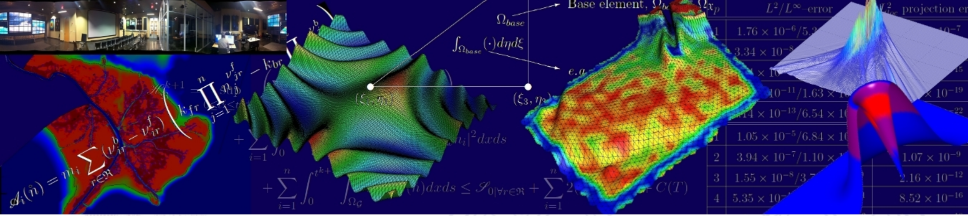


# A primer on DG methods, and DG-SWEM

STORM Kickoff meeting, October 21 2014

Craig Michoski

COMPUTATIONAL HYDRAULICS GROUP
THE UNIVERSITY OF TEXAS AT AUSTIN

ICES INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES

1. A primer on discontinuous Galerkin methods

2. An overview of DG-SWEM

# A basic spiel on DG methods

## Some Marketable features

### Why DG is useful ...

- Immediate high order accuracy with *p*-adaptivity (*h*-adaptivity)
- Local isolation of error
- Extendible representation of generalized nonlinear systems (extreme flexibility of model design)
- Robust solutions, locally conservative by construction
- Recovery of weak (realistic?) function spaces
- Can be made computationally competitive ....
- Lots of flavors: nodal, modal, mixed, hybrid, HDG, DPG, RDG, etc.

# How does the DG method work?

## Vanilla discontinuous Galerin method from scratch

Consider linear advection–diffusion in one spatial dimension:

$$u_t + a u_x - b u_{xx} = 0,$$

where $u = u(x, t)$, and $a$ and $b$ are positive constants.

Since mathematical analysis shows that such an equation in general may have no solution $u \in C^2$, we look for solutions (those that hopefully both *exist* and are *unique*) in the so-called *weak form* or variational form.

# The weak form of an equation

To get to the weak form of an equation, we multiply by some "sufficiently smooth" test function $\phi(x)$,

$$\phi u_t + \phi a u_x - \phi b u_{xx} = 0,$$

and then integrate in space:

$$\int_G \phi u_t dx + \int_G \phi a u_x dx - \int_G \phi b u_{xx} dx = 0$$

where $G$ is the domain/line we are solving the solution over.

# The classical weak form

Since the test function we chose $\phi = \phi(x)$ does not depend on time, we rewrite this equivalently as:

$$\frac{d}{dt}\int_G \phi u dx + \int_G \phi a u_x dx - \int_G \phi b u_{xx} dx = 0.$$

Finally, the weak form attempts to “remove” the derivatives from the solution $u$, and place them onto the test function $\phi$. To do this, we integrate by parts,

$$\frac{d}{dt}\int_G \phi u dx - \int_G \phi_x a u dx + \int_G (\phi a u)_x - \int_G (\phi b u_x)_x dx + \int_G \phi_x b u_x dx = 0,$$

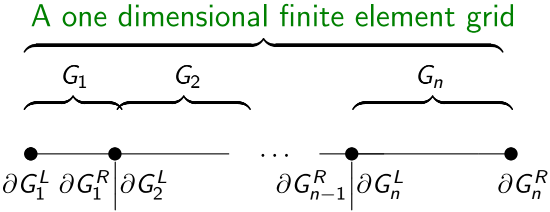
which is just applying the product rule on the derivatives, e.g.

$$\begin{aligned}(\phi a u)_x &= \phi_x a u + \phi (a u)_x \\ &= \phi_x a u + \phi a u_x.\end{aligned}$$

# Moving to the local discrete weak form

We restrict to the single element $G_i$ of our mesh, which is cut up into $n$ elements,

$$\frac{d}{dt}\int_{G_i}\phi u dx - \int_{G_i}\phi_x a u dx + \int_{G_i}(\phi a u)_x - \int_{G_i}(\phi b u_x)_x dx + \int_{G_i}\phi_x b u_x dx = 0$$




Using the fundamental theorem of calculus,

$$\int_{G_i} f_x dx = f|_{\partial G_i^L} - f|_{\partial G_i^R}$$

# The local discrete weak form

Vanilla DG

Then for every element $G_i$ we solve the local problem:

$$\frac{d}{dt}\int_{G_i}\phi u dx - \int_{G_i}\phi_x a u dx + \phi a u|_{\partial G_i^L} - \phi a u|_{\partial G_i^R}$$
$$- \phi b u_x|_{\partial G_i^L} + \phi b u_x|_{\partial G_i^R} + \int_{G_i}\phi_x b u_x dx = 0.$$

Now that we have a formula that makes sense on every element, we are presented with a set of *approximation choices*

$\rightarrow$ a specific flavor of DG.

# The mixed DG method

In DG-SWEM, first we use a so-called mixed method. This entails linearizing the second order (laplacian-type) operator $\partial_{xx}$ relative to an auxiliary variable $\sigma$. That is, choose an auxiliary variable $\sigma = u_x$. Then in our previous equation we can substitute in $\sigma$ wherever $u_x$ appears:

$$\frac{d}{dt}\int_{G_i}\phi u dx - \int_{G_i}\phi_x a u dx + \phi a u|_{\partial G_i^L} - \phi a u|_{\partial G_i^R}$$

$$- \phi b\sigma|_{\partial G_i^L} + \phi b\sigma|_{\partial G_i^R} + \int_{G_i}\phi_x b\sigma dx = 0.$$

Now, to complete our linearization, we solve the equation $\sigma = u_x$ to recover $\sigma$, by casting it into its own weak form (that is multiplying by a test function $\zeta$ and integrating by parts):

$$\int_{G_i}\zeta\sigma dx = \zeta u|_{\partial G_i^L} - \zeta u|_{\partial G_i^R} - \int_{G_i} u\zeta_x dx$$

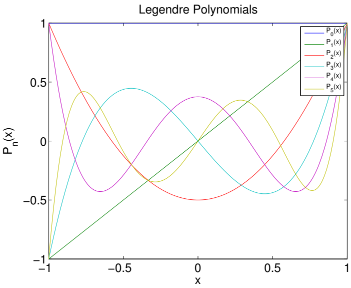
Notice that we have completely removed all the derivatives from $u$ in our formulation.

# The modal representation

As a general rule, the test space we define our test functions over a polynomial space. Here we have choices, but in the case of DG-SWEM we choose a peicewise polynomial modal respresentation. What does this mean?

A popular choice of FE shape functions in DG are the Legendre polynomials $P$, given by:

$$P_{n+1}(x) = \frac{1}{n+1}\left((2n+1)xP_n - nP_{n-1}\right).$$




$$P_0 = 1, \; P_1 = x$$
$$P_2 = \tfrac{1}{2}(3x^2 - 1)$$
$$P_3 = \tfrac{1}{2}(5x^3 - 3x)$$
$$P_4 = \tfrac{1}{8}(35x^4 - 30x^2 + 3)$$
$$P_5 = \tfrac{1}{8}(63x^5 - 70x^3 + 15x)$$

# The modal representation, cont.

We can write the Legendre polynomails of degree $p$ over the modal "degrees of freedom" $n_p$ as

$$P_p(x) = \sum_{l=0}^{n_p} \alpha_l x^l = \sum_{l=0}^{n_p} P_i(x), \quad \forall x \in G_i$$

Then we will define the DG modal test functions $\phi$ as

$$\phi(x) = \sum_{l=0}^{n_p} \phi_l P_l(x), \quad \forall x \in G_i,$$

The approximate DG solution $u_h \approx u$ is then projected into this modal basis as

$$u_h(t,x) = \sum_{l=0}^{n_p} u_l(t) P_l(x), \quad \forall x \in G_i.$$

Thus dimensions work out for multiplying $u_h\phi$, etc.

# But how do we integrate over stencil?

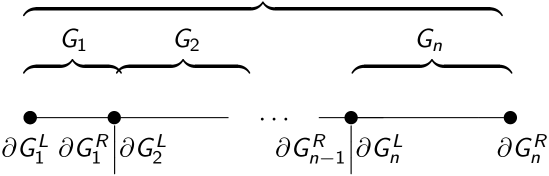
The remaining question: how to compute the interelement terms:

$$\frac{d}{dt}\int_{G_i}\phi u_h dx - \int_{G_i}\phi_x a u_h dx + \phi a u_h|_{\partial G_i^L} - \phi a u_h|_{\partial G_i^R}$$
$$- \phi b\sigma_h|_{\partial G_i^L} + \phi b\sigma_h|_{\partial G_i^R} + \int_{G_i}\phi_x b\sigma_h dx = 0.$$

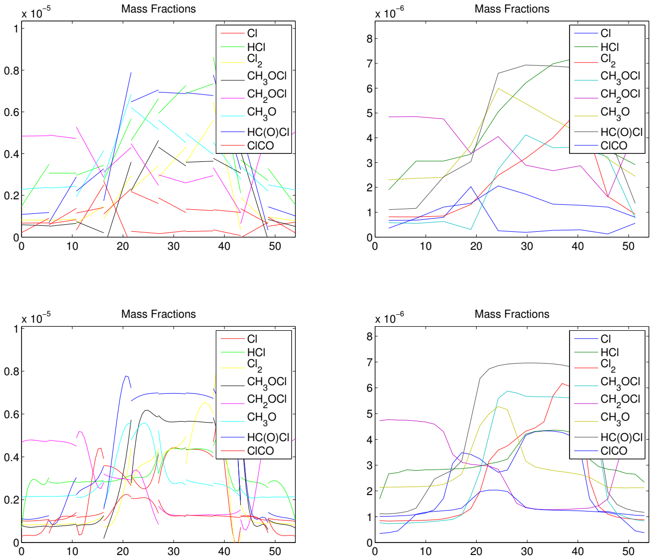
$$\int_{G_i}\zeta\sigma_h dx = \zeta u_h|_{\partial G_i^L} - \zeta u_h|_{\partial G_i^R} - \int_{G_i} u_h\zeta_x dx$$

Why? $\rightarrow$ because the basis $\phi, \zeta$ is a locally discontinuous polynomial!
So the solution $u_h$ is naturally multivalued at the edged of elements.

A one dimensional finite element grid



# Eight component gas, $p = 1$ (top) and $p = 3$



# The concept of numerical fluxes in DG

Consequently, we have to derive mathematically consistent notions of generating "single-valued" evaluations of the solution at these points. This is accomplished using the concept of "numerical fluxes."

The numerical flux depends on the operator type (for example, $\partial_x$ or $\partial_{xx}$). Then the interelement terms are apporximated by numerical fluxes $\hat{\sigma}_h$, $\hat{u}_h$:

$$\phi a u_h|_{\partial G_i^L} \approx \phi a \hat{u}_h|_{\partial G_i^L}$$

$$\phi b \sigma_h|_{\partial G_i^R} \approx \phi b \hat{\sigma}_h|_{\partial G_i^R}$$

There are many choices here, but common ones include standard upwinding

$$\hat{u}_h|_{\partial G_i^L} = u_h|_{\partial G_{i-1}^R} \quad \text{since } a > 0,$$

and averaging,

$$\hat{\sigma}_h = \left( \frac{\sigma_h|_{\partial G_i^R} + \sigma_h|_{\partial G_{i+1}^L}}{2} \right)$$

etc.

# The final form of the DG method

With this we arrive at the formal DG scheme:

$$\frac{d}{dt}\int_{G_i}\phi u_h dx - \int_{G_i}\phi_x a u_h dx + \phi a\hat{u}_h|_{\partial G_i^L} - \phi a\hat{u}_h|_{\partial G_i^R}$$
$$- \phi b\hat{\sigma}_h|_{\partial G_i^L} + \phi b\hat{\sigma}_h|_{\partial G_i^R} + \int_{G_i}\phi_x b\sigma_h dx = 0.$$

noticing that the auxiliary flux $\hat{u}_h$ in the mixed form induces a double stencil (neighbors of neighbors!):

$$\int_{G_i}\zeta\sigma_h dx = \zeta\hat{u}_h|_{\partial G_i^L} - \zeta\hat{u}_h|_{\partial G_i^R} - \int_{G_i} u_h\zeta_x dx,$$

hence the use of the term "local." All that remains is a time-stepper.

# DG-SWEM
## As a DG method

### So let's take a look at DG-SWEM from this vantage

- DG-SWEM uses a modal DG method in two-dimensions
- Dubiner basis up to order $p = 7$ functions are used on triangular elements
- Explicit timestepping (the Runge Kutta method) is used for temporal integration
- LDG-type fluxes are used for diffusion/viscous terms
- Upwinding fluxes are used for convective terms
- A number of methods are used to "stabilize" the solution

1. A primer on discontinuous Galerkin methods

2. An overview of DG-SWEM

# Discontinuous Galerkin Shallow Water Equation Model (DG-SWEM)

As a coastal engineering code

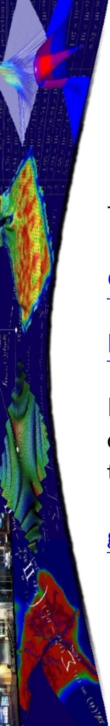
## The physics of DG-SWEM

- Solves a 2D shallow water equation
- Uses a wetting/drying algorithms along coastlines
- Uses the same imput/output formats as ADCIRC
- Couples the Exner equation for soil erosion, bathymetry that evolves in space and time $b(x, t)$
- Solves for advecting pollutants, etc.
- Solves chemically coupled components
- Supports coupling to SWAN

# DG-SWEM
Basic requirements

## Running DG-SWEM requires the following

- intel/13.0.2.146 or newer
- mvapich2/1.9a2 or newer
- Python 2.7.3 (with SymPy support)
- git, plus a github account



# DG-SWEM

## Taking a look under the hood

The documentation and help for DG-SWEM is available online:

Computational Hydraulics Group (CHG) wiki

DG-SWEM sphinx documentation

Moreover, the documentation itself is written using the sphynx python documentation generator, which can be accessed and improved/added to from the online repository:

git to sphynx documentation