Runge-Kutta Discontinuous Galerkin methods for Plasma by Alec Johnson, September 24, 2008

1 My motivation.

I am trying to develop a fast solver to simulate fast magnetic reconnection in collisionless space plasmas. Magnetic reconnection is the cancellation and reconfiguration of field lines. In the ideal magnetohydodynamics (MHD) model of plasma, magnetic flux is convected with the plasma, just as flux of vorticity is convected with an ideal gas. In focused regions where magnetic field gradients become sufficiently strong, however, the frozen-in flux condition breaks down, and reconnection ensues, resulting in the most powerful explosions in the solar system.

Ideal MHD is our macroscale model. It is of the form

$$\partial_t \underline{q} + \underset{\sim}{\nabla} \cdot (\underset{\underline{\sim}}{f}(\underline{q})) = 0;$$

this model is hyperbolic, i.e., $(\forall \underset{\sim}{n})$ $\underset{\sim}{n} \cdot \underset{\underline{\sim}}{f}_{\underline{q}}$ is diagonalizable with real eigenvalues which are the wave speeds.

Our microscale model is a two-fluid collisionless model of plasma. It is of the form

$$\partial_t \underline{q} + \underset{\sim}{\nabla} \cdot (\underset{\underline{\sim}}{f}(\underline{q})) = \frac{1}{\epsilon}\underline{s}(\underline{q});$$

this model also has hyperbolic flux, but it also has a stiff oscillatory source term (i.e., ϵ is small and $\underline{s}_{\underline{q}}$ has imaginary eigenvalues). In addition, it has fast waves (i.e., $\underset{\sim}{n} \cdot \underset{\underline{\sim}}{f}_{\underline{q}}$ has large eigenvalues for all direction vectors $\underset{\sim}{n}$). MHD is derived from the microscale model by taking $\epsilon \to 0$ (and imposing additional approximating assumptions).

Solutions to the plasma models conserve various quantities. and maintain constraints on the magnetic and electric fields called *divergence constraints*. In order to ensure that the features of our numerical solution are physically correct, we seek to develop numerical methods which somehow enforce or maintain a discretized version of these physical constraints. We hope thereby to make sure not only that our numerical solution diverges slowly from the physical solution but also that the numerical solution remains close to the manifold of physically possible solutions for all time and exhibits physically correct behavior (in particular, shocks should travel at the right speeds).

Numerical methods which (up to numerical precision) maintain discretized versions of physical constraints are called **mimetic**. Mimetic methods are generally derived by integrating the corresponding physical laws over mesh cells (which leads to an underdetermined system) and then closing the system by positing a discrete constitutive relation (such as an estimate of discretized flux based on discretized state values). For an excellent high-level discussion of conservative methods based on mimetic differencing operators, see [2].

2 Shock-capturing.

In problems where the hyperbolic term is strong (i.e., where convection plays an important role), we seek methods which are **shock-capturing**. This means that shock waves (1) should move at the correct speed and (2) should have the correct shape. The Rankine-Hugoniot jump condition tells us that to ensure that shock waves travel at the right speed, the numerical method should be *conservative*, i.e., the change in the amount of stuff in each cell should be representable in terms of fluxes across boundary elements. To ensure that we get clean shocks without smearing or artificial oscillations near the shock, we apply *limiters* to high-order estimates of flux.

3 Classes of methods.

Shock-capturing methods tend to be explicit (or perhaps semi-implicit). Explicit methods are simple and allow for a small stencil, but a small stencil requires a time step short enough to allow information to propagate. In hyperbolic systems information can only propagate as fast as the fastest wave, so explicit methods are a natural choice.

To obtain high-order conservative methods, one needs to calculate high-order estimates of the flux across cell boundaries. Two approaches are: (1) to widen the stencil to include information from more cells (collocation methods), or (2) to include more information in each cell (Galerkin methods). There are also two approaches to space-time coupling: (1) first discretize in space to get a system of ODEs in time and then discretize in time (this is known as the *method of lines* (MOL)), or (2) discretize space and time together. Combining possibilities gives four types of conservative methods:

high-order mechanism	**time discretization**	**prominent example**
collocation	MOL	WENO
collocation	space-time	Lax-Wendroff
Galerkin	MOL	RK-DG
Galerkin	space-time	STE-DG

Collocation methods represent states using point values and estimate fluxes using interpolating polynomials. Galerkin methods represent states as a linear combination of a finite set of basis functions and estimate fluxes by numerical integration.

Not all methods fit neatly into one of these categories. In particular, a representation in terms of point values may be equivalent to a representation in terms of a linear combination of a basis of interpolating polynomials. For example, "nodal" discontinuous Galerkin methods represent solutions in terms of values

3.1 Advantages of MOL-DG

The advantages of MOL-DG are generally related to its high level of decoupling. First, the use of MOL decouples space and time and so decouples the order of accuracy in space and time. For MOL methods the instantaneous flux is estimated. This means that the order of accuracy does not depend on the order of accuracy of flux estimates. [1] Second, DG has minimal coupling between cells. The stencil of a given cell never extends beyond its nearest neighbors regardless of the order of accuracy. (This is made possible by high-order representation and the explicit scheme). Since the evolution stencil only depends on neighboring cells, DG makes it easy to get high-order accuracy for arbitrary meshes.

3.2 Disadvantages of DG

Finite Volume methods have the advantage that it is easy to formulate mimetic operators which preserve divergence constraints up to numerical precision. Mimetic ("mixed") DG methods have to use a staggered mesh, which couples all cells and therefore requires inverting a nondiagonal matrix. We instead resort to divergence cleaning.

4 DG framework

The FV framework begins by integrating the balance law $\underline{q}_t + \underset{\sim}{\nabla} \cdot \underset{\simeq}{f} = \underline{s}$ over each mesh cell. The DG framework, however, is based on the variational formulation of the balance law. We first multiply the balance law by an arbitrary test function $\underline{\mathbf{v}}$ and then integrate over an arbitrary region C (which we anticipate to be a mesh cell):

$$\begin{aligned}
&\int_C (\underline{q}_t + \underset{\sim}{\nabla} \cdot \underset{\simeq}{f} = \underline{s}) \cdot \underline{\mathbf{v}}, \text{ i.e.,} \\
&\partial_t \int_C \underline{q} \cdot \underline{\mathbf{v}} + \int_C (\underset{\sim}{\nabla} \cdot (\underset{\simeq}{f} \cdot \underline{\mathbf{v}}) - \underset{\simeq}{f} : \underset{\sim}{\nabla} \underline{\mathbf{v}}) = \int_C \underline{s} \cdot \underline{\mathbf{v}}, \text{ i.e.,} \\
&\partial_t \int_C \underline{q} \cdot \underline{\mathbf{v}} + \int_{\partial C} \underset{\sim}{n} \cdot \underset{\simeq}{f} \cdot \underline{\mathbf{v}} - \int_C \underset{\simeq}{f} : \underset{\sim}{\nabla} \underline{\mathbf{v}} = \int_C \underline{s} \cdot \underline{\mathbf{v}}.
\end{aligned}$$

If we choose $\underline{\mathbf{v}}$ to be the identity matrix, this reduces to the finite volume assertion that the amount of stuff in each mesh cell is conserved.

To discretize the method, we project the equation onto a finite-dimensional subspace spanned by a basis of functions consisting of polynomials up to a given order multiplied by the characteristic function of each cell. These basis functions are not continuous (thus the name *discontinuous* Galerkin).

Let $\{\underline{v}_j\}_{j=1}^N$ denote such a basis, and let $\{\underline{v}^j\}_{j=1}^N$ denote its reciprocal basis. Define the inner products for tuple-valued and matrix-valued functions by $\langle \underline{u}, \underline{v} \rangle := \int \underline{u} \cdot \underline{v}$ and $\langle \underset{\simeq}{f}, \underset{\sim}{\nabla} \underline{v} \rangle := \int \underset{\simeq}{f} : \underset{\sim}{\nabla} \underline{v}$, and define a boundary inner product by $\langle \underset{\simeq}{f}, \underline{v} \rangle_\partial := \int_\partial \underset{\sim}{n} \cdot \underset{\simeq}{f} \cdot \underline{v}$, where ∂ denotes the boundary of the region in question and $\underset{\sim}{n}$ denotes the unit normal. Integration by parts tells us that $\langle \underset{\sim}{\nabla} \cdot \underset{\simeq}{f}, \underline{v}^j \rangle = \langle \underset{\simeq}{f}, \underline{v}^j \rangle_\partial - \langle \underset{\simeq}{f}, \underset{\sim}{\nabla} \underline{v}^j \rangle$.

So our variational formulation reads

$$\partial_t \langle \underline{q}, \underline{\mathbf{v}} \rangle + \langle \underset{\simeq}{f}, \underline{\mathbf{v}} \rangle_\partial - \langle \underset{\simeq}{f}, \underset{\sim}{\nabla} \underline{\mathbf{v}} \rangle = \langle \underline{s}, \underline{\mathbf{v}} \rangle.$$

The projection of the evolution equation onto the space of test functions is:

$$\begin{aligned}
&\sum_{j=1}^N \underline{v}_j \langle \underline{v}^j, \partial_t \underline{q} + \underset{\sim}{\nabla} \cdot \underset{\simeq}{f} = \underline{s} \rangle, \text{ i.e.,} \\
&\sum_{j=1}^N \underline{v}_j \left(\partial_t \langle \underline{q}, \underline{v}^j \rangle + \langle \underset{\sim}{\nabla} \cdot \underset{\simeq}{f}, \underline{v}^j \rangle = \langle \underline{s}, \underline{v}^j \rangle \right), \text{ i.e.,} \\
&\sum_{j=1}^N \underline{v}_j \left(\partial_t \langle \underline{q}, \underline{v}^j \rangle + \langle \underset{\simeq}{f}, \underline{v}^j \rangle_\partial - \langle \underset{\simeq}{f}, \underset{\sim}{\nabla} \underline{v}^j \rangle = \langle \underline{s}, \underline{v}^j \rangle \right). \qquad (1)
\end{aligned}$$

To close this evolution equation for the projection of the solution onto the space of test functions, we need constitutive relations that specify the projection of the divergence of the flux and of the source term. We obtain them by estimating the inner products involving the flux and the source term, which we do by pretending that the projection of the solution onto the test space *is* the solution and using Gaussian quadrature to approximate the integrals. Then $\underline{q} = \sum_{j=1}^N \underline{v}_j \langle \underline{v}^j, \underline{q} \rangle$, and equation (1) is a system of N ordinary differential equations in N unknowns, which we can solve using an ODE solver (e.g. high-order explicit Runge-Kutta or Spectral Deferred Correction).

Limiting is done by limiting the solution itself. We limit the polynomial basis functions from high order to low order. Each basis function is limited by differencing the coefficients of the next lower order basis function.

References

[1] B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional Systems*, Journal of Computational Physics **141**, 199224 (1998).

[2] V. Subramanian, J.B. Perot, *Higher-order mimetic methods for unstructured meshes*, Journal of Computational Physics **219**, 68-85 (2006).

[1] For Finite Volume methods, the left and right states are the cell center values, and the Riemann solver's task is to estimate the average flux over the course of a time step. Since the flux rate changes over the course of a time step, flux estimates must be of high order in order to get a high-order method.

For MOL, the left and right states are the solution representation evaluated at the left and right sides of the boundary, and the Riemann solver's task is to estimate the instantaneous flux at the boundary. If the representation of the solution is n-th order accurate then the left and right states at a boundary should agree to n-th order and so should the left flux, right flux, and any stable flux estimate based on the left and right states.