

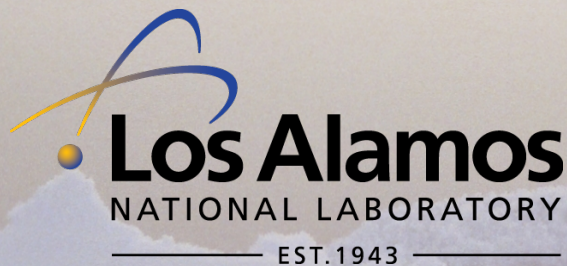
# MPAS-CICE development progress

Adrian Turner

COSIM

Los Alamos National Laboratory

Thanks to Doug Jacobsen, Mark Petersen, Todd Ringler, Elizabeth Hunke (LANL), Michael Duda and Bill Skamarock (NCAR)



U.S. DEPARTMENT OF  
**ENERGY**

Office of Science



# MPAS-CICE: Project Goals

- COSIM has developed an ocean model that uses an unstructured grid using the DOE/NCAR MPAS framework



- Traditionally sea ice models have used same grid as underlying ocean model
- CESM coupler requires this
- Need sea ice model that uses same grid as MPAS-O!
- Aim of project is essentially to port CICE to work on the MPAS-O grid

# MPAS

Model for Prediction Across Scales

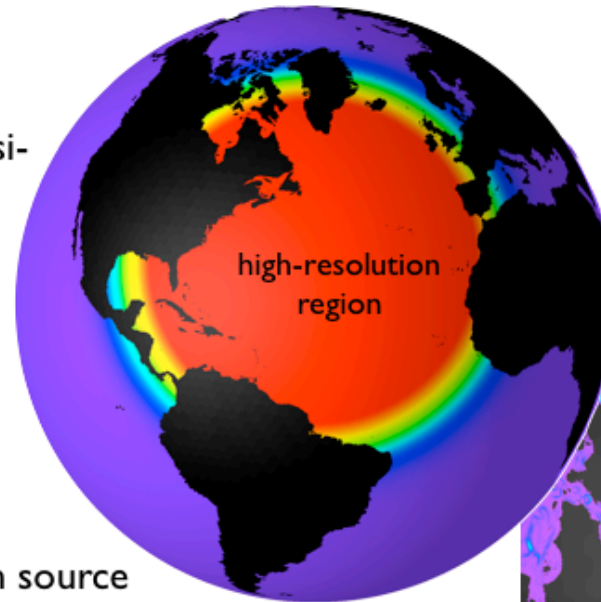
## Building a Global, Multi-Scale Ocean Model

MPAS-Ocean supports both quasi-uniform and variable resolution meshing of the sphere.

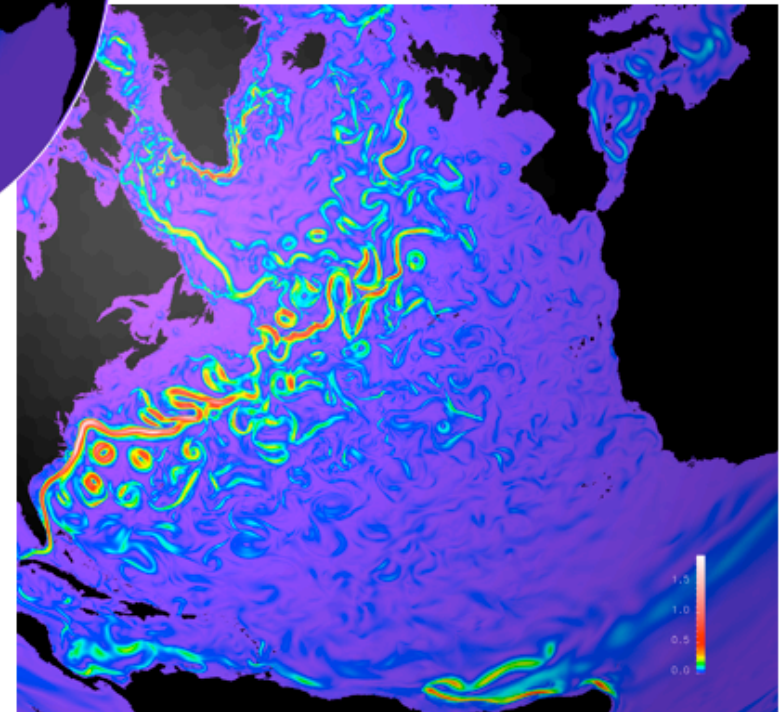
MPAS-O uses a generalized vertical coordinate that supports z-level, z-star, z-tilde, hybrid definitions.

MPAS-O will be released as open source software on Github in June 2013.

We hope to build partnerships within the international modeling community for the ongoing development and scientific application of MPAS-O.



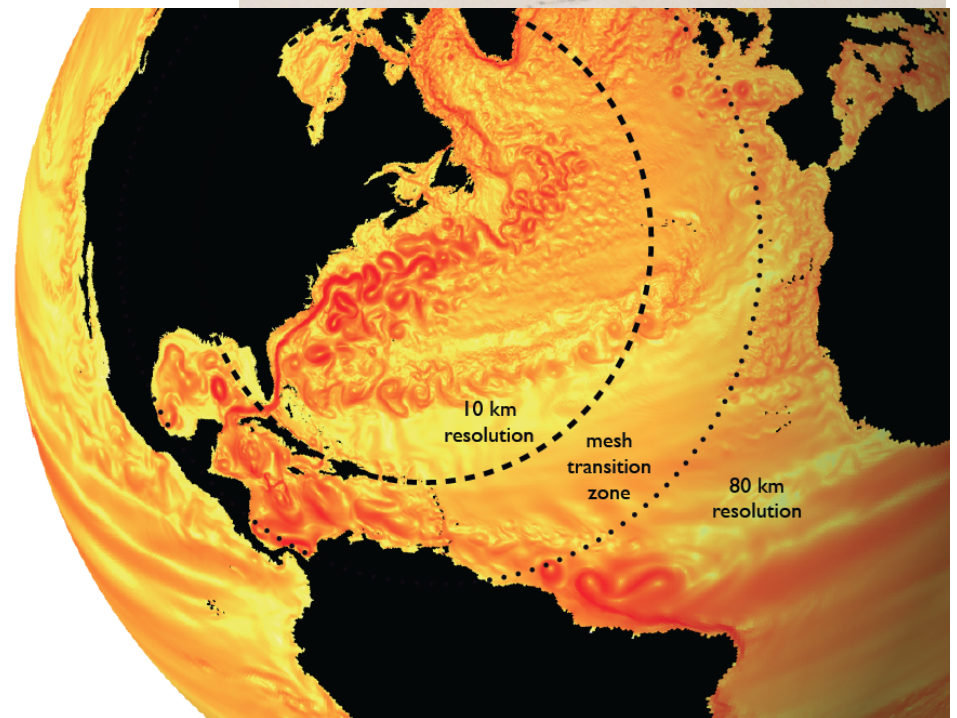
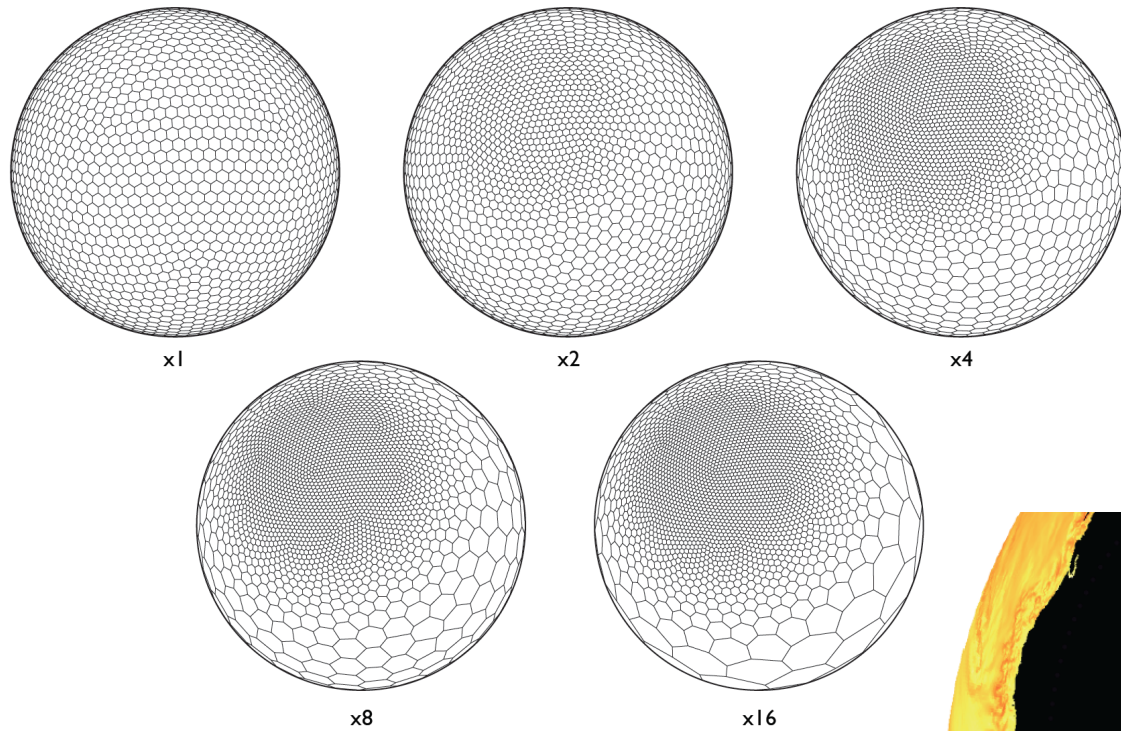
Below: Snapshot of kinetic energy from a global ocean simulation with 7.5 km resolution in the North Atlantic. The rest of the global ocean is resolved with a 40 km mesh.



Courtesy Todd Ringler (LANL)



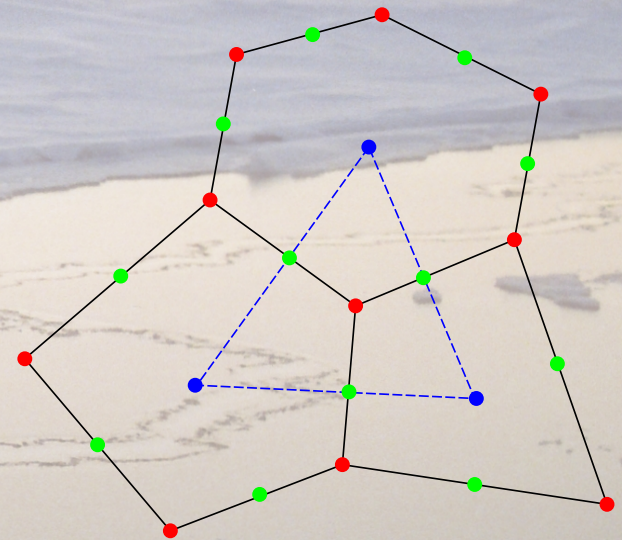
# MPAS grid allows variable resolution





# MPAS Grid

- Spherical Centroidal Voronoi Tessellation (SCVT)
- Voronoi Tessellation
  - Cell is formed from points closest to the generator points (cell centers)
  - Dual grid is Delaunay Triangulation
- Centroidal
  - Cell center is at the ‘centre of mass’ of the cell (achieved through iteration during generation)
- Supports quadrilateral grids
- Unstructured – variables stored as unordered 1D arrays not 2D structured arrays





# Anatomy of CICE

- Framework
  - IO
  - Coupling/Forcing
  - Block/processor communication
- Physics
  - Column physics (shortwave radiation, vertical thermodynamics, mechanical redistribution, ice thickness distribution,...)
  - Velocity solver
  - Advection



# MPAS Framework

- Framework
  - IO
  - Coupling
  - Block/processor communication
- Physics
  - Column physics (shortwave radiation, vertical thermodynamics, mechanical redistribution, ice thickness distribution,...)
  - Velocity solver
  - Advection



# Column Physics Package

- Framework
  - IO
  - Coupling
  - Block/processor communication
- Physics
  - Column physics (shortwave radiation, vertical thermodynamics, mechanical redistribution, ice thickness distribution,...)
  - Velocity solver
  - Advection



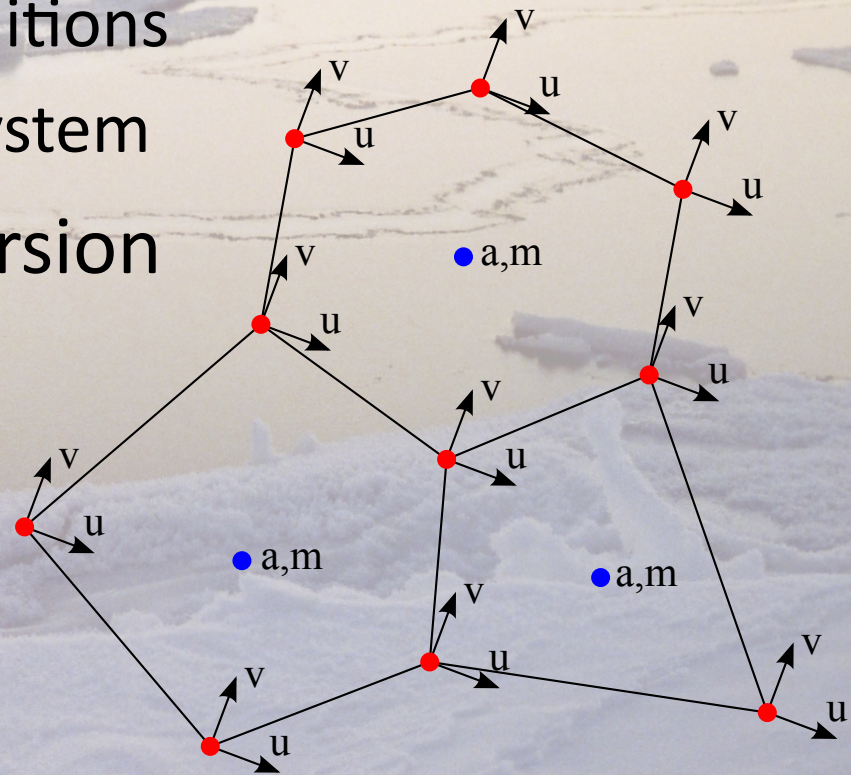
# New Development

- Framework
  - IO
  - Coupling
  - Block/processor communication
- Physics
  - Column physics (shortwave radiation, vertical thermodynamics, mechanical redistribution, ice thickness distribution,...)
  - Velocity solver
  - Advection



# Velocity solver: grid choice

- Use B-grid – both U and V velocity co-located at grid vertices
  - On quads becomes original CICE grid
  - Natural boundary conditions
  - Need local Cartesian system
- May develop C-grid version
  - More issues

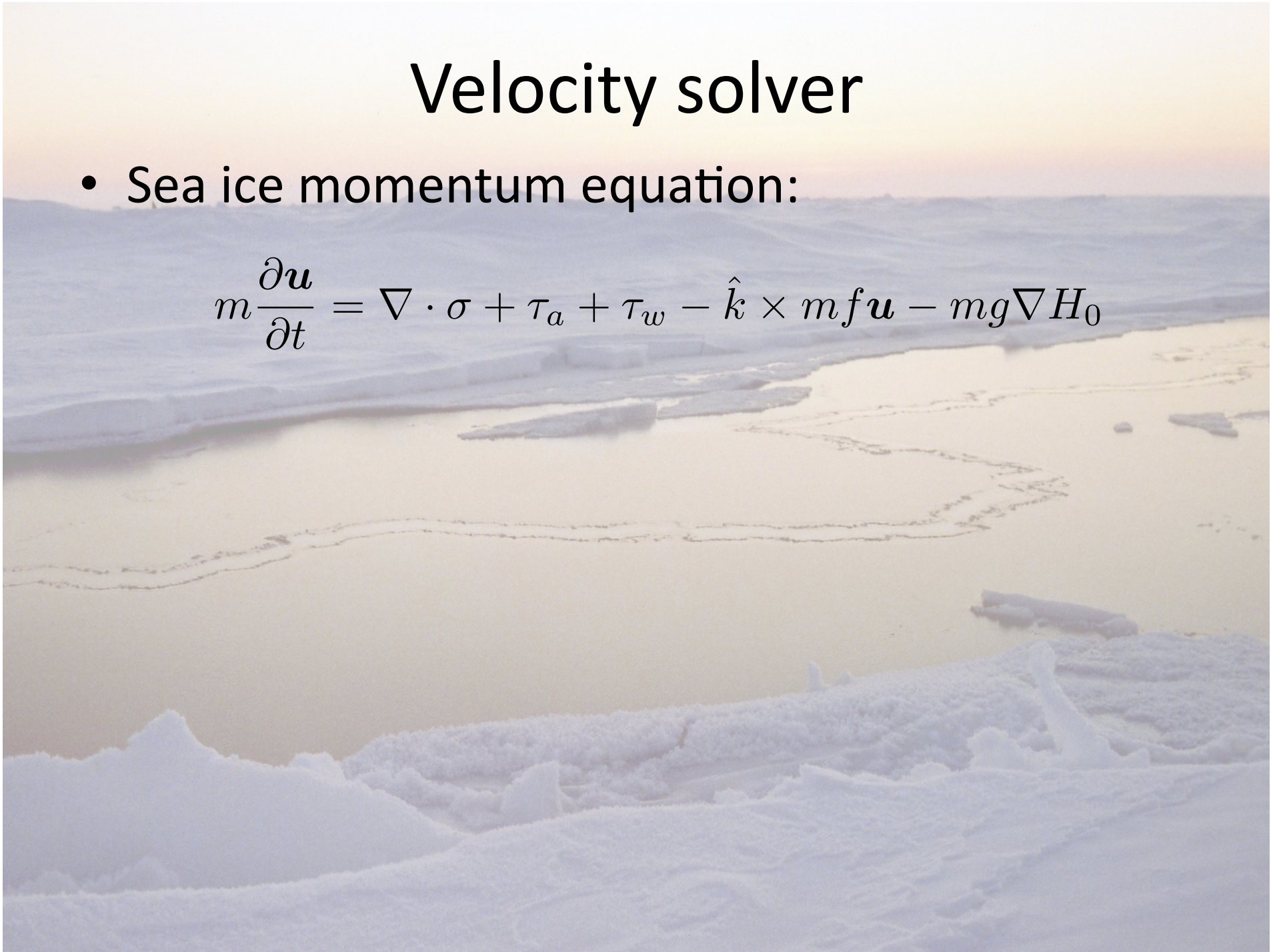




# Velocity solver

- Sea ice momentum equation:

$$m \frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{\tau}_a + \boldsymbol{\tau}_w - \hat{\mathbf{k}} \times m f \mathbf{u} - mg \nabla H_0$$





# Velocity solver

- Sea ice momentum equation:

$$m \frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{\tau}_a + \boldsymbol{\tau}_w - \hat{\mathbf{k}} \times m f \mathbf{u} - m g \nabla H_0$$

Inertia

Air stress

Coriolis Force

Internal stress

Ocean Stress

Sea surface tilt



# Velocity solver

- Sea ice momentum equation:

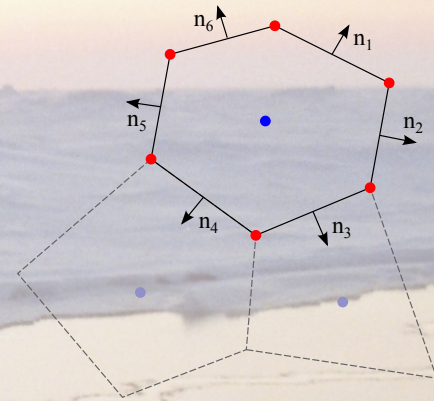
$$m \frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \boldsymbol{\sigma} + \tau_a + \tau_w - \hat{k} \times m f \mathbf{u} - m g \nabla H_0$$

- Only divergence of stress tensor depends on horizontal grid
  - Need to calculate the strain rate tensor from the velocity field
  - (Grid independent EVP constitutive relationship converts strain to stress)
  - Then need to calculate divergence of the stress tensor



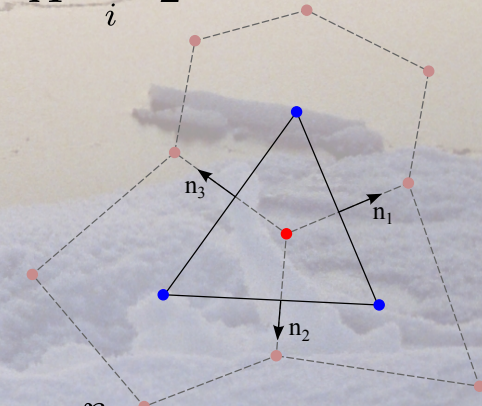
# Operators: Weak Method

- Line integrals
  - strain rate tensor
    - located at cell centers



$$\dot{\epsilon} = \nabla_s \mathbf{v} = \lim_{A \rightarrow 0} \frac{1}{A} \oint \frac{1}{2} [\mathbf{n}\mathbf{v} + \mathbf{v}\mathbf{n}] \partial l \quad \xrightarrow{\text{Discretize}} \quad \dot{\epsilon} = \frac{1}{A} \sum_i^{n_e} \frac{1}{2} [\mathbf{n}_i \mathbf{v}_i + \mathbf{v}_i \mathbf{n}_i] l_i$$

- divergence of stress operator
  - located at vertices



$$\nabla \cdot \boldsymbol{\sigma} = \lim_{A \rightarrow 0} \frac{1}{A} \int_C [\boldsymbol{\sigma} \cdot \mathbf{n}] \partial l \quad \xrightarrow{\text{Discretize}} \quad \nabla \cdot \boldsymbol{\sigma} = \frac{1}{A} \sum_i^{n_e} [\boldsymbol{\sigma}_i \cdot \mathbf{n}_i] l_i$$



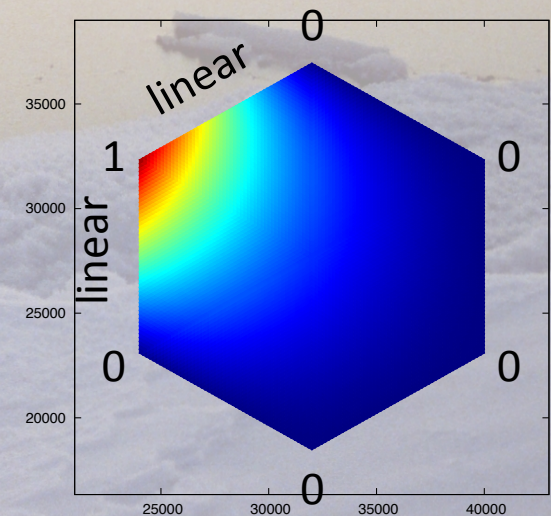
# Operators: Variational method

- Over whole domain: Work done by internal stress = Dissipation of mechanical energy

$$\int_{\Omega} v \cdot (\nabla \cdot \sigma) dA = - \int_{\Omega} (\sigma_{11} \dot{\epsilon}_{11} + \sigma_{22} \dot{\epsilon}_{22} + 2\sigma_{12} \dot{\epsilon}_{12}) dA$$

- Evaluate dissipation in terms of discretized velocities and take variation with respect to one of those velocity points
- Use basis functions,  $\phi$ , to evaluate dissipation across domain
- “Wachspress” basis to start
  - Expensive in memory
- Will explore “PieceWise Linear (PWL)” basis

$$\sum_i \phi_i = 1 \quad u = \sum_i u_i \phi_i \quad \frac{\partial u}{\partial x} = \sum_i u_i \frac{\partial \phi_i}{\partial x}$$

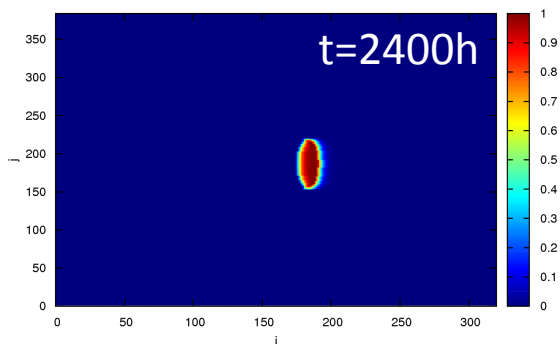
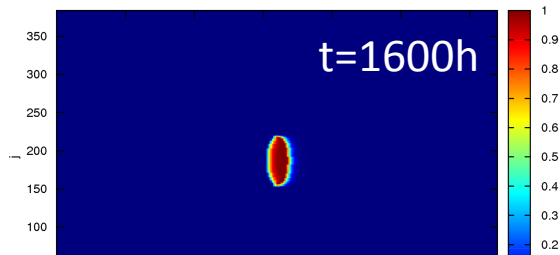
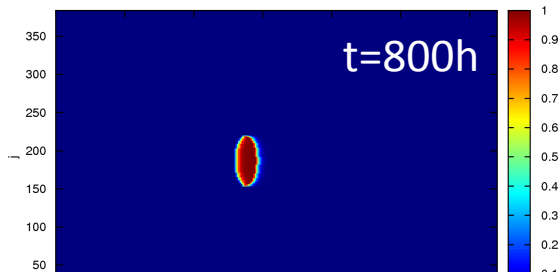
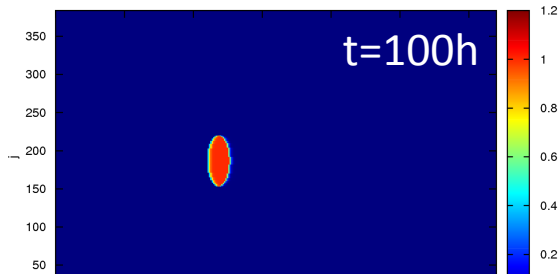




# Column physics

- Separate column physics from other parts of the code in CICE (e.g. IO/Dynamics/MPI)
- Package into “Column Physics Package” library
- Will be called from both MPAS-CICE and CICE as a separate library
- Available to other sea ice models – aim to be agnostic of what grid type called from
- This is a big task!
- Many questions remain on how to do this – just beginning scoping of work





# Advection

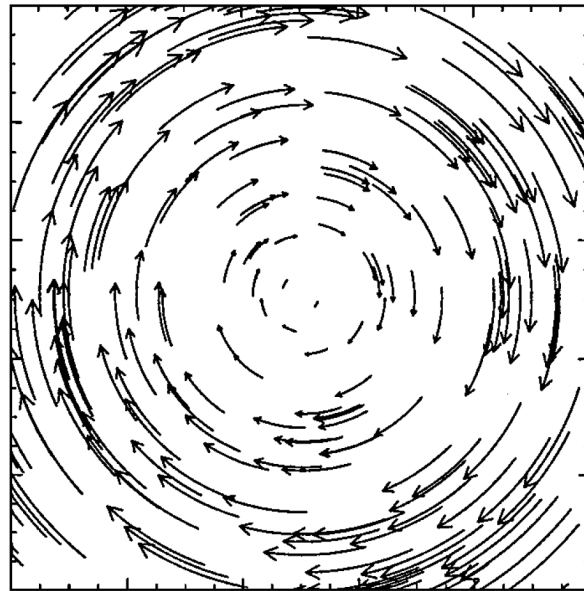
- Initially implemented MPAS FCT monotonic scheme
- True testing only after column physics tracers included
- May need to use other scheme (incremental remapping) for efficiency

One degree POP grid,  $v=1\text{m/s}$  eastwards

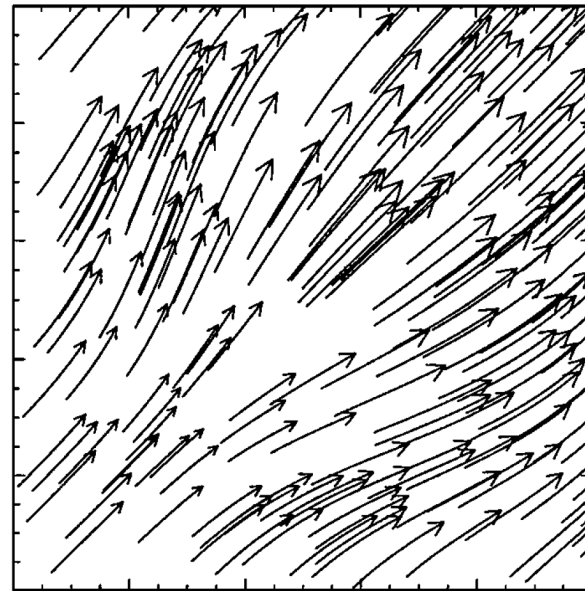


# Planar test case – Square domain

- 80km × 80km – Constant 2m thick ice with area varying linearly in u-direction from 0 to 1
- Circular ocean currents
- Uniform winds with added divergence



Ocean currents



Atmospheric winds

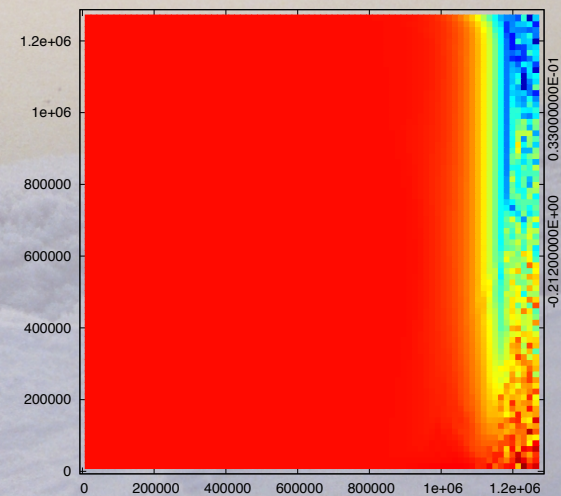
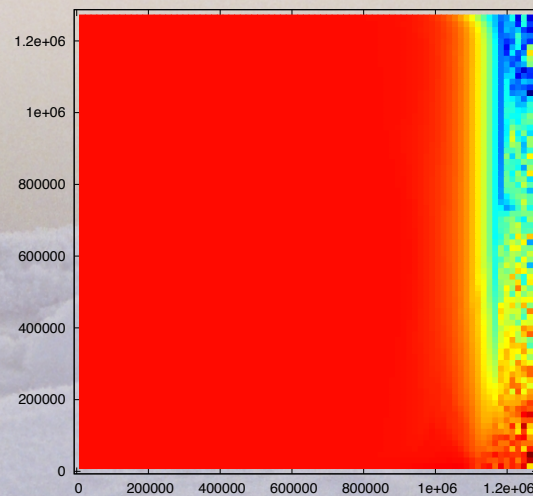
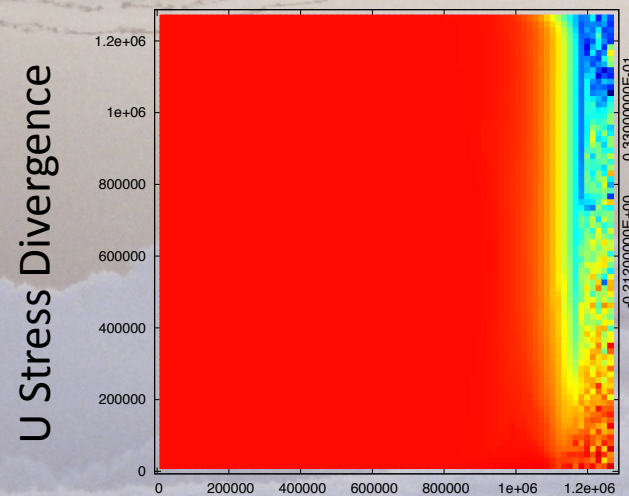
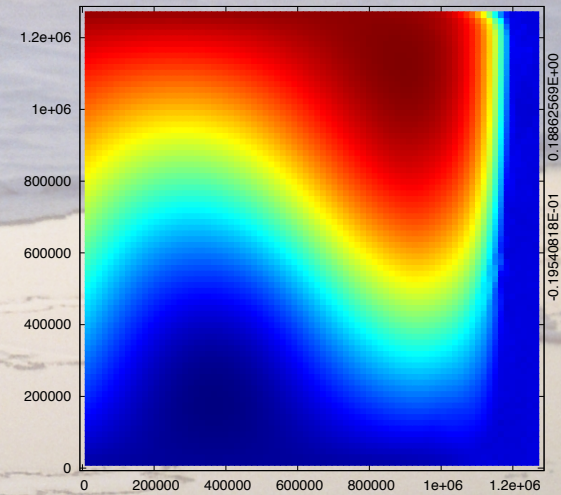
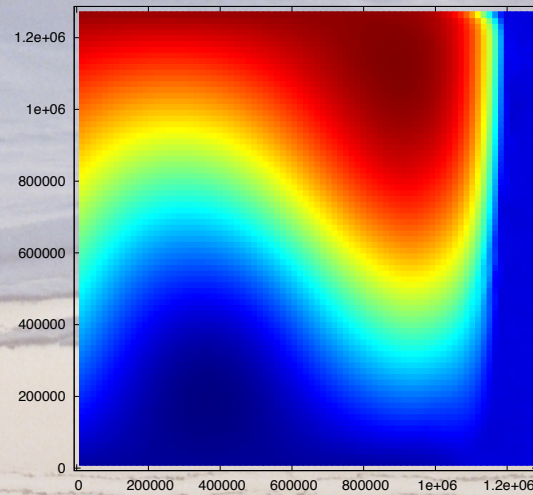
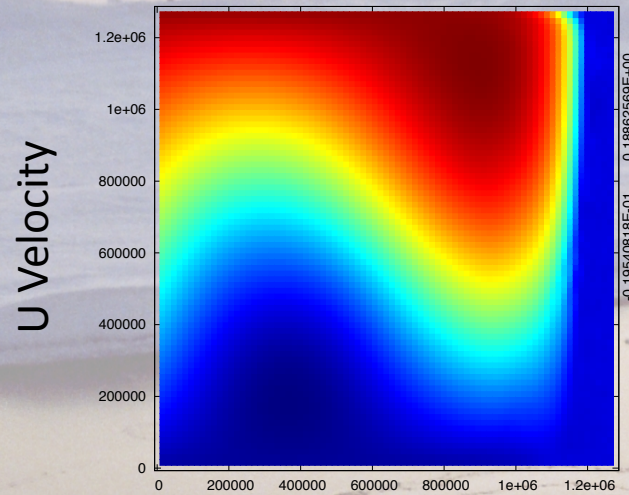


# Square test case: results I

CICE

MPAS: Variational

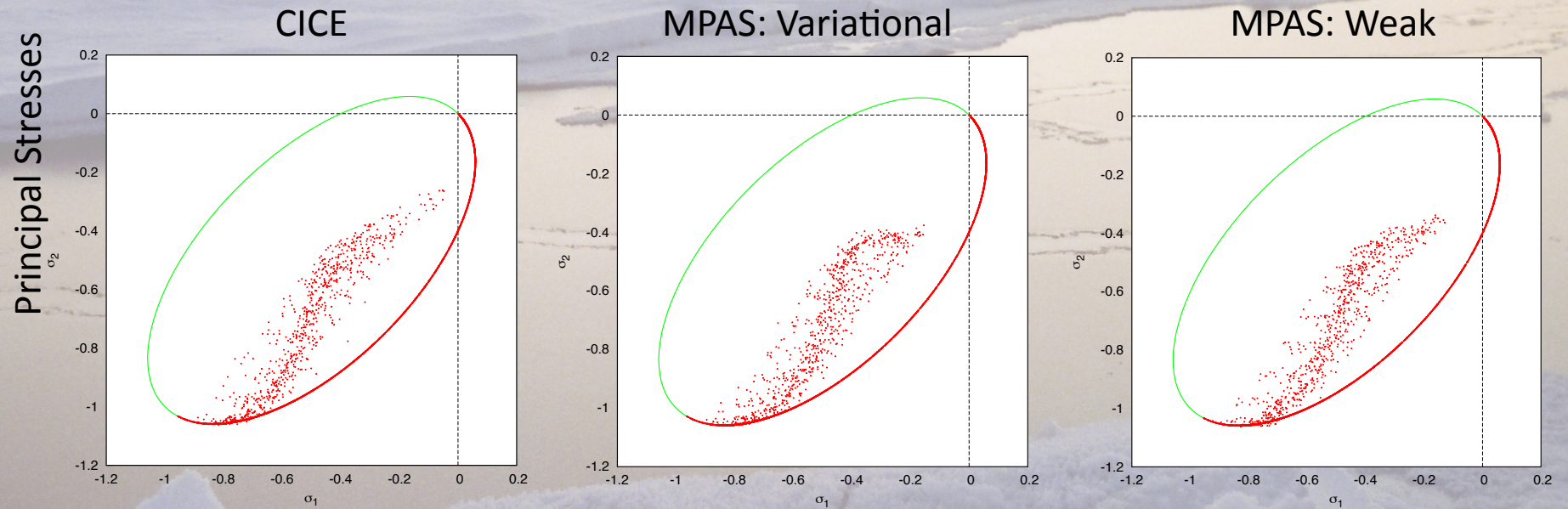
MPAS: Weak





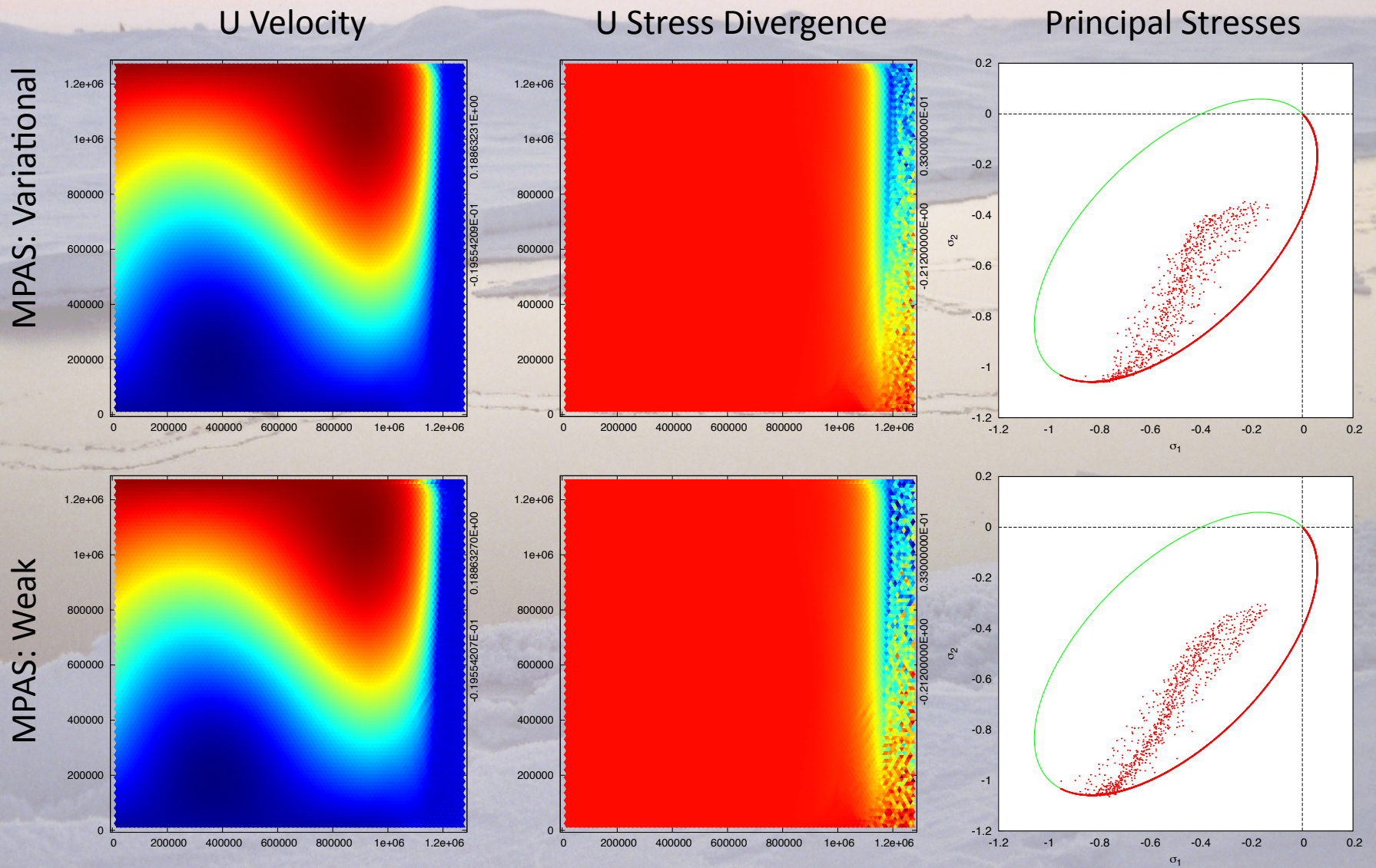
# Square test case: results II

- Can also look at the yield curve





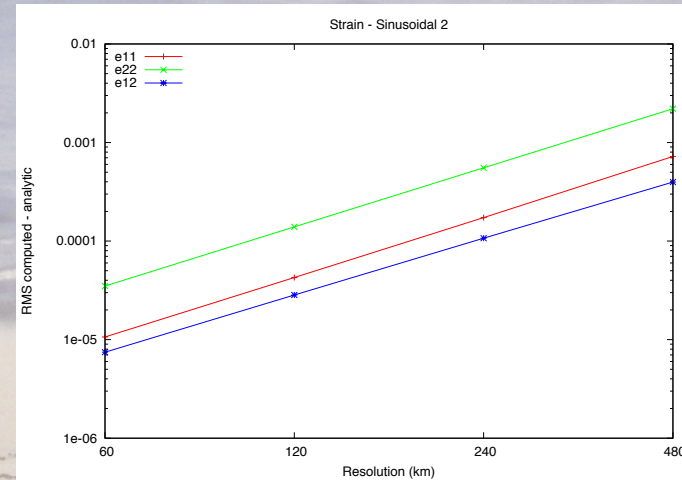
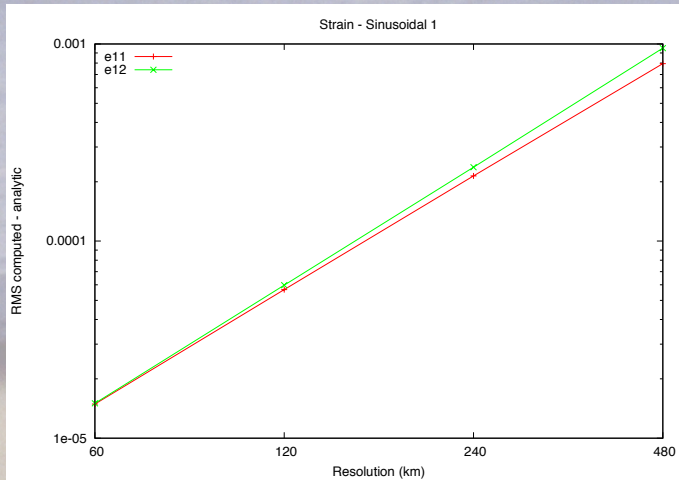
# Square test case: results III



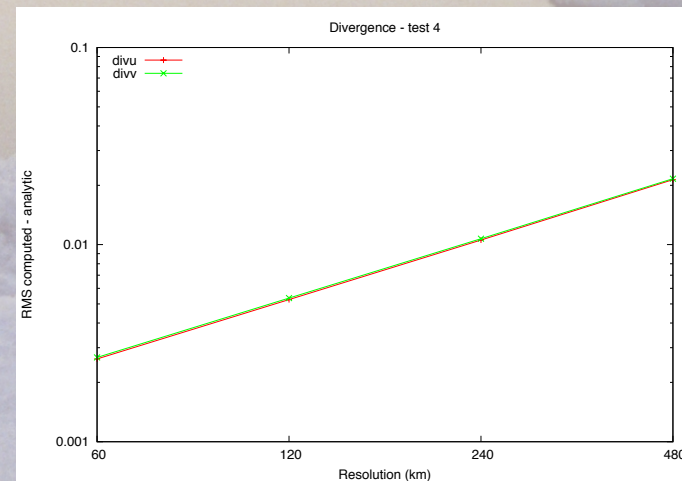
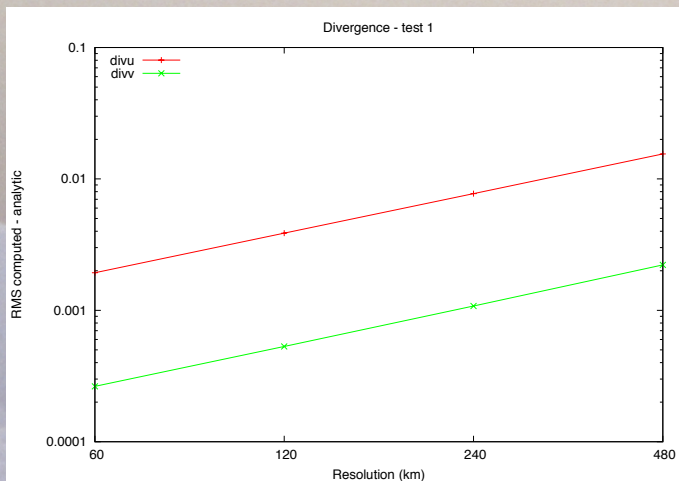


# Weak operator errors on sphere

- Quadratic convergence of strain rate error



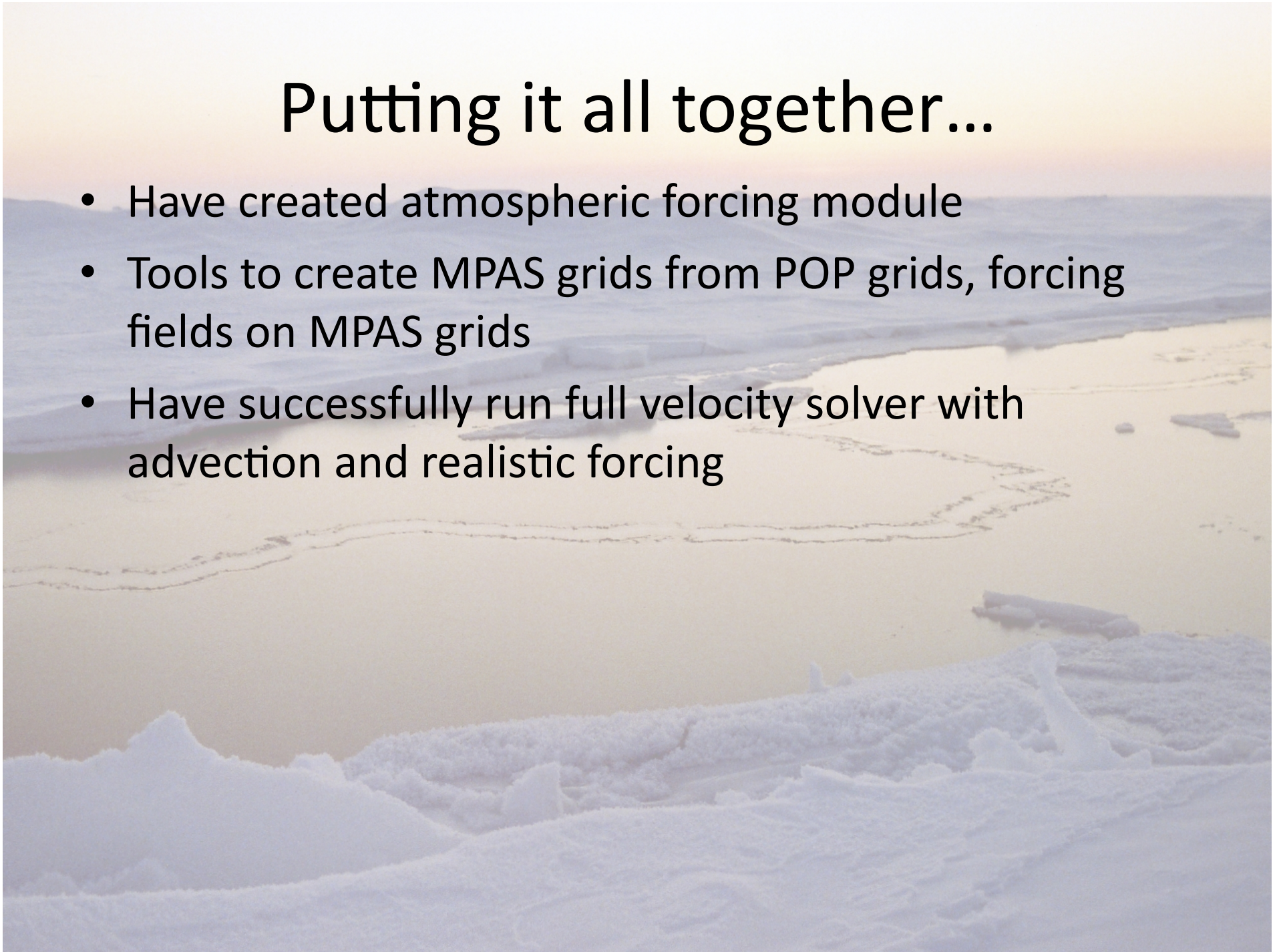
- Linear convergence of divergence of stress





# Putting it all together...


- Have created atmospheric forcing module
- Tools to create MPAS grids from POP grids, forcing fields on MPAS grids
- Have successfully run full velocity solver with advection and realistic forcing









A wide, flat, icy landscape under a pale sky, with a winding path of water or meltwater cutting through the ice. The word "Thanks!" is overlaid in the center.

Thanks!



# Variational divergence of stress

- Over whole domain work done by internal stress = equal to dissipation of mechanical energy

$$\int_{\Omega} v \cdot (\nabla \cdot \sigma) dA = - \int_{\Omega} (\sigma_{11} \dot{\epsilon}_{11} + \sigma_{22} \dot{\epsilon}_{22} + 2\sigma_{12} \dot{\epsilon}_{12}) dA$$

- Break domain into cells, with each cell associated with a  $v$  point

$$\sum_i \int_i (u F_1 + v F_2) dA = D(u_1, u_2, \dots, u_n, v_1, v_2, \dots, v_n)$$

- Assume velocity and divergence are constant in cell

$$\sum_i (u_i F_{1i} + v_i F_{2i}) A_{ui} = D(u_1, u_2, \dots, u_n, v_1, v_2, \dots, v_n)$$

- Take variation with respect to  $u_j$

$$F_{1j} = \frac{1}{A_{uj}} \frac{\partial}{\partial u_j} D(u_1, u_2, \dots, u_n, v_1, v_2, \dots, v_n)$$



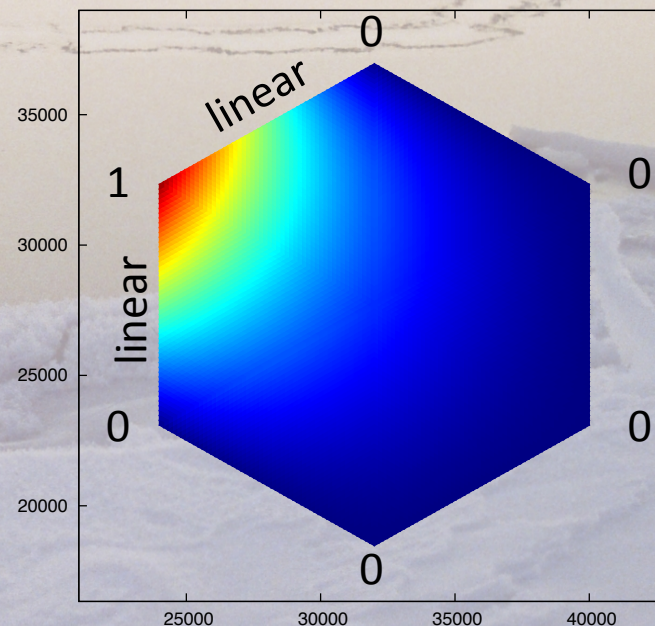
# Variational divergence of stress II

- Need to calculate global dissipation as function of discretized velocities – then take derivative
- Use basis functions,  $\phi_i$ , within the cells
- $u_i$  is velocity at vertex  $i$
- $\phi_i$  is 1 at vertex  $i$  and 0 at other vertices
- Use “Wachspress” basis functions to cope with arbitrary polygons

$$\sum_i \phi_i = 1$$

$$u = \sum_i u_i \phi_i$$

$$\frac{\partial u}{\partial x} = \sum_i u_i \frac{\partial \phi_i}{\partial x}$$





# Variational divergence of stress III

- Work through the math....
- Sum of terms like:

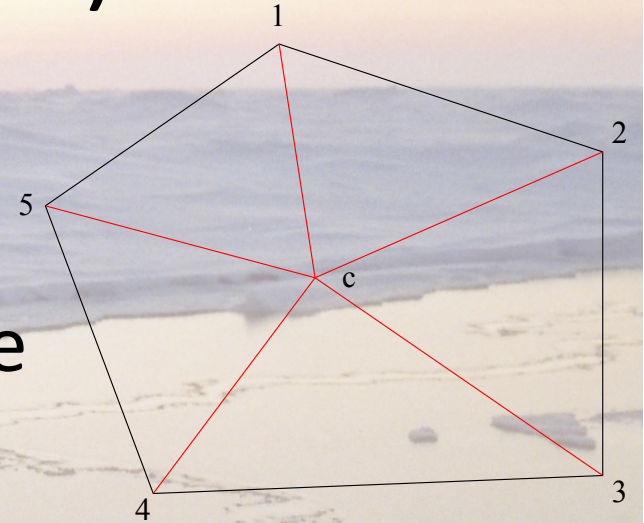
$$(\nabla \cdot \sigma)_j = \frac{1}{2} \sum_i^n \sigma_{1i} \underbrace{\int \phi_i \frac{\partial \phi_j}{\partial x} dx dy}_{S_x(i,j)}$$

- Calculate  $S_x$  at initialization – just geometrical factor of unmoving grid
- Requires quite a bit of storage – looking at PieceWise Linear (PWL) basis as alternative to Wachspress



# PieceWise Linear (PWL) basis

- Alternative to Wachpress
- Break polygon into triangles
- Linear basis within each triangle
- Can save lots of memory



$$\mathbf{x}_c = \sum_i^n \alpha_i \mathbf{x}_i \quad \sum_i^n \alpha_i = 1$$

- $\alpha_i$  is weight at vertex  $i$
- $\varphi_i$  equals  $\alpha_i$  at position  $c$
- Choose  $\alpha_i = 1/n$  to start