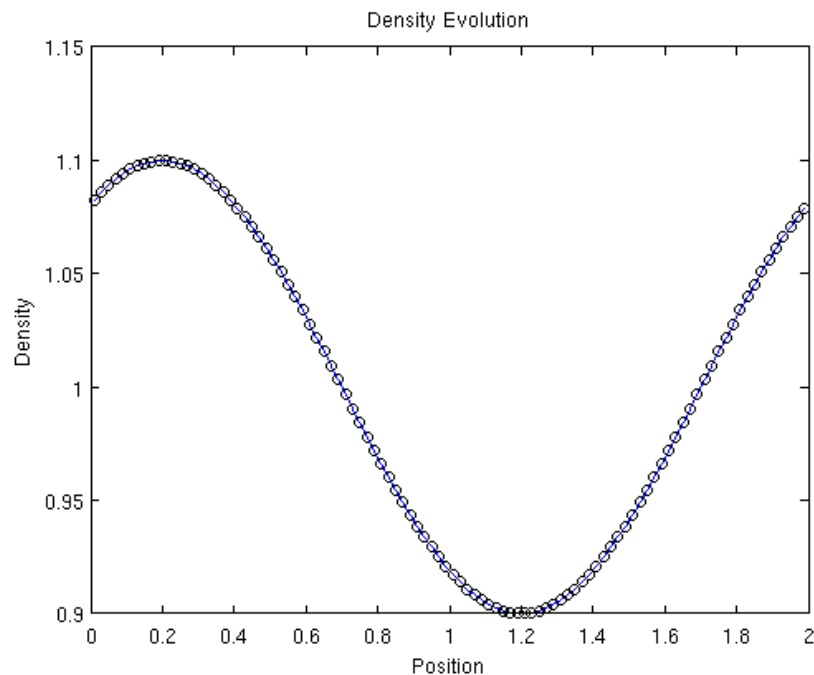


Figure 4. Density Wave at t=2 secs

6. Conclusion

A one-dimensional Galerkin Method was implemented with some success in capturing the discontinuities that arise naturally in the Euler Equations. However because of the apparent worse behavior of the third order scheme, there is a bug in the third order implementation. The overall goal of this work is to develop a two-dimensional flow solver with multiple boundary condition implementations. The immediate focus is to fix the issue causing the third order scheme to perform

worse than the second order scheme. All the modifications will be tested using the density wave convection which has an exact solution and also has smooth, varying feature that make it ideal for grid convergence studies. Once this is complete, work on the two-dimensional version will begin. The original timeline proposed is shown below. Some unforeseen problems were encountered causing a delay in the progress of the work.

7. Initial and Revised Timeline

Initially proposed timeline for the project is given below

10/31/12-	One dimensional version. Apply to one dimensional problem with a known solution to test accuracy and shock capturing abilities. Sod shock tube problem. Will validate the 1-D version (serial)
12/15/13-	Two dimensional version. Apply to 2-D airfoil problem using provided grids (serial)
02/15/13-	Validation of the two dimensional version using experimental airfoil results as well as the results published in literature
03/15/13-	Parallelization of the two dimensional. Validate using results from the serial version

Currently the one dimensional 2nd order implementation is working correctly (grid convergence study results pending). The third order method has a bug that renders it unstable even for smooth solutions. This is being investigated. The complete and validated implementation up to 3rd order will be delivered by January 20th, 2013.

The two dimensional problem has not been addressed as of yet. The revised timeline for uncompleted work for one and two dimensional problems is given below. The details of the timeline are in a separate revised proposal for the project.

01/20/13-	Complete and validated one dimensional Discontinuous Galerkin Code up to 3 rd order accurate in space
03/01/13-	Two dimensional solution of the vortex convection problem to test spatial order of accuracy without limiting (analogous to entropy convection in one dimension). With validation.
04/01/13?-	Two dimensional solution of vortex-shock interaction problem to test the shock capturing capabilities as well as the limiter. (Estimate delivery date because of possible unforeseen complications). With validation
Time permitting-	Boundary treatment such as inviscid wall applied to allow for solutions to airfoils

Delivarables:

Mid- year report

1/20/13-Complete and validated flow solver for one dimensional Euler Equations

03/01/13-Summary of the vortex convection problem including grid convergence studies

04/01/13-05/01/13-Summary of vortex shock interaction problem

05/15/13-Final Report and two dimensional code. Time permitting with boundary treatments.

References:

1. Bernardo Cockburn, Chi-Wang Shu, *TVB Runge-Kutta Local Projection Discontinuous Galerkin Method for Conservation Laws III: One-Dimensional System*, *Journal of Computational Physics* **84** 90-133 (1989)
2. Bernardo Cockburn, Chi-Wang Shu, *TVB Runge-Kutta Local Projection Discontinuous Galerkin Method for Conservation Laws II: General Frame Work* *Mathematics of Computation* Volume **52** 186 (1989)
3. Culbert B. Laney. Computational Gasdynamics. Cambridge University Press. 1998
4. M.P. Martin E.M Taylor, *A bandwidth-optimized WENO scheme for the effective direct simulation of compressible turbulence*, *Journal of Computational Physics* **220** 270-289 (2006)
5. D. Gosh, *Compact-Reconstruction Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws*. Doctor of Philosophy Thesis, University of Maryland, College Park 2012.