Acceleration Techniques for the Spectral Element Ocean Model Methodology

Craig C. Douglas
University of Kentucky
and Yale University

Gundolf Haase
Johannes Kepler
University Linz, Austria

with contributions by

Mohamed Iskandarani and Stefan Reitzinger

Supported in part by the National Science Foundation.

Shallow Water Equations

- Good approximation to fluid motion equations
  - When fluid density is homogenous and depth is small in comparison to characteristic horizontal distances
  - Coastal areas
  - Reduces complicated 3D problem to 2D
- In solving 3D primitive hydrostatic equations with top surface of the fluid free to move
  - Free surface allows gravity wave propagation at speed $\sqrt{gh}$
  - Gravity wave speed greatly exceeds advective velocity in deep of oceans and is culprit of restrictions on time steps

Derivation of Numerical Model

$$\vec{u}_t + g\nabla\zeta = \vec{F} \quad \text{and} \quad \zeta_t + \nabla\cdot\left[(h+\zeta)\vec{u}\right] = Q.$$

- $\vec{u} = (u, v)$ is the velocity vector.
- ζ is the sea surface displacement
- g is the gravity acceleration
- h is the resting depth of the fluid
- Q is a mass source/sink term
- $\vec{F} = (f^x, f^y)$ is the generalized forcing term for the momentum equations: Coriolis force, nonlinear advection, viscous dissipation, and wind forcing
- No slip boundary conditions are imposed to complete the system

Cost Cutting by Dynamics Splitting

- Barotropic (external modes)
  - Akin to shallow water equations
  - Govern evolution of gravity waves
  - Solved implicitly to avoid excessively small time steps
- Baroclinic (internal waves)
  - Eliminate gravity wave speed restrictions
  - Solved explicitly using large time steps

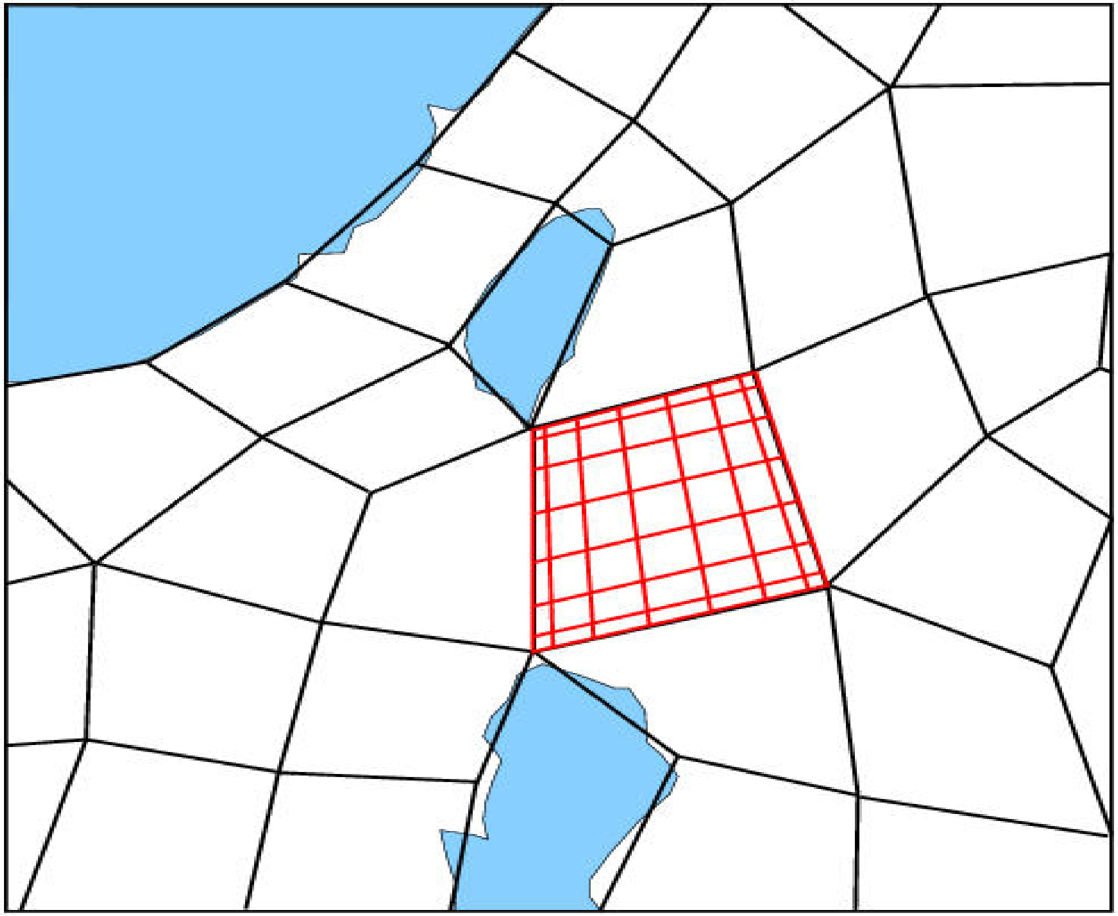
Time Splitting for Implicit Terms Only

- Uzawa-like splitting used for pressure and velocity unknowns. Pressure equation results in a SPD system
  - Can use an inexpensive, robust CG solver
  - Matrix never stored
  - Only a matrix-vector operation available
- Preconditioners
  - Standard is a diagonal based one
  - New one specifically tailored to Spectral Ocean Element Model (SEOM) using an Additive Schwarz Method (ASM)

Spectral Ocean Element Method (SEOM)

- Novel feature
  - Isopycnal coordinates vertically, spectral horizontally
- Benefits of spectral discretization
  - Geometric flexibility through unstructured and quasi-unstructured grid
  - Dual *h-p* paths to convergence
  - Low numerical dispersion and dissipation errors
  - Dense computational kernels lead to splendid parallel scaling

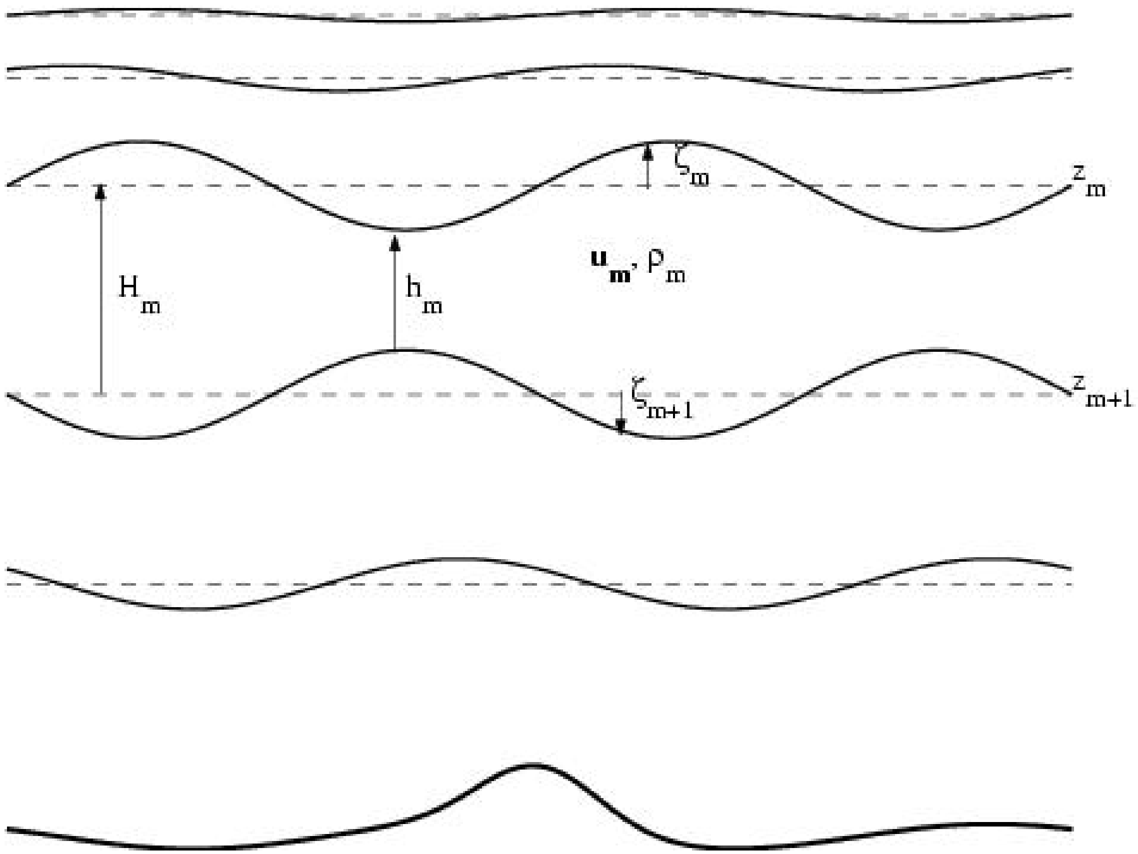
Dual h-p Spectral Finite Element Methods



Multiple Layer Rationale

- Benefits
  - Ease of development (stacked shallow water equations)
  - Minimization of cross-isopycnal diffusion
  - Eliminates pressure gradient errors
  - Baroclinic process representation
  - Cost savings over full 3D model
- Appropriate for
  - Wind driven simulations, eddy formulations, and (in part) flow/topology interaction

Five Layer Example



Variational Form

In Cartesian coordinates:

$$\int_A \phi u_t dA + \int_A g\zeta_x \phi dA = \int_A f^x \phi dA,$$

$$\int_A \phi v_t dA + \int_A g\zeta_y \phi dA = \int_A f^y \phi dA,$$

$$\int_A \psi\zeta_t dA - \int_A (\psi_x u - \psi_y v)(h+\zeta) dA = \int_A Q\psi dA,$$

where ϕ and ψ are the velocity and pressure test functions. The integration by parts leads to a SPD system.

Semi-Implicit Integration

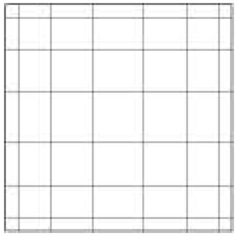
- Gravity waves
  - Terms $g\zeta$ and $\nabla \cdot (h\vec{u})$ cause trouble.
  - Use Crank-Nicolson scheme to solve implicitly.
- Rest of the terms
  - Do not cause trouble.
  - Use third order Adams-Bashforth scheme to solve explicitly.

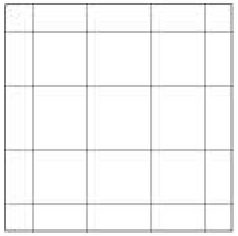
Time Discretized Equations

- $$\tau^{-1}\int_A u^{n+1}\phi dA + .5\int_A g\zeta_x^{n+1}\phi dA = \tau^{-1}\int_A u^n \phi dA -$$
  $$.5\int_A g\zeta_x^n \phi dA + \sum_{p=0}^{2}\alpha_p \int_A f^{x(n-p)}\phi dA$$
- $$\tau^{-1}\int_A v^{n+1}\phi dA + .5\int_A g\zeta_y^{n+1}\phi dA = \tau^{-1}\int_A v^n \phi dA -$$
  $$.5\int_A g\zeta_y^n \phi dA + \sum_{p=0}^{2}\alpha_p \int_A f^{y(n-p)}\phi dA$$
- $$\tau^{-1}\int_A \zeta^{n+1}\psi dA - .5\int_A \nabla\psi \cdot h\vec{u}^{n+1} dA = \tau^{-1}\int_A \zeta\psi dA +$$
  $$.5\int_A \nabla\psi \cdot h\vec{u}^n dA + \sum_{p=0}^{2}\alpha_p \int_A \left(Q^{n-p}\psi + \nabla\psi \cdot \vec{u}^{n-p}\psi^{n-p}\right) dA$$

Spatial Discretization

- Spectral formulation with $N^p = N^v - 2$.

velocity: 

pressure: 

- For each element, we can write

$$\vec{u}(\xi,\eta) = \sum_{k,l=1}^{N^v} \vec{u}_{k,l}\,\mu_k^v(\xi)\,\mu_l^v(\eta),$$

$$\zeta(\xi,\eta) = \sum_{i,j=1}^{N^v} \zeta_{i,j}\,\mu_i^p(\xi)\,\mu_j^p(\eta)$$

Galerkin Formulation

- Reduces problem to algebraic equations with a, b, c obvious substitution. Let M^v and M^p be the mass matrices, G^x and G^y be the discrete gradient operators, and D^x and D^y are components of the discrete gradient operators. Then the Galerkin formulation is simply

$$\tau^{-1}M^v u + .5gG^x\zeta = a$$

$$\tau^{-1}M^v v + .5gG^y\zeta = b$$

$$-.5D^x u - .5D^y v + \tau^{-1}M^p\zeta = c$$

Solution of the Reduced System

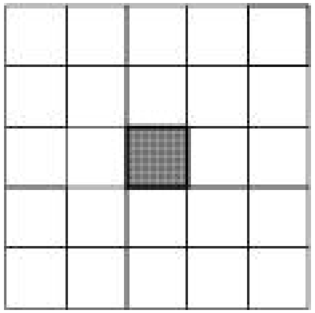
- By substitution, $D^s = \left(G^s\right)^T H$, and the fact that M^v is a diagonal matrix, the original system reduces to solving $S\zeta = f$, where the Schur complement S is only available through the matrix-vector operation Sw.

$$S = \left[\frac{M^p}{\tau} + \frac{\tau g}{4} \left(\left(G^x\right)^T H \left(M^v\right)^{-1} G^x + \left(G^y\right)^T H \left(M^v\right)^{-1} G^y \right) \right]$$

$$\mathrm{f} = \left[c - \frac{\tau}{2} \left(\left(G^x\right)^T H \left(M^v\right)^{-1} a + \left(G^y\right)^T H \left(M^v\right)^{-1} b \right) \right]$$

Matrix S Not Stored

- S is *extremely* memory consuming.
  - Many entries from nodes that are two layers of elements away.



  - S is never stored. The implemented implicit matrix-vector multiplication is fast and takes advantage of the local tensor product structure of the matrix.

Schoedinger-like Equation

- Schur complement system similar to Schoedinger-like equation

$$-\frac{g}{4}\nabla^T\left(\frac{h(x)}{m(x)}\nabla\zeta(x)\right)+\tau^{-2}\zeta(x)=\frac{1}{\tau}f(x),\quad x\in\Omega$$

- S symmetric, positive definite
  - Can use preconditioned conjugate gradient method
  - Preconditioner defined using matrix-vector multiply

$$\left\{C_{ii}\right\}_{i=1}^{N} = \left\{\left(S\cdot 1\right)_i\right\}_{i=1}^{N}$$

lumped Schur complement.

A Much Better Preconditioner

- Additive Schwarz Method (ASM)
  - Define restriction from a global vector to one on a local element r by $A_r : \mathbb{R}^N \rightarrow \mathbb{R}^{N_r}$.
  - Original matrix S is no longer expressible by element matrices, which are no longer defined just on an element. This is a result of the long connections between nodes connecting elements.

Local Element Contributions

- Consider only local contributions to generation of local matrices $\hat{S}_r$, i.e., restrict the support of the transformed pressure basis functions to the elements. Then $\hat{S} = \sum_{\mathrm{r}=1}^{\mathrm{nelem}} A_{\mathrm{r}}^{\mathrm{T}} \hat{S}_r A_r =$

$$\left[\frac{M_r^p}{\tau} + \frac{\tau g}{4} \left(\left(G_r^x \right)^T H \left(M_r^v \right)^{-1} G_r^x + \left(G_r^y \right)^T H \left(M_r^v \right)^{-1} G_r^y \right) \right]$$

- Local matrices are similar to Schroedinger-like equations in Ω_r with homogeneous Neuman b.c.

Calculating $\hat{S}$ Matrices

- Method 1: Derive $\hat{S}$ as a stiffness matrix of the local Schroedinger-like equation
- Method 2: Use modified matrix-vector routine that operates on a single element. Multiply all unit vectors to get columns of $\hat{S}_{r,i,j}$ for all local rows i.
  - Consumes more CPU time, but is easy to do and robust.
  - We opted for Method 2. Time spent is a once per run throwaway expense. After all, each run is *O(10,000)* to *O(1,000,000)* time steps.

Local Properties of S Representation

- $\hat{S}_r$ really represents local properties of S.
  - $S \cdot 1 \equiv \hat{S} \cdot 1$ holds.
  - $\hat{S}$ is some sort of blockwise lumped S.
  - All $\hat{S}_r$ are nonsingular as long as $|\tau| \neq \infty$.
- ASM preconditioner can be derived for S and $\hat{S}$.

$$\hat{C}^{-1} = W^{-1} \sum_{r=1}^{nelem} A_r^T \hat{S}_r^{-1} A_r$$

which is not block diagonal (blocks overlap only in the rows/columns shared by multiple elements).

Preconditioner

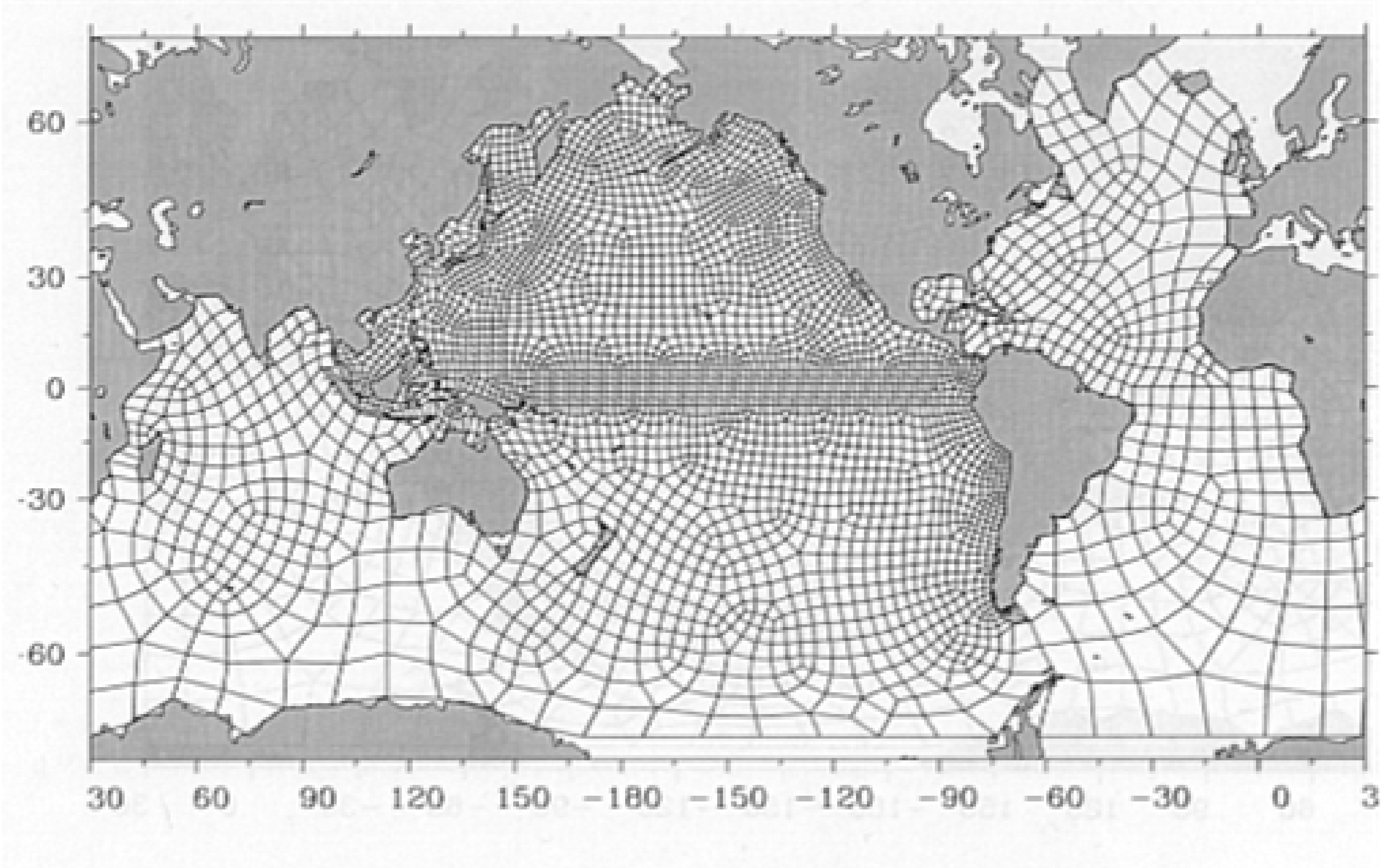
- We require the diagonal weight matrix $W = \sum_{r=1}^{nelem} A_r^T A_r$

  containing the number of elements a node belongs to, i.e., these are the weights needed for the partition of unity.
- Parallelization of $\hat{C}^{-1}$ is
  – Straightforward
  – Requires only one next neighbor communication per application.

Numerical Example (NEP08)

- Compare old and new preconditioners
  - Reduce norm by 4 orders of magnitude as stopping criteria in CG iteration.
- 14-16 year studies: several minute time steps.
- 5 layers in oceans.
- 7^{th} and 5^{th} order spectral elements for velocity and pressure.
- Wind stress data sets from National Center for Environmental Prediction (NCEP) drive examples.

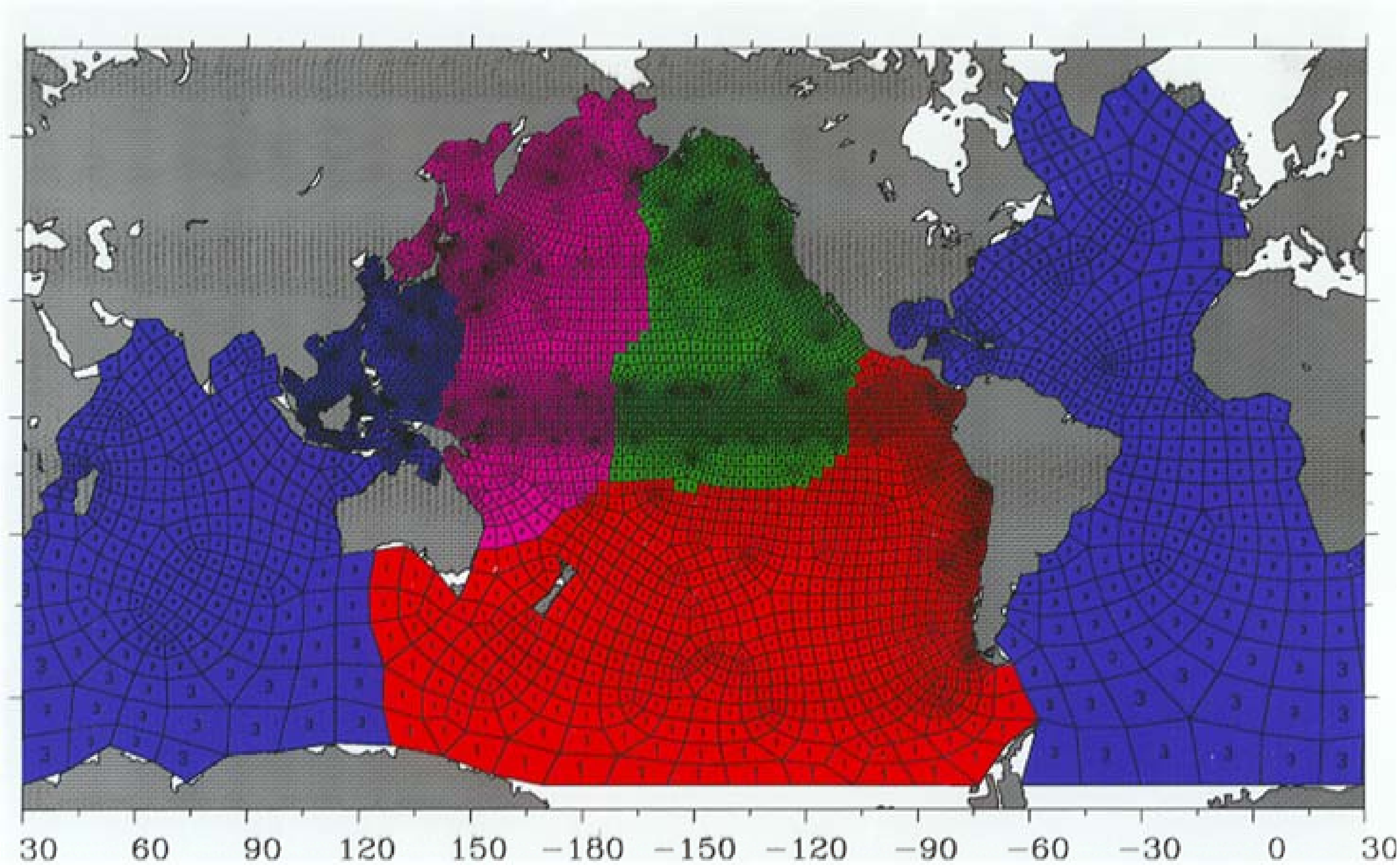
NEP08 Example



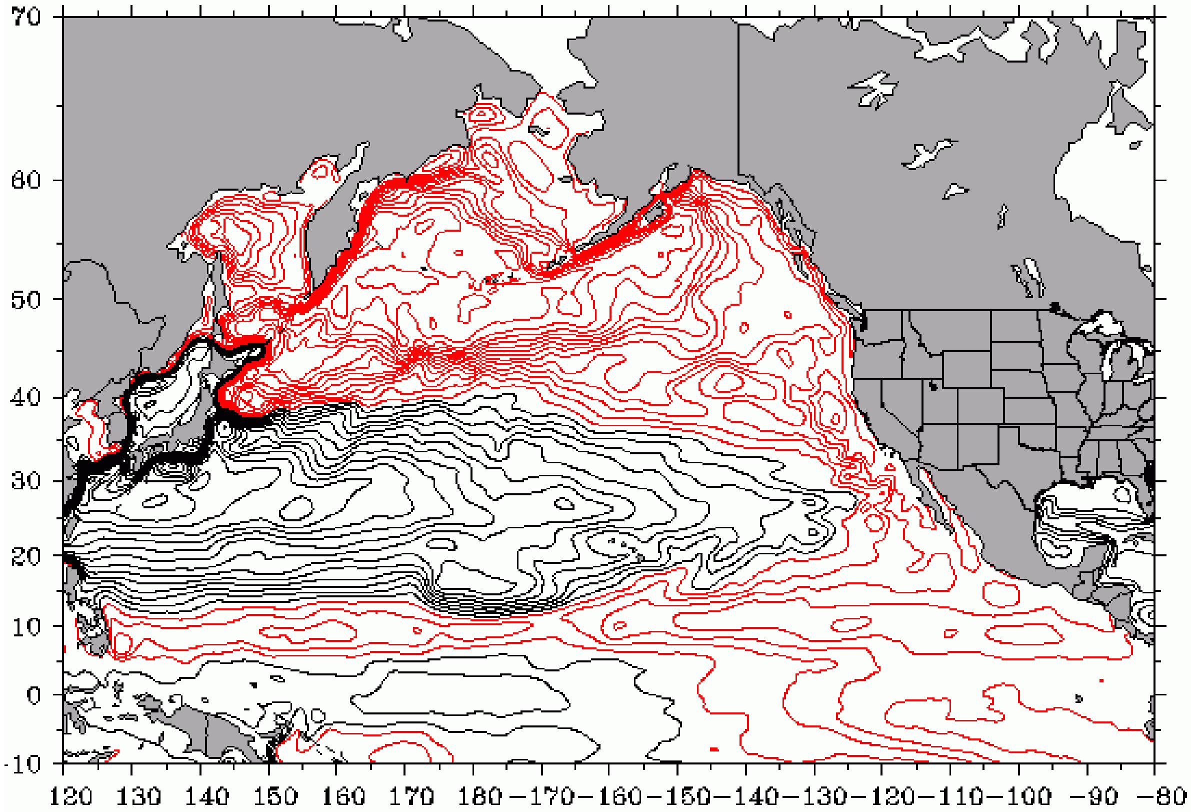
NEP08 Details

- Larger grid and greater disparity of element sizes.
  - 3552 elements, 176377/90459 velocity/pressure nodes.
- Gulf of Alaska, waveguide, and equatorial regions of interest.
- Global ocean included (solves open boundary condition problem).
- 20 km Gulf of Alaska, 35 km in waveguide region, much more elsewhere.

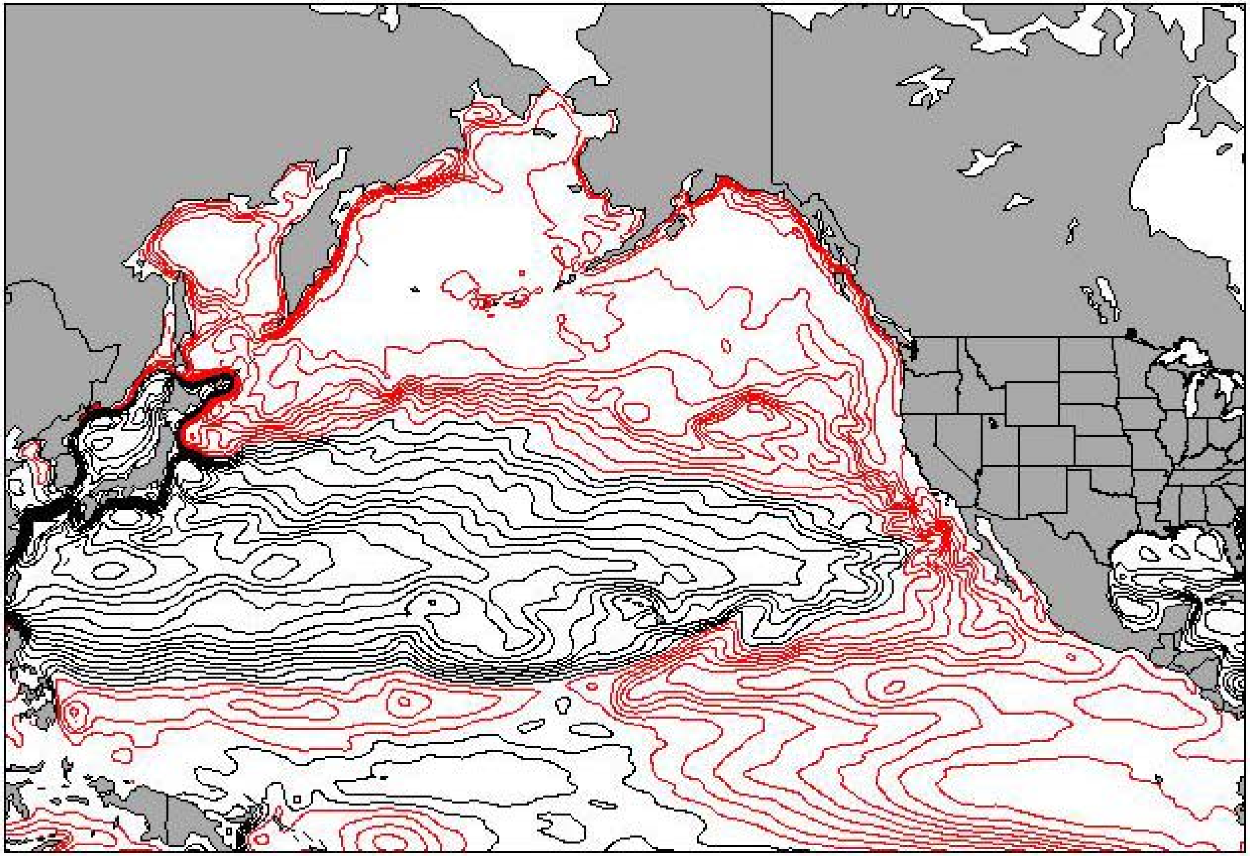
NEP08 p = 4 Domain Decomposition



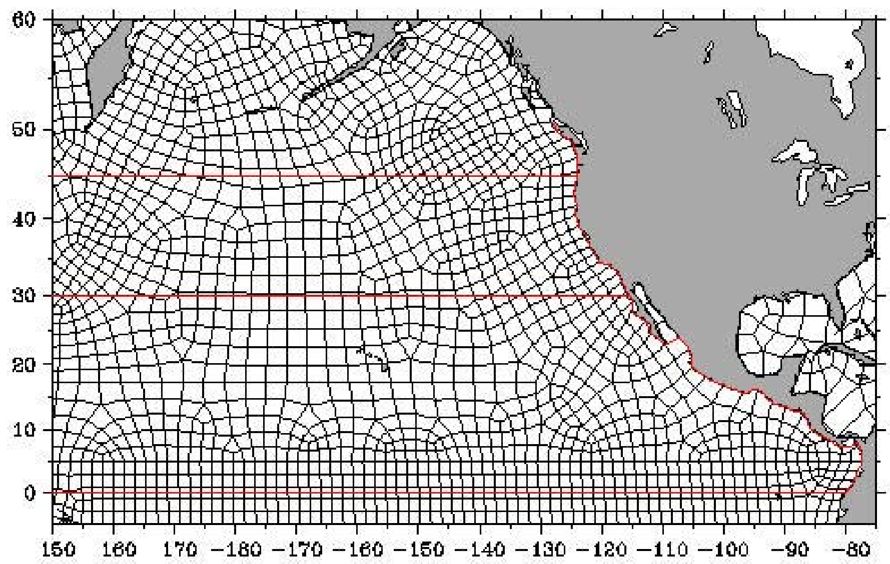
NEP08 Sea Surface Height



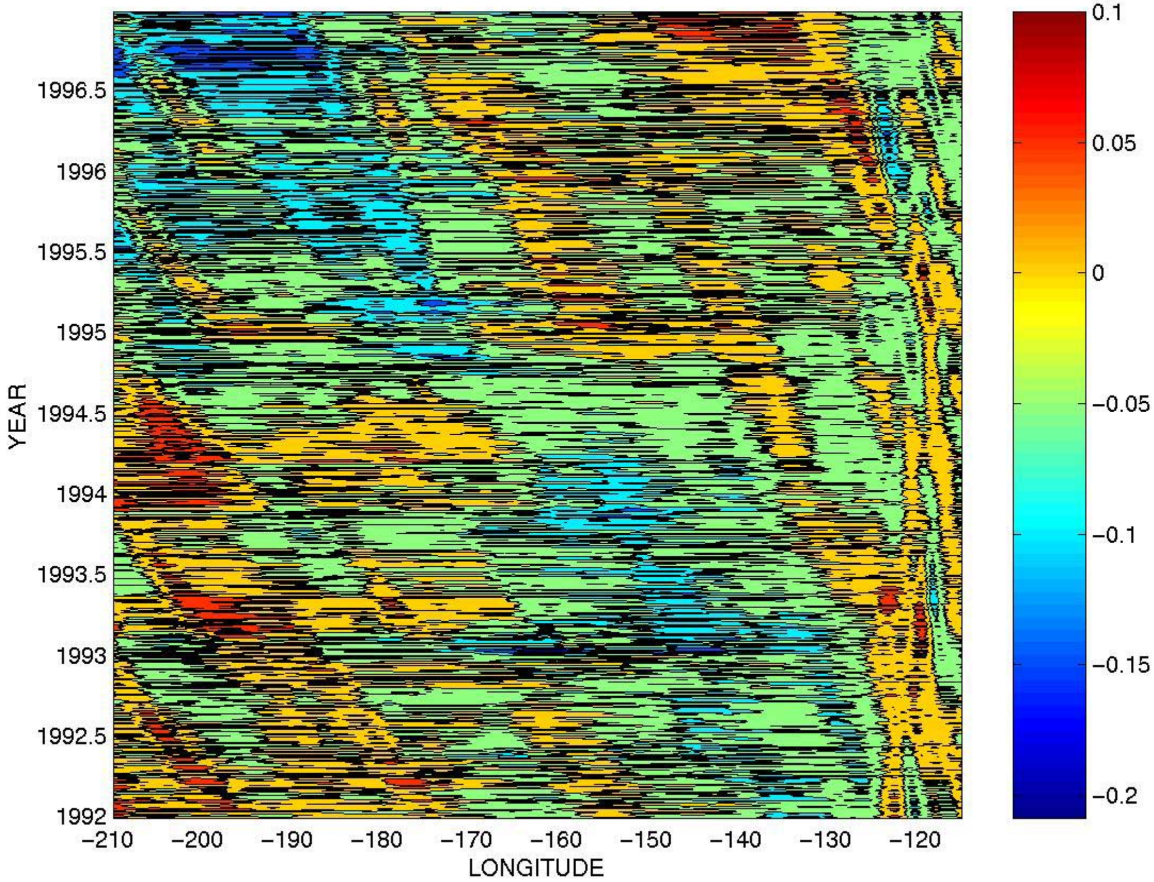
NEP08 Layer 1 Anomaly



NEP08 Wave Guide



NEP08 Wave Guide



NEP08 Results

p	2	4	6	8	12	16	32	48
SGI Origin 3800								
C	410.3	212.2	140.4	98.2	72.7	55.3	29.5	26.5
$\frown{C}$	146.1	88.4	45.1	43.1	20.1	17.5	12.1	8.9
Acceleration	2.8	2.4	3.1	2.3	3.6	3.2	2.4	3.0
SGI Origin 2000								
C	940.1	378.6	230.4	164.4	110.7	85.2	47.0	40.0
$\frown{C}$	306.4	141.5	89.0	64.8	46.0	28.6	17.4	11.6
Acceleration	3.1	2.7	2.6	2.5	2.4	3.0	2.7	3.4

Multilayer Filtering (Details Missing)

- Solve a pair of 2D Poisson problems per layer on multiscale grids in order to pass information between layers. To be realistic, a sequence of inverse problems should be solved, too, occasionally.
- We developed a number of parallel Schur complement and parallel algebraic multigrid (AMGe style) methods. PEBBLES was highly modified and speeded up a lot during the work.

Filtering Numerics: Iterations

- AMGe* is the clear winner by iterations. The Schur-PCG method is not even close.
- Now when have I ever cared about anything other than wall clock time? (uh, oh)…

Method	Iterations
AMGe	100
Schur-PCG	278

* Linz style AMGe (not LLNL style)

Filtering Numerics: Iterations + Time

- Schur complement is the clear winner.
- AMGe does not provide speed up due to nasty properties of coarsened systems (non M-matrices)
- Oh, well… future research

Method	Iterations	Time
AMGe	100	26.10
Schur-PCG	278	5.36

Conclusions

- No explicit matrix produced in reduced Schur complement procedure
  - Local approximation of reduced system matrix required
  - Costs ~ 10 CG iterations, a throwaway cost.
- New preconditioner $\widehat{C}^{-1}$
  - Reduces Uzawa iterations by a factor of ~ 4
  - Reduces run time by a factor of ~ 3.
- New preconditioners that improve with respect to time discretization is part of active research.