Title Slide

Good morning, everyone. Thank you for coming to my dissertation defense. I've been developing a space-time discontinuous Petrov-Galerkin finite element for applications to transient fluid mechanics.

Table of Contents

I'll begin by explaining our motivation of in some sense automating scientific computing with DPG. Then we'll look at what discontinuous Petrov-Galerkin finite elements are before jumping into some work I did developing a locally conservative DPG formulation. Next I'll develop a theory for space-time DPG as applied to transient convection-diffusion and apply this by extension to the incompressible and compressible Navier-Stokes equations.

Navier-Stokes Equations

Basically what I want to say on this slide is that Navier-Stokes is hard. Despite the decades of research that have been invested in simulating fluid flows, robust simulation is still a challenge.

- Numerical methods need to resolve a wide variety of flow features from shocks to boundary layers to turbulence.
- On top of that, most numerical methods have minimum mesh resolution requirements before convergence can be guaranteed. Meshes that are too coarse for the numerical method to converge on are termed to be in the pre-asymptotic regime. This means that in some sense, the mesh for the simulation to the phenomena on the right needs to anticipate the flow features before they are known.

Initial Mesh Design

This requirement can present additional challenges to the CFD practitioner designing a mesh. At the minimum level, the surface mesh needs to adequately represent the physical geometry of the problem under consideration. But the stability of the numerical methods presents additional requirements on the volume mesh surrounding the surface mesh. This often leads to a trial and error process for the CFD engineer.

DPG on Coarse Meshes

My goal with this research is to remove one of those constraints on computational mechanics practitioners. In contrast to most other methods out there, the DGP method does not suffer convergence issues on coarse meshes. We are able to start on the coarsest mesh possible which sufficiently resolves the geometry and adaptively refine towards a resolved solution. These are results from Jesse Chan's thesis on supersonic flow over a flat plate. This is not a remarkably difficult problem, but what is notable is that we were able to start the simulation on a mesh of only two elements, sit back, and let the method take over as it solved, refined, and converged to a resolved solution after 11 adaptive steps.

Lessons from Other Methods

Before I jump into the particulars of DPG, I wanted to briefly touch on some of the methods that influenced our work and the lessons we learned from them.

- The streamline Upwind Petrov-Galerkin method was really the first finite element method to successfully solve fluid problems. Put simply, SUPG was able to solve convection-diffusion type problems by adaptively upwinding the test functions based on the elements Peclet number. This introduced the idea that you could improve the stability of a finite element method by modifying your test space.
- Despite the similar name, DG methods only really have one significant influence on the DPG method. The realization that discontinuous basis functions or broken Sobolev spaces were fair game for finite element methods allowed the DPG method to progress from being an interesting trick

to a practically computable method.
- Hybridized DG, or HDG is a newer variation of the DG method that seeks to address the main criticism of DG methods: that of proliferation of unknowns. For the lowest order DG method on a 3D hex mesh, the global solve will have 8 times as many unknowns as an equivalent continuous Galerkin method. HDG addresses this by adding new interface unknowns that couple elements together. Internal dofs no longer directly communicate with each other, and can be statically condensed out of the global solve, resulting in a global solve of comparable size to a continuous discretization. DPG likewise makes use of interface unknowns and static condensation.
- Least-Squares finite element methods attempt to turn any PDE into a least-squares solve, bypassing the LBB inf-sup stability conditions and producing a symmetric, positive definite global stiffness matrix. The insight is that finite element methods are most powerful in a Ritz type framework. DPG can be interpreted as a generalized least-squares method where instead of minimizing the residual in L2, we minimized it in a user-defined dual norm.
- Finally, space-time finite elements were proposed as a means to handle some of the disadvantages to attempting to time step a highly adapted mesh. Analysis shows that a unified treatment of space and time within one method can produce superior results for moving boundary problems, but producing a method which is stable both spatially and temporally has historically been a challenge.

Overview of DPG

I'll start with a brief mathematical overview of the DPG method. I call DPG a framework for computational mechanics because it provides a methodology to derive families of stable finite element methods for any well posed variational problem.

This is a derivation of DPG for some abstract variational problem.

- So we are looking for a solution u in some trial space, big U.
- We've multiplied both sides by some test function v and somehow come out with a bilinear form on u and v.

- This bilinear form defines an operator which maps values from the trial space to the dual of the test space.
- We can then write the operator equation in V'
- Now we wish to find the solution u_h which minimizes the residual for some finite dimensional subset of big U
- The dual space norm is not especially convenient to work with computationally, but another functional analysis construct makes this more accessible.
- The inverse Riesz map, which is an isometry, turns this into a standard norm.
- Before I move on with the derivation of the method, I wanted to point out that really, this is a generalization of the least squares finite element method. If we chose the L2 topology for V, we would get the standard least squares method. But now we are free to define the test space topology however we want.

Optimal Petrov-Galerkin Methods

- Since we have a minimization problem, we want to find the critical points, so we take the Gateaux derivative.
- We can then reverse one of the Riesz maps to give this duality pairing, which if you squint your eyes, kind of looks like a new bilinear form with a very special test function defined through the inverse Riesz map and operator B.
- The one complication is that the Riesz map is defined through the test space inner product, so evaluation of the optimal test functions requires an auxiliary problem with the test space inner product on the LHS and the bilinear form operator on the RHS.

Mixed Formulation

In addition to a minimum-residual method and a Petrov-Galerkin method with optimal test functions, there is a third interpretation of DPG as a mixed problem. If we identify the error representation function as the inverse Riesz map of the residual, we can write an equivalent saddle-point mixed formulation where you are simultaneously solving for the error representation function and the approximate solution. The curious thing about this formulation is that the approximate solution, uh, plays the role of Lagrange multiplier for the error representation function. This particular interpretation carrier much larger global solve, but is useful for analytical purposes and when defining a non-Hilber Lp version of DPG which I will touch on later.

The Most Stable Petrov-Galerkin Method

Babuska's theorem guarantees that discrete stability and approximability imply convergence. So given a bilinear form with discrete inf-sup constant gamma_h, then the Galerkin error will be bounded by M, gamma_h, and the best approximation error. But the optimal test functions are defined such that they realize the supremum and the discrete gamma_h is greater than or equal to the continuous constant gamma. If continuous problem is well-posed, OPG will be as well. Furthermore, using the energy norm in the above estimates gives M = gamma = 1, and the optimal Petrov-Galerkin method is the most stable Petrov-Galerkin method possible.

Other Features

A few other interesting features of the DPG method.

The test space so far has not been defined, but if we were to use continuous test functions, then the auxiliary problem would turn into a global solve of similar complexity to the original problem. Introducing a broken test space localizes these solves making it embarrassingly parallel. The downside is that this introduces additional interface unknowns, which I will illustrate later with an example.

As a minimum residual method, the method always produces a hermitian positive definite stiffness matrix (SPD for real valued problems). We haven't really explored the implications for iterative solvers, but this seems like it would be a positive feature.

We can also evaluate the residual error without knowing the exact solution. This can be used to robustly drive adaptivity.

High Performance Computing

Finally a few notes on some of the implications for high performance computing. The goal is to design a method that eliminates human intervention as much as possible.

- Exceptional stability properties prevent a solve from crashing, eliminating expensive restarts
- The method runs robustly on a wide range of Reynolds numbers
- Adaptivity allocates degrees of freedom efficiently to allow larger simulations with fewer resources
- Automaticity means that you can start a simulation and let it solve and adapt without needing to jump in and fix things part way through
- DPG is very compute intensive, with a large portion of the work being done in the embarrassingly parallel local solves and stiffness matrix assembly.
- In our code, we do the local solves via QR factorization. But the local solve can be viewed as having multiple right hand sides, so that QR factorization can be recycled multiple times.
- Degrees of freedom can be separated into two categories: internal and trace. The internal degrees of freedom can be condensed out and the global solve can be conducted purely in terms of the trace variables which have limited coupling, reducing fill in of the stiffness matrix. The internal DOFs can then be solved for in an embarrassingly parallel post-processing phase. This is a similar process behind the hybridized DG method.
- As I mentioned earlier, an SPD stiffness matrix seems like a promising thing for iterative solvers, but we haven't really explored the benefits.
- Many supercomputer simulations are increasingly coupling multiphysics. The only stability requirement for a DPG method is that the continuous problem is well-posed. As such, it has successfully been applied to a wide range of problems from Helmholtz to solid mechanics to Navier-Stokes and Maxwell's equations.

Locally Conservative DPG

Now I'll go over some of my work developing a locally conservative DPG formulation using convection-diffusion as a model problem. We prefer to work with a system of first order PDEs, but it's possible to do DPG directly on the second order problem. The first equation represents our conservation law, and the second a constitutive law for stress sigma.

Proceed by multiplying by test functions v and tau, then integrating by parts over each element K. Note that the test functions are discontinuous between elements. We seek field variables u and sigma in L2, but this leaves their traces between elements undefined. So we introduce new flux and trace unknowns t hat and u hat. Putting it all together we get our new bilinear form.

Local Conservation for Convection-Diffusion

What does local conservation mean in the context of convection-diffusion? We essentially just want the net flux through element edges to be balanced by any source terms internal. If we were to look at the bilinear form on the previous slide, this is equivalent to having a particular test function in the test space, namely v = 1 and tau = 0. Unfortunately, due to the dynamically computed nature of DPG, we can't guarantee that this will happen with standard DPG. However, we can explicitly augment our test space via Lagrange multipliers. The Lagrangian for our newly conservative scheme appears below.

Locally Conservative Saddle Point System

Proceeding as before, we find the critical points, but now we end up with a saddle point system rather than the symmetric positive definite stiffness matrix (which still resides in the upper left hand corner). We've also added an additional unknown for each element. An additional nicety is that since we've explicitly added constants to the test space, we only need to search for optimal test functions in a space orthogonal to constants.

Optimal Test Functions

Here is how we do our optimal test function solve. For each trial function u, we need to find it's complementary optimal test function v_u such that the inner product of v_u with some w equals the bilinear form acting on u and w for all w in V. Technically space V is infinite dimensional, but in order to make this computationally tractable, we use an enriched space to approximate the optimal test functions. This approximation can be troublesome when the inner product contains boundary layers as it can with convection-diffusion. Fortunately, the topology of the V space is our choice, and we can choose the test norm that we want. We've found that the following one works well for convection-diffusion.

Stability and Robustness Analysis

We were able to extend our stability and robustness theory for standard DPG to the locally conservative version via a Brezzi style analysis. I'm not going to reproduce the proof here in this presentation since it can get a little tedious, but you can refer to our ICES report for details.

Stokes Cylinder

That was an easy problem, but let's see how the method does on a notoriously bad one. This is Stokes flow over a cylinder with a narrow channel on either side. Least squares methods will lose nearly 100% of the mass flux around the cylinder, and since DPG can be viewed as a kind of generalized least squares, we might expect a similar issue. On coarse meshes, standard DPG does indeed produce a poor solution, but converges with resolution. THe locally conservative one produces a decent solution on even very coarse meshes.

Here you can see measures of mass loss at different points in the domain, and indeed you do see the standard DPG method getting better with refinement. Conservative DPG on the other hand bumps around at almost exactly zero mass loss.

Stokes Step

The reentrant corner in this problem again causes problems for least squares methods, but the conservative DPG method handles in beautifully. It's not until the standard method resolves the singularity at the corner that it really gets the flow profile correct.

And we see the same kind of story with the mass loss profiles.

Space-Time DPG

But before I put some equations down, let's briefly explore why we might want to work in a space-time framework rather than a traditional finite difference based time stepper. Adaptivity has been an integral part of DPG from day one, and we typically get refined elements which are several orders smaller magnitude than other elements in the mesh. Classical time stepping techniques propagate the entire solution forward in lockstep at the pace of the most restrictive element. Implicit techniques allow you to take larger steps but for the sake of temporal accuracy, you may not want to. Now, there are techniques out there that do allow different elements to proceed at different time steps, such as asynchronous variational integrators, but for us, this is not an ideal solution. Bolting on a different temporal integrator to a DPG spatial integrator produces a kind of Frankenstein of properties. We know we have great stability in space, but where does that leave us temporally? If we instead decide to just treat time as another dimension to be discretized with a DPG method, then we get a unified treatment and preserve all of our nice stability and adaptivity properties. We get automatic local time stepping and a kind of parallel-in-time integration as we can solve an entire time slab at once, distributing different space-time elements within the slab to different processors. I mentioned that space-time presents a challenge for classical finite elements as equations may have different spatial and temporal characteristics, but DPG addresses this concern. Is your space-time formulation well-posed? Yes, then we are in business. The big complication is on the computational and implementation side. Your code now has to support higher dimensional meshes or as we are implementing, a kind of tensor product of spatial and temporal elements.

Space-Time DPG for Convection-Diffusion

The heat equation is the simplest transient fluid problem we can think of. As we will see later, it actually serves as a decent model problem for something as complicated as compressible Navier-Stokes. Just like Navier-Stokes, it is parabolic in space-time. I am going to derive one variational problem for this equation, but there are other choices we could make. In particular, early in DPG's history, we decided to focus on the ultra-weak first order formulation because we believed it presented the most flexibility, but increasingly we have been working with the original second order problem. So as we decompose this into a system of first order equations, we get a constitutive law in space and conservation of mass. The key insight is that we can rewrite our conservation law as a space-time divergence operator.

Ultra-Weak Formulation

This is still in strong form, so let's multiply by test functions tau and v, and integrating by parts over each element. IBP of the stress equation introduces a trace unknown u hat. If we were working with a conforming test space, the integration of this term against tau dot n over element boundaries would cancel, but the cost of a discontinuous test space is new interface unknowns. Similarly, IBP of the conservation equation introduces flux t hat. The trace only exists on spatial boundaries because the stress equation is only integrated by parts over spatial dimensions. Note that the flux changes nature depending on whether it is on a temporal or spatial boundary or a mix of the two.

Robust Norms

So you're probably thinking, this sounds awesome: any formulation I like, choose the norm I converge in, unlimited stability, that's great. Here is where I bring up the little caveat in the method. Let's say we want to develop a method that converges in a norm that is bounded by L2. So let's write our bilinear form in group variables where u represents the volume parts and uhat the interface parts. And since we are working with the ultra-weak first order system, we have the adjoint on the volume integrated test functions. Now the space of continuous test functions is a subspace of our discontinuous test functions, so lets take a conforming test function v *which satisfies the adjoint equation. If we plug this into our bilinear form, we get the L2 norm of u. Then multiplying and dividing by the V norm of v*, we can bound this by the supremum, which is by definition of the energy norm in which we are guaranteed best approximation. So as long as the V norm of v* is bounded by the L2 norm of u, then convergence in the energy norm implies convergence in L2. I've put together a little space-time test problem to numerically check this assertion.

Norm Comparison

There are an infinite number of norms with this property, so lets pick two of them and compare. The graph norm, defined by the adjoint equation is on the left and another norm, I will call the robust norm is on the right. In the plots, we see the L2 and energy error during an adaptive solve for decreasing diffusion parameters. Ideally the L2 error (dashes) and energy error (dots) should parallel each other and monotonically decrease. Yet we only see the desired behavior in robust norm. This comes down to conditioning and approximability of the optimal test functions.

Ideal Optimal Test Functions

The very desirable properties of DPG assume that we are using optimal test functions from an infinite dimensional space V, but in order to make this computationally tractable, in practice we use a finite dimensional enriched space in the auxiliary problem. Different test norms will produce different ideal optimal test functions. Here we see the optimal test functions corresponding to a linear trial basis function with the two norms. The graph norm optimal test functions have very strong boundary layers that we could not hope to approximate with any reasonable enriched space.

Approximated Optimal Test Functions

Here we see what the actual approximated test functions would look like for the two norms. There is little difference between the ideal and approximated shape functions with the robust norm, but a cubic approximation of the sharp boundary layers in the graph norm is not sufficient, and our overall convergence suffers. Most of the labor of designing a DPG method goes into analysis of the auxiliary solve to make sure we get boundedness by L2, approximable test functions, and good conditioning. But ultimately, this boils down to some fairly basic functional analysis and understanding of PDEs.

Two Robust Norms for Steady Convection-Diffusion

We've successfully used the following norms for steady state convection-diffusion problems. The first norm was derived and proved to be robust by Jesse Chan, Norbert Heuer, and Leszek Demkowicz. Jesse and I later realized that for some problems we could get much better results by replacing the Div Tau term with a term coupling Div Tau and Beta dot Grad V. Based on the previous proof of robustness, this norm was also trivially robust.

Two Robust Norms for Transient Convection-Diffusion

In adapting these norms for transient convection-diffusion, we treat dv/dt as if it were just convection in time and write Beta tilde as the spatial convection field augmented with 1 for temporal convection while defining a space-time gradient operator. The only change we make to our norms is replacing Beta dot Grad v with Beta tilde dot Gradxt V.

Adjoint Operator

Now I'm going to outline to proof of robustness. We begin by writing the first order system with homogeneous boundary conditions: flux conditions on the inflow and trace on the outflow. I can go through the derivation later if you care, but the adjoint can essentially be thought of as reversing any convection directions. So we get our adjoint operator A*.

Controlling Different Field Variables

We begin by splitting our adjoint into two parts, one with a forcing term on the conservation law and the other with forcing on the constitutive law. We then try to derive bounds on the two parts independently. Bounds on v1 and tau1 correspond to having robust control of our primary field variable u while v2 and tau2 corresponds to controlling the stress sigma.

Proved Bounds at Our Disposal

In our recently submitted paper, we prove the following lemmas. The first one holds for both v1 and v2 while the second we could only prove for v1.

Control of u

We can use the previous two lemmas and definition of the first split adjoint equation to develop the following bounds. These account for all the terms in both the robust and the coupled robust norm guaranteeing at the very least that we get robust control of our primary field variable u.

Control of sigma

However for v2 and tau2, we only have lemma 1 at our disposal which means that we can only develop bounds on the following terms which means that we can not mathematically guarantee that either of these norms gives us robust control of sigma.

Transient Analytical Solution

We developed the following analytical solution to test our claims of robustness. We have a transient impulse that dies away to a solution of the steady state convection-diffusion problem.

Robust Convergence to Analytical Solution

We try this out for a range of different diffusion coefficients from 10^-2 to 10^-8 and plot the reported energy error and L2 error for different polynomial orders with the two norms. As expected, we get robust convergence in every case.

Space-Time Incompressible Navier-Stokes

Space-Time Compressible Navier-Stokes

I'll get into some of our results applying the space-time methodology to compressible Navier-Stokes. I mentioned that space-time support in Camellia is still experimental, so the numerical results so far are limited to 1D space + time, but we are on the cusp of getting a tensor topology thing working for 2D and 3D computations. That is basically the last thing I need to do to graduate, actually. But here are the preliminary 1D results.

First Order System

One interesting thing about the DPG framework is that it takes stability out of the question. One long standing issue when it comes to simulating fluid flow is whether it's best to work with primitive variables, conservation variables, or entropy variables. Primitive variables have the benefit of being simple, intuitive and the least nonlinear formulation. Conservation variables make time stepping much more straightforward. And Entropy variables were introduced in the belief that a symmetric system was best. I'm going to perform a simple comparison of these three formulations within our space-time DPG framework.

For the sake of simplicity, we assume Stokes hypothesis, ideal gas, and constant viscosity, but if we wanted to choose something more realistic for these, it would just make the linearization process a little messier. We start by splitting up our conservation equations into a first order system as we did with convection-diffusion, but here we jump straight into the space-time divergence form. Since the Navier-Stokes equations are incompletely parabolic, we only get 2 constitutive laws to the 3 conservation equations.

Compact Notation

We now introduce compact notation defining the conserved quantities, the inviscid fluxes, the viscous fluxes, viscous terms, and viscous relations. At this point we could perform a change of variables to either conservation or entropy variables, but I'm going to hide those details in this talk.

Define Group Variables

We further compact our notation by introducing group variables. We can now write our Navier-Stokes problem in a way that we can easily draw analogies to convection-diffusion. I'm going to skip the details, but we then linearize this and develop test norms.

Sod Problem

We start with everyone's favorite shock tube problem, the Sod problem. This is a spatially 1D problem from 0 to 1 with a final time of 0.2. So we introduce a starting space-time mesh with 4 quadratic elements where the y=0 axis corresponds to the initial conditions and y=0.2 is the final time. Now we are going to track the refinement process as the method automatically resolves the solution features. After 14 refinements, we've pretty much gotten our mesh down to the viscous scale along the shock. I want to point out that we are now getting to the point where we can apply a multigrid solution strategy to the series of adaptive meshes that we generate here.