An Energy-Conserving Formulation of Maxwellian-Weighted Basis Functions for a Discontinuous Galerkin Kinetic Code*

Greg Hammett, Ammar Hakim, Eric Shi

Princeton Plasma Physics Laboratory

APS-DPP Meeting, Denver, Nov. 11, 2013

* Original submitted title: Tests of Maxwellian-Weighted Basis Functions in a Discontinuous Galerkin Kinetic Code

Improving Confinement Can Significantly Lower Cost of a Fusion Reactor

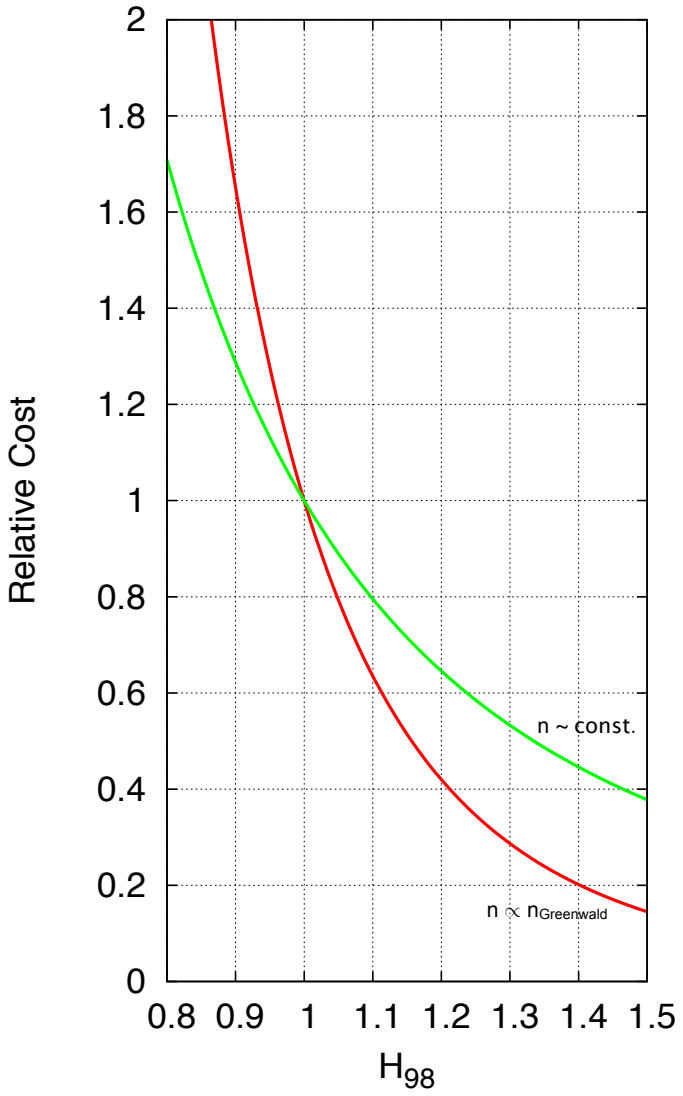
Well known that improving confinement factor H & beta limit can significantly lower cost of electricity at fixed power output.

H has even stronger impact on construction cost at fixed fusion gain Q, because higher H allows a smaller machine to achieve same Q.

Even with a conservative estimate: cost $\propto R^2$,
get cost $\propto 1/H^{4.76}$ (if $n \propto n_{Greenwald} \propto 1/R$).

If H can be improved just 25%, can reduce cost by x3.
(Lower bounds on device size set by blanket & coil thickness, $\langle\sigma v\rangle \sim T^2$ assumption, but can go smaller than present.)

ITER conservatively designed with H=1. Experiments have achieved better confinement via various mechanisms that are understood qualitatively. Working to develop better computer simulations, particularly near plasma edge, to predict extrapolation to reactors.



Many Interesting Ideas To Improve Fusion

* Liquid metal (lithium, tin) coatings on walls: (1) protects solid wall (2) absorbs incident hydrogen ions, reduces recycling of cold neutrals back to plasma, raises edge temperature & improves global performance. TFTR found: ~2 keV edge temperature. NSTX, LTX: more lithium is better, where is the limit?

* Spherical Tokamaks (STs) appear to be able to suppress much of the ion turbulence: PPPL & Culham upgrading 1 --> 2 MA to test scaling

* Advanced tokamaks, alternative operating regimes (reverse magnetic shear or "hybrid"), methods to control Edge Localized Modes, higher plasma shaping. Will beam-driven rotation be more important than previously thought?

* Tokamaks spontaneously spin: can reduce turbulence and improve MHD stability. Can we enhance this with up-down-asymmetric tokamaks or non-stellarator-symmetric stellarators with quasi-toroidal symmetry?

* Many possible stellarator designs, room for further optimization: Quasi-symmetry / quasi-omnigenity improvements discovered relatively recently, after 40 years of fusion research. Stellarators fix disruptions, steady-state, density limit.

* Robotic manufacturing advances: reduce cost of complex, precision, specialty items

Project Motivation

Continuum gyrokinetic codes (like GENE, GYRO, and GS2) have been very successful in the core region of tokamaks. Here we are trying to develop a new continuum gyrokinetic code that can handle the additional complications of the edge region: large amplitude fluctuations, steep variations of profiles, separatrix, open and closed field lines, interactions with wall, strong fuelling.

Code needs to be full F, non-local, with good algorithms for avoiding negative overshoots that occur in vicinity of steep gradients with standard centered, Arakawa, or spectral algorithms. Try recent development in advanced algorithms, including a version of the Discontinuous Galerkin algorithm, to help with these challenges, and to try to help with overall efficiency on this challenging kinetic problem.

MPPC goal to also apply to basic laboratory experiments (LAPD, Vineta in Greiswald) studying drift-wave turbulence & nonlinear Alfvén dynamics for astro applications. Benchmark with GENE when ready. Eventually could also benchmark with SOLPS. compare with ASDEX-U edge measurements.

Code being developed with a general framework, could eventually be applied to a wide range of non-fusion gyrokinetic/kinetic problems, in astrophysics, ... Possible non-plasma applications (rarified gas dynamics: hypersonic vehicles, MEMS (Micro-Electromechanical Systems, semiconductor modelling when electron mean free path is larger than feature size.)

DG Algorithm Motivation

- Discontinuous Galerkin (DG) algorithms: hot topic in CFD & Applied Math in recent years. (Cockburn & Shu JCP & SIAM 1998, >1000 citations)
- DG combines key advantages of Finite Element (low-phase error, high accuracy, flexible geometry) with Finite Volume algorithms (limiters to preserve positivity/monotonicity --> avoid unphysical overshoots, locality --> parallelizes well).
- Gaussian integration methods (like core gyrokinetic codes in velocity space): use optimal location of points to interpolate p points to get 2p-1 order accuracy, ~ twice the accuracy of standard finite volume interpolation (p order accurcy) with non-optimal point locations.
- Also using:
  - certain forms of discontinuous Galerkin have excellent conservation properties, conserve the energy invariant of the Poisson bracket of the Vlasov equation,
  - optimized basis functions (Maxwellian-weighted) could improve efficiency
  - sub-grid turbulence models, efficient use of massively parallel computers, ...
- Edge/pedestal gyrokinetic turbulence very challenging, 5D problem, not yet solved. Benefits from all tricks we can find: Factor of 2 reduction in resolution --> 64x speedup.

Essential idea of Galerkin methods: L_2 minimization of errors on a finite-dimensional subspace

Consider a general time-dependent problem

$$f'(x,t) = G[f]$$

where $G[f]$ is some operator. To approximate it expand $f(x)$ with a finite set of basis functions $w_k(x)$,

$$f(x,t) \approx f_h(x,t) = \sum_{k=1}^{N} f_k(t) w_k(x)$$

This gives discrete system

$$\sum_{k=1}^{N} f_k' w_k(x) = G[f_h]$$

Question

How to determine f_k' in an optimum manner?

Essential idea of Galerkin methods: L_2 minimization of errors on a finite-dimensional subspace

Answer: Do an L_2 minimization of the error, i.e. find f'_k such that

$$E_N = \int \left[\sum_{k=1}^{N} f'_k w_k(x) - G[f_h] \right]^2 dx$$

is minimum. For minimum error $\partial E_N / \partial f'_m = 0$ for all $k = 1, \ldots, N$. This leads to the linear system that determines the coefficients f'_k

$$\int w_m(x) \left(\sum_{k=1}^{N} f'_k w_k(x) - G[f_h] \right) dx = 0$$

for all $m = 1, \ldots, N$.

Key Idea

Projection of residual on the basis set chosen for expansion leads to minimum errors in the L_2 sense. For this reason DG/CG schemes are constructed by projecting residuals of PDEs on basis sets.

Essential idea of Galerkin methods: L_2 minimization of errors on a finite-dimensional subspace

Answer: Do an L_2 minimization of the error, i.e. find f'_k such that

$$E_N = \int \left[\sum_{k=1}^{N} f'_k w_k(x) - G[f_h] \right]^2 dx$$

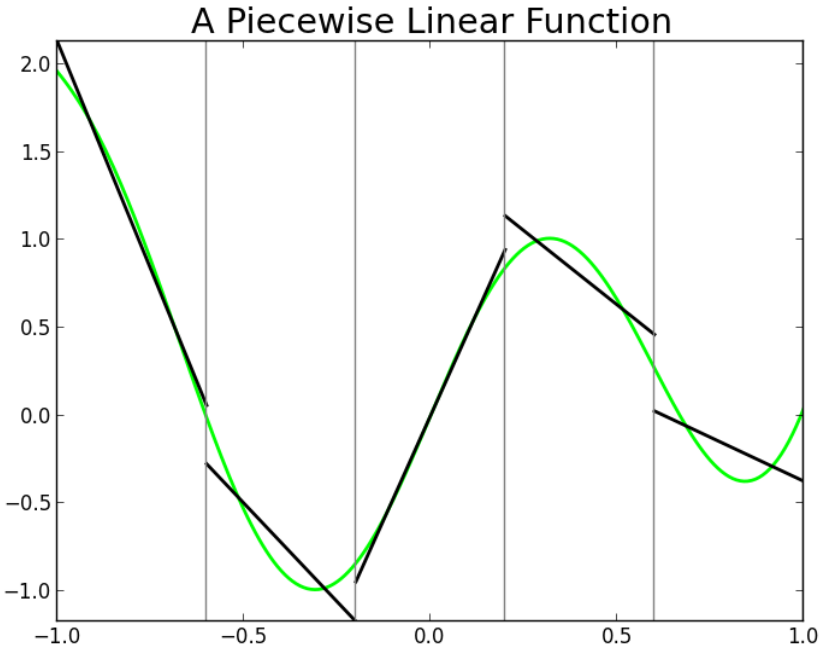
is minimum. For minimum error $\partial E_N / \partial f'_m = 0$ for all $k = 1, \ldots, N$. This leads to the linear system that determines the coefficients f'_k

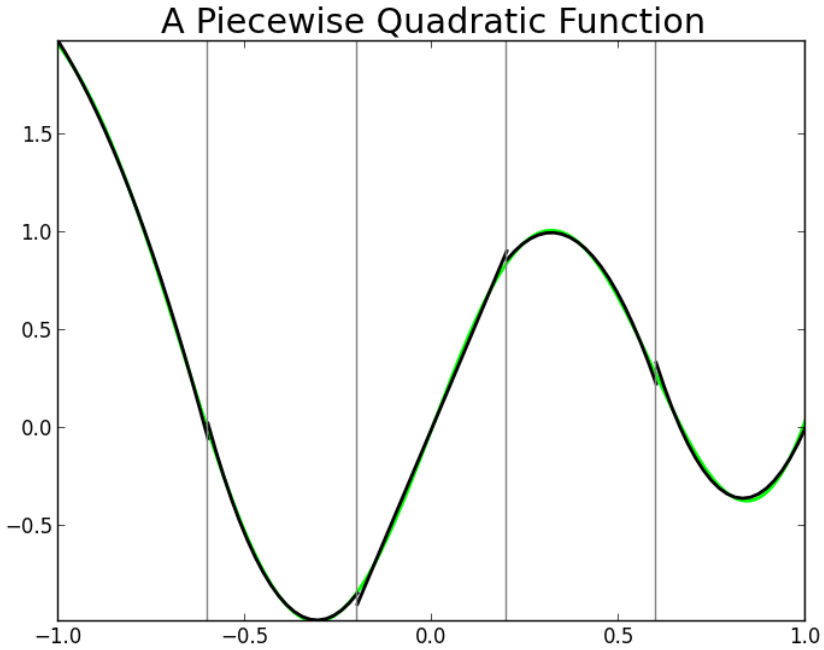
$$\int w_m(x) \left(\sum_{k=1}^{N} f'_k w_k(x) - G[f_h] \right) dx = 0$$

for all $m = 1, \ldots, N$.

Key feature of Discontinuous Galerkin: basis functions are non-zero only in non-overlapping cells, continuity at boundaries is not required (& thus is local like finite volume methods, unlike standard Finite Element (Continuous Galerkin) method).

Discontinuous Galerkin (DG) Combines Attractive Features of Finite-Volume & Finite Element Methods





Standard finite-volume methods evolve just the average value in each cell (piecewise constant), combined with interpolations.

DG evolves higher-order basis functions like finite-element methods, but, like finite-volume methods, doesn't force continuity at cell boundaries. --> (1) can use flux limiters like shock-capturing finite-volume methods (2) keeps the calculations local so one doesn't have to invert a global mass matrix, easier to parallelize.

Can Efficient Maxwellian-Weighted Basis Functions Be Implemented While Conserving Energy?

One of the reasons for the success of continuum gyrokinetic codes (GS2, GENE, GYRO) designed for the core region is their usage of efficient Gaussian integration techniques (high-order integration techniques that provide $n = 2p-1$ order accuracy with just p points). Perhaps more importantly, integrates a Maxwellian times polynomials v^n exactly even on a coarse velocity mesh.

Maxwellian-weighted basis functions in a standard DG algorithm would lose energy conservation (and even particle conservation)....

To allow Maxwellian-weighted basis functions, while still preserving important form of conservation laws, generalize inner-product norm to include a weighting function:

$$E_n = \int \mathcal{W}(v) \left[\sum_{k=1}^{N} f_k' w_k(v) - G[f_h] \right]^2 dv$$

Then the best fit leads to:

$$\int \mathcal{W}(v) w_m(v) \left(\sum_{k=1}^{N} f_k' w_k(v) - G[f_h] \right) dv = 0$$

If $w_m(v) = \exp(-v^2)v^m$, then by choosing $\mathcal{W}(v) \sim 1/\exp(-v^2)$, this corresponds to ensuring that the standard conservation laws are satisfied for v^0, v^1, v^2, ... (i.e., conservation of density, momentum, and energy, ...). (Conservation properties also depend on how the field equations are discretized.)

The above is a Galerkin method with a modified inner-product weight. Alternatively, can think of this as a "Petrov-Galerkin method", where the error is projected onto a different subspace than the basis functions.

Passive advection is a good prototype to study DG schemes

Consider the 1D passive advection equation on $I \in [L, R]$

$$\frac{\partial f}{\partial t} + \lambda \frac{\partial f}{\partial x} = 0$$

with λ the constant advection speed. $f(x,t) = f_0(x - \lambda t)$ is the exact solution, where $f_0(x)$ is the initial condition. Designing a good scheme is much harder than it looks.

- Discretize the domain into elements $I_j \in [x_{j-1/2}, x_{j+1/2}]$
- Pick a finite-dimensional function space to represent the solution. For DG we usually pick polynomials in each cell but allow discontinuities across cell boundaries
- Expand $f(x,t) \approx f_h(x,t) = \sum_k f_k(t) w_k(x)$.

Discrete problem can be stated as finding the coefficients that minimize the L_2 norm of the residual

The discrete problem in DG is stated as: find f_h in the function space such that for each basis function φ we have

$$\int_{I_j} \varphi \left(\frac{\partial f_h}{\partial t} + \lambda \frac{\partial f_h}{\partial x} \right) dx = 0.$$

Integrating by parts leads to the discrete *weak-form*

$$\int_{I_j} \varphi \frac{\partial f_h}{\partial t} dx + \lambda \varphi_{j+1/2} \hat{f}_{hj+1/2} - \lambda \varphi_{j-1/2} \hat{f}_{hj-1/2} - \int_{I_j} \frac{d\varphi}{dx} \lambda f_h \, dx = 0.$$

Here $\hat{f}_h = \hat{f}(f_h^+, f_h^-)$ is the consistent *numerical flux* on the cell boundary. Integrals are performed using high-order quadrature schemes.

Conditions for Hamiltonian Conservation Discovered Relatively Recently.

Unlike fluid equations which directly express conservation laws for particles, momentum, and energy, and so are preserved by finite-volume methods (modulo field equation issues), the conservation laws for Vlasov problems are indirect.

For problems of the form $\partial f/\partial t = \{H, f\}$, like the Vlasov / Gyrokinetic equations, can multiply by H or by f integrate over all space to get conservation laws for energy or for an entropy/enstrophy-like quantity.

Liu and Shu (2000) showed that a discrete energy is conserved if the basis functions for H are a continuous subset of the basic functions for f, even if an upwind flux is used. (Not widely appreciated, next paper was Bernsen et al. JCP 2006. Actually, these two papers were only for 2D incompressible hydro and don't point out the generalization to Hamiltonian systems.) We later discovered an extension to allow discontinuous potential ϕ (this could be useful for gyrokinetics to preserve the nature of the gyrokinetic Poisson equation, as a set of uncoupled 2-D elliptic equations).

Entropy is conserved with DG if central fluxes are used (and thus is like a high-order generalization of the well-known Arakawa scheme). However, want to use upwind / flux-limiters to avoid negative / unphysical overshoots in the solution, models collisions at unresolved scales that cause entropy to increase, while conserving energy.

Momentum conservation subtleties

Conservation properties also depends on how the field equations are solved. We have found ways for the discretized equations to exactly conserve energy or momentum, but have not found way to conserve both simultaneously.

Plan to use the energy conserving algorithm (at least for now). Although momentum conservation is not exact, it does converge as the spatial grid is refined, even if the velocity grid is coarse.

Might eventually solve a separate momentum conservation law (similar to Parra and Catto's suggestion), and could apply small corrections (of order the truncation errors) to the kinetic solution each time step to remove components of the error that are in momentum.

Separate issue: Parra and Catto are correct that accurate calculation of momentum transport in a low-flow gyro-Bohm regime requires extreme accuracy, $H_3 \sim (\rho/L)^3$ T. Discussed more in a recent technical report J.A. Krommes and G.W. Hammett (2013), PPPL-4945.

Our code here will focus first on edge / pedestal regimes that might break this ordering, where H_2 may be sufficient.

Initial Results: Full-F Gyrokinetic Continuum Code

PPPL work initiated by LDRD funds (January 2012), built on by MPPC funds this year.

Good progress:

Carried out a range of well-documented 2D tests:
http://www.ammar-hakim.org/sj/ (starting w/ JE12)

Code now able to handle 1x+2v, i.e. (z, $v_{||}$, $v_{\perp}$), Lenard-Bernstein model collision operator, electrostatic kinetic electrons and ions, 1D sheath boundary conditions. (a little more work to integrate it all together.)

Focussing on demonstrating full 5D capability of gyrokinetics (x, y, z, $v_{||}$, $v_{\perp}$) quickly (by end of 2014), with some simplifications at first (atomic physics, magnetic geometry).

Prototype code named Gkeyll is being developed

- Gkeyll is written in C++ and is inspired by framework efforts like Facets, VORPAL (Tech-X Corporation) and WarpX (U. Washington). Uses structured grids with arbitrary dimension/order nodal basis functions.
- Package management and builds are automated via `scimake` and `bilder`, both developed at Tech-X Corporation.
- Linear solvers from Petsc[1] are used for inverting stiffness matrices.
- Programming language Lua[2], used in widely played games like World of Warcraft, is used as an embedded scripting language to drive simulations.
- MPI is used for parallelization via the `txbase` library developed at Tech-X Corporation.

[1] http://www.mcs.anl.gov/petsc/
[2] http://www.lua.org

Good Progress To Date:

- 2012, 1st half (LDRD began Jan. 9, 2012):
  - successful benchmark on 2-D vortex problems, verified excellent conservation properties of DG algorithm for Poisson-bracket Hamiltonian problems.
  - Invented extensions to allow discontinuous potentials and preserve separability of independent 2D gyrokinetic Poisson solvers (will lead to paper).
- 2012, 2cd half:
  - extended tests to 1x/1v Vlasov Eq., linear and nonlinear Landau damping.
  - implemented Lenard-Bernstein collision operator, good model because it has essential features of full collision operator: conservation properties and preferential diffusive smoothing of small scales.
  - implemented Hasewaga-Wakatani 2D 2-field drift-wave eqs., collaboration with IFS (F. Waelbroeck et al.)
- 2013, 1st half:
  - discovered and fixed surprising problem with widely-used DG algorithm for diffusion. will lead to nice algorithm paper (draft written).
  - extended collision operator to 2D in velocity: ($v_{||}$, $v_{\perp}$)
  - now have 3D capability (z, $v_{||}$, $v_{\perp}$).
  - Good documentation of progress: http://www.ammar-hakim.org/sj/ (starting w/ JE12)

Plans: Continue Main Code Development

- rest of FY13:
  - Add options for sheath boundary conditions & sources for model of scrape-off layer. Simulations of ELM propagation along SOL, compare:
    - "Comparison of fluid and kinetic models of target energy fluxes during edge localized modes", Havlickova, Fundamenski, et al, PPCF (2012)
    - (done, with grad student Eric Shi)
    - extending to include magnetic fluctuations
  - Exploring best approach to implement Maxwellian-weighted basis functions (already in GENE & other core δf codes, not in any edge full-F codes)
- Winter 2013-14:
  - Upgrade to 4-D (2-space + 2-v), in simple axisymmetric geometry on open or closed field lines.
- 2014:
  - complete prototype 5D gyrokinetic code, with simple magnetic geometries (axisymmetric, open or closed field lines) for now, to demonstrate feasibility.
  - initial comparisons with LAPD / VINETA
- Then:
  - extend to more complicated geometries, better atomic physics models, ...

EXTRA SLIDES

A good strategic fit for the MPPC

Interactions Between Fusion, Basic Plasma Physics, and Astrophysics:

- Early application (by end of 2014) to basic laboratory experiments LAPD (UCLA) & Vineta (Greiswald): for fundamental studies of drift-wave turbulence & of nonlinear Alfven dynamics, astrophysics applications

- Continuum gyrokinetic codes very successful in the core region of tokamaks, can these algorithms be useful in tokamak edge and on non-fusion problems?

Max-Planck / Princeton Interactions:

- Close benchmarking & expertise sharing w/ core fusion gyrokinetic code GENE (developed at Garching)

- Compare with data from Vineta and later Asdex (Garching). Asdex group has extensive experience on fundamental edge measurements and modelling.

Dream: a robust code applicable for a wider range of fusion and non-fusion problems, capable of relatively fast simulations at low velocity resolution but with qualitatively-good results, or fully converged high velocity resolution w/ massive computing.

How To Apply DG to Diffusion Problems.

Consider calculating

$$g(x) = f_{xx}(x),$$

given a known $f(x)$. The "local DG" algorithm replaces this with two first order equations

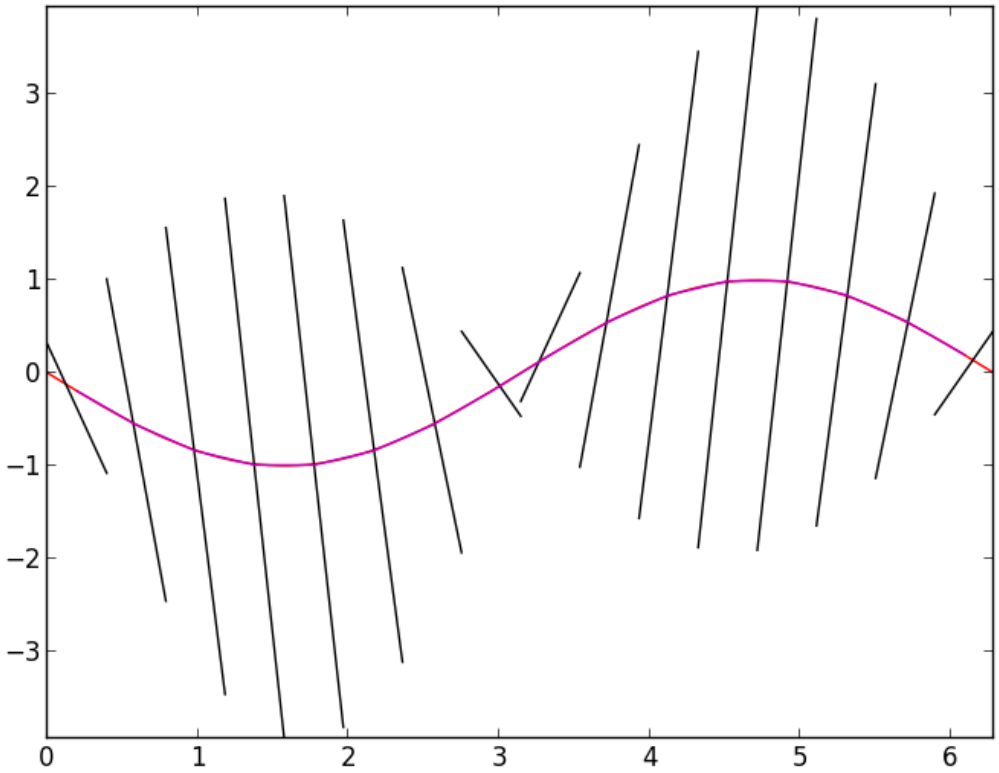
$$g(x) = dw/dx$$

$$w(x) = df/dx$$

and solving each equation with a DG method using upwind fluxes based on analogies with first-order advection equations.

However, there are problems with this approach, and we find it is better to reconstruct a smooth function that interpolates the DG moments in adjacent cells, an extension of how diffusion is treated in finite volume methods, where one only knows the cell average in each cell.

Discovered surprising problem in widely used DG algorithm for diffusion, and a fix.



$d^2 \sin(x) / dx^2$

computed with Local DG (LDG) algorithm (black curve) compared with exact answer (magenta).

The cell averages are correct, but slope within each cell is wrong & diverges as grid is refined.

We discovered this problem and a fix to it (similar to recent work by van Leer).

Have a draft of a very nice applied math paper on this.

This error was not noticed in most previous work in part because it it less severe at higher order or for elliptic problems where the Laplacian is inverted. More noticeable in transient diffusion problems.