MAE 766 Written Preliminary Exam #1

Kyle B. Thompson

March 4, 2016

1 Introduction

Discontinous Galerkin (DG) methods have seen significant development in the last three decades. Incorporating aspects of finite-volume and finite-element methods, DG schemes offer a compact way of achieving higher order accuracy by solving for polynomial basis coefficients that are defined uniquely on each element. Discontinuities at the element boundaries are reconciled via "flux functions", which provide the means that information is propogated between elements. This is a significant advantage when solving hyperbolic partial differential equations (PDEs) with discontinuities in the solution, since these discontinuities can be handled properly by flux functions using an upwind mechanism. Unfortunately, DG method becomes problematic when dealing with elliptic PDEs. Since there is not an upwinding mechanism for the diffusion operator, a central difference seems to be an intuitive choice; however, this has been shown to result in an inconsistent scheme. The central difference flux does not capture jumps in the solution, only in the solution gradients. To overcome this issue, Bassi and Rebay developed a flux scheme to account for jumps in the solution via "local lifting operators", which "lift" the solution to account for C_0 continuity.

2 Problem Definition and Method of Solution

Potential theory states that for incompressible flows that are irrotational, the velocity potential function of a flow is governed by the Laplace Equation

$$\Delta\phi = 0 \tag{1}$$

In two space dimensions this can be written out explicitly as

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

and the velocity is equivalent to the gradient of the potential function

$$u = \frac{\partial\phi}{\partial x}, \qquad v = \frac{\partial\phi}{\partial y} \tag{3}$$

The second scheme derived by Bassi and Rebay (BR2) solves this problem by rewriting Eq. (1) as a first order system to be solved

$$\nabla \cdot q = 0 \tag{4}$$

$$q - \nabla\phi = 0 \tag{5}$$

with q being a vector auxilliary variable with dimensionality of the problem. We proceed by defining the solution ϕ as a polynomial

$$\phi = \sum \phi_i B_i \tag{6}$$

where ϕ_i are the basis weights for the basis set B_i. The solution can now be iterpolated anywhere in the computational domain, and the basis set used here is a linear finite-element basis based on barycentric coordinates

$$\begin{gathered} B_i = \frac{a_i x + b_i y + c_i}{D} \\ a_i = y_j - y_k, \quad b_i = -(x_j - x_k), \quad c_i = x_j y_k - x_k y_j \\ D = c_1 + c_2 + c_2 \end{gathered} \tag{7}$$

where the subscripts $_i$, $_j$, and $_k$ denote different nodes around the triangle. This is a nodal basis, and is useful in that the exact integrals of the basis functions are given by

$$\int_{\Omega_i} B_1^m B_2^n B_3^l d\Omega = D \frac{m!\ n!\ l!}{(m+n+l+2)!} \tag{8}$$

$$\int_{\Gamma_{ij}} B_1^m B_2^n d\Gamma = L \frac{m!\ n!}{(m+n+1)!} \tag{9}$$

Where Ω_i denotes the domain integral over the element i, and Γ_{ij} denotes the contour integral over the interface or boundary of an element. By introducing the "lift operator" concept and recasting Eq. (5) in integral form, it is possible to arrive at the "primal formulation" that solves the system in a single equation as a function of the solution basis weights and the lifting operators. The first Bassi and Rebay scheme (BR1) defined a "global lifting operator", $\boldsymbol{\delta}$, as

$$\int_{\Omega_i} \boldsymbol{\delta} \cdot \boldsymbol{\tau} d\Omega + \sum_{j=1}^{3} \int_{\Gamma_{ij}} \frac{1}{2} [\![\phi]\!] \boldsymbol{\tau} \cdot \mathbf{n} d\Gamma = 0 \tag{10}$$

where $\boldsymbol{\tau}$ is an arbitrary basis function with the dimensionality of the problem. However, using this global lifting operator was found to be non-compact, and unstable for purely elliptic problems, such as the Laplace equation. The BR2 scheme overcomes both of these issues by defining a "local lifting operator", $\boldsymbol{\delta_l}$, defined at each element interface by

$$\begin{gathered} \int_{\Omega_i} \boldsymbol{\delta_l} \cdot \boldsymbol{\tau} d\Omega + \eta \int_{\Gamma_{ij}} \frac{1}{2} [\![\phi]\!] \boldsymbol{\tau} \cdot \mathbf{n} d\Gamma = 0 \\ [\![\phi]\!] = \phi_i - \phi_j \end{gathered} \tag{11}$$

Where η is a stability parameter and $[\![\cdot]\!]$ is a jump operator. It has been proven that $\eta \geq N_{edges}$ results in a stable scheme. Additionally, since $\boldsymbol{\tau}$ is an arbitrary basis, the global lifting operator can be related to the local lifting operators by summing the local lift contributions

$$\boldsymbol{\delta} = \sum_{e=1}^{3} \boldsymbol{\delta_l}^{(e)} \tag{12}$$

With this, the primal form is given by

$$\int_{\Omega_i} (\nabla \phi - \boldsymbol{\delta}) \cdot \nabla B d\Omega - \sum_{j=1}^{3} \int_{\Gamma_{ij}} \{(\nabla \phi - \boldsymbol{\delta_l}) \cdot \mathbf{n}\} B d\Gamma = 0 \tag{13}$$

where $\{\cdot\}$ is an average operator.

3 Implementation

Since the Laplace equation is a linear problem, it is possible to construct a global system an solve for all unknowns simultaneously. This is reminscient of the continuous Galerkin (CG) method, where a stiffness matrix was formed for the left hand side and a load vector for the right hand side. In the Discontinuous Galerkin formulation, however, continuity is not enforced

across elements; therefore, the number of unknowns going from CG(P1) to DG(P1) increases from the number of points in the mesh to the number of elements times each elements' verticies. Additionally, if the system is to be solved implicitly, each local lifting operator $\boldsymbol{\delta_l}$ must be solved for as well, increasing the total number of equations to $3(N_{elem}) + N_{faces}$ at a minimum. This drastically increases the problem size and complexity for an implicit solver, since a sparsity pattern must be exploited to formulate an efficient linear solver.

An alternative to solving the system implicitly is to explicitly evolve the solution in pseudo-time to steady state. Adding a time derivative to Eq. (13) gives

$$\int_{\Omega_i} \frac{\partial \phi}{\partial t} B d\Omega + \int_{\Omega_i} (\nabla \phi - \boldsymbol{\delta}) \cdot \nabla B d\Omega - \sum_{j=1}^{3} \int_{\Gamma_{ij}} \{(\nabla \phi - \boldsymbol{\delta_l}) \cdot \mathbf{n}\} B d\Gamma = 0 \tag{14}$$

This results in a much simpler problem to solve, and the notation can be contracted to

$$\frac{\partial \phi}{\partial t} = -\mathbf{M}^{-1}\mathbf{r} \tag{15}$$

where $\mathbf{r}$ is the residual vector and $\mathbf{M}$ is the block diagonal mass matrix. If the boundaries are not time-dependent, then we recover the solution to the Laplace equation upon convergence, since $\mathbf{r} = 0$. The mass matrix components can be easily computed using Eq.s (8-9), and the system can be inverted by hand to signifcantly save computational time. The time integration is done via Runge-Kutta and the time step can be computed locally or globally based on a CFL prescribed. Thus, this explicit DG scheme can be evolved from an initial state to the solution of the laplace equation by iteratively updating each degree of freedom until the residual is less than a prescribed tolerance, with the local lift operators $\boldsymbol{\delta_l}$ simply computed as an intermediate variable.

4 Numerical Results

The case being solved is a channel with a circular bump. Figure 1 shows triangulated mesh and the Neumann boundary conditions prescribed. Flow in and out of the channel is the freestream velocity (1,0), and no flux is permitted through the channel walls. The solution initial state

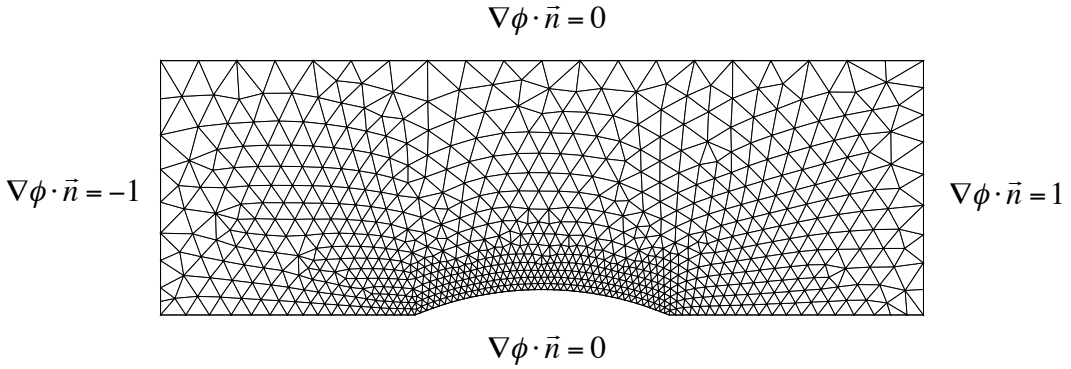
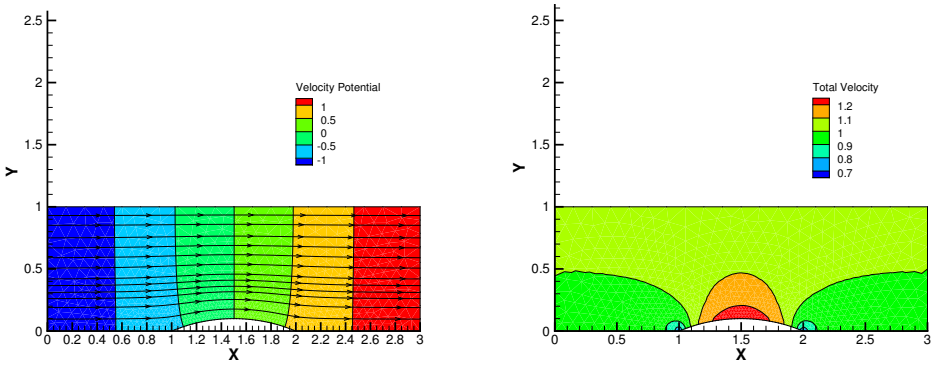


Figure 1: Channel Mesh and Boundary Conditions

was set as zero throughout the domain and evolved to a steady state solution through explicit runge-kutta time integration. The time step used in the integration was computed locally, based on element area and the prescribed CFL number, and the system was considered converged when the absolute magnitude of the L2 norm of the residual was less than 10^{-12}. Figure 2a shows the



(a) Velocity Potential with Streamlines (b) Total Velocity Contours

Figure 2: Channel Contour Plots

velocity potential, ϕ, contours across the computational domain. It is clearly seen that there is a great deal of symmetry to this particular problem, since the geometry is mirrored across the half-plane. The potential behaves as expected, with the difference between the edges of the domain and the centerline being equal in magnitude. Figure 2b also supports the prevalence of symmetry, and shows that the flow does indeed accelerate as it passes over the spherical bump, as expected by the venturi effect. Figure 3 breaks the symmetry seen in the contour plots, as the velocity and start and end on the bump are of equal magnitude. Since this is potential flow over a supposedly symmetric problem, this is unexpected, but also seen and documented by the CG(P1) solver. Upon examining the mesh over the lower surface, it is seen that the domain is

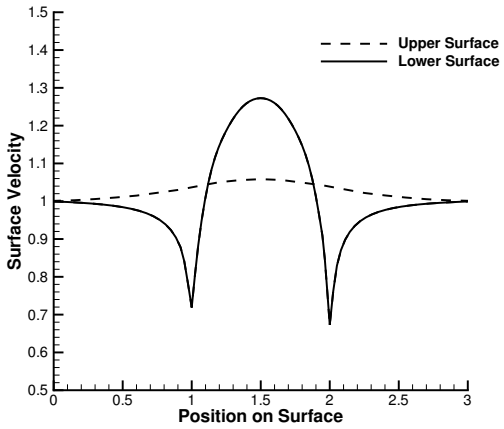


Figure 3: Channel Upper and Lower Surface Velocity

not perfectly symmetric. Figure