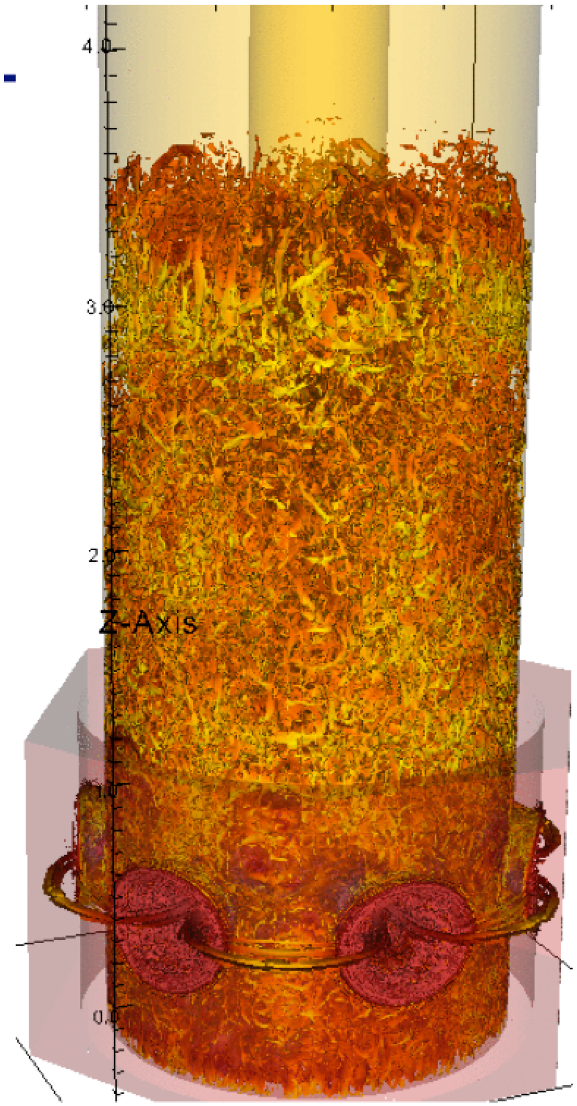
Nek5000 and Spectral Element Tutorial

Paul Fischer
Mathematics and Computer Science Division
Argonne National Laboratory

Joint work with:

Christos Frouzakis	*ETHZ*
Stefan Kerkemeier	*ETHZ / ANL*
Katie Heisey	*ANL*
Frank Loth	*U. Akron*
James Lottes	*ANL / Oxford*
Elia Merzari	*ANL*
Aleks Obabko	*ANL*
Tamay Ozgokmen	*U. Miami*
David Pointer	*ANL*
Philipp Schlatter	*KTH*
Ananias Tomboulides	*U. Thessaloniki*

and many others…



Turbulence in an industrial inlet.

Overview

0. Background

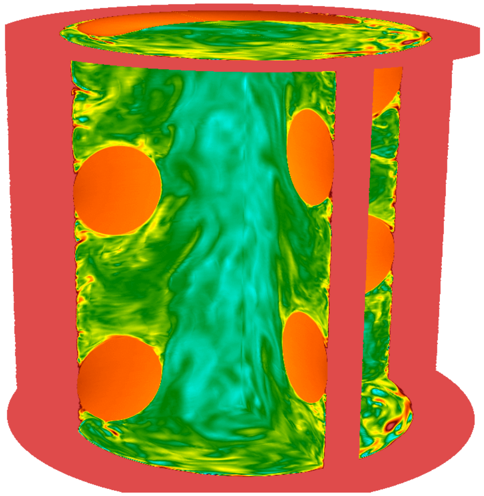
I. Scalable simulations of turbulent flows

- Discretization
- Solvers
- Parallel Implementation

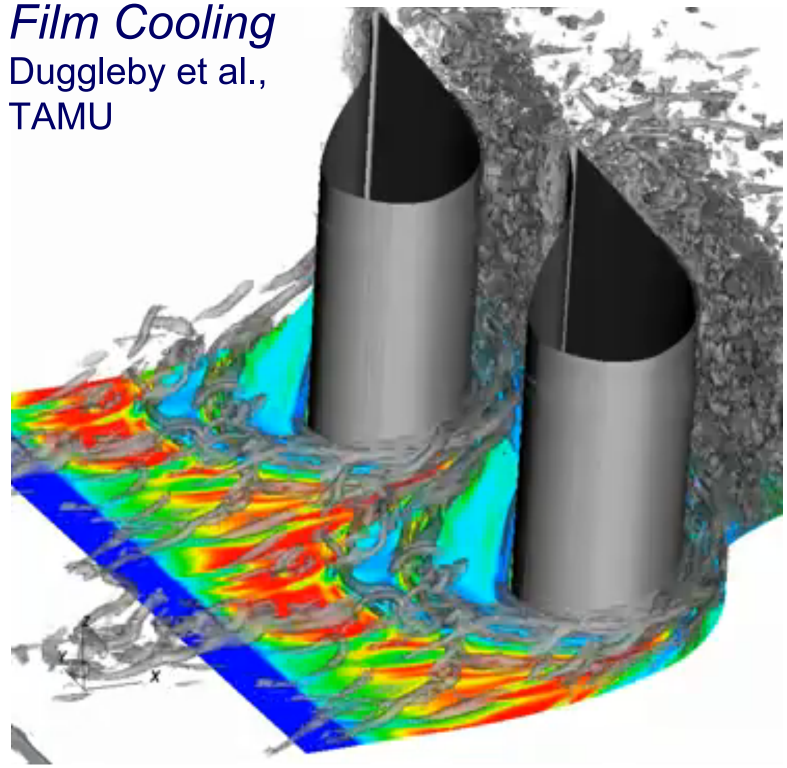
II. A quick demo…

Recent SEM-Based Turbulence Simulations

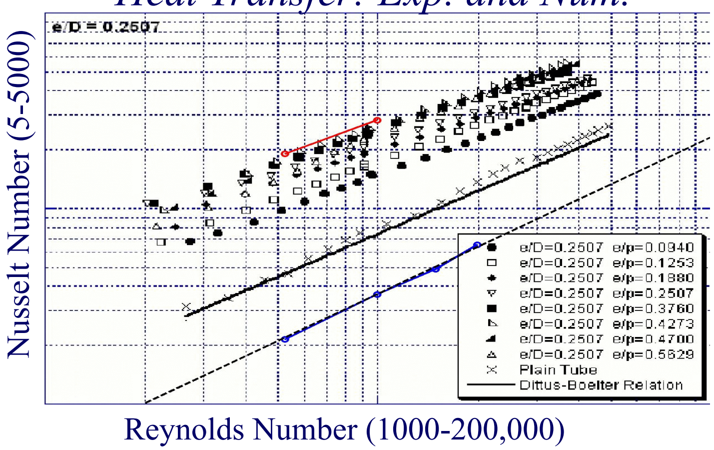
Enhanced Heat Transfer with Wire-Coil Inserts w/ J. Collins, ANL

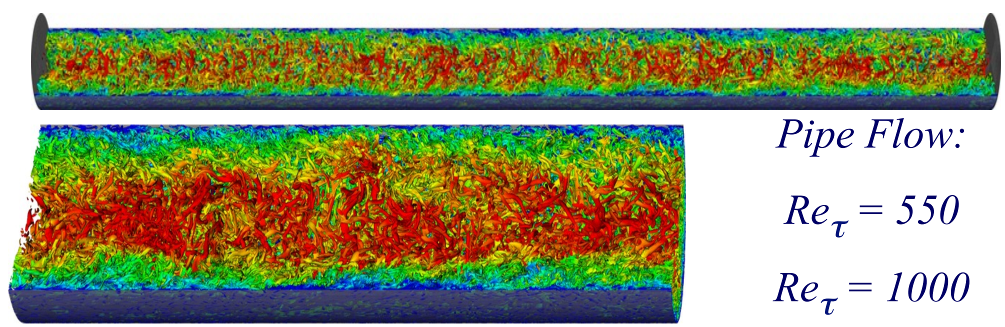


Film Cooling
Duggleby et al., TAMU



Heat Transfer: Exp. and Num.





Pipe Flow:

$Re_\tau = 550$

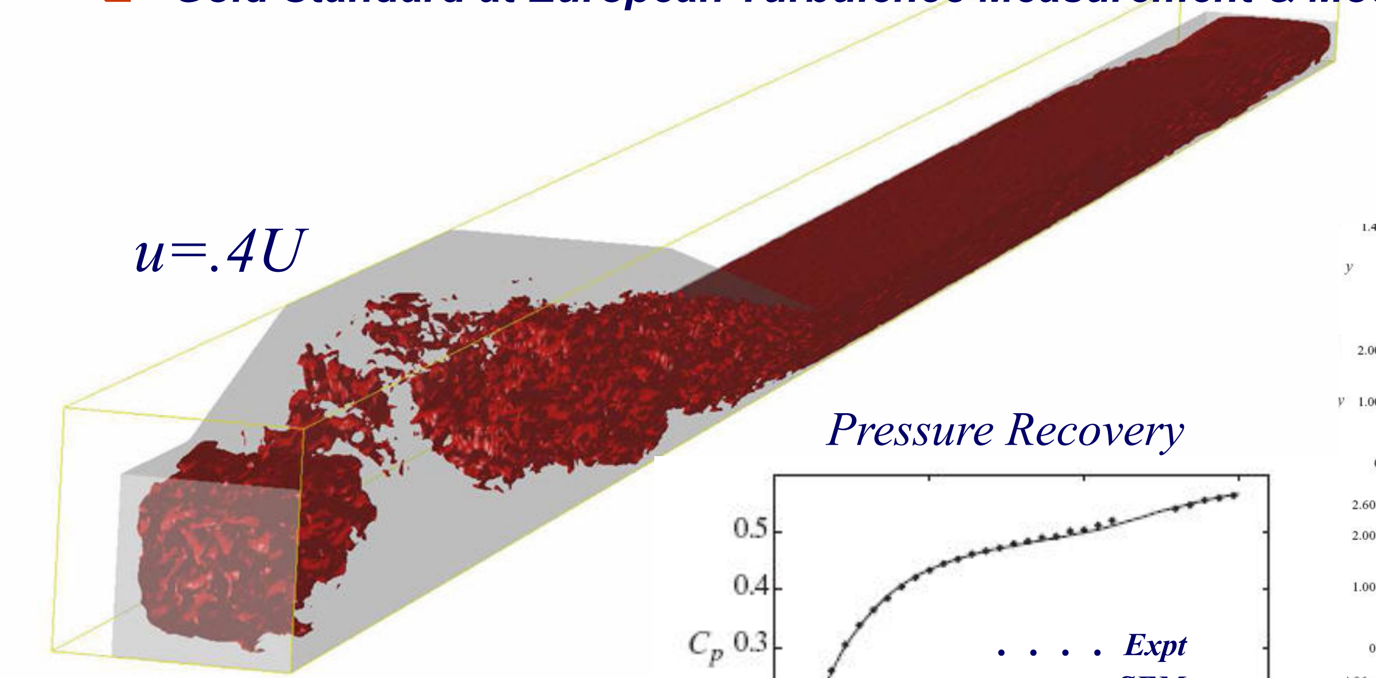
$Re_\tau = 1000$

G. El Khoury, KTH

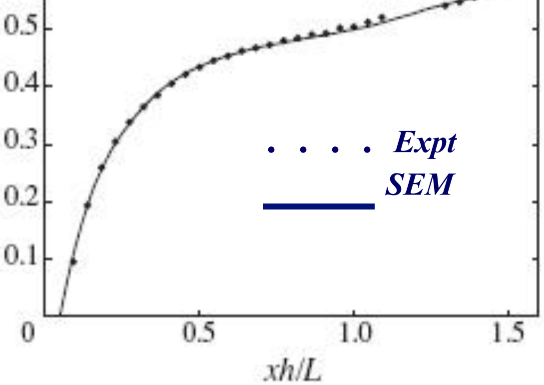
Validation: Separation in an Asymmetric Diffuser

Johan Ohlsson*, KTH

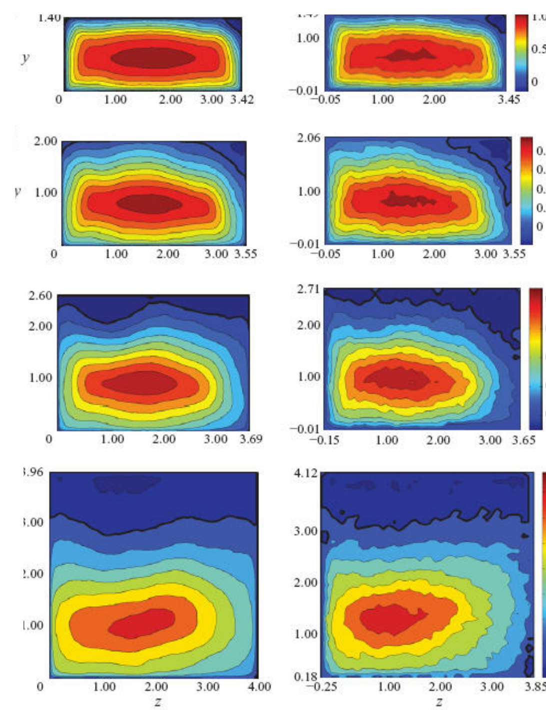
- Challenging high-Re case with ***flow separation and recovery***
- DNS at Re=10,000: E=127750, N=11, 100 convective time units
- Comparison with experimental results of Cherry et al.
- ***Gold Standard at European Turbulence Measurement & Modeling '09.***



Pressure Recovery



Axial Velocity



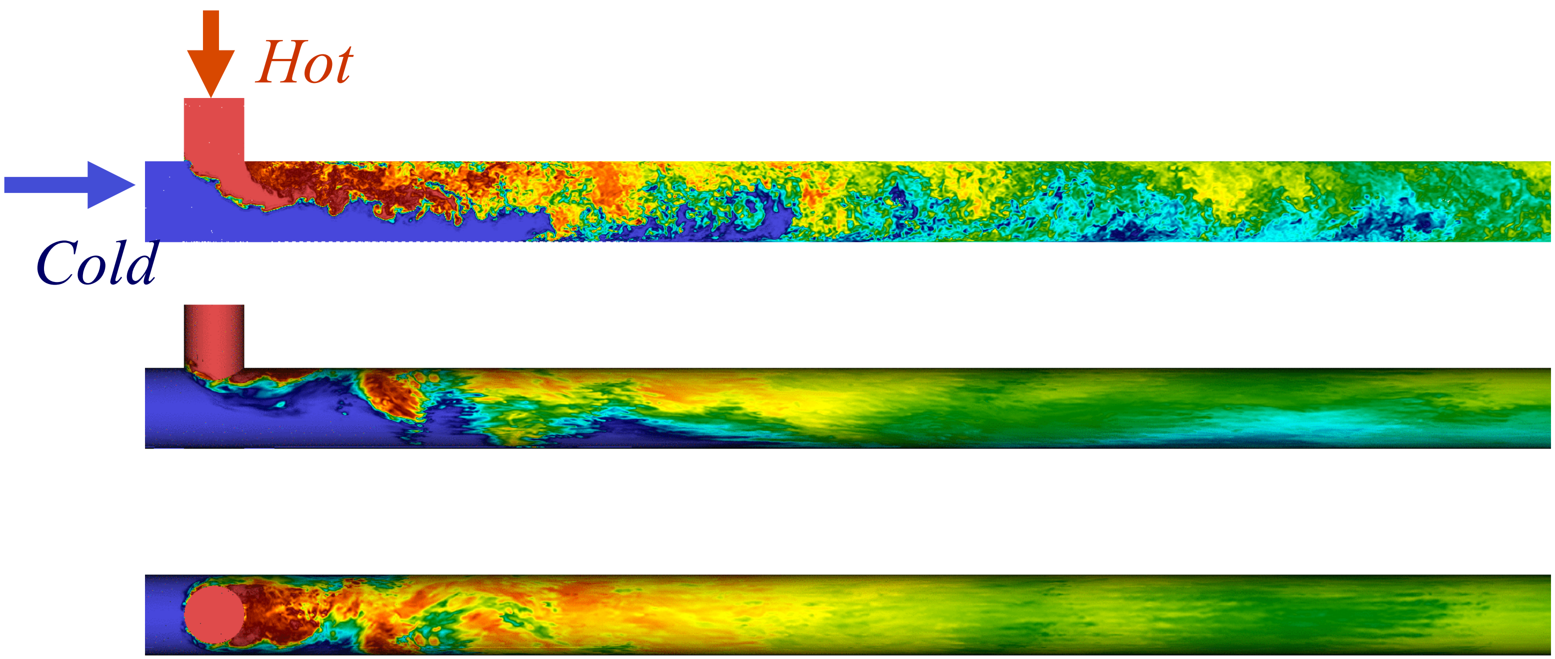
SEM   Expt.

*J. Fluid Mech., **650** (2010)

OECD/NEA T-Junction Benchmark

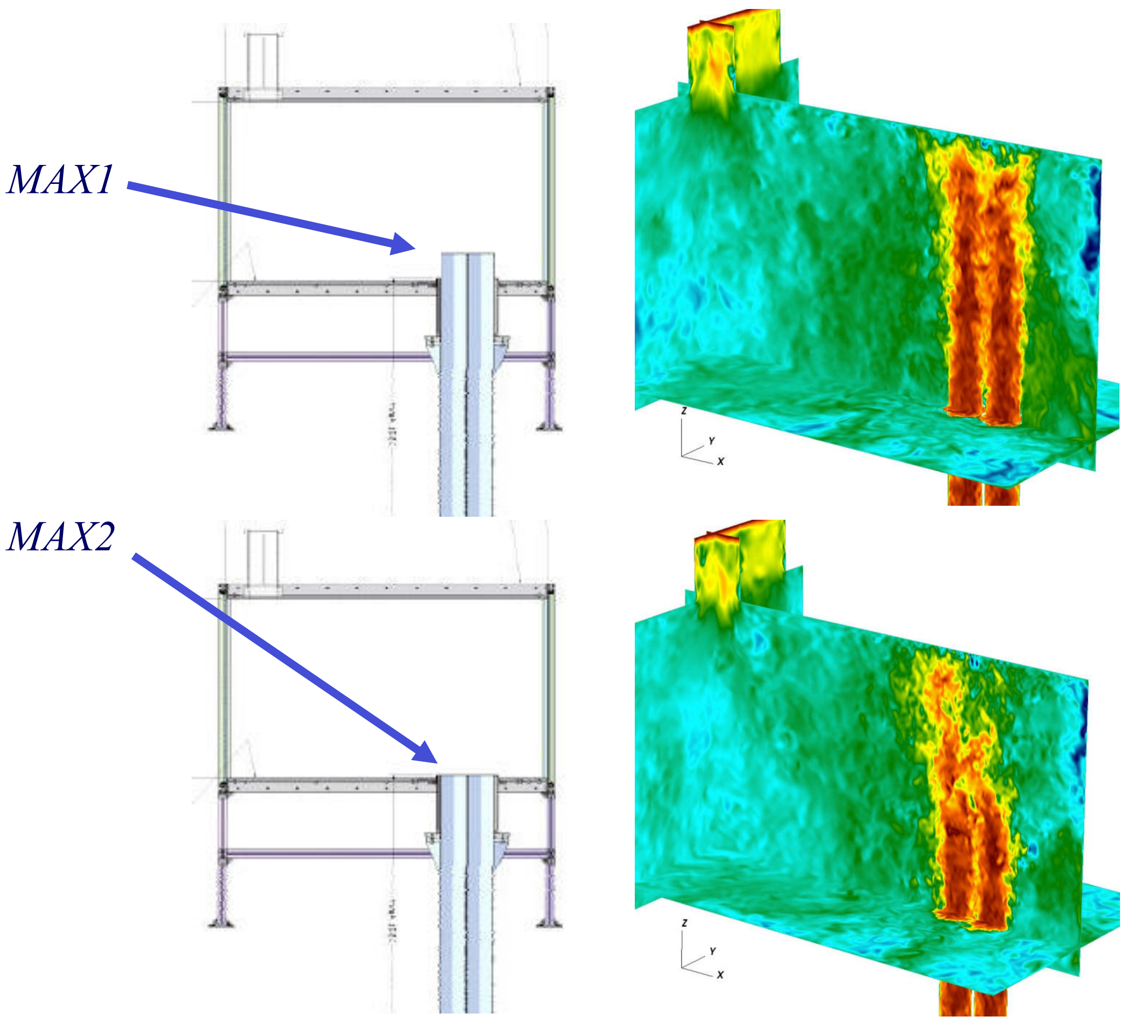
F., Obabko, Tautges, Caceres

- *E=62000 spectral elements of order N=7 (n=21 million)*
  - *Mesh generated with CUBIT*
- *Subgrid dissipation modeled with low-pass spectral filter*
- *1 Run: 24 hours on 16384 processors of BG/P (850 MHz) ~ 33x slower than uRANS*
- *SEM ranked **#1** (of 29) in thermal prediction.*



Centerplane, side, and top views of temperature distribution

LES Predicts Major Difference in Jet Behavior for Minor Design Change



Results:

- *Small perturbation yields O(1) change in jet behavior*
- *Unstable jet, with low-frequency (20 – 30 s) oscillations*
- *Visualization shows change due to jet / cross-flow interaction*
- ***MAX2 results NOT predicted by RANS***

Nek5000: Scalable Open Source Spectral Element Code

- Developed at MIT in mid-80s (Patera, F., Ho, Ronquist)
- Spectral Element Discretization: High accuracy at low cost
- Tailored to LES and DNS of turbulent heat transfer, but also supports
  - Low-Mach combustion, MHD, conjugate heat transfer, moving meshes
  - New features in progress: compressible flow (Duggleby), adjoints, immersed boundaries (KTH)
- Scaling: 1999 Gordon Bell Prize; Scales to over a million MPI processes.
- Current Verification and validation:

  > 900 tests performed after each code update

  > 200 publications based on Nek5000

  > 175 users since going open source in 2009

  > …

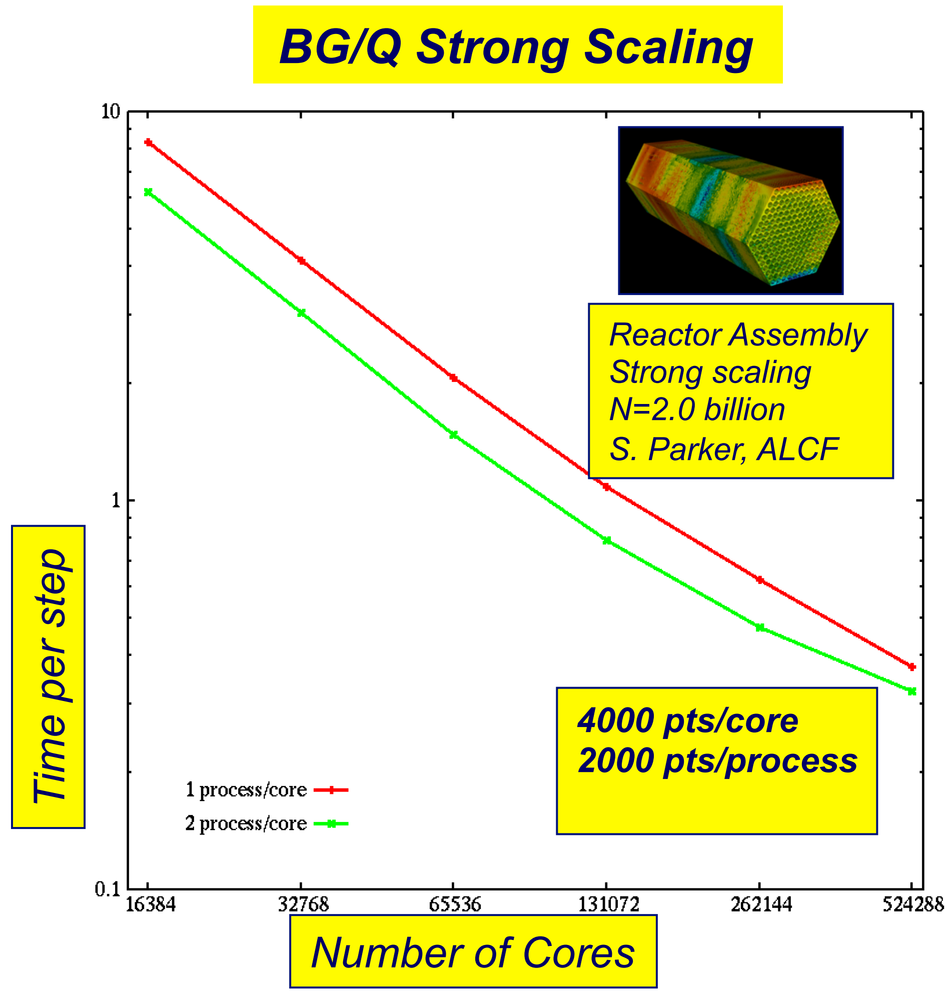
Scaling to a Million Processes

w / Scott Parker, ALCF

217 Pin Problem, N=9, E=3e6:

- *2 billion points*
- *BGQ – 524288 cores*
  - *1 or 2 ranks per core*
- *60% parallel efficiency at 1 million processes*
- ***2000 points/process***

  → Reduced time to solution for a broad range of problems



Influence of Scaling on Discretization

Influence of Scaling on Discretization

Large problem sizes enabled by peta- and exascale computers allow propagation of small features (size λ) over distances L >> λ. If speed ~ 1, then t_{final} ~ L/ λ.

- Dispersion errors accumulate linearly with time:

  *~|correct speed – numerical speed| * t* (for each wavenumber)

  → $error_{t\_final}$ *~ (L / λ) * | numerical dispersion error |*

- For fixed final error ε_f, require: numerical dispersion error ~ $(\lambda / L)\varepsilon_f$, << 1.

Influence of Scaling on Discretization

Large problem sizes enabled by peta- and exascale computers allow propagation of small features (size λ) over distances L >> λ. If speed ~ 1, then t_{final} ~ L/ λ.

- Dispersion errors accumulate linearly with time:

  *~|correct speed – numerical speed| * t* (for each wavenumber)

  → $error_{t\_final}$ *~ (L / λ) * | numerical dispersion error |*

- For fixed final error ε_f, require: numerical dispersion error ~ $(\lambda / L)\varepsilon_f$, << 1.

High-order methods can efficiently deliver small dispersion errors.

(Kreiss & Oliger 72, Gottlieb et al. 2007)

Our objective is to realize the advantage of high-order methods, at low-order costs.

Motivation for High-Order

High-order accuracy is uninteresting unless

- Cost per gridpoint is comparable to low-order methods
- You are interested in simulating interactions over a broad range of scales…

Precisely the type of inquiry enabled by HPC and leadership class computing facilities.

Incompressible Navier-Stokes Equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}$$

$$\nabla \cdot \mathbf{u} = 0$$

- Key algorithmic / architectural issues:
  - Unsteady evolution implies many timesteps, significant reuse of preconditioners, data partitioning, etc.
  - Div u = 0 implies long-range global coupling at each timestep
    → iterative solvers
    communication intensive
    opportunity to amortize adaptive meshing, etc.
  - Small dissipation → large number of scales → large number of gridpoints for high Reynolds number *Re*

Navier-Stokes Time Advancement

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}$$

$$\nabla \cdot \mathbf{u} = 0$$

- Nonlinear term: *explicit*
  - k th-order backward difference formula / extrapolation (k =2 or 3)
  - k th-order characteristics (Pironneau '82, MPR '90)
- Linear Stokes problem: pressure/viscous decoupling:
  - 3 Helmholtz solves for velocity *("easy" w/ Jacobi-precond.CG)*
  - (consistent) Poisson equation for pressure *(computationally dominant)*
- For LES, apply grid-scale spectral filter (F. & Mullen 01, Boyd '98)
  – in spirit of HPF model (Schlatter 04)

Timestepping Design

- ***Implicit:***
  - *symmetric and (generally) linear terms,*
  - *fixed flow rate conditions*

- ***Explicit:***
  - *nonlinear, nonsymmetric terms,*
  - *user-provided rhs terms, including*
    - Boussinesq and Coriolis forcing

- ***Rationale:***
  - *div **u** = 0 constraint is fastest timescale*
  - *Viscous terms: explicit treatment of 2nd-order derivatives* → $\Delta t \sim O(\Delta x^2)$
  - *Convective terms require only* $\Delta t \sim O(\Delta x)$
  - *For high Re, temporal-spatial accuracy dictates* $\Delta t \sim O(\Delta x)$
  - *Linear symmetric is "easy" – nonlinear nonsymmetric is "hard"*

BDF2/EXT2 Example

Consider the convection-diffusion equation,

$$\frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u \;=\; \nu \nabla^2 u.$$

Discretize in space:

$$B\frac{d\underline{u}}{dt} + C\underline{u} \;=\; -\nu A\underline{u}, \qquad (A \text{ is SPD})$$

Evaluate each term at t^n according to convenience:

$$B\frac{d\underline{u}}{dt}\bigg|_{t^n} \;=\; B\frac{3\underline{u}^n - 4\underline{u}^{n-1} + \underline{u}^{n-2}}{2\Delta t} + O(\Delta t^2)$$

$$C\underline{u}\bigg|_{t^n} \;=\; 2C\underline{u}^{n-1} - C\underline{u}^{n-2} + O(\Delta t^2)$$

$$\nu A\underline{u}\bigg|_{t^n} \;=\; \nu A\underline{u}^n$$

BDFk/EXTk

- BDF3/EXT3 is essentially the same as BDF2/EXT2
  - $O(\Delta t^3)$ accuracy
  - essentially same cost
  - accessed by setting Torder=3 (2 or 1) in .rea file

- For convection-diffusion and Navier-Stokes, the "*EXTk*" part of the timestepper implies a CFL (Courant-Friedrichs-Lewy) constraint

$$\max_{\mathbf{x}\in\Omega} \frac{|\mathbf{u}|\Delta t}{\Delta x} \approx 0.5$$

- For the spectral element method, $\Delta x \sim N^{-2}$, which is restrictive.
  - We therefore often use a characteristics-based timestepper. (IFCHAR = T in the .rea file)

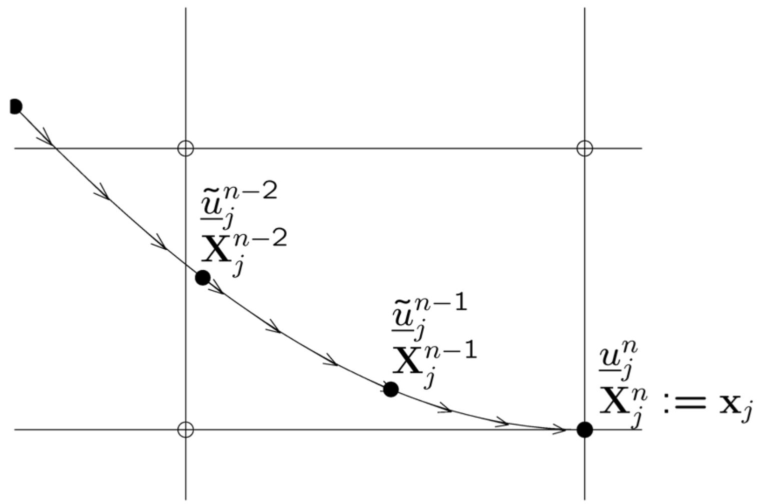
Characteristics Timestepping

- Apply BDFk to material derivative, e.g., for k=2:

$$\frac{Du}{Dt} := \frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u$$

$$= \frac{3u^n - 4\tilde{u}^{n-1} + \tilde{u}^{n-2}}{2\Delta t} + O(\Delta t^2)$$

- Amounts to finite-differencing along the characteristic leading into x_j



Characteristics Timestepping

- *Δt can be >> Δt_{CFL}* *(e.g., Δt ~ 5-10 x Δt_{CFL})*

- *Don't need position (e.g., X_j^{n-1}) of characteristic departure point, only the value of $u^{n-1}(x)$ at these points.*

These values satisfy the pure hyperbolic problem:

$$\begin{aligned} \frac{\partial \tilde{u}}{\partial s} + \mathbf{c} \cdot \nabla \tilde{u} &= 0, \qquad s \in [t^{n-1}, t^n] \\ \tilde{u}(\mathbf{x}, t^{n-1}) &:= u^{n-1}(\mathbf{x}), \end{aligned}$$

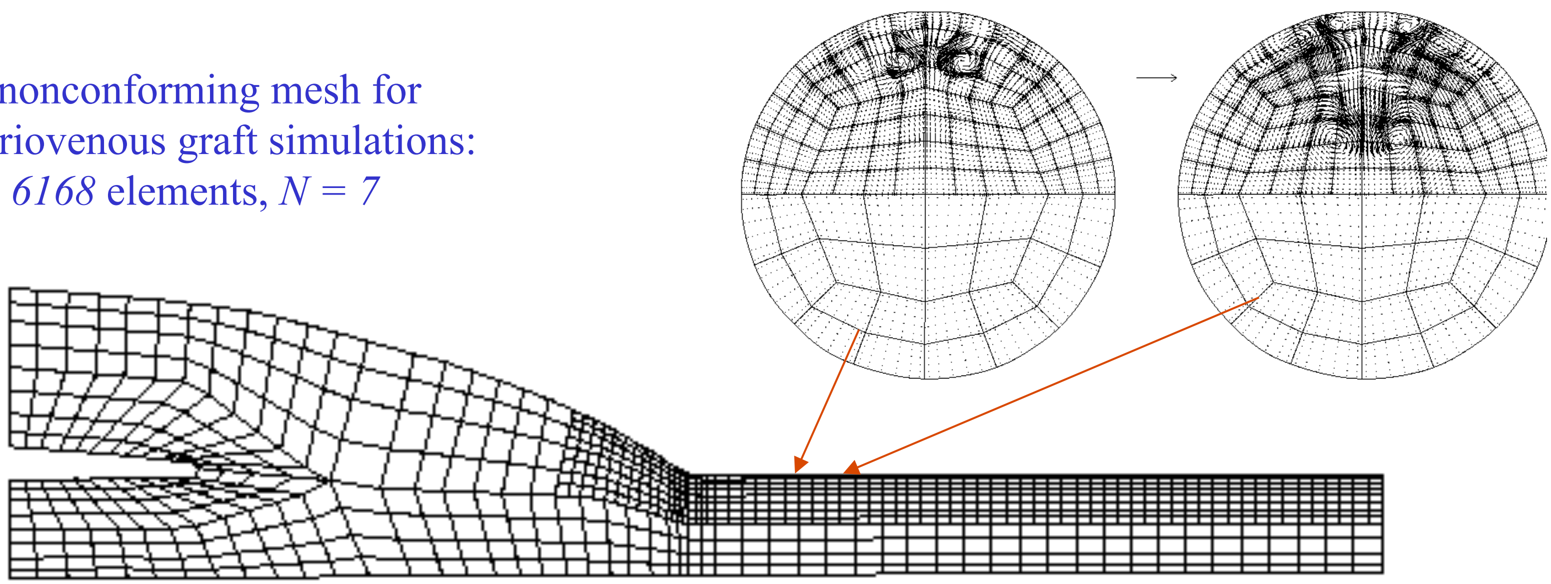
which is solved via explicit timestepping with Δs ~ Δt_{CFL}

Spatial Discretization: *Spectral Element Method*

(Patera 84, Maday & Patera 89)

- Variational method, similar to FEM, using *GL* quadrature.
- Domain partitioned into E high-order quadrilateral (or hexahedral) elements (decomposition may be nonconforming - *localized refinement*)
- Trial and test functions represented as Nth-order tensor-product polynomials within each element. (N ~ 4 -- 15, typ.)
- EN^3 gridpoints in 3D, EN^2 gridpoints in 2D.
- Converges *exponentially fast* with N for smooth solutions.

3D nonconforming mesh for arteriovenous graft simulations: $E = 6168$ elements, $N = 7$



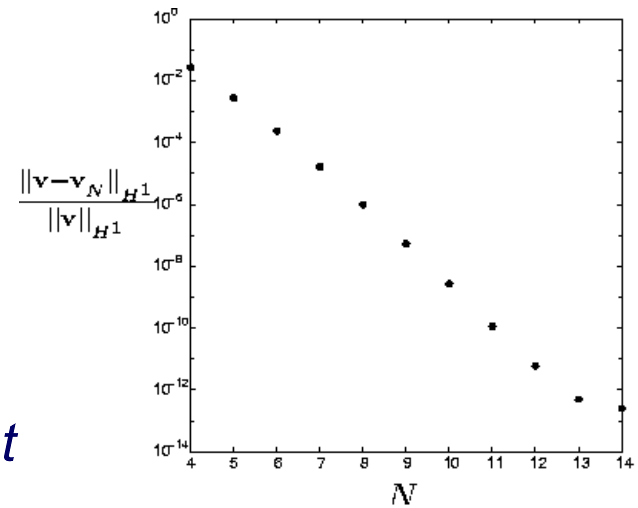
Spectral Element Convergence: Exponential with N

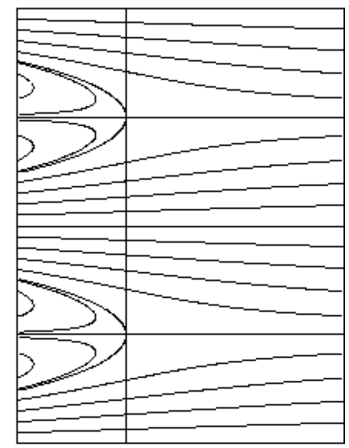
- *4 orders-of-magnitude error reduction when doubling the resolution in each direction*

- *Benefits realized through tight data-coupling.*

- *For a given error,*
  - *Reduced number of gridpoints*
  - *Reduced memory footprint.*
  - *Reduced data movement.*

Exact Navier-Stokes Solution (Kovazsnay '48)





$$v_x = 1 - e^{\lambda x} \cos 2\pi y$$

$$v_y = \frac{\lambda}{2\pi} e^{\lambda x} \sin 2\pi y$$

$$\lambda := \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}$$

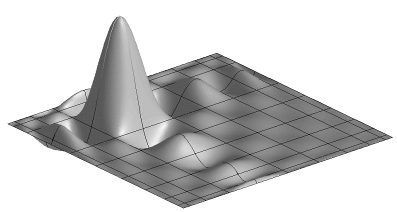
Spectral Element Discretization

$$u_t + \mathbf{c} \cdot \nabla u = \nu \nabla^2 u$$

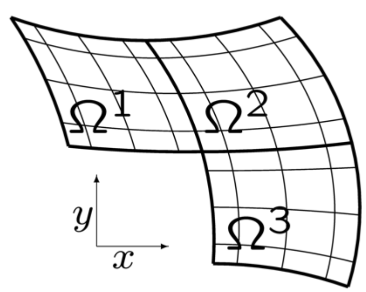
Find $u \in X_0^N \subset H_0^1$ *such that*

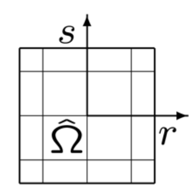
$$(v, u_t)_N + (v, \mathbf{c} \cdot \nabla u)_M = \nu (\nabla v, \nabla u)_N \quad \forall\, v \in X_0^N,$$

- $(f,g)_M := \sum_{j=0}^{M} \rho_j^M f(\xi_j^M) g(\xi_j^M)$, (1-D, $\Omega = [-1,1]$)
- ξ_j^M, ρ_j^M—Mth-order Gauss-Legendre points, weights.



2D basis function, N=10





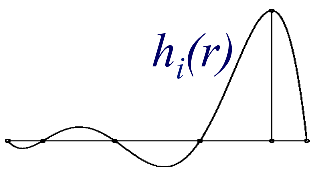
E=3, N=4

Spectral Element Basis Functions

- Tensor-product nodal basis:

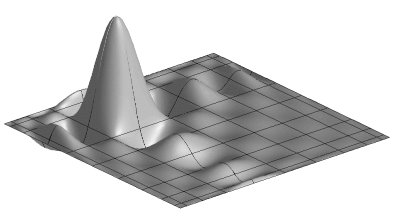
$$u(x,y)|_{\Omega^e} = \sum_{i=0}^{N} \sum_{j=0}^{N} u_{ij}^e \, h_i(r) \, h_j(s)$$

$$h_i(r) \in \mathcal{P}_N(r), \qquad h_i(\xi_j) = \delta_{ij}$$

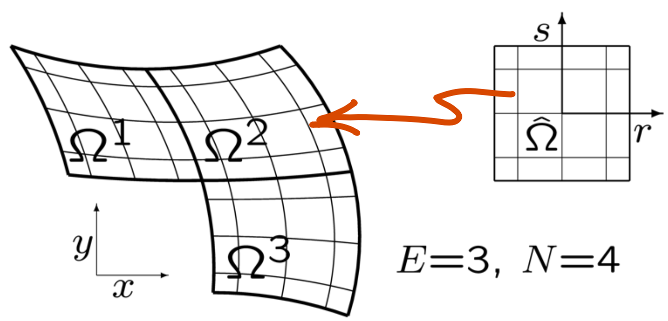


- ξ_j = Gauss-Lobatto-Legendre quadrature points:

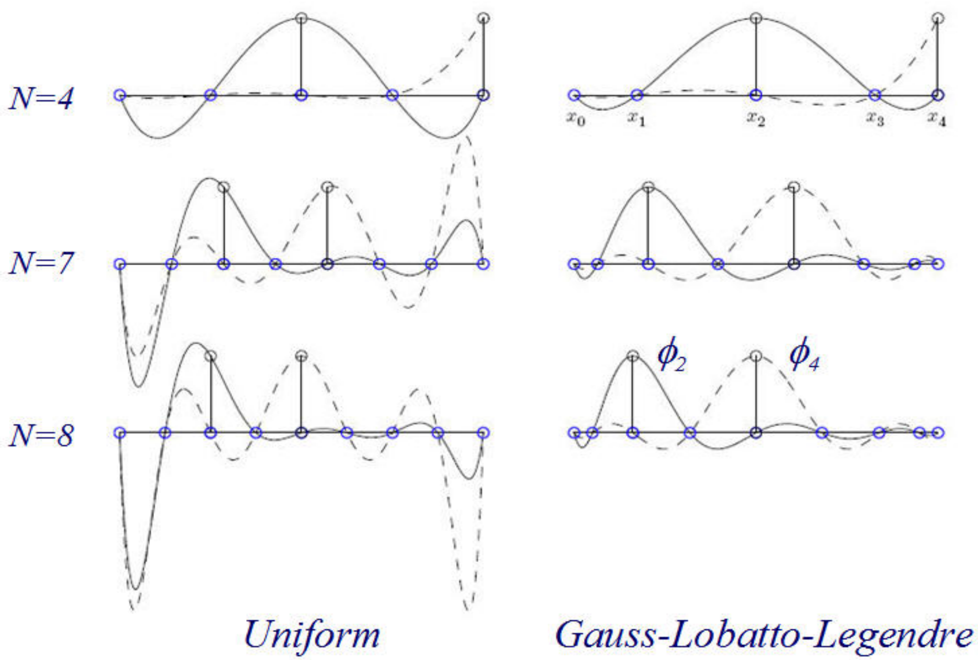
  - stability (*not* uniformly distributed points)
  - allows pointwise quadrature (for *most* operators…)
  - easy to implement BCs and C^0 continuity

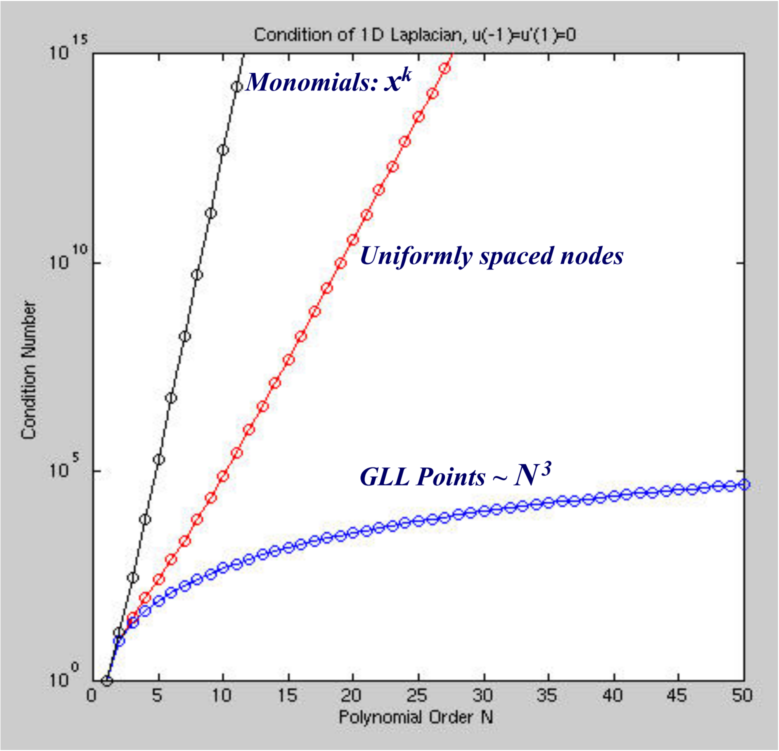


2D basis function, N=10



Influence of Basis on Conditioning





- *Monomials and Lagrange interpolants on uniform points exhibit exponentional growth in condition number.*
- *With just a 7x7 system the monomials would lose 10 significant digits (of 15, in 64-bit arithmetic).*

Attractive Feature of Tensor-Product Bases (quad/hex elements)

- *Local tensor-product form* (2D),

$$u(r,s) = \sum_{i=0}^{N} \sum_{j=0}^{N} u_{ij} h_i(r) h_j(s), \quad h_i(\xi_p) = \delta_{ip}, \; h_i \in \mathbb{P}_N$$

allows derivatives to be evaluated as ***fast*** *matrix-matrix products:*

$$\left.\frac{\partial u}{\partial r}\right|_{\xi_i,\xi_j} = \sum_{p=0}^{N} u_{pj} \left.\frac{dh_p}{dr}\right|_{\xi_i} = \sum_p \underbrace{\hat{D}_{ip} u_{pj}}_{mxm} =: D_r \underline{u}$$

Fast Operator Evaluation

Local matrix-free stiffness matrix in 3D on Ω^e,

$$A^e \underline{u}^e = \begin{pmatrix} D_r \\ D_s \\ D_t \end{pmatrix}^T \begin{pmatrix} G^e_{rr} & G^e_{rs} & G^e_{rt} \\ G^e_{rs} & G^e_{ss} & G^e_{st} \\ G^e_{rt} & G^e_{st} & G^e_{tt} \end{pmatrix} \begin{pmatrix} D_r \\ D_s \\ D_t \end{pmatrix} \underline{u}^e$$

Matrix free form :
$\cdot\ 7N^3$ memory ref′s.
$\cdot\ 12N^4 + 15N^3$ op′s.

$$D_r = (I \otimes I \otimes \hat{D}) \qquad G^e_{rs} = J^e \circ B \circ \left(\frac{\partial r}{\partial x}\frac{\partial s}{\partial x} + \frac{\partial r}{\partial y}\frac{\partial s}{\partial y} + \frac{\partial r}{\partial z}\frac{\partial s}{\partial z} \right)^e$$

- Operation count is only $O(N^4)$ not $O(N^6)$ *[Orszag '80]*
- Work is dominated by fast matrix-matrix products (D_r , D_s , D_t)
- Memory access is 7 x number of points
  – because of GLL quadrature, G_{rr} ,G_{rs}, etc., are *diagonal*

Spectral Filter

Boyd '98, F. & Mullen '01

- Expand in modal basis:

$$u(x) = \sum_{k=0}^{N} \hat{u}_k \, \phi_k(r)$$

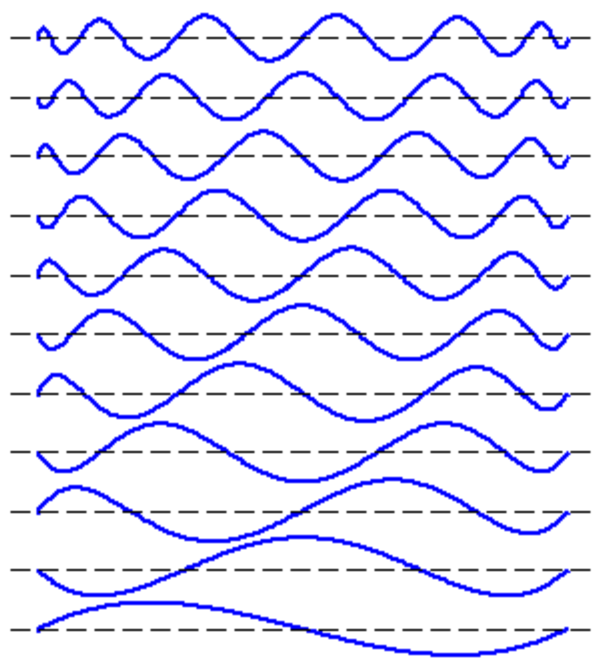
- Set filtered function to:

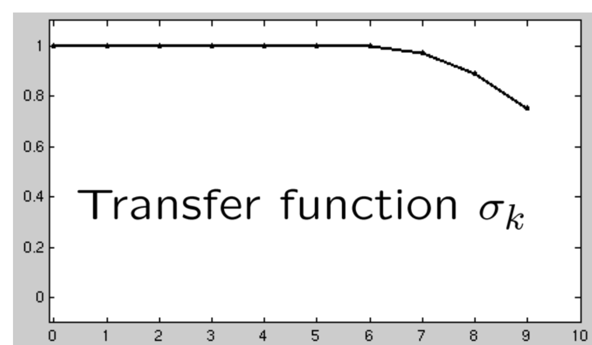
$$\bar{u}(x) = \hat{F}(u) = \sum_{k=0}^{N} \sigma_k \, \hat{u}_k \, \phi_k(r)$$

- Spectral convergence and continuity preserved. (Coefficients decay exponentially fast.)

- In higher space dimensions:

$$F = \hat{F} \otimes \hat{F} \otimes \hat{F}$$





Filtering Cures High Wavenumber Instabilities

- Free surface example:

Error in OS Growth Rate

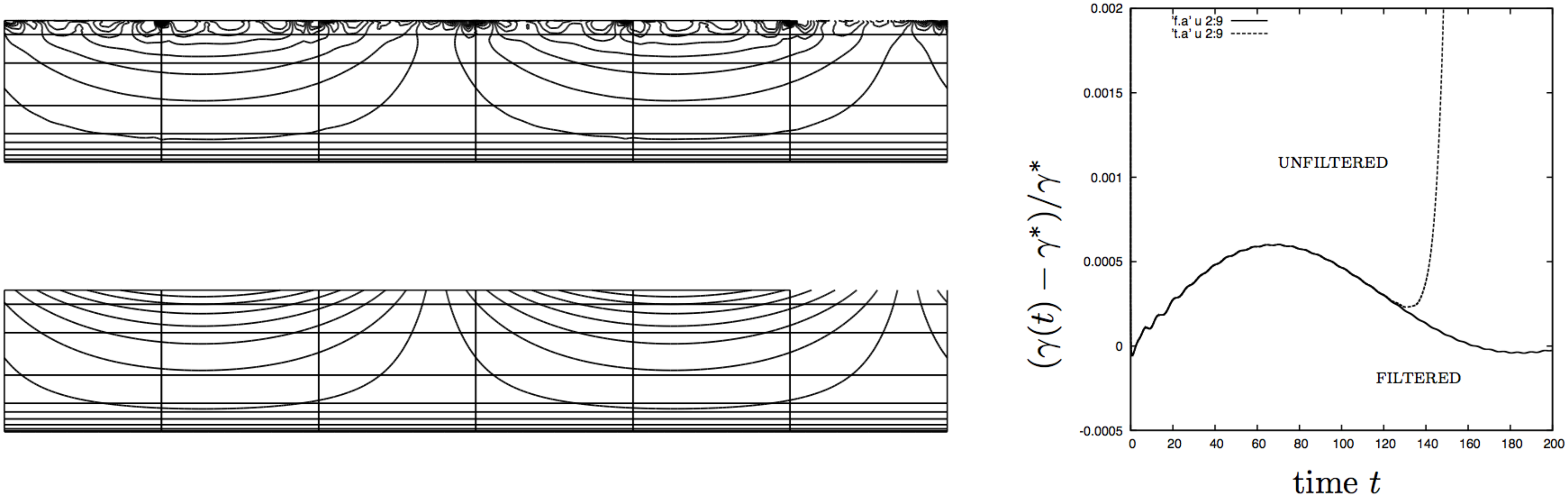


Figure 6: Eigenmodes for free-surface film flow: (left, top) contours of vertical velocity v for unfiltered and (left, bottom) filtered solution at time $t = 179.6$; (right) error in growth rate vs. t.

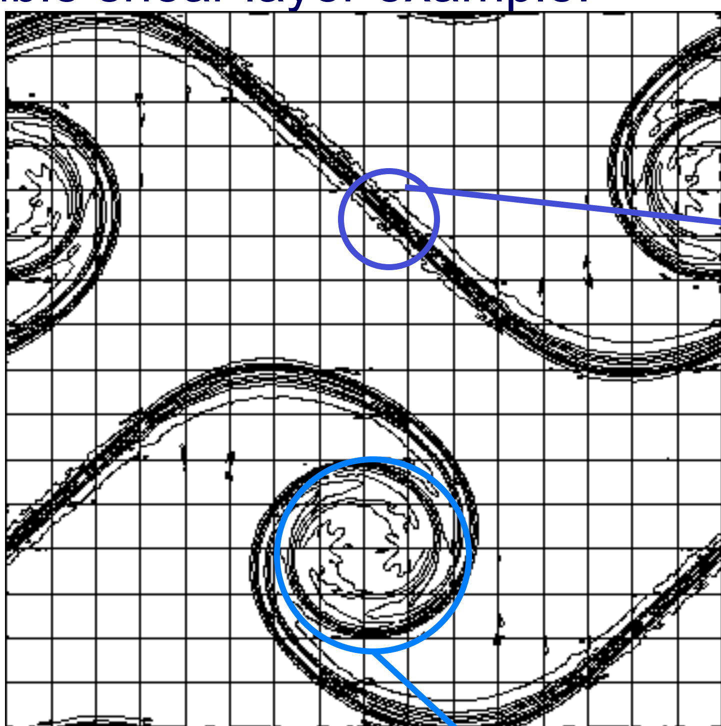
---

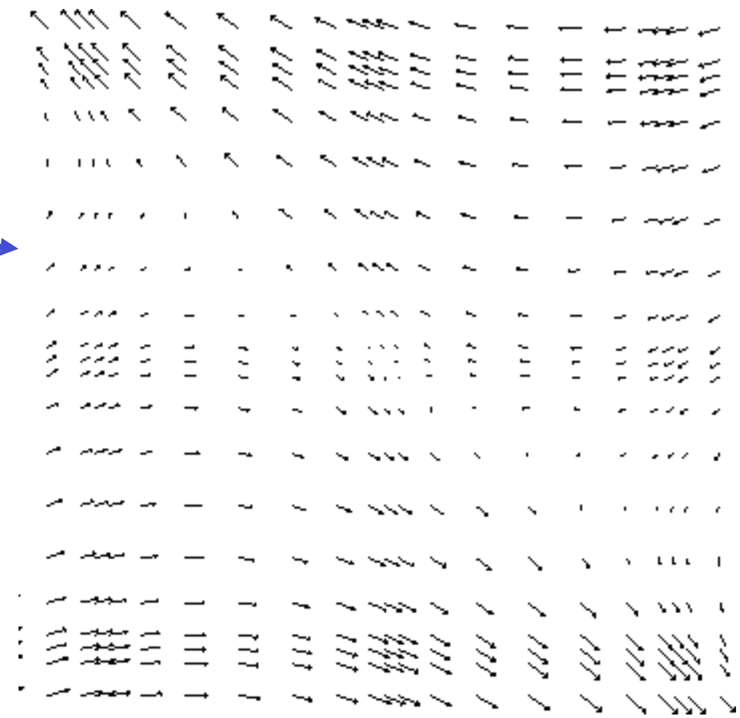
[10] Instabilities in free-surface Hartmann flow at low magnetic Prandtl numbers. Giannakis, D., Rosner, R., & Fischer, P.F. 2009, J. Fluid Mech., 636, 217-277

Dealiasing

When does straight quadrature fail ??

Double shear layer example:





OK

High-strain regions are troublesome…

When Does Quadrature Fail?

Consider the model problem: $$\frac{\partial u}{\partial t} = -\mathbf{c} \cdot \nabla u$$

Weighted residual formulation: $$B\frac{d\underline{u}}{dt} = -C\underline{u}$$

$$\begin{aligned}
B_{ij} &= \int_\Omega \phi_i \phi_j \, dV = \text{symm. pos. def.} \\
C_{ij} &= \int_\Omega \phi_i \, \mathbf{c} \cdot \nabla \phi_j \, dV \\
&= -\int_\Omega \phi_j \, \mathbf{c} \cdot \nabla \phi_i \, dV - \int_\Omega \phi_j \phi_j \nabla \cdot \mathbf{c} \, dV \\
&= \text{skew symmetric, if } \nabla \cdot \mathbf{c} \equiv 0. \\
B^{-1}C &\longrightarrow \text{imaginary eigenvalues}
\end{aligned}$$

Discrete problem should never blow up.

When Does Quadrature Fail?

Weighted residual formulation vs. spectral element method:

$$C_{ij} = (\phi_i, \mathbf{c} \cdot \nabla \phi_j) = -C_{ji}$$

$$\tilde{C}_{ij} = (\phi_i, \mathbf{c} \cdot \nabla \phi_j)_N \neq -\tilde{C}_{ji}$$

This suggests the use of over-integration (dealiasing) to ensure that skew-symmetry is retained

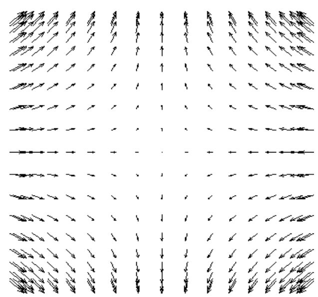
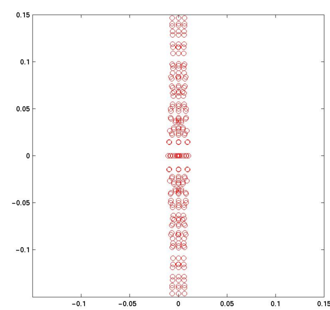
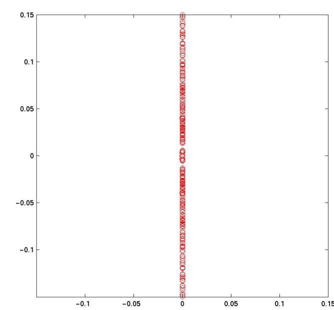
$$C_{ij} = (J\phi_i, (J\mathbf{c}) \cdot J\nabla \phi_j)_M$$

$$J_{pq} := h_q^N(\xi_p^M) \quad \text{interpolation matrix (1D, single element)}$$

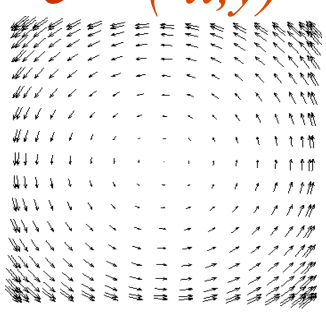
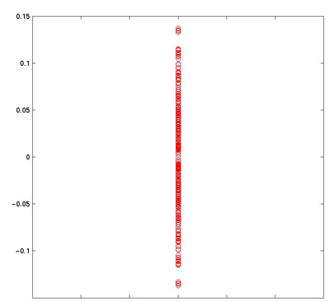
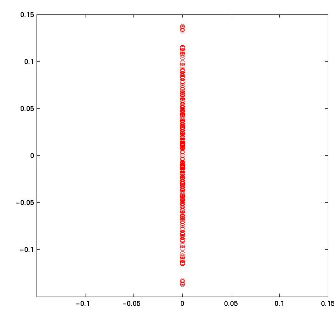
Aliased / Dealiased Eigenvalues: $u_t + \mathbf{c} \cdot \nabla u = 0$

- Velocity fields model first-order terms in expansion of straining and rotating flows.
  - Rotational case is skew-symmetric
  - Over-integration restores skew-symmetry (Malm et al, JSC 2013)

N=19, M=19 *N=19, M=20*

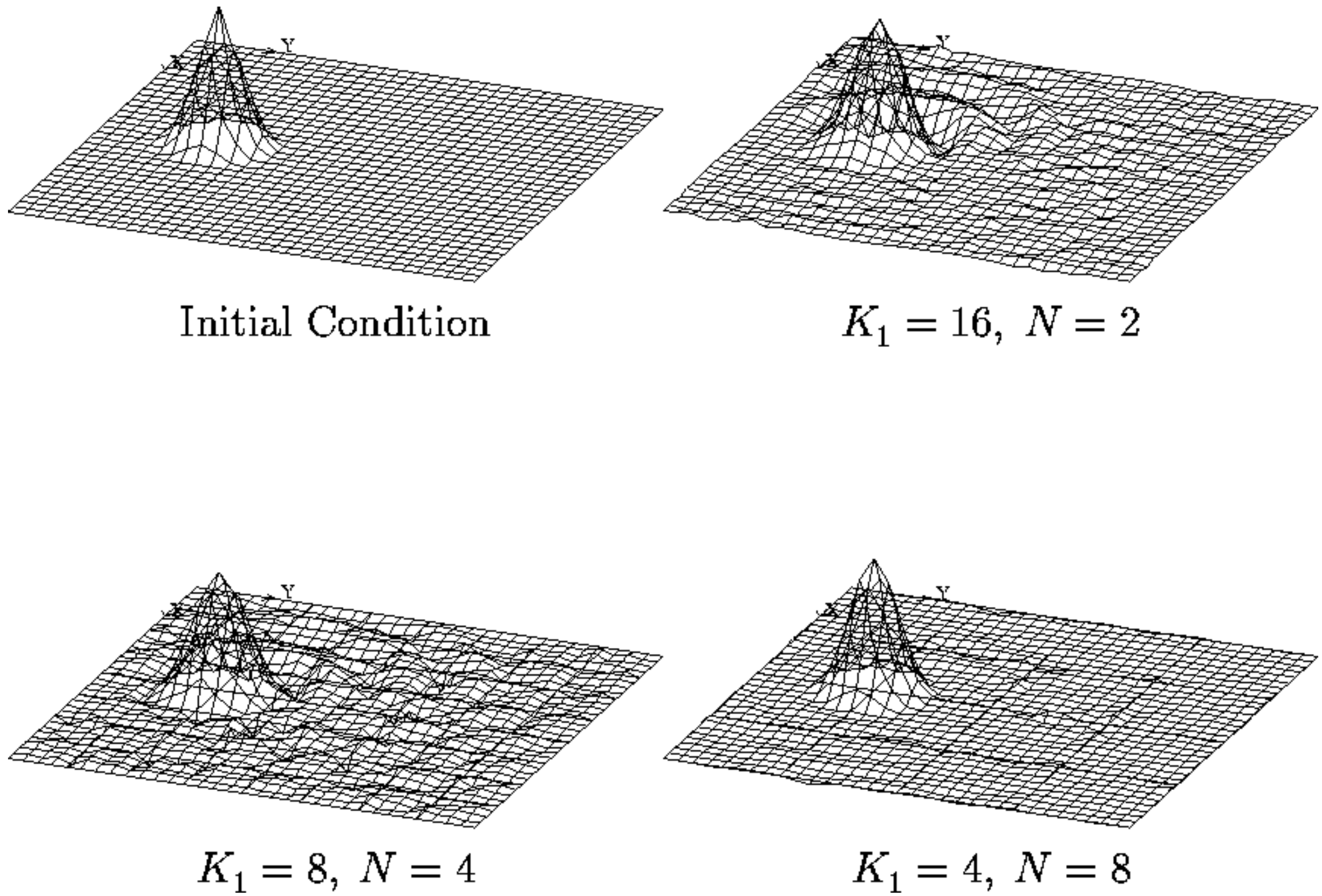




c = (-x,y)

c = (-y,x)

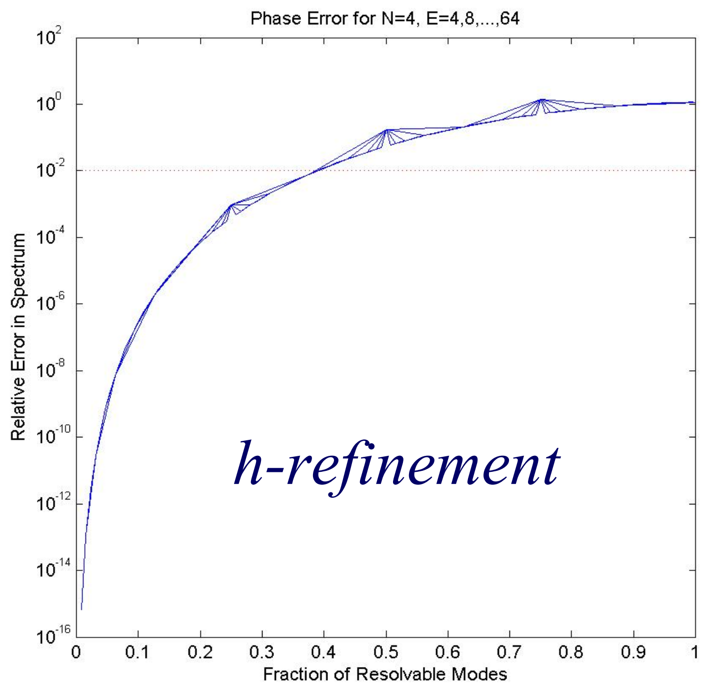
Excellent transport properties, even for *non-smooth* solutions

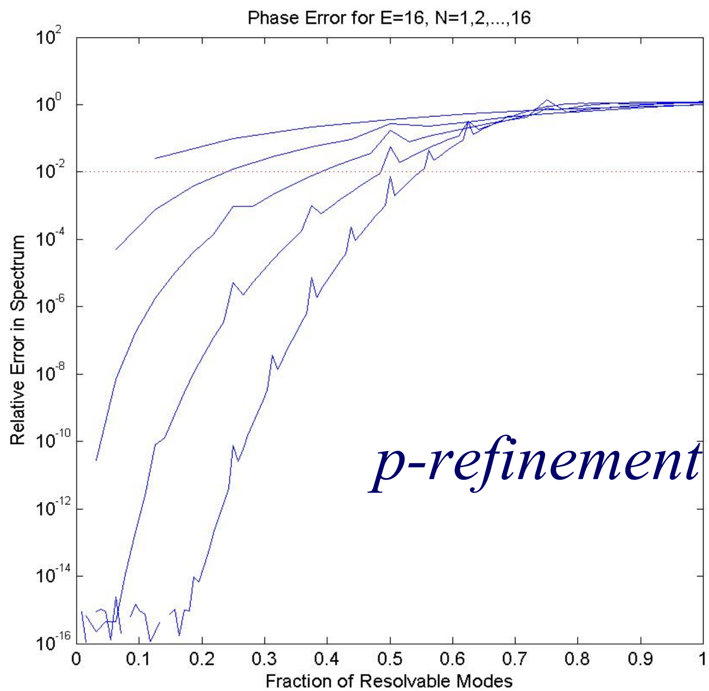


Convection of non-smooth data on a 32x32 grid (K_1 x K_1 spectral elements of order N).

(cf. Gottlieb & Orszag 77)

Relative Phase Error for h vs. p Refinement: $u_t + u_x = 0$





- $x\text{-axis} = k / k_{max}$, $k_{max} := n / 2$ (Nyquist)
- Fraction of resolvable modes increased only through p-refinement
  - *– dispersion significantly improved w/ exact mass matrix* (Guermond, Ainsworth)
- Polynomial approaches saturate at $k / k_{max} = 2 / \pi$

→ $N = 8\text{-}16$ ~ point of marginal return

Impact of Order on Costs

- To leading order, cost scales as number of gridpoints, regardless of approximation order.   WHY?

Impact of Order on Costs

- To leading order, cost scales as number of gridpoints, regardless of SEM approximation order. WHY?

- Consider Jacobi PCG as an example:

$$\underline{z} = D^{-1}\,\underline{r}$$
$$\underline{r} = \underline{r}^{t}\,\underline{z}$$
$$\underline{p} = z + \beta\,\underline{p}$$
$$\underline{w} = A\,\underline{p} \quad \longleftarrow$$
$$\sigma = \underline{w}^{t}\,\underline{p}$$
$$\underline{x} = \underline{x} + \alpha\,\underline{p}$$
$$\underline{r} = \underline{r} - \alpha\,\underline{p}$$

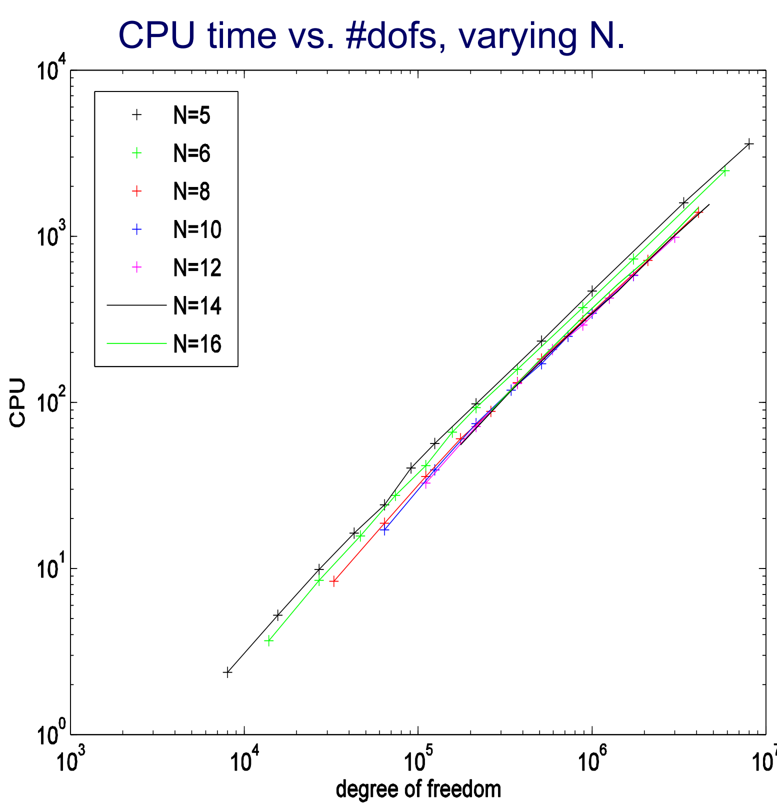
- Six O(n) operations with order unity computational intensity.

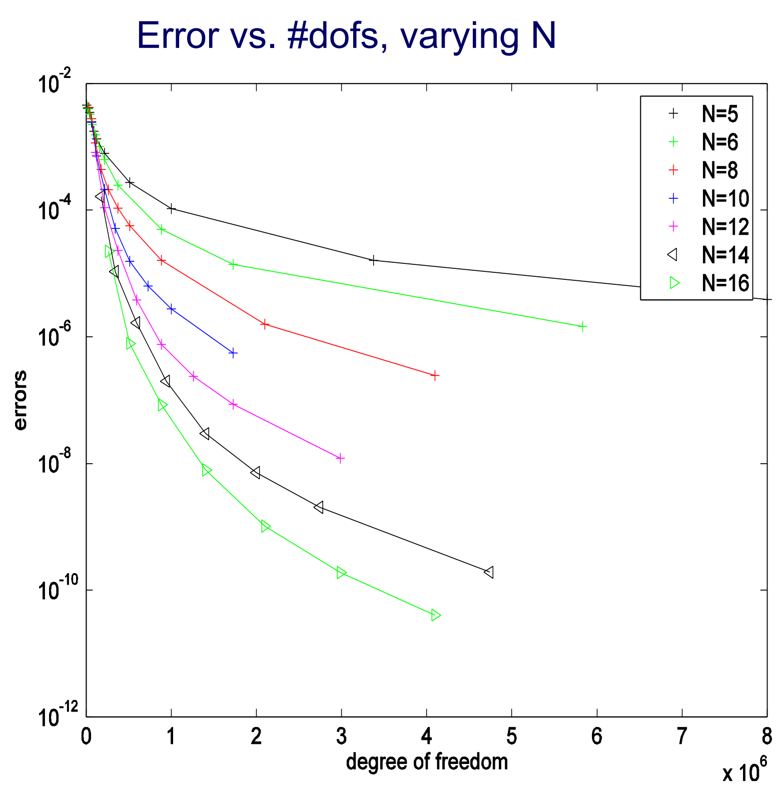
- *One* matrix-vector product dependent on approximation order

- ***Reducing n is a direct way to reduce data movement.***

Cost vs. Accuracy: Electromagnetics Example

- For SEM, ***memory*** scales as number of gridpoints, *n*.
- Work scales as *nN*, but is in form of (*fast*) matrix-matrix products.





Periodic Box; 32 nodes, each with a 2.4 GHz Pentium Xeon

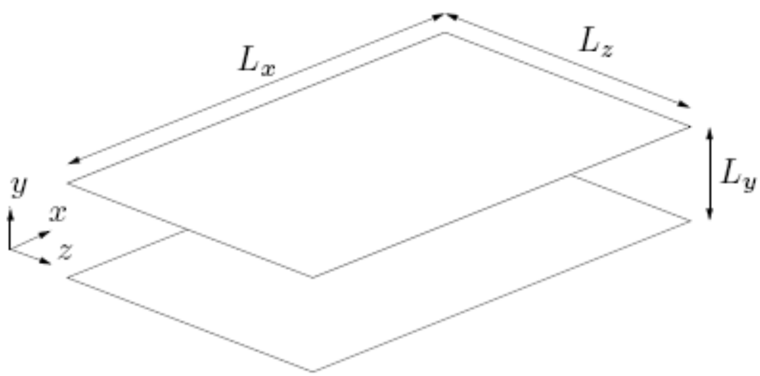
What About Nonlinear Problems?

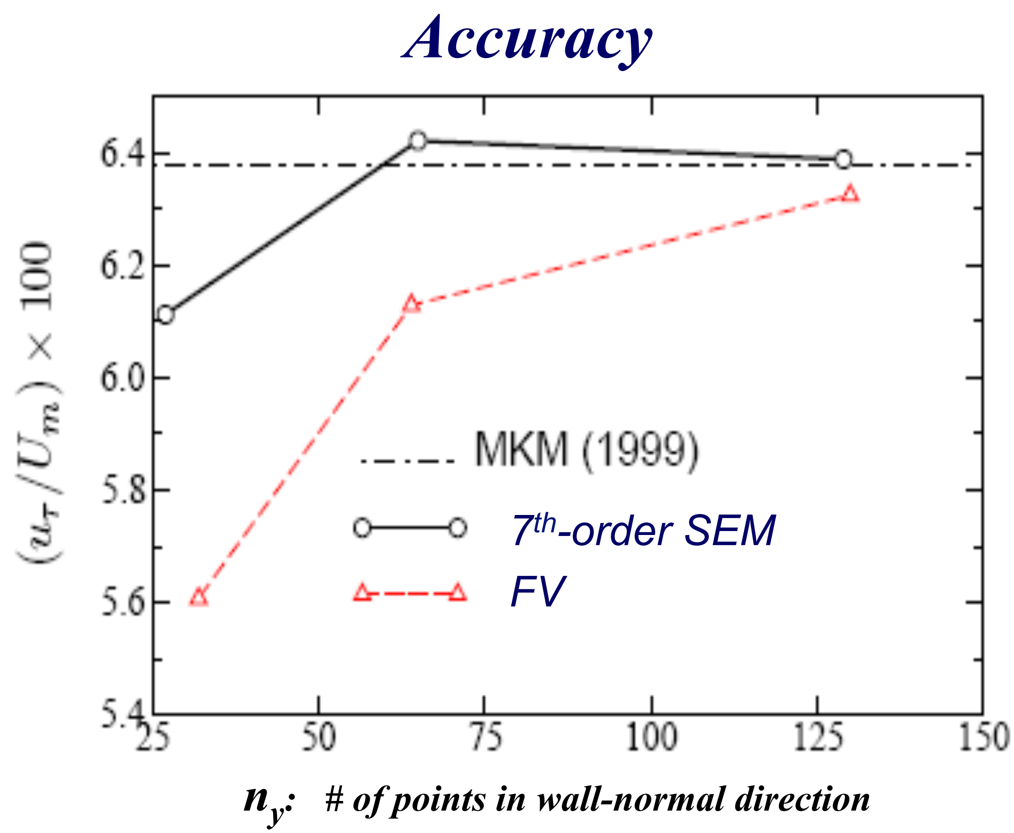
Are the high-order phase benefits manifest in linear problems evident in turbulent flows with nontrivial physical dispersion relations?

Nonlinear Example: NREL Turbulent Channel Flow Study

Sprague et al., 2010

- **Accuracy:** Comparison to several metrics in turbulent DNS, $Re_\tau = 180$ (MKM' 99)



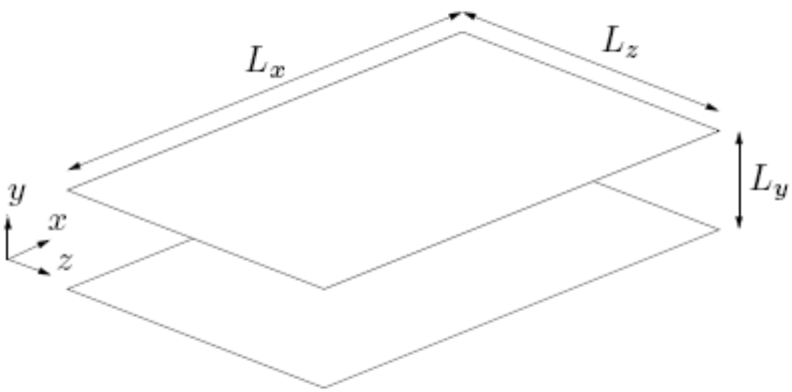


- **Results:** 7^{th}-order SEM needs an *order-of-magnitude* fewer points than 2^{nd}-order FV.

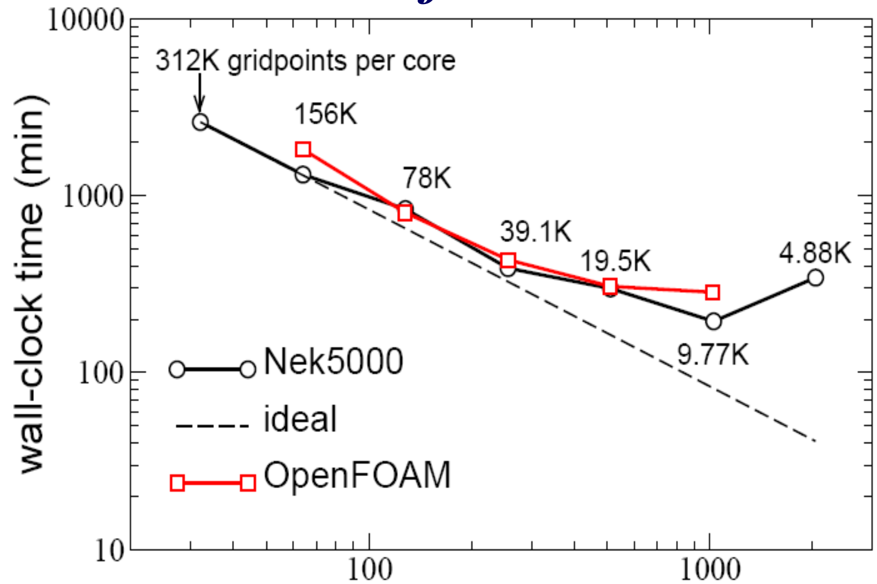
Nonlinear Example: NREL Turbulent Channel Flow Study

Sprague et al., 2010

- **Test case:** Turbulent channel flow comparison to DNS of MKM '99.



Performance



P: # of processors

- **Costs:** Nek5000 & OpenFOAM have the *same* cost per gridpoint

Overview

I. Scalable simulations of turbulent flows

- Discretization
- Solvers
- Parallel Implementation

II. A quick demo…

Scalable Linear Solvers

- Key considerations:
  - Bounded iteration counts as n→infinity
  - Cost that does not scale prohibitively with number of processors, P
- Our choice:
  - Projection in time: extract available temporal regularity in $\{\underline{p}^{n-1}, \underline{p}^{n-2}, \ldots, \underline{p}^{n-k}\}$
  - CG or GMRES, preconditioned with multilevel additive Schwarz
  - Coarse-grid solve:
    - XX^T projection-based solver
    - single V-cycle of well-tuned AMG *(J. Lottes, 2010)*

Projection in Time for $A\underline{x}^n = \underline{b}^n$ (A - SPD)

Given $\cdot\ \underline{b}^n$

$\cdot\ \{\underline{\tilde{x}}_1, \ldots, \underline{\tilde{x}}_l\}$ satisfying $\underline{\tilde{x}}_i^T A \underline{\tilde{x}}_j = \delta_{ij}$,

- Set $\underline{\bar{x}} := \Sigma\, \alpha_i \underline{\tilde{x}}_i, \quad \alpha_i = \underline{\tilde{x}}_i^T \underline{b}$ (best fit solution)
- Set $\Delta\underline{b} := \underline{b}^n - A\underline{\bar{x}}$
- Solve $A\Delta\underline{x} = \Delta\underline{b}$ to $tol\ \epsilon$ (black box solver)
- $\underline{x}^n := \underline{\bar{x}} + \Delta\underline{x}$
- If $(l = l_{\max})$ then (update X^l)

  $\underline{\tilde{x}}_1 = \underline{x}^n / ||\underline{x}^n||_A$

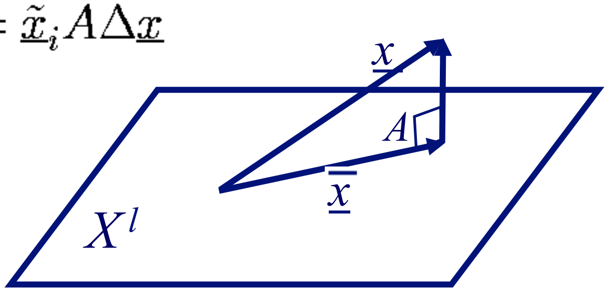
  $l = 1$

  else

  $\underline{\tilde{x}}_{l+1} = (\Delta\underline{x} - \Sigma\, \beta_i \underline{\tilde{x}}_i)/(\Delta\underline{x}^T A \Delta\underline{x} - \Sigma\, \beta_i^2)^{\frac{1}{2}}, \quad \beta_i = \underline{\tilde{x}}_i A \Delta\underline{x}$

  $l = l + 1$

  endif

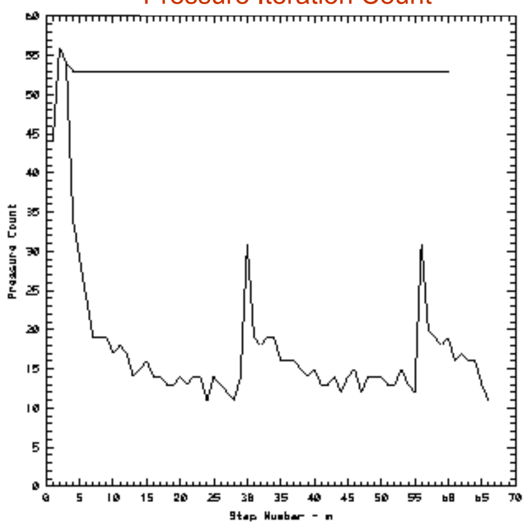


Initial guess for $A\underline{p}^n = \underline{g}^n$ via projection onto previous solutions

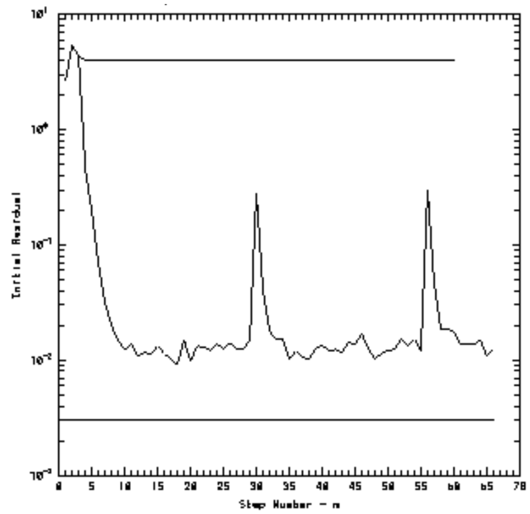
- $\| \underline{p}^n - \underline{p}^* \|_A = O(\Delta t^l) + O(\, \varepsilon_{tol} \,)$

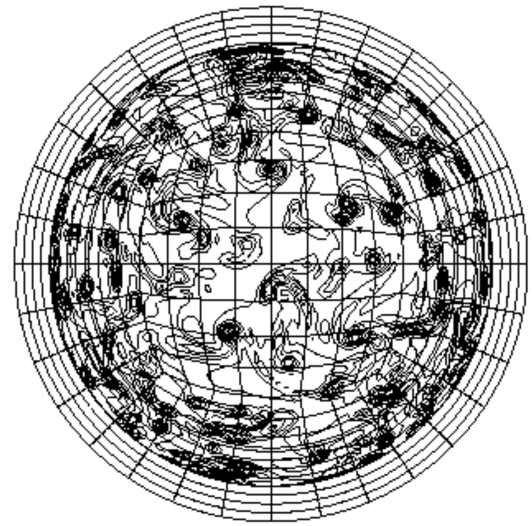
- Results with/without projection (1.6 million pressure nodes):

Pressure Iteration Count



Initial Pressure Residual





• 4 fold reduction in iteration count, 2 – 4 in typical applications

- Similar results for pulsatile carotid artery simulations –
  10^8-fold reduction in initial residual

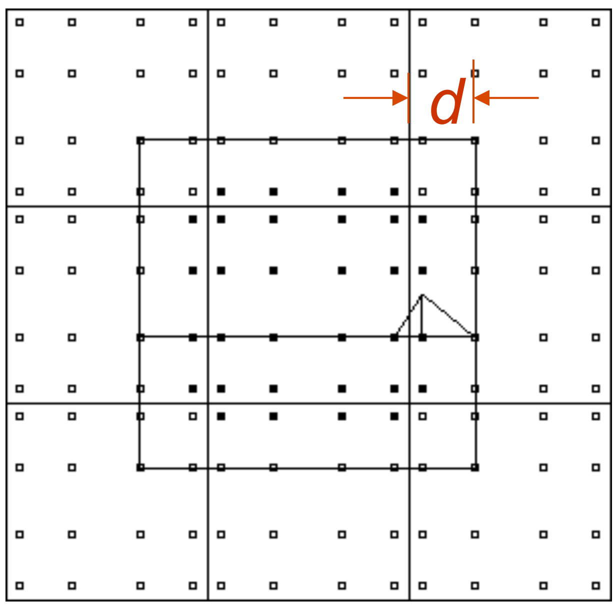
Scalable Linear Solvers

- Key considerations:
  - Bounded iteration counts as n→infinity
  - Cost that does not scale prohibitively with number of processors, P
- Our choice:
  - Projection in time – extract available temporal regularity in $\{\underline{p}^{n-1}, \underline{p}^{n-2}, \ldots, \underline{p}^{n-k}\}$
  - CG or GMRES, preconditioned with multilevel additive Schwarz
  - Coarse-grid solve:
    - FOR SMALL PROBLEMS: XX^T projection-based solver (default).
    - FOR LARGE PROBLEMS: single V-cycle of well-tuned AMG (Lottes)

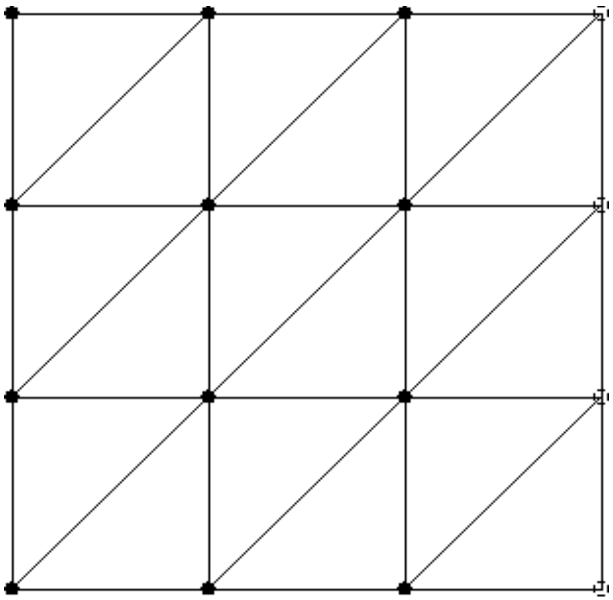
Multilevel Overlapping Additive Schwarz Smoother

(Dryja & Widlund 87, Pahl 93, F 97, FMT 00, F. & Lottes 05)

$$\underline{z} = M\underline{r} = \sum_{e=1}^{E} R_e^T A_e^{-1} R_e \, \underline{r} \; + \; R_0^T A_0^{-1} R_0 \underline{r}$$



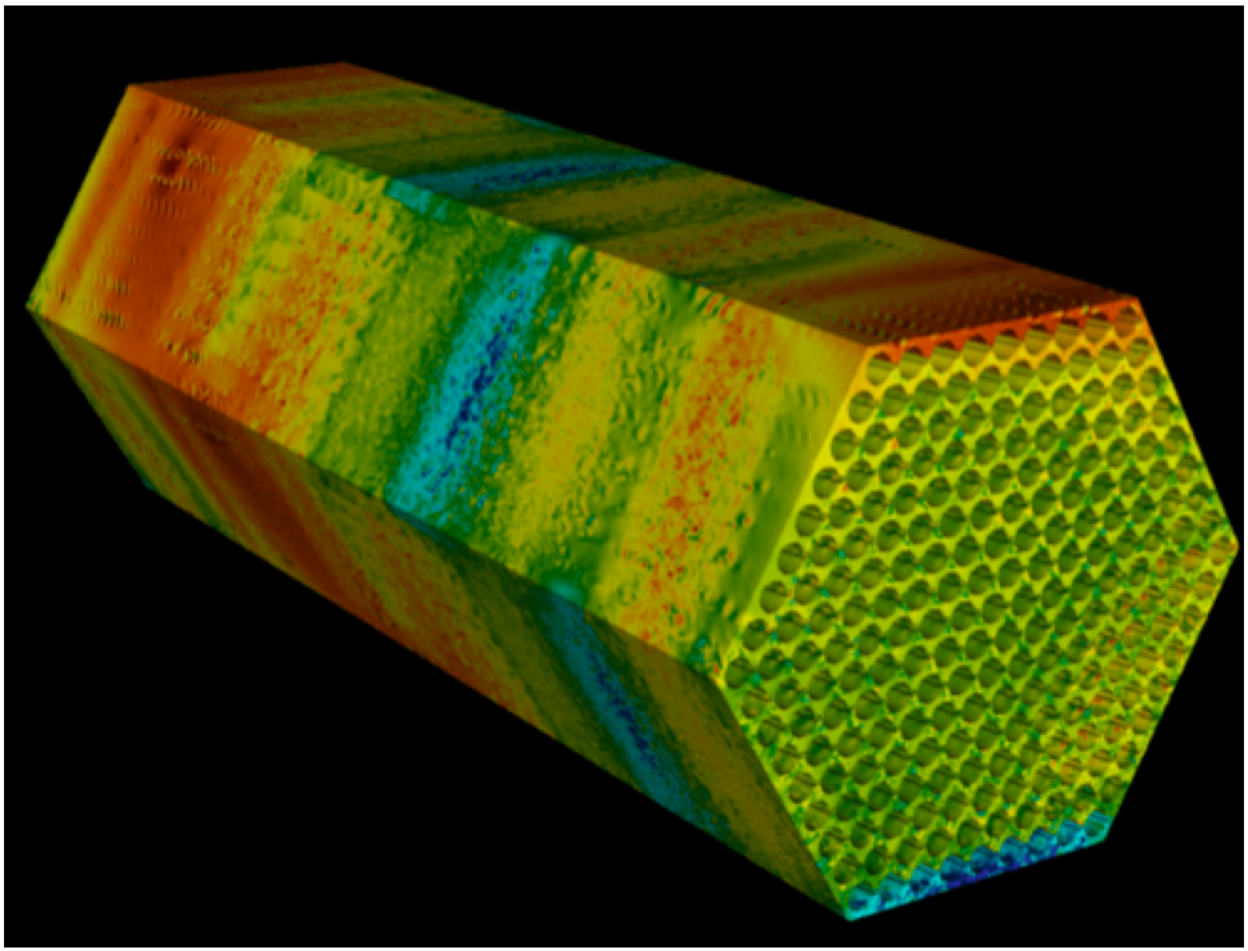
Local Overlapping Solves: *FEM-based Poisson problems with homogeneous Dirichlet boundary conditions,* A_e .

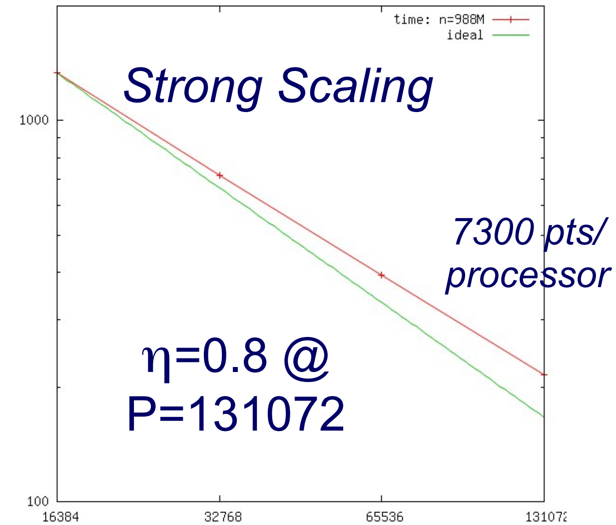


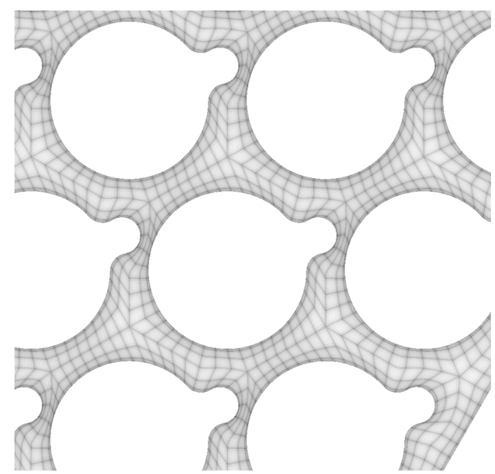
Coarse Grid Solve: *Poisson problem using linear finite elements on entire spectral element mesh,* A_0 *(GLOBAL).*

Scaling Example: Subassembly with 217 Wire-Wrapped Pins

- 3 million 7th-order spectral elements (n=1.01 billion)
- 16384–131072 processors of IBM BG/P
- 15 iterations per timestep; 1 sec/step @ P=131072
- Coarse grid solve < 10% run time at P=131072







Some Limitations of Nek5000

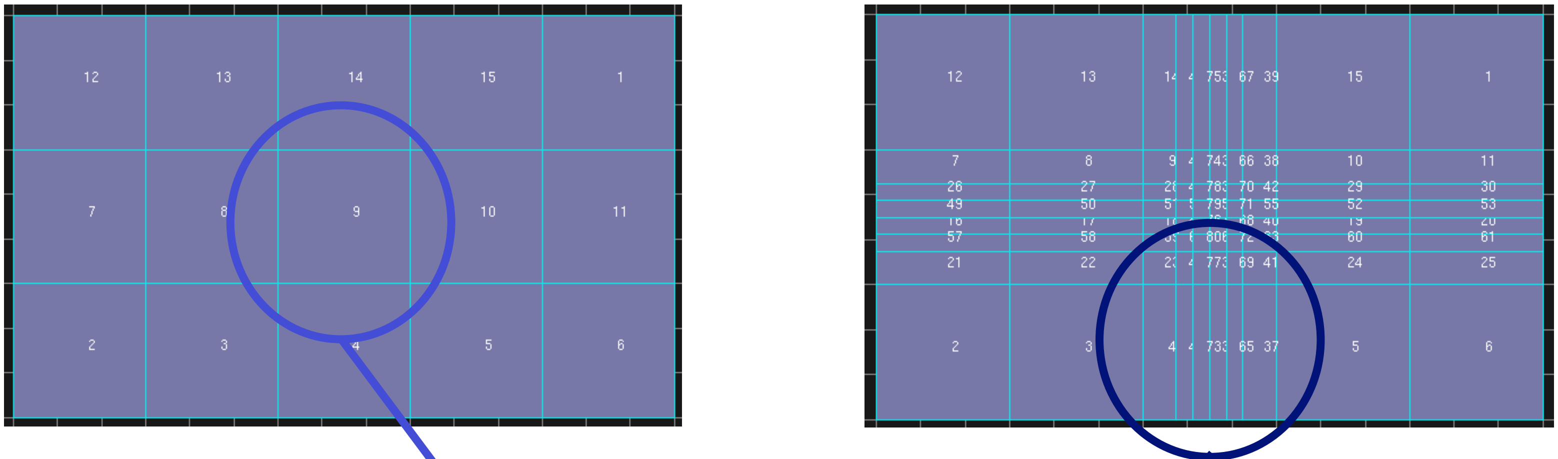
- ***No steady-state NS or RANS****:*
  - *unsteady RANS under development / test – Aithal*

- ***Lack of monotonicity for under-resolved simulations***
  - *limits, e.g., LES + combustion*
  - *Strategies under investigation: DG (Fabregat), Entropy Visc.*

- ***Meshing complex geometries****:*
  - *fundamental: meshing always a challenge;*
    hex-based meshes intrinsically anisotropic
  - *technical: meshing traditionally not supported as part of advanced modeling development*

Mesh Anisotropy

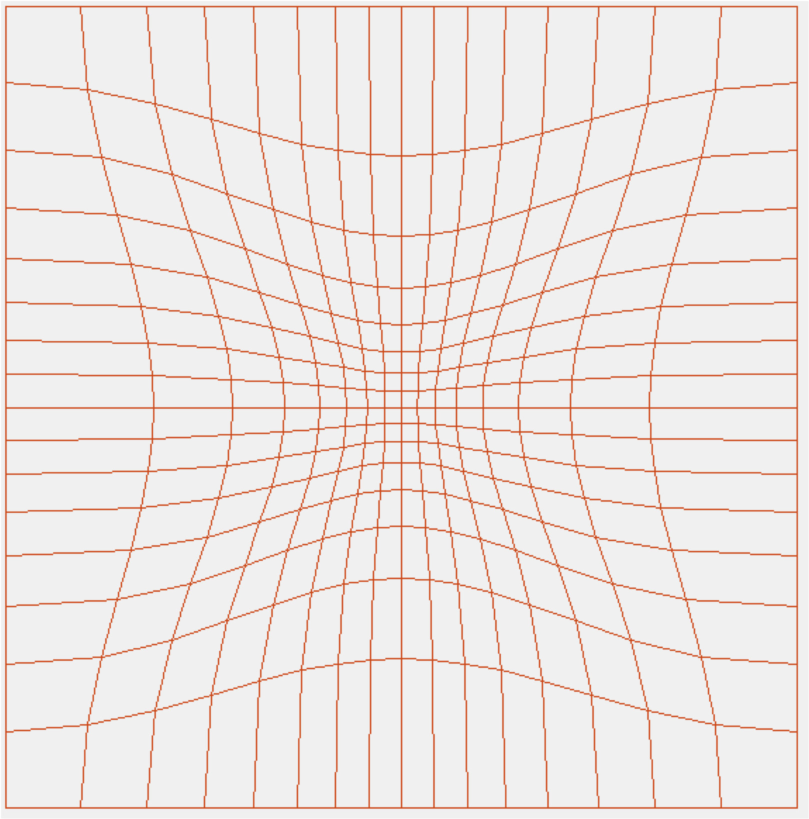
A common refinement scenario (somewhat exaggerated):

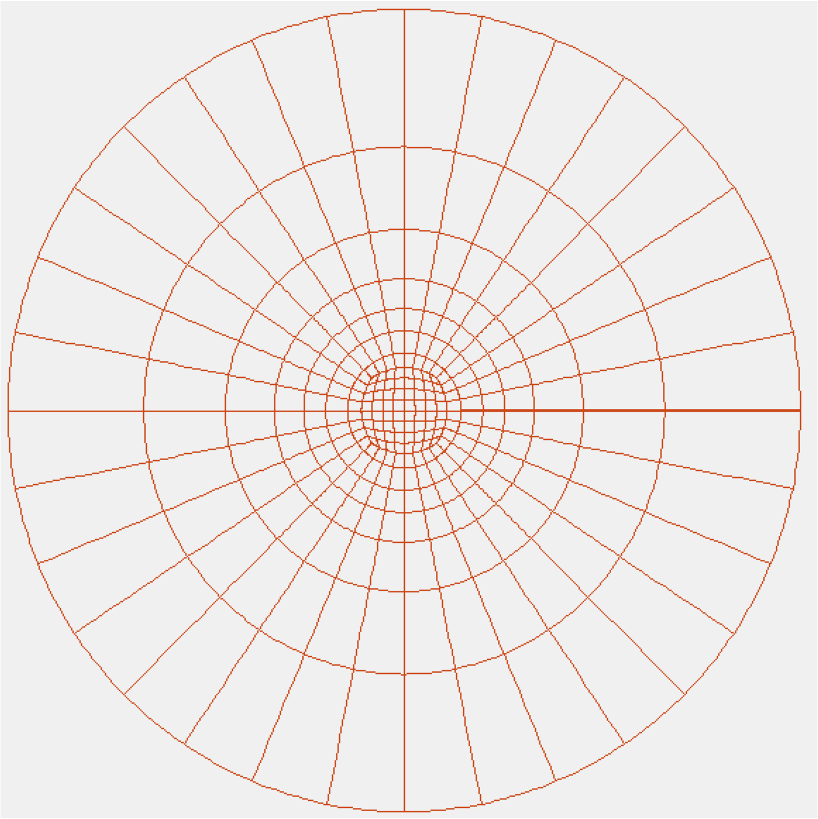


Refinement in region of interest yields unwanted high-aspect-ratio cells.

- ***Refinement propagation leads to***
  - unwanted elements in far-field
  - high aspect-ratio cells that are detrimental to iterative solver performance (F. JCP' 97)

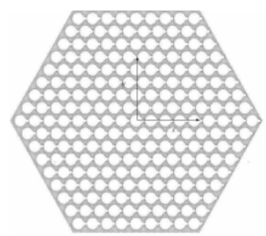
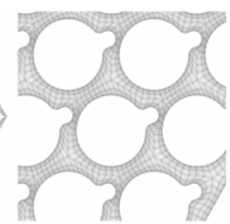
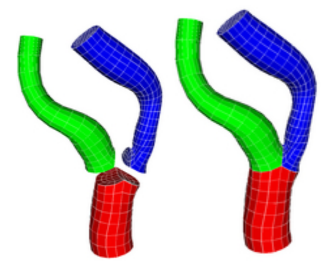
Alternative Mesh Concentration Strategies



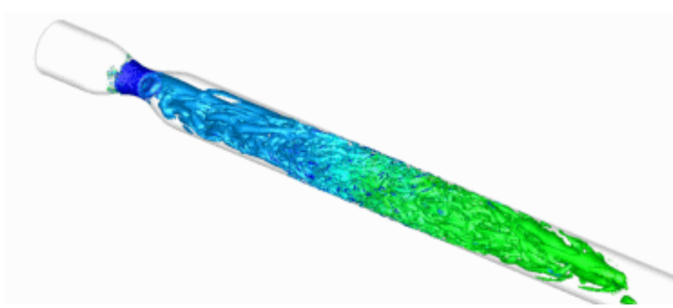


Meshing Options for More Complex Domains

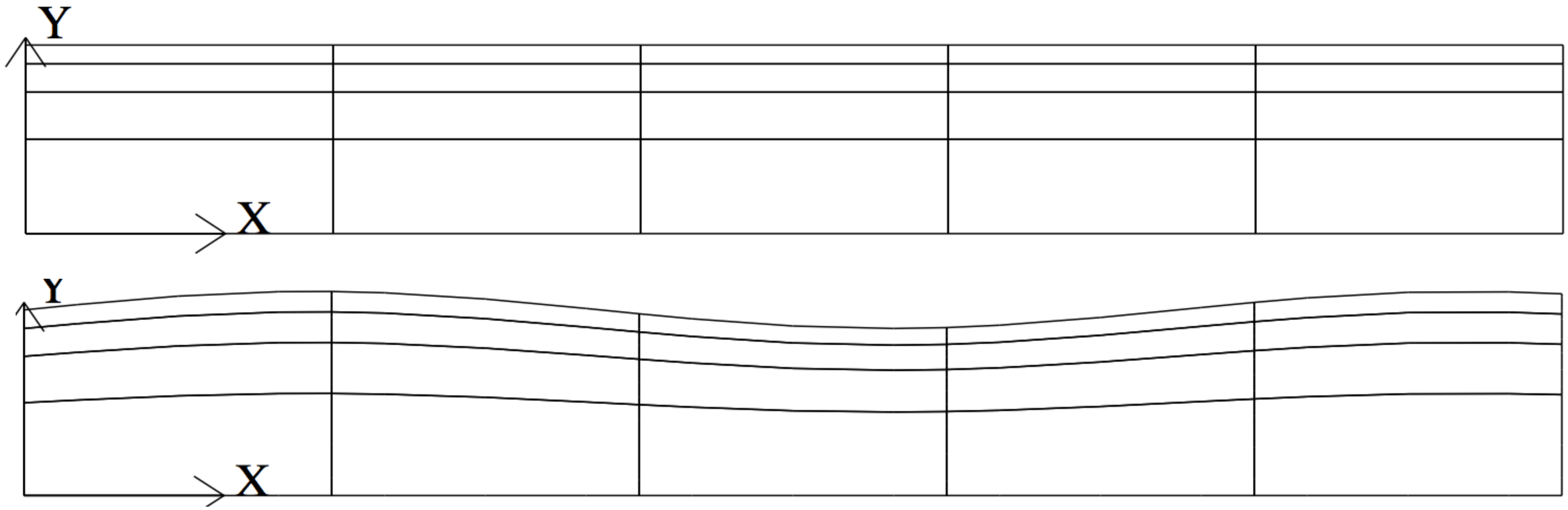
- *genbox: unions of tensor-product boxes*
- *prenek: basically 2D + some 3D or 3D via extrusion (n2to3)*
- *Grow your own: 217 pin mesh via matlab; BioMesh*

- *3^{rd} party: CUBIT + MOAB, TrueGrid, Gambit, Star CD*
- *Morphing:*



Morphing to Change Topography




```
do i=1,ntot
   argx          = 2*pi*xm1(i,1,1,1)/lambda
   ym1(i,1,1,1) = ym1(i,1,1,1) + ym1(i,1,1,1)*A*sin(argx)
enddo
```

Stratified Flow Model

- *Blocking phenomena – Tritton*
- *Implemented as a rhs forcing:*

$$
\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} &= -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} - \frac{1}{Fr^2}(\rho' - y) \\
\nabla\cdot\mathbf{u} &= 0 \\
\frac{\partial \rho'}{\partial t} + \mathbf{u}\cdot\nabla\rho' &= \frac{1}{Pr\,Re}\nabla^2\rho'.
\end{aligned}
$$

```
c-----------------------------------------------
      subroutine userf  (ix,iy,iz,ieg)
      include 'SIZE'
      include 'TOTAL'
      include 'NEKUSE'
c
      Fr2 = param(4)  ! Froude number squared
      ffx = 0.0
      ffy = (temp - y) / Fr2
      ffz = 0.0
      return
      end
c-----------------------------------------------
```

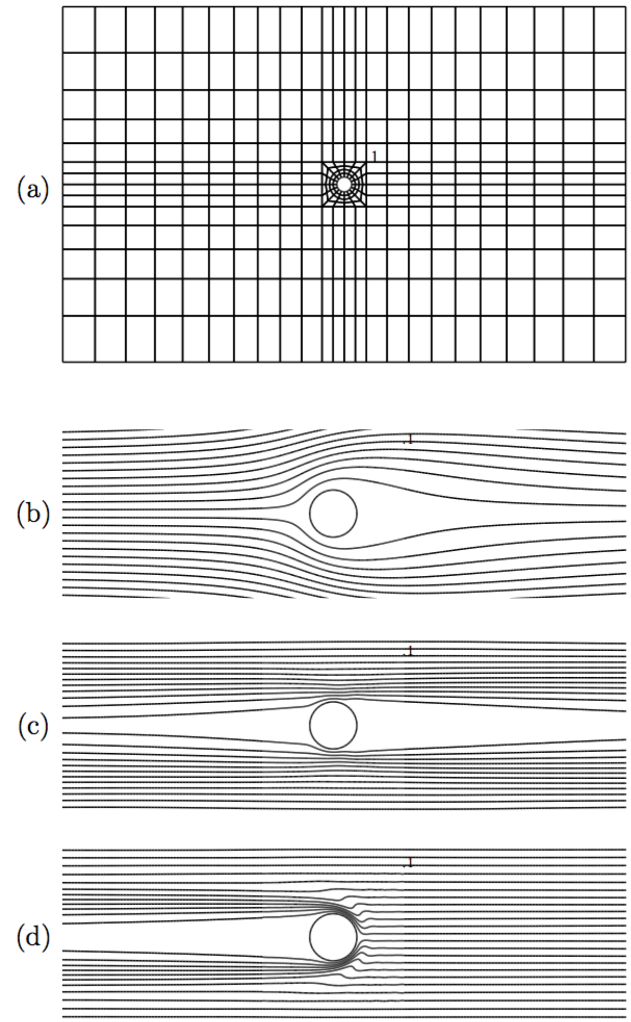


Figure 7: Examples of blocking phenomena in stratified flow at $Re = 10$: (a) spectral element mesh, (E, N)=(384, 7), and steady-state streamfunction distribution for (b) no stratification, (c) Fr^{-2}=1000, $Pr = 1$, and (d) Fr^{-2}=1000, $Pr = 1000$.

High Richardson Number Can Introduce Fast Time Scales

- *Fast waves in stratified flow can potentially lead to additional temporal stability constraints.*

- *Also, must pay attention to reflection from outflow.(Same issue faced in experiments…)*

$$
\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} &= -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} - \frac{1}{Fr^2}(\rho' - y)\hat{\jmath} \\
\nabla\cdot\mathbf{u} &= 0 \\
\frac{\partial \rho'}{\partial t} + \mathbf{u}\cdot\nabla\rho' &= \frac{1}{PrRe}\nabla^2\rho'.
\end{aligned}
$$

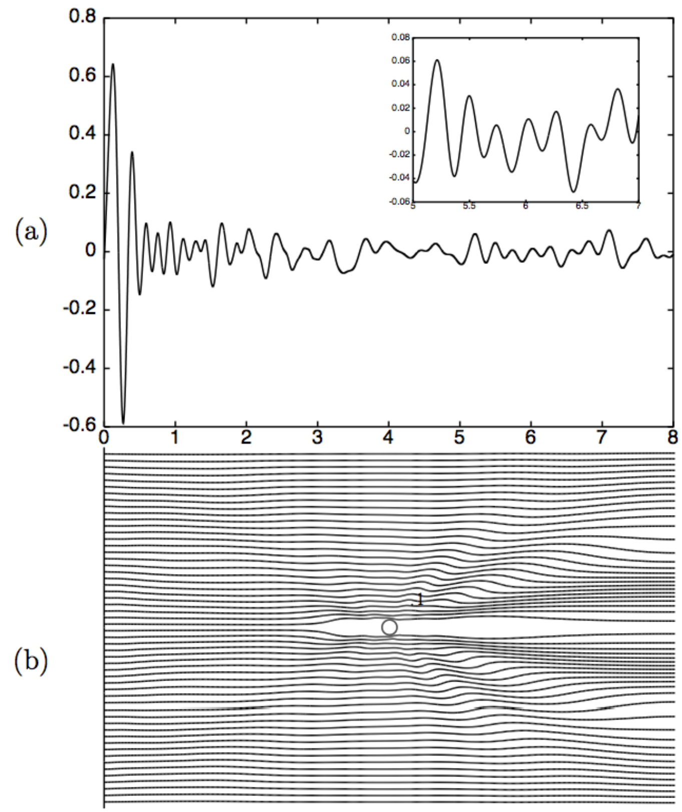


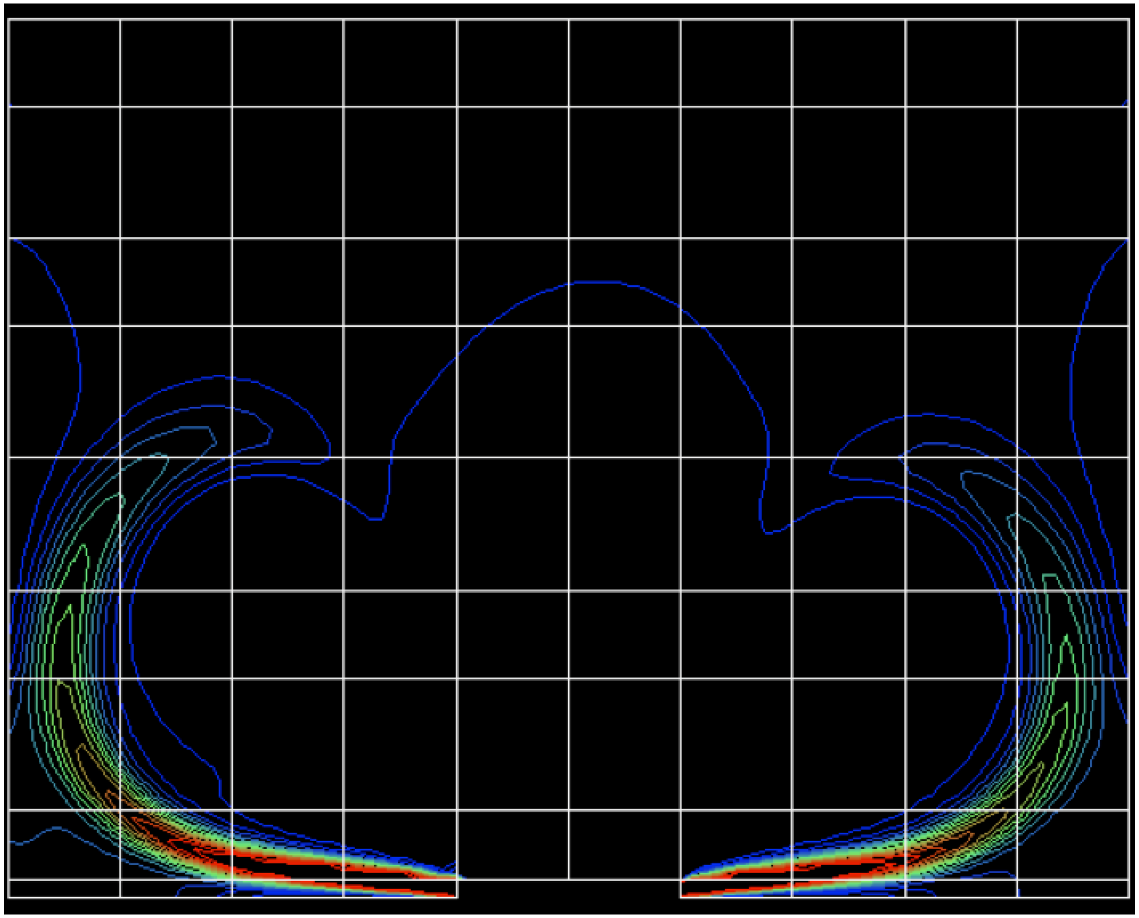
Figure 8: Wave-like response to sudden application of gravitation forcing for Fr^{-2}=1000, Pr = 1000: (a) time trace of v at point "1" indicated in (b); (b) instantaneous streamline pattern at $t = 0.5$.

Moving Mesh Examples

- **peristaltic flow model**
  nek5_svn/examples/peris

- **2D piston, intake stroke:**
  (15 min. to set up and run)

- More recent 3D results by Schmitt and Frouzakis

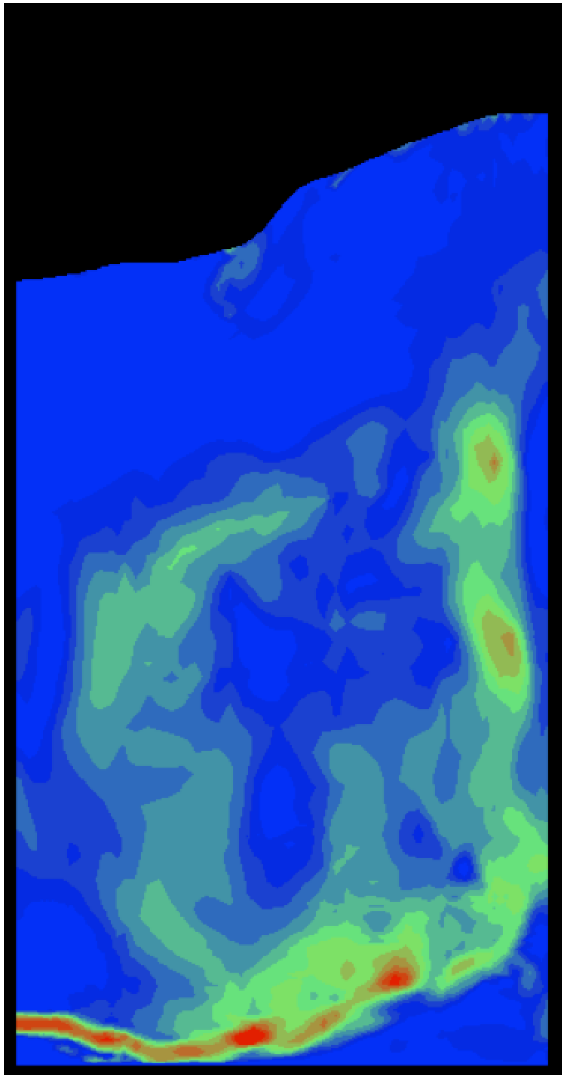


Moving Mesh Examples

- **Free surface case**

  (Lee W. Ho, MIT Thesis, ‘89)

- **Nominally functional in 3D, but needs some development effort.**



A (hopefully) Quick Demo

Thank You!