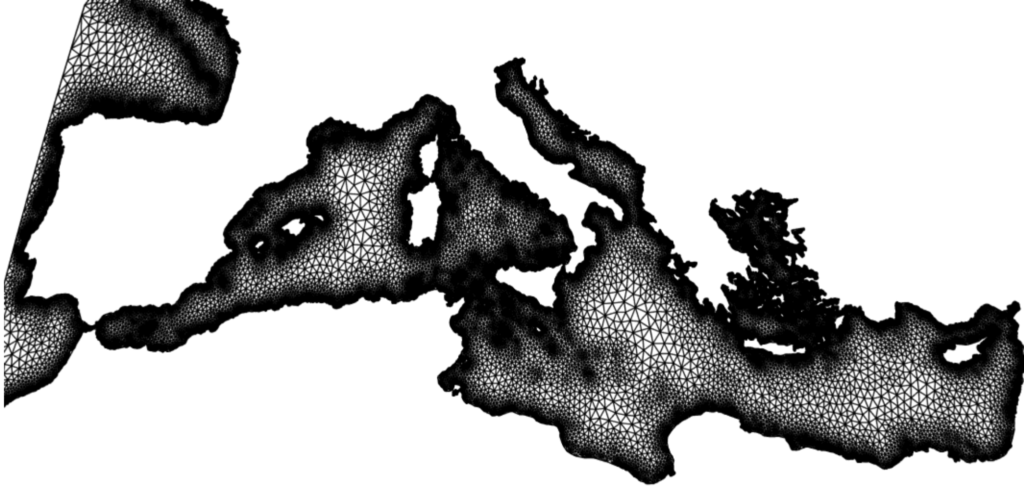
Imperial College London



Modelling multiple scales in geophysical systems using adaptive grids

Matthew Piggott
m.d.piggott@imperial.ac.uk

Applied Modelling and Computation Group & Grantham Institute for Climate Change
Department of Earth Science and Engineering, Imperial College London, UK

Acknowledgements: Chris Pain, Stephan Kramer, Rhodri Nelson, Patrick Farrell, Colin Cotter, David Ham, Gerard Gorman, Hannah Hiester, Flora MacTavish, Simon Funke, Alastair McVicar, Adam Candy, Alex Avdis, Yusuke Oishi, Axelle Vire, Fangxin Fang, James Maddison, David Marshall, Gareth Collins, Christian Jacobs, Simon Mouradian, Rhodri Davies, Peter Allison, Tim Bond, ...

Plan

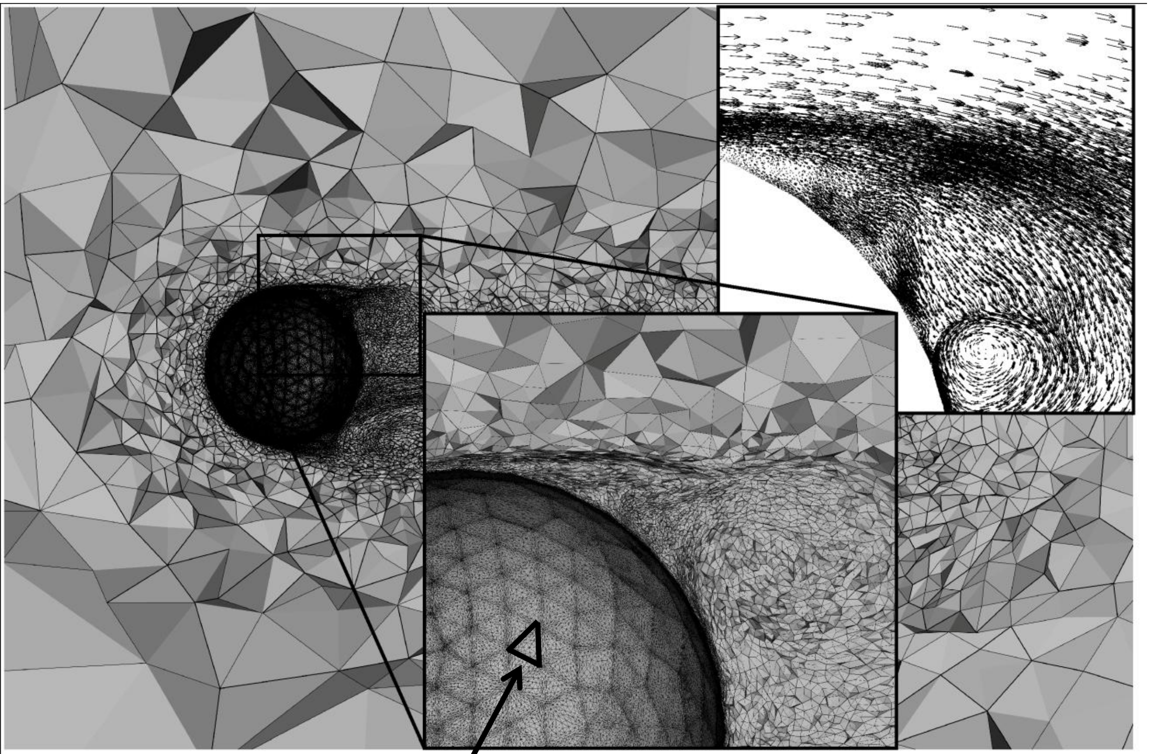
- Motivation
- Adaptivity via mesh optimisation
  - Mesh optimisation operations
  - Metric calculation
  - Mesh-to-mesh interpolation
  - Parallelisation
- Discretisations
- Mesh anisotropy
- Geophysical balance and aspect ratios

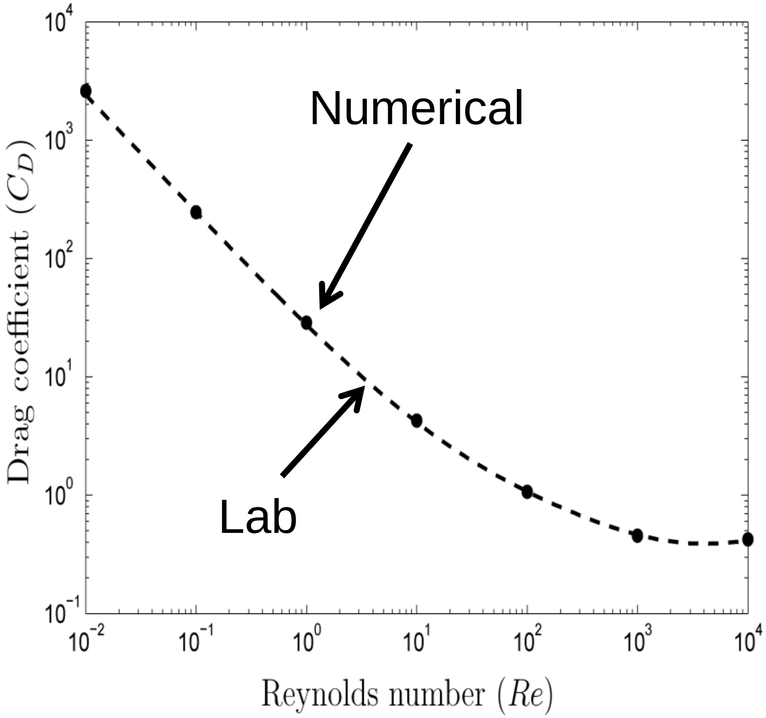
- Free surface and inundation
- Compressible
- Multi-material
- Multi-phase

Motivation

- Interested in tough problems characterised by multiple scales, complex geometries, coupled multi-physics, ..., e.g.:
  - linking estuary, coastal, shelf and deep ocean regions;
  - explicitly representing (more, not all) processes crucial to the overturning circulation;
  - merging industrial CFD and GCM-type applications of numerical models
- Flexibility of unstructured meshes and the power of finite element/control volume methods
- BUT these come with additional computational costs!
- Can we off-set the extra costs by making maximal use of mesh flexibility?
- For transient systems this points to the use of dynamic mesh adaptivity in response to the discrete system's changing resolution requirements
- Use/develop a highly flexible modelling approach (mesh, discretisation, physics, coding) to allow a wide range of industrial & geophysical systems to be simulated

CFD motivation/validation: drag calculation in flow past a sphere at a range of Re numbers





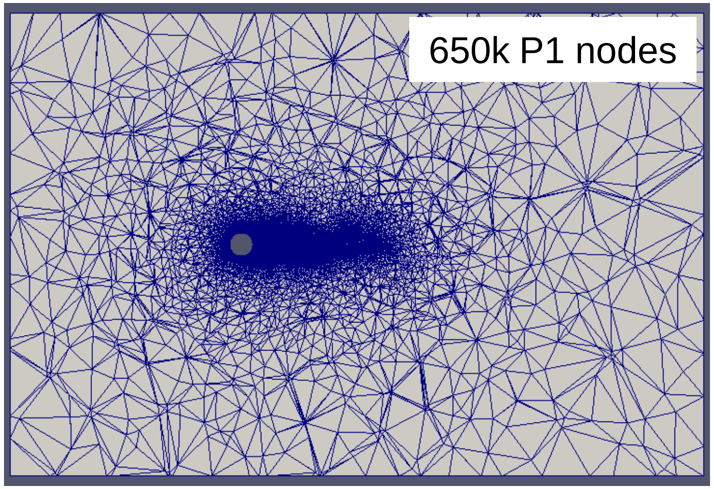
Resolution of the geometry/initial mesh is preserved through simulation, more resolution is added on original facets – conserving domain volume

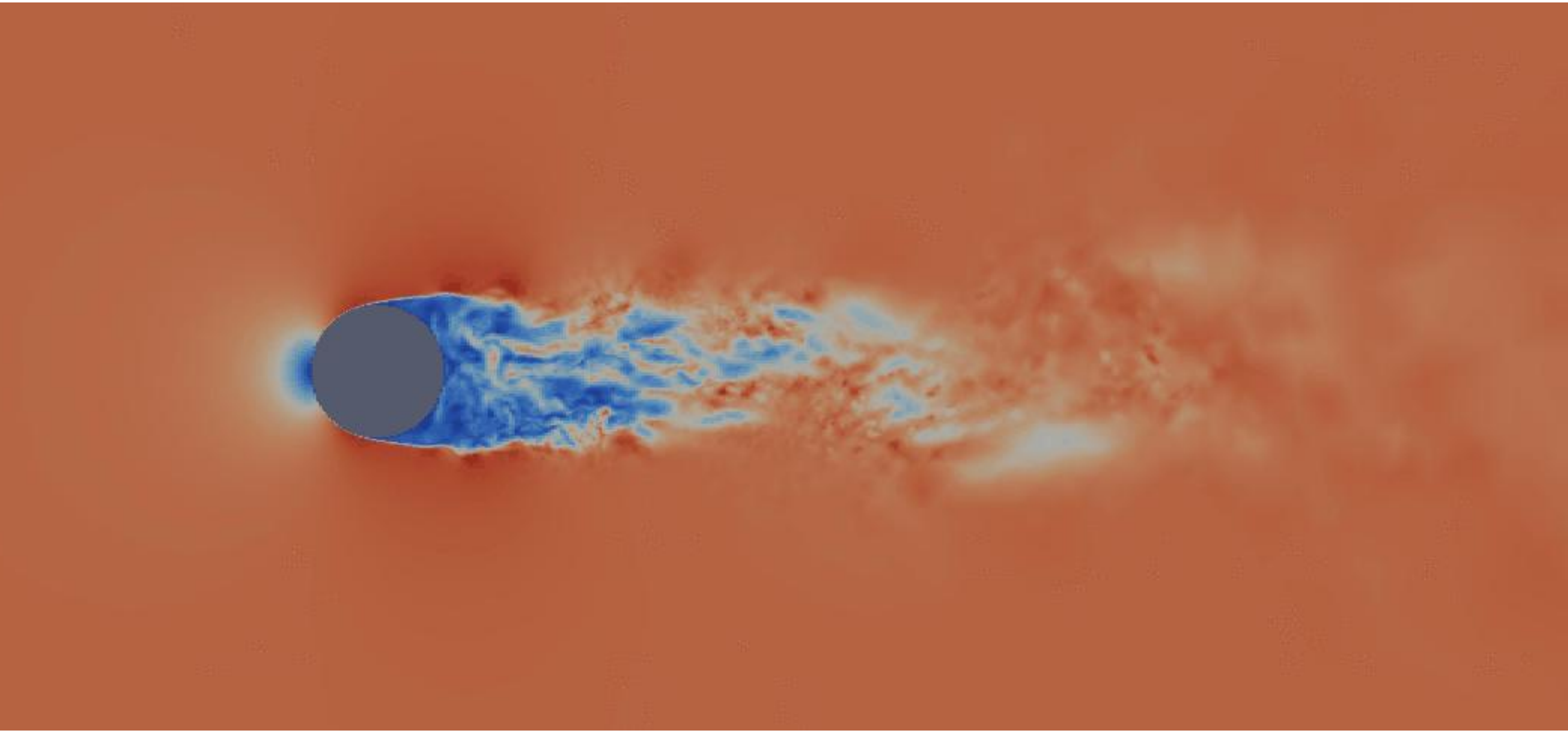
Computed drag coefficient compared against correlation (from Brown and Lawler, 2003) with lab data

$$C_D = \frac{24}{Re}\left(1 + 0.15 Re^{0.681}\right) + \frac{0.407}{1 + \frac{8710}{Re}}.$$

Movie of the velocity magnitude on a slice

Error metric guiding adaptivity has been chosen here to focus resolution on the boundary layer and separation region, not on the wake

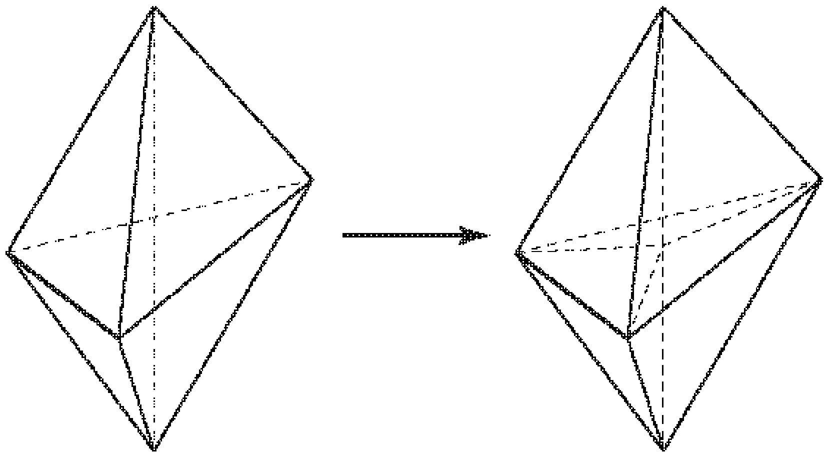




Mesh optimisation operations: altering the node-count

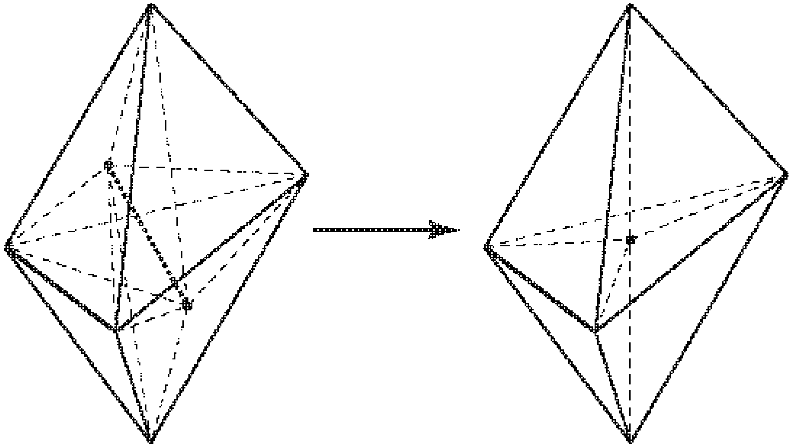
<u>Edge split</u>

If an edge length is too large split it and create new elements.
For simplicity here place the new node at the mid-point of the edge.



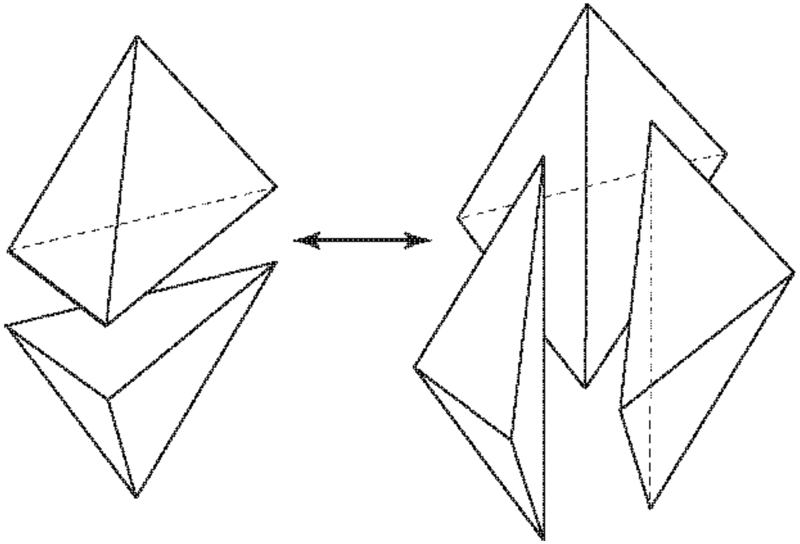
<u>Edge collapse</u>

If an edge length is too small collapse it and delete all elements that contained the edge.
For simplicity here collapse to the mid-point of the edge, unless (internal or external) geometrical constraints apply.

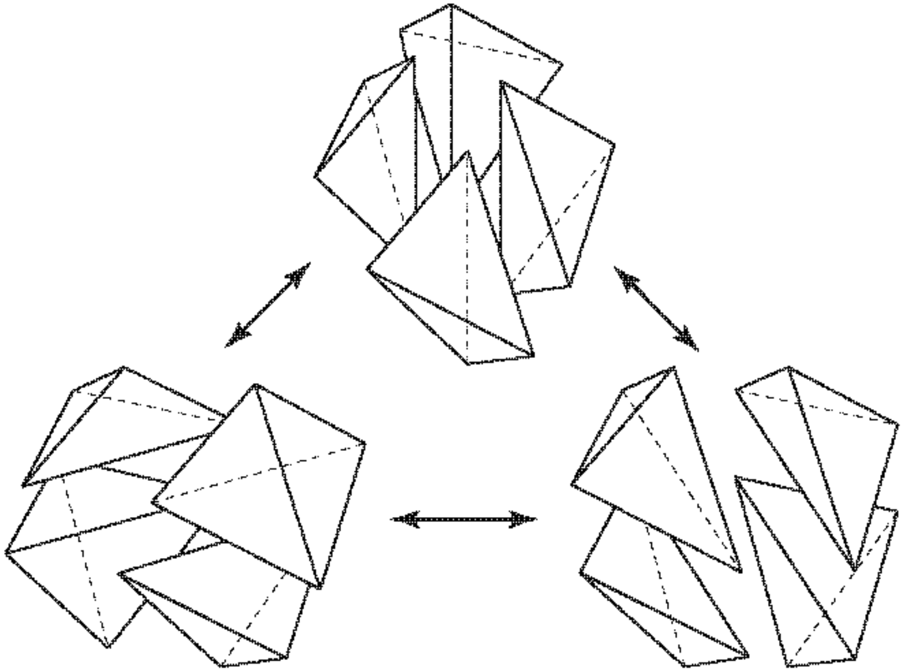


Mesh optimisation operations: altering element alignment
Very important for anisotropy, non-hierarchical approach

<u>Edge-to-face and face-to-edge swapping</u>
For two elements sharing a face whose union is convex, delete the face and connect the other two nodes, creating three elements.
If three elements surround an edge the inverse operation can be considered.



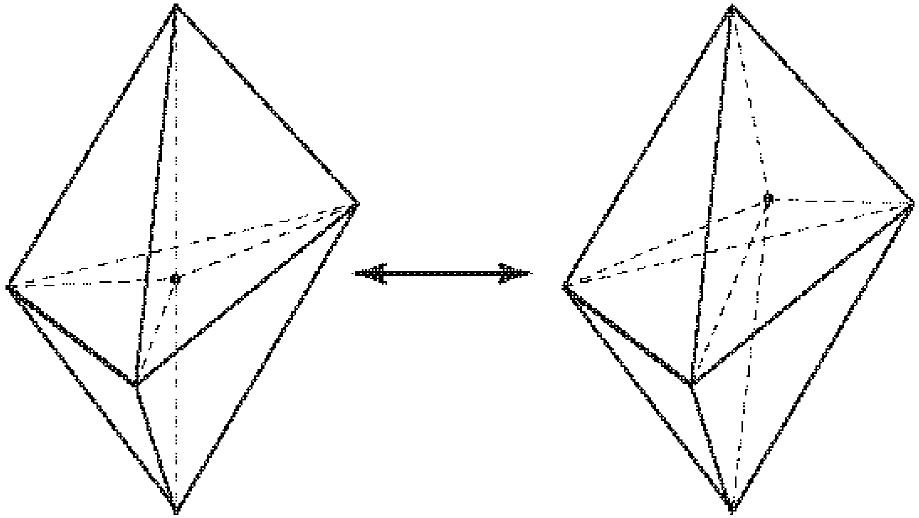
<u>Edge swapping</u>
For four elements surrounding an edge consider deleting that edge and replacing with one of the other two options. The number of elements is preserved.



NB. These operations do not change the node-count and make minimal changes to the element-count, but importantly they do change the anisotropic resolution

Mesh optimisation operations: smoothing

<u>Node movement</u>
Move nodes so as to improve the quality of the surrounding elements.
Also allow movement constrained to interior or exterior surfaces



In combination, the above operations alter both the resolution and alignment of the mesh, yielding a mesh closer to optimal in terms of required edge lengths

See Pain et al., 2001 for further details of the approach used here.
And work by George, Freitag, Ollivier-Gooch, Shephard, ... for other similar approaches and improvements.

NB. We are currently developing a new OpenMP based optimisation library in 1,2,3D for hybrid (OpenMP+MPI) use – large number of functional evaluations and topological operations offers scope for significant speed-ups

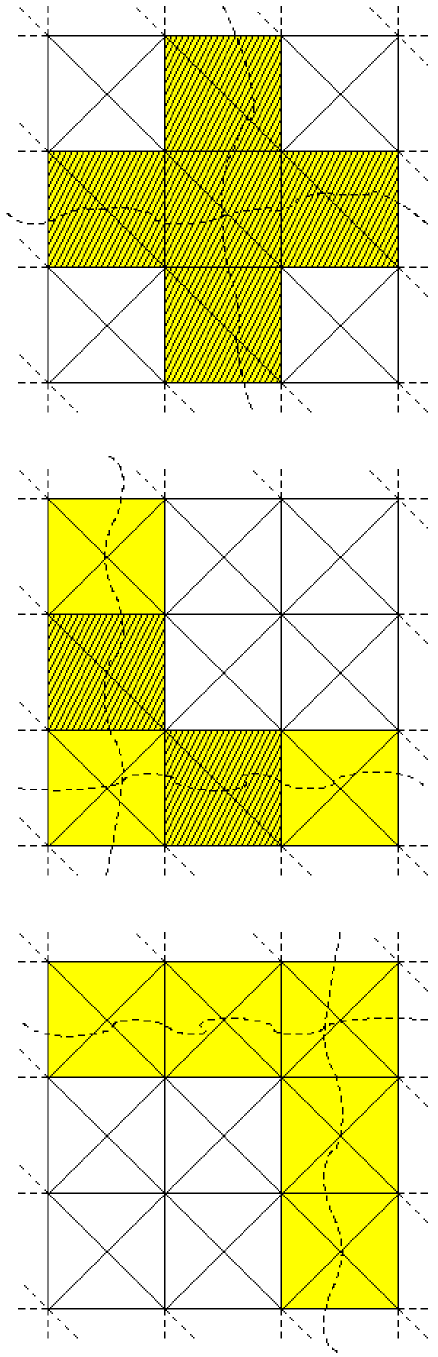
MPI parallelisation of mesh optimisation

- Lock halos: disallow all operations affecting halo elements

- Each process optimises elements it owns
  - Some processes will now have more nodes/elements than others – poor load-balancing

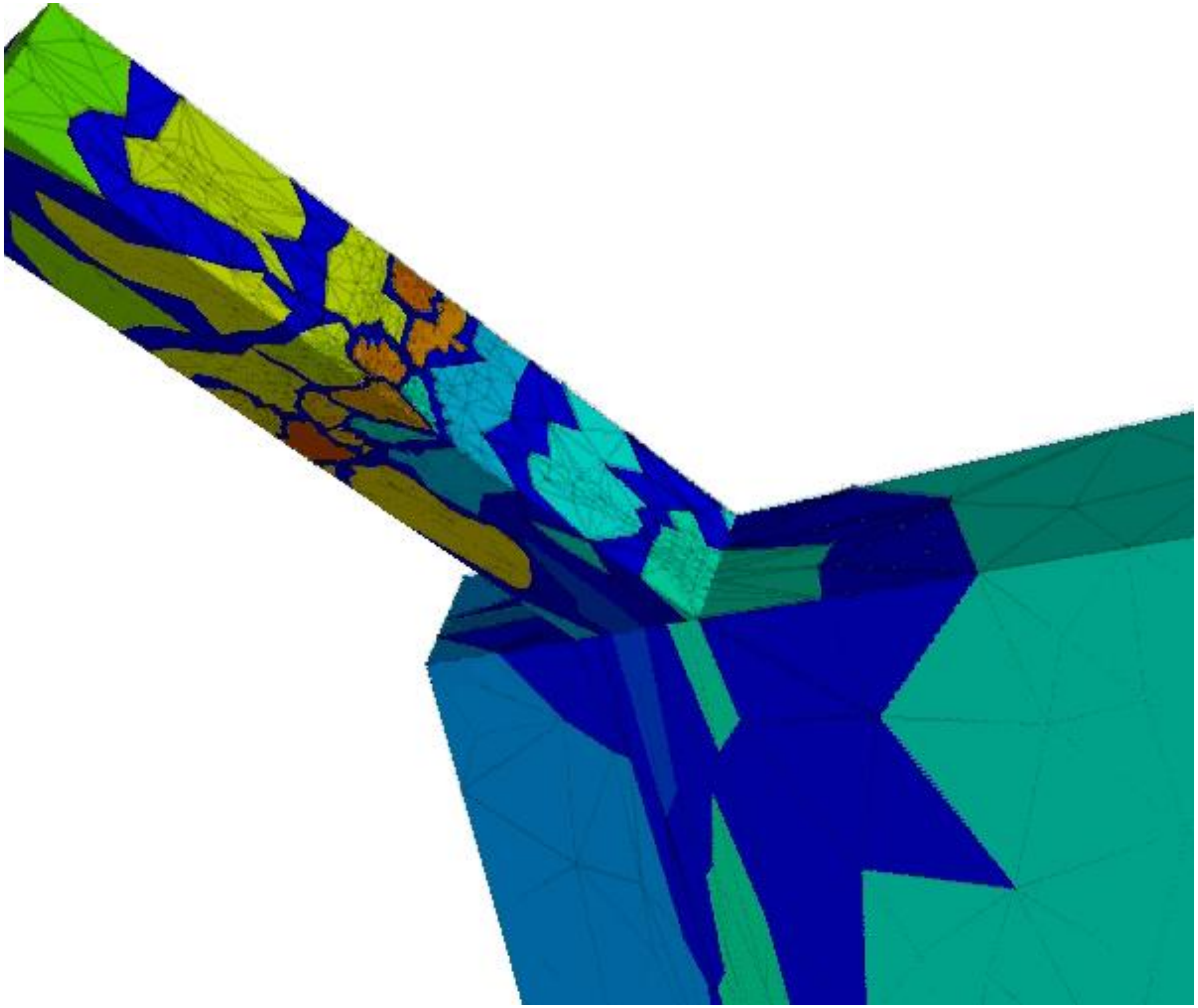
- Perform load-balancing and data migration (Zoltan)
  - Optimise the equality of work across processes
  - Subject to the constraint of a small edge-cut (to minimise communication)
  - Weight the edges in terms of element quality – dissuade the new partition from going through previously un-optimised (halo) elements

- Repeat
  - To ensure all elements have been considered for optimisation



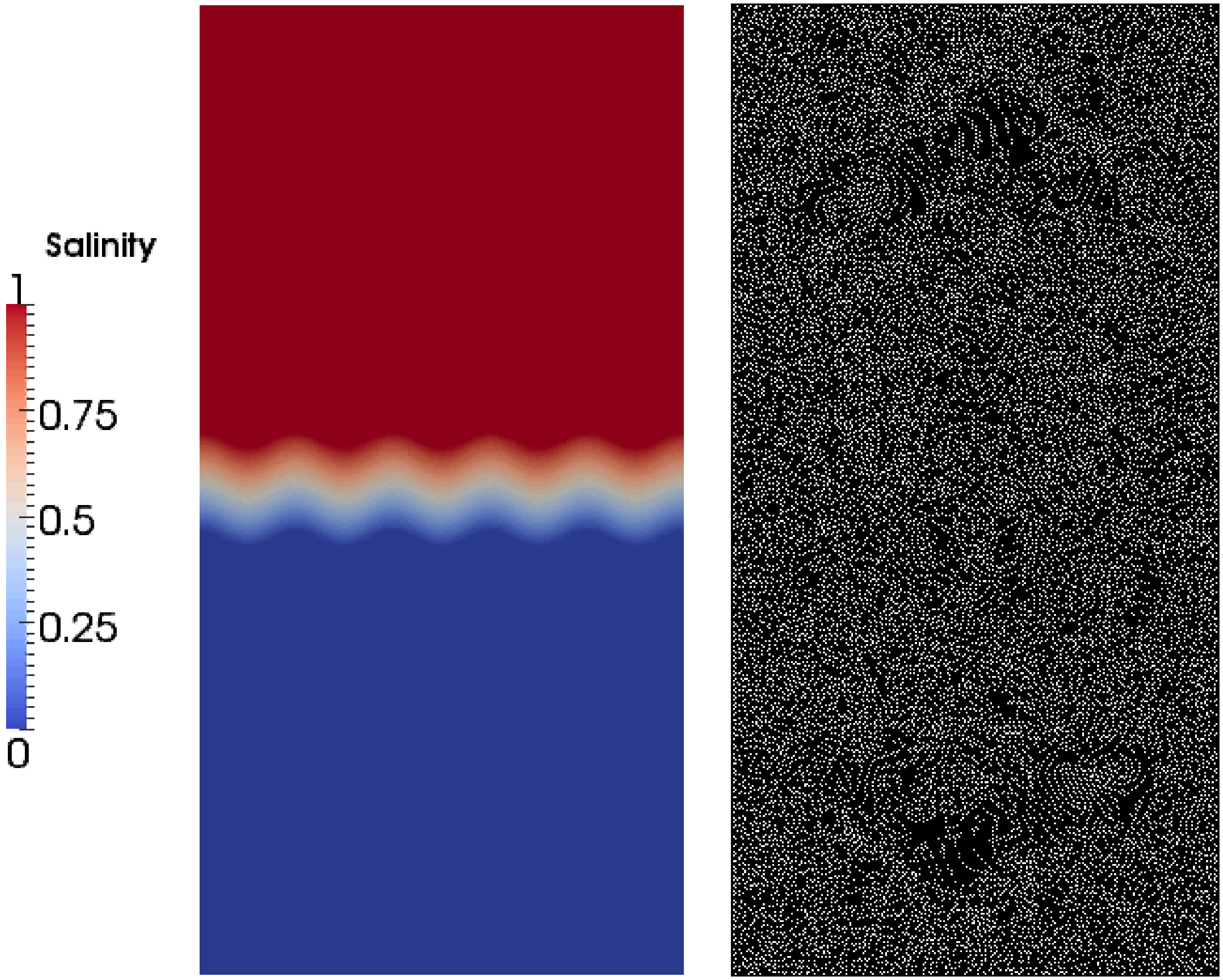
This combines the parallelisation of adaptivity with load-balancing. Data migration costs shown to be minimal, later optimsation iterations have less work

Dynamically load-balanced parallel adaptivity example (blue = halo)



Aside: a similar boundary perturbation approach has been used to allow us to perform mesh adaptivity on periodic domains

(lots of ‘corner cases’ to consider with parallel+adaptive+periodic!)



How to describe desired edge length: Interpolation errors and metrics

- Interpolation error of scalar u over element e is given by

$$\epsilon_e \equiv ||u - \Pi_h u||_{\infty,e} \leq \tilde{\gamma} \max_{\boldsymbol{x}\in e} \max_{\boldsymbol{v}\subset e} \{\boldsymbol{v}^T |H(\boldsymbol{x})| \boldsymbol{v}\}$$

Basically a max curvature times the size of the element squared – cf. 1D

$$H = \nabla^T \nabla u = Q\Lambda Q^T, \quad |H| := Q|\Lambda|Q^T$$

where H is the Hessian matrix associated with u. Straightforward extension for L_p norms.

- For practical use: drop the (O(1) mesh independent) constant, replace the vectors with element edges, and assume we have an edge/element centred H

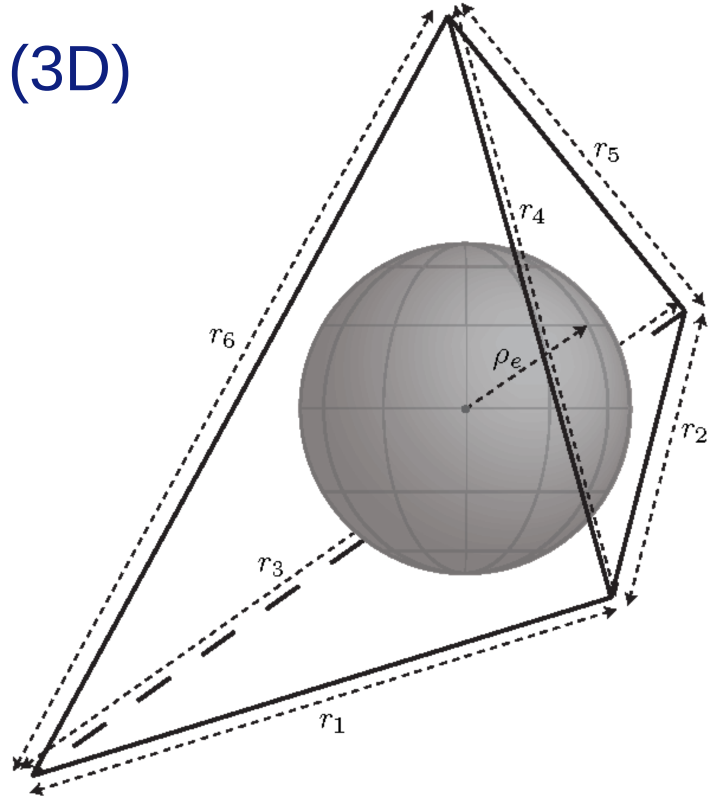
- Equi-distribution of a (user-defined) interpolation error now looks like

$$\epsilon = \boldsymbol{v}^T |H(\boldsymbol{x})| \boldsymbol{v} \iff 1 = \boldsymbol{v}^T M \boldsymbol{v} = ||\boldsymbol{v}||_M^2, \quad M := \frac{|H|}{\epsilon}$$

- We now have the definition of a metric space where equi-distribution of errors in physical space is equivalent to having a mesh of unit equilateral triangles/tetrahedra in metric space

- Use this definition of length in a mesh generator, or a mesh optimisation algorithm

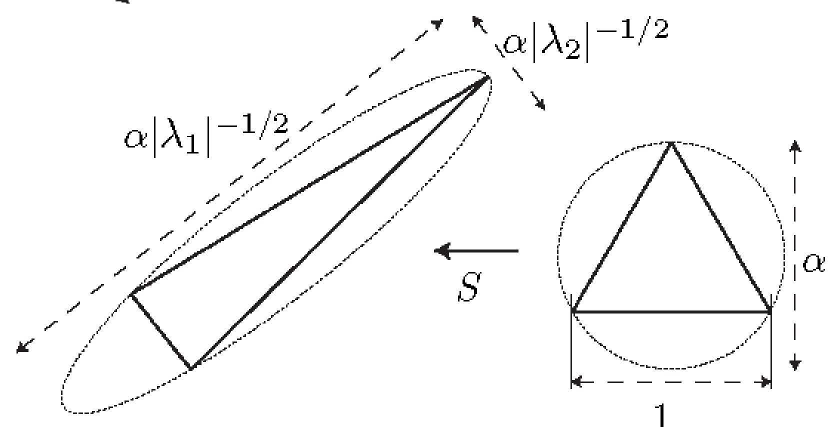
Formulating the optimisation problem (3D)

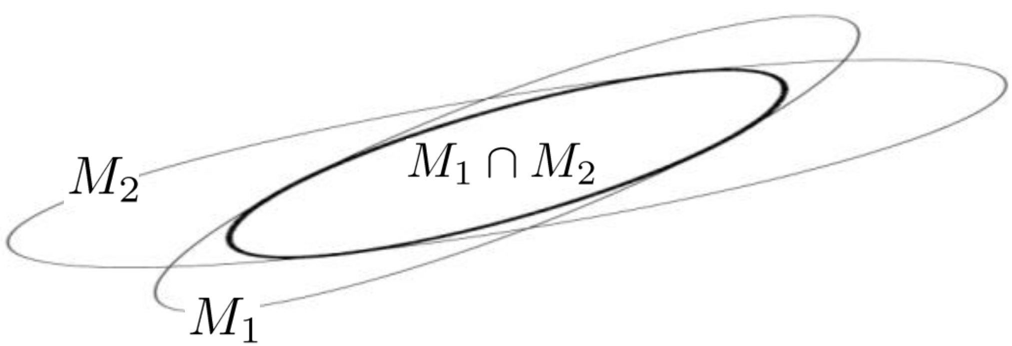


- Form an optimisation functional, in 3D we use:

$$F_e = \sum_{l \in \mathcal{L}_e} (r_l - 1)^2 + \left(\frac{\alpha}{\rho_e} - 1 \right)^2$$

- First term ensures good edge lengths of an element e, second term helps with element shape
- Evaluate lengths with respect to the metric $M = \dfrac{|H|}{\epsilon}$
- Minimise the functional through the above series of local mesh operations
- NB. total metric is formed by combining metrics for multiple solution fields, with user-defined weightings for each, and applying constraints on max/min edge lengths, max. number of nodes and gradation to ensure smooth variation of mesh sizes

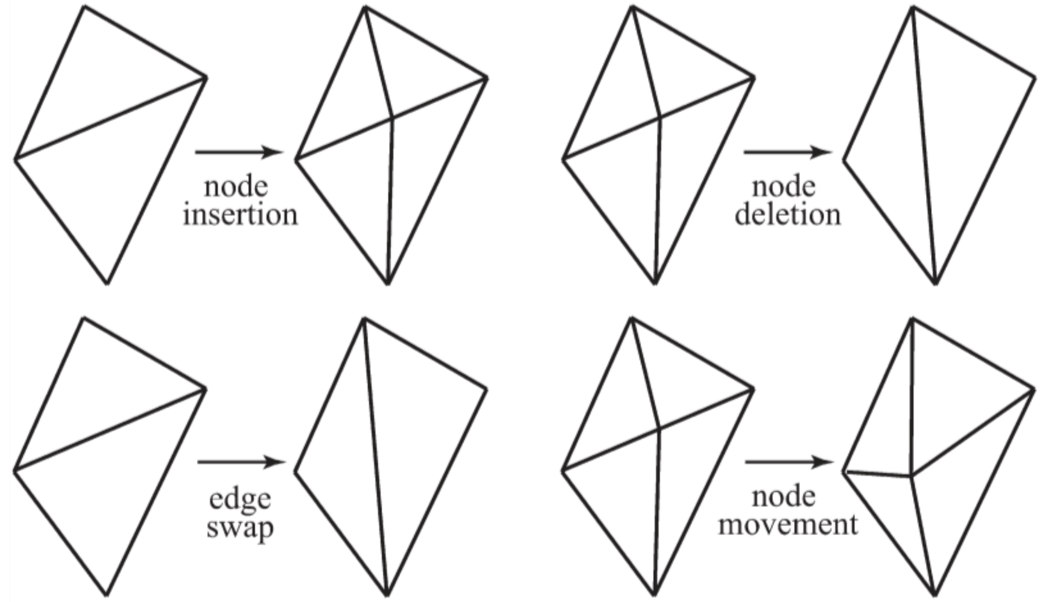




Following rotation to coordinate axis, edge lengths still satisfy:

$$\Delta x^2 \frac{\partial^2 u}{\partial x^2} = \Delta y^2 \frac{\partial^2 u}{\partial y^2} = \epsilon$$

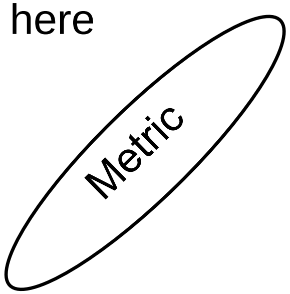
Two-dimensional mesh optimisation

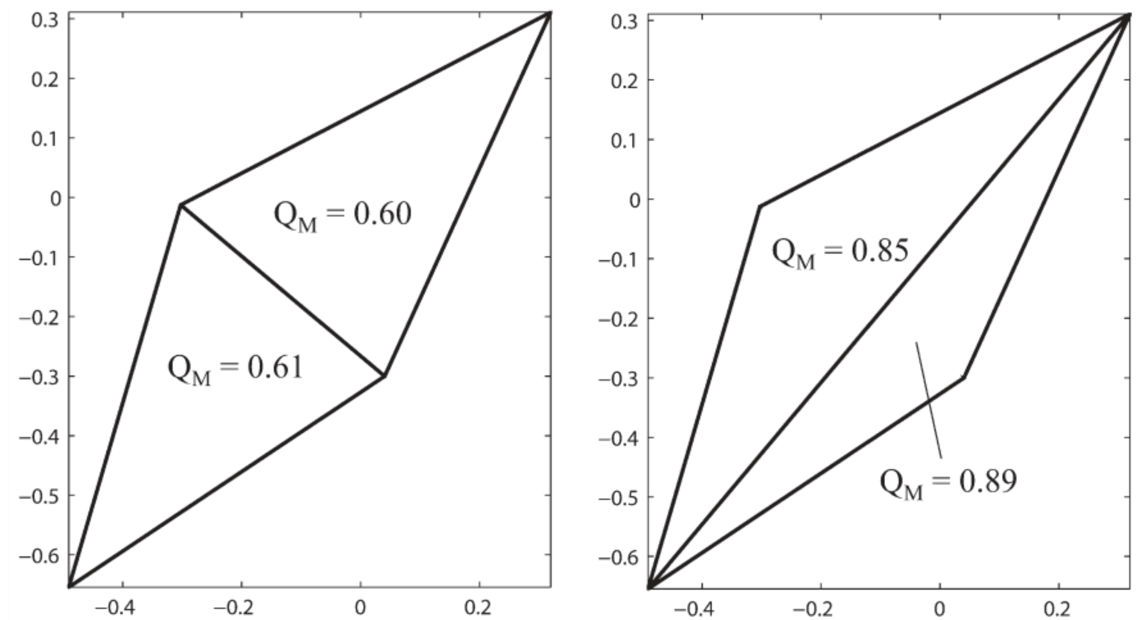


For 2D applications we use the Ani2D-MBA library
http://sourceforge.net/projects/ani2d/
e.g. Vasilevski and Lipnikov, 1999

E.g. An example of the effect of an edge swap on the element quality functional for a pair of elements in 2D.

NB. The 2D functional is optimal for a value of unity here





Conservative Interpolation: crucial for many applications, with adaptivity, model coupling

- Start from fact that Galerkin projection is conservative, second-order accurate, non-dissipative and well-defined for DG fields

- Galerkin projection leads to the linear system:

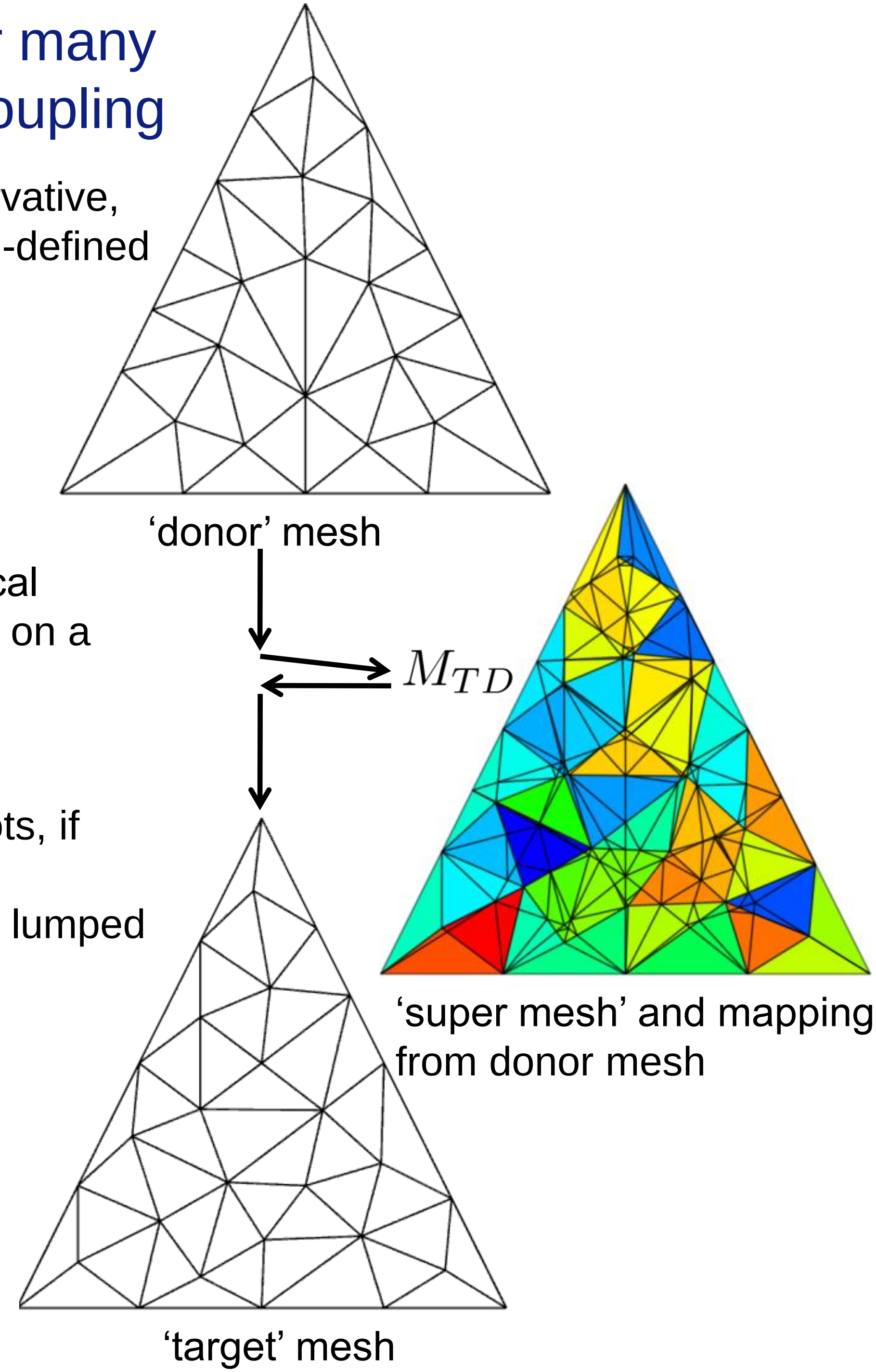
$$M_T u_T = M_{TD} u_D$$

- Compute the 'mixed' mass matrix using numerical integration of the T&D shape function evaluated on a 'supermesh' of simplices constructed from the intersection of T&D

- Galerkin projection can result in under/overshoots, if boundedness important 'diffuse' these using the diffusive (but still conservative) properties of the lumped mass matrix:

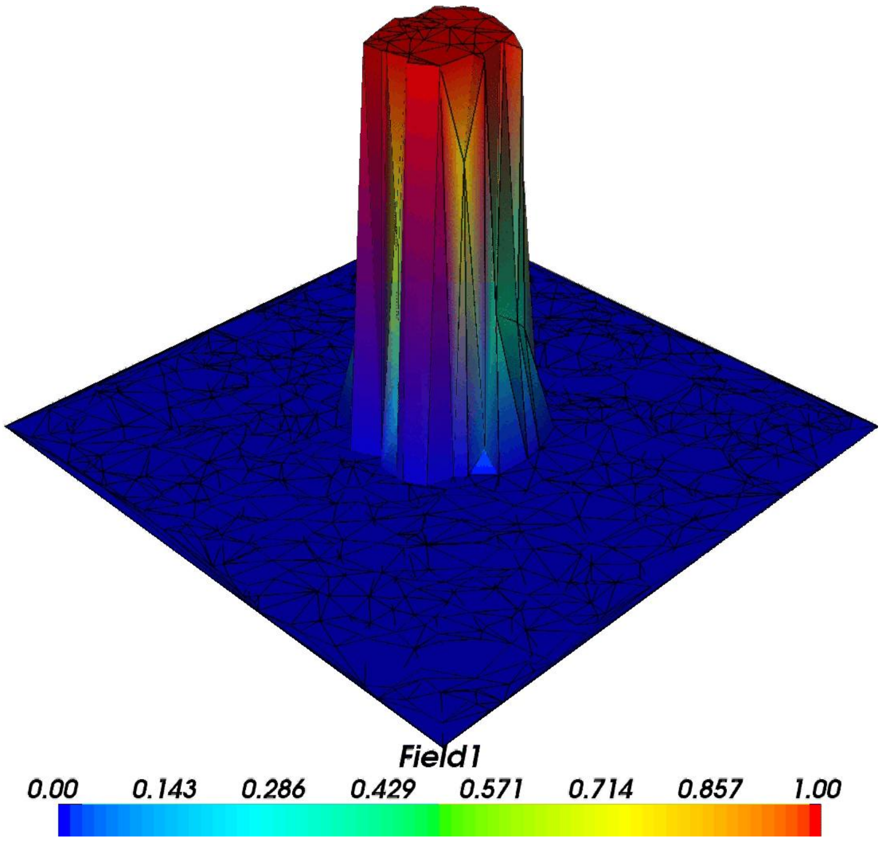
$$M_T^L u_{\text{alt}} = M_T u_{\text{dev}}$$

$$u_T \leftarrow u_T - u_{\text{dev}} + u_{\text{alt}}$$

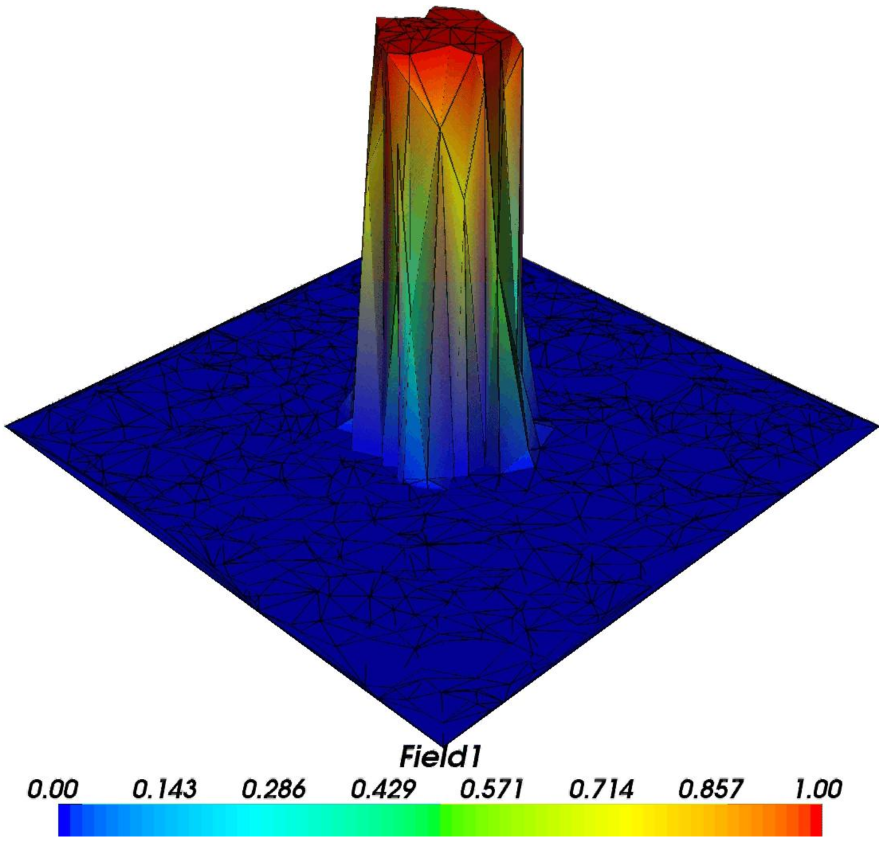


2D + bdd correction: Farrell et al., 2009; 3D: Farrell, Maddison, 2009

A synthetic example – interpolating a top hat back and forth between 100 arbitrary meshes (not refined to represent the top hat in any way – artificially difficult problem)

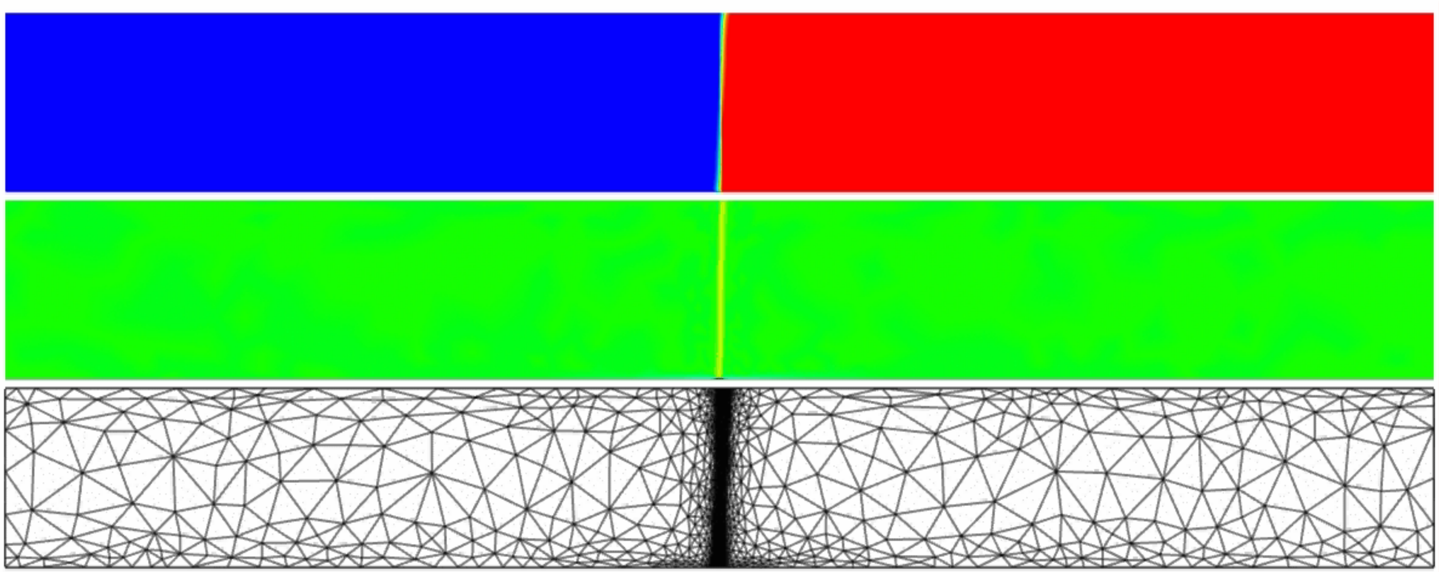


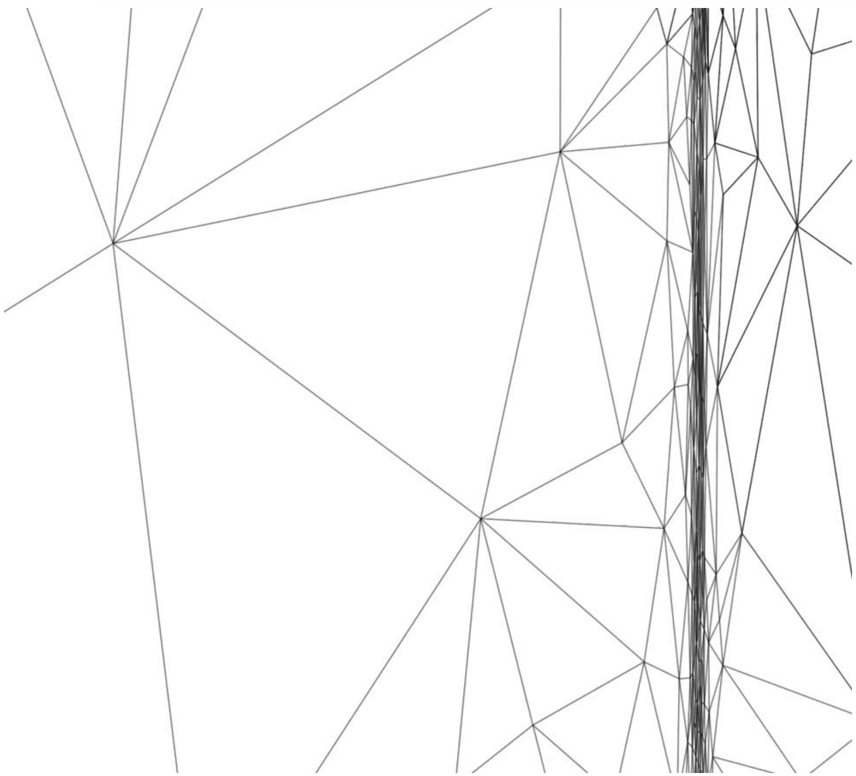
‘Consistent’ FE interpolation

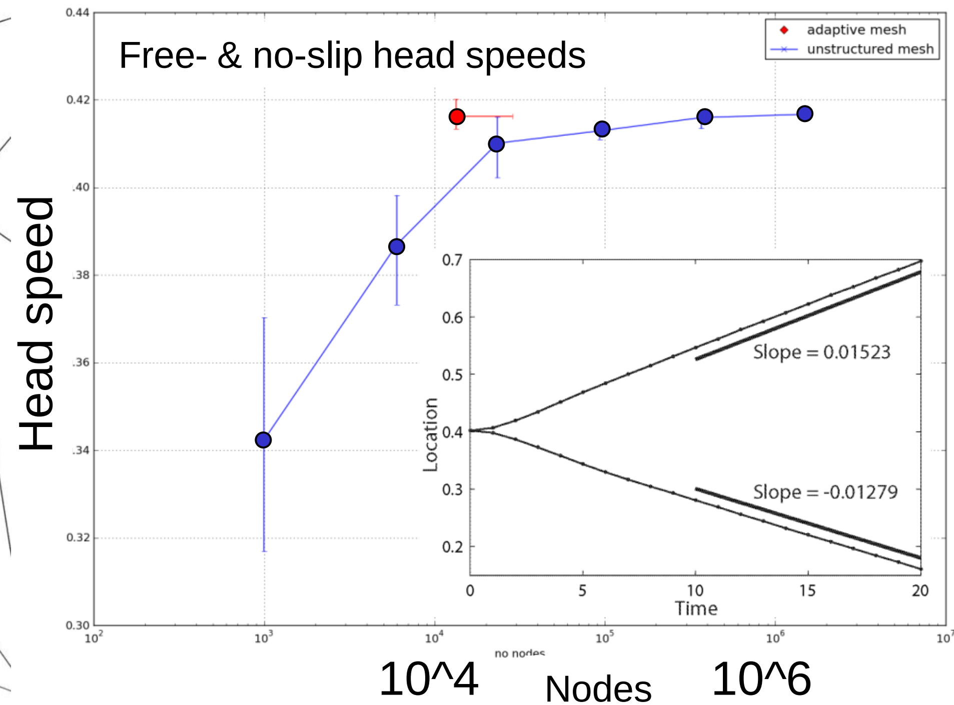


Galerkin projection

2D example with buoyancy: lock exchange problem

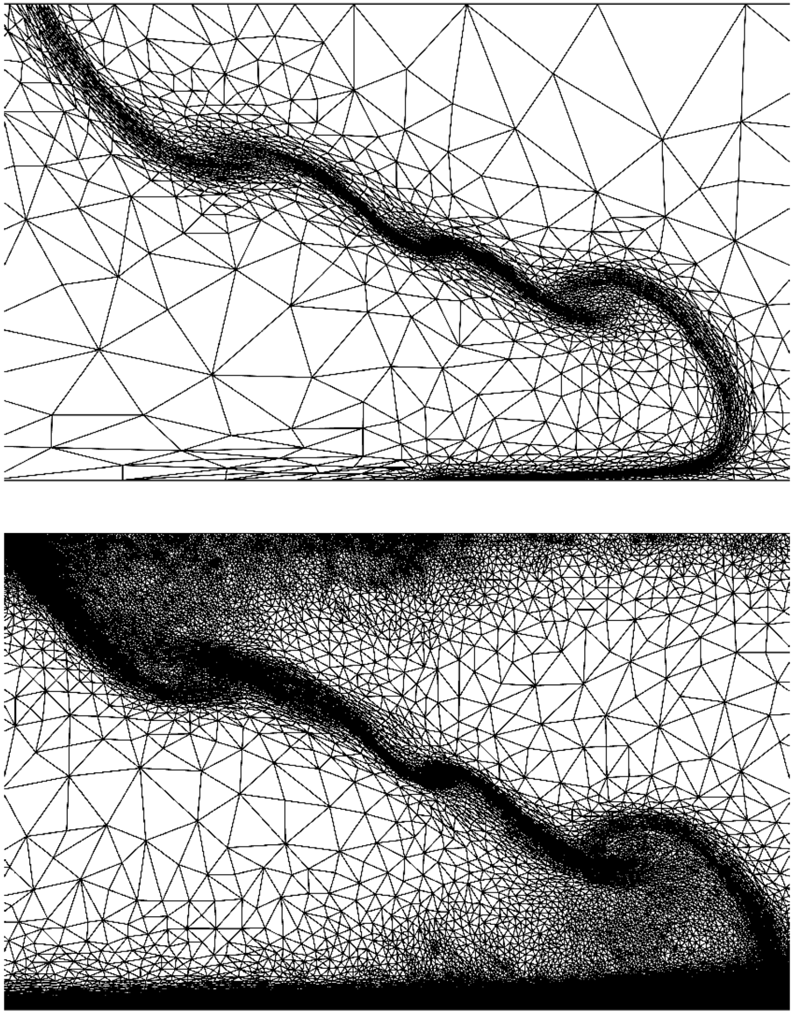




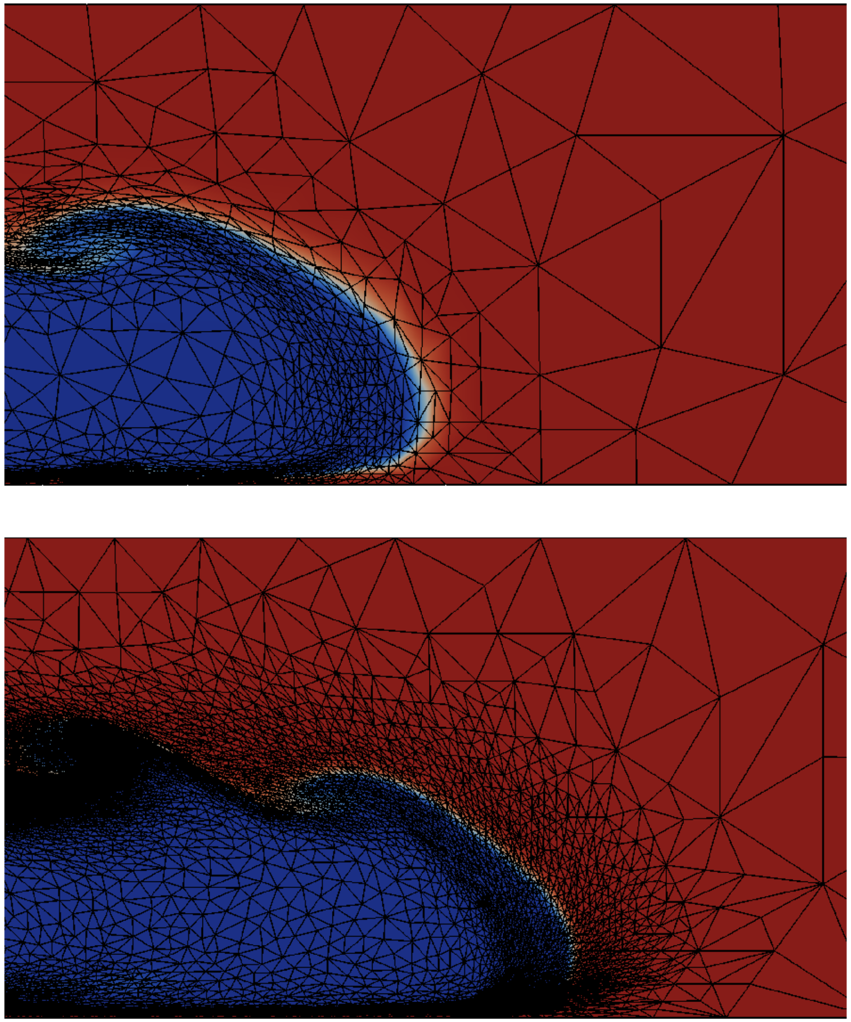


Consider the impact of two enhancements to standard adaptivity

Extra boundary refinement – spatially dependent error

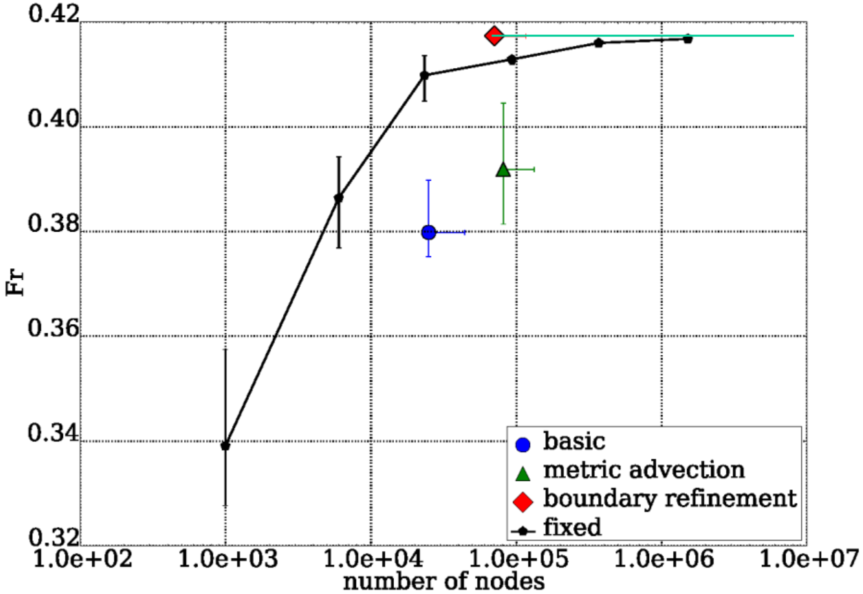


Metric advection – advect components of the metric forward in time and superimpose

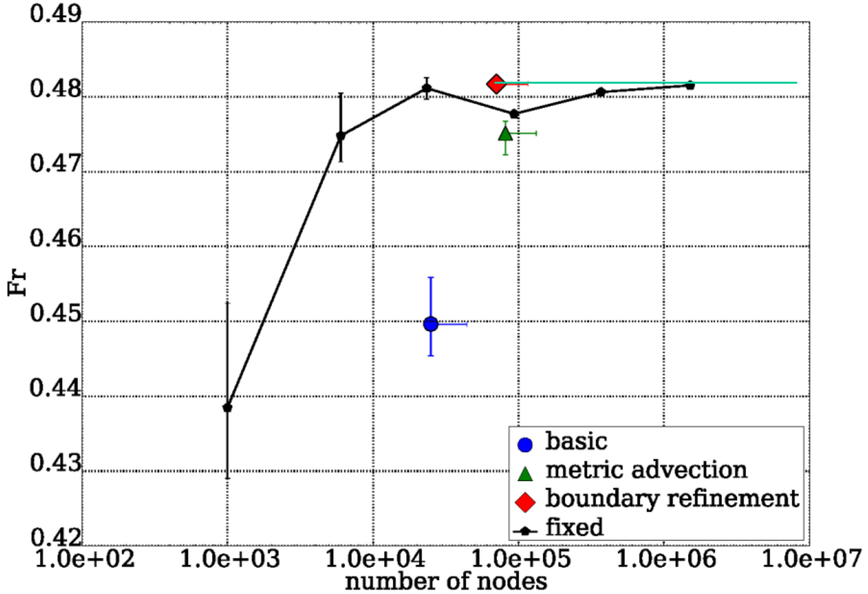


Impact on head speed

No-slip front (bottom)



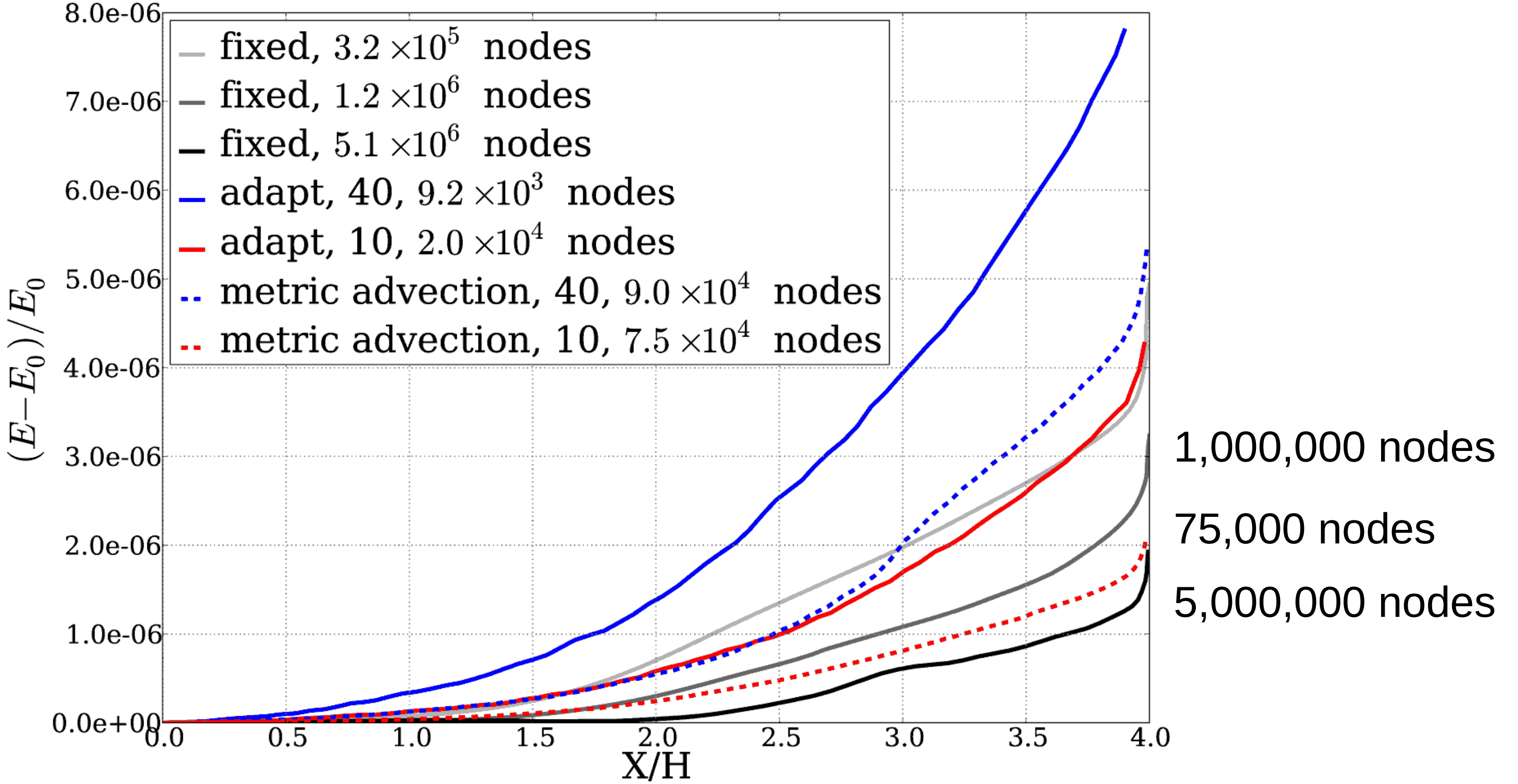
Free-slip front (top)



With boundary refinement we get a result as good as a uniform mesh with over an order of magnitude more resolution

(Hiester et al., 2011)

Impact on mixing/entrainment (the ability to keep spurious diapycnal mixing levels low one of the primary concerns for large scale models)



Comments:

- Perception that adaptivity itself can be dissipative
- This case shows that this can certainly be true if adaptivity not appropriate – advecting a front out of a region of high resolution
- Not really adaptivity's fault, just need appropriate error metrics/measures - difficult topic

Typical solver time loop

<u>Time step:</u>

<u>Nonlinear iteration:</u>

- Solve scalar equations – e.g. T, S, e, k-ε, biology, sediment, material/phase vol. fracs
- Evaluate density via an EoS
- Solve the coupled momentum/pressure system

Repeat a fixed number of nonlinear iterations or until convergence

<u>Diagnostics/output if appropriate</u>

<u>Adapt the mesh if appropriate (simulated time, # time steps, mesh quality measure):</u>

- Metric calculation
- Mesh optimisation
- Mesh-to-mesh interpolation
- Load balance

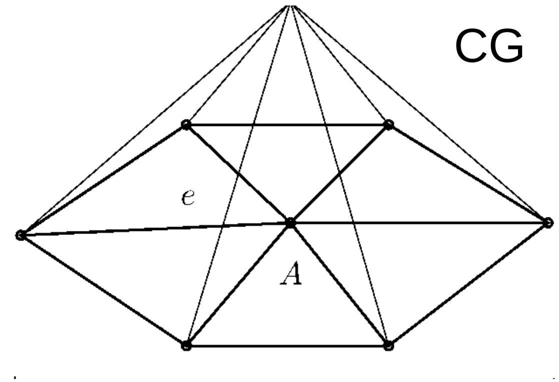
Next time step

Discretisation options

Spatial discretisation

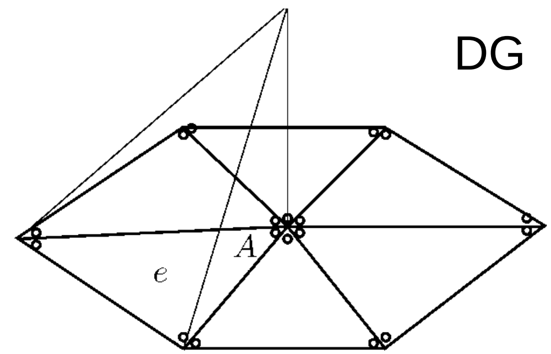
- CG (continuous Galerkin)
  - Streamline upwind (Petrov-Galerkin) stabilisation

CG



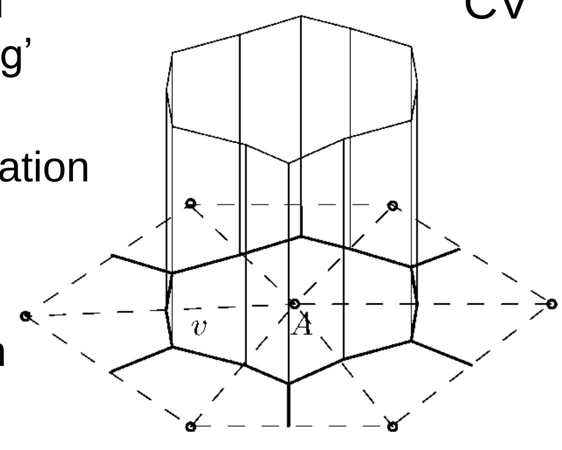
- DG (discontinuous Galerkin)
  - Upwind advective flux
  - CDG diffusive flux
  - Slope limiters (vertex based – Kuzmin 2010)
  - Explicit sub-cycling of advection

DG



- CV (control volume)
  - Flux limit between upwind flux and high-order flux obtained from FE interpolation, Sweby limiter, options for ‘sharpening’ interface limiters for multi-material simulations
  - Use of finite element solution space for diffusive flux calculation

CV



Temporal discretisation

- Simple theta-method - forward/backward Euler, Crank-Nicolson
- Possibly with explicit sub-cycling

Coupled momentum/pressure

$$M\frac{du}{dt} + A(u)u + Ku + Cp = 0$$
$$C^T u = \ldots$$

- Solve a linearised system, using the best available guess for pressure, for a new guess at velocity

$$M\frac{u_*^{n+1} - u^n}{\Delta t} + A(u^{n+\theta_{nl}})u_*^{n+\theta_u} + Ku_*^{n+\theta_u} + Cp_* = 0$$

- Consider the equation that the velocity will satisfy following a pressure correction step

$$M\frac{u^{n+1} - u^n}{\Delta t} + A(u^{n+\theta_{nl}})u_*^{n+\theta_u} + Ku_*^{n+\theta_u} + Cp^{n+1} = 0$$

- Subtract and multiply through by $C^T M^{-1}$, solve for Δp and then update u and p

$$\begin{aligned} C^T M^{-1} C \Delta p &= -C^T(u^{n+1} - u_*^{n+1})/\Delta t \\ &= C^T u_*^{n+1}/\Delta t + \underline{\text{other terms}} \end{aligned}$$

- The <u>other terms</u> come from the precise form continuity we'd like the new velocity to satisfy
  - Boundary conditions applied via continuity: normal velocities, free surface, inundation
  - 'Balanced pressure' decomposition
  - Compressible continuity equation

Discretisation – velocity/pressure

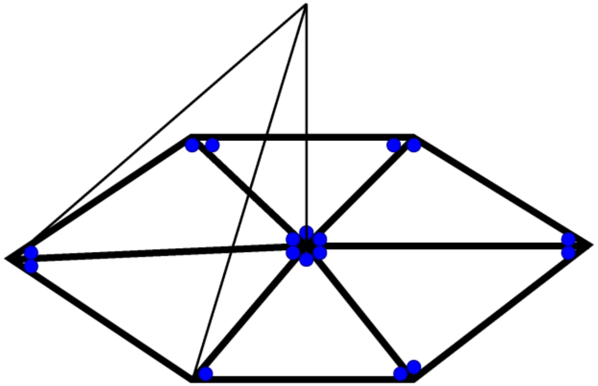
- The big remaining choice is how to represent velocity and pressure – so-called element choice, cf. finite difference grid staggering

- A whole zoo of options with different properties and appropriateness to represent certain physical regimes (we use P1P1, P2P1, P0P1, P0P1cv, P1dgP2, ...)

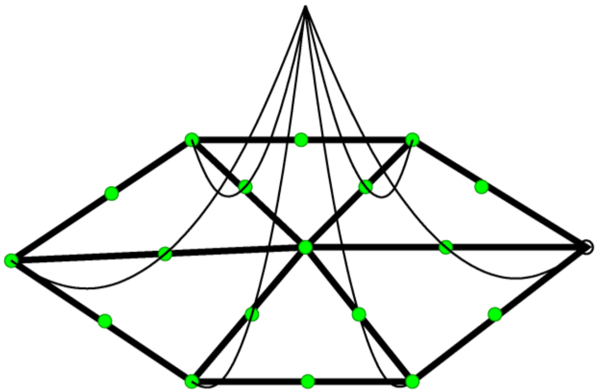
- Loosely, LBB stability (presence of pressure (checkerboard) modes) implies one should use a larger space to represent velocity than pressure

- But accurate representation of balance requires the opposite

- We tackle this with two approaches –
  – separating out a more accurate 'balance' pressure from the 'div-free' pressure
  – Using a mixed combination of CG and DG spaces, e.g. (u,p) ϵ (P1dg,P2)



$P1_{DG}$

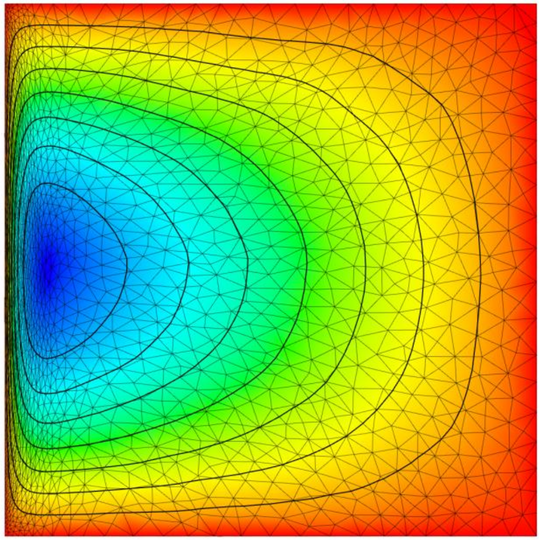


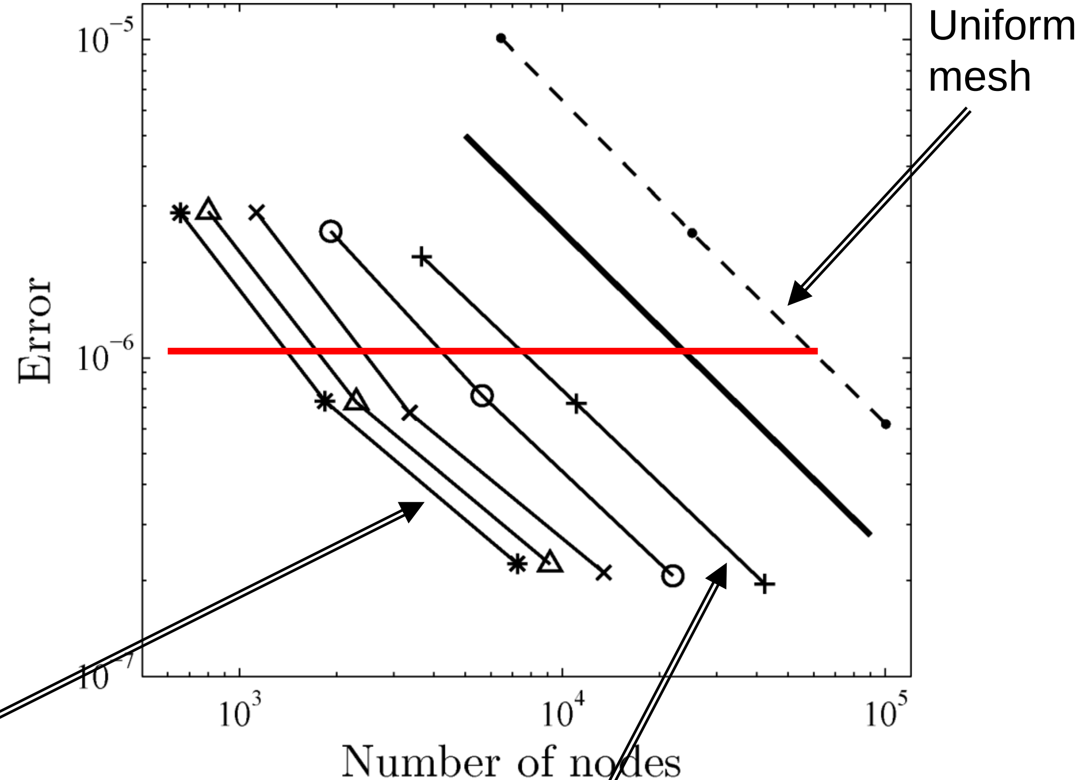
P2

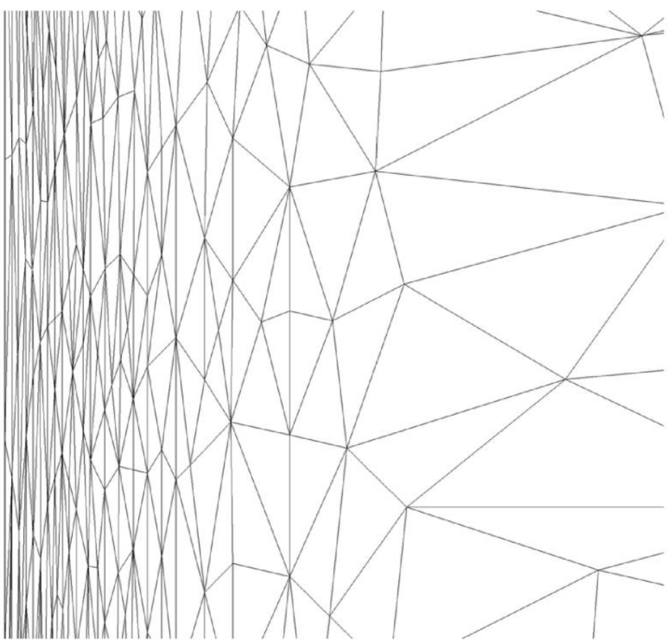
- LBB stable
- Balance
- No mass lumping
- Good wave dispersion properties

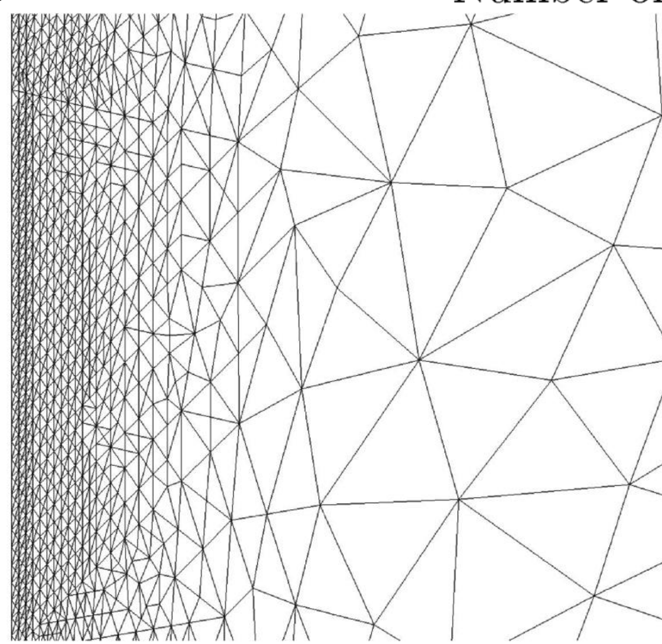
Cotter et al., 2009

Anisotropy and boundary currents (WBC: Stommel 1948)



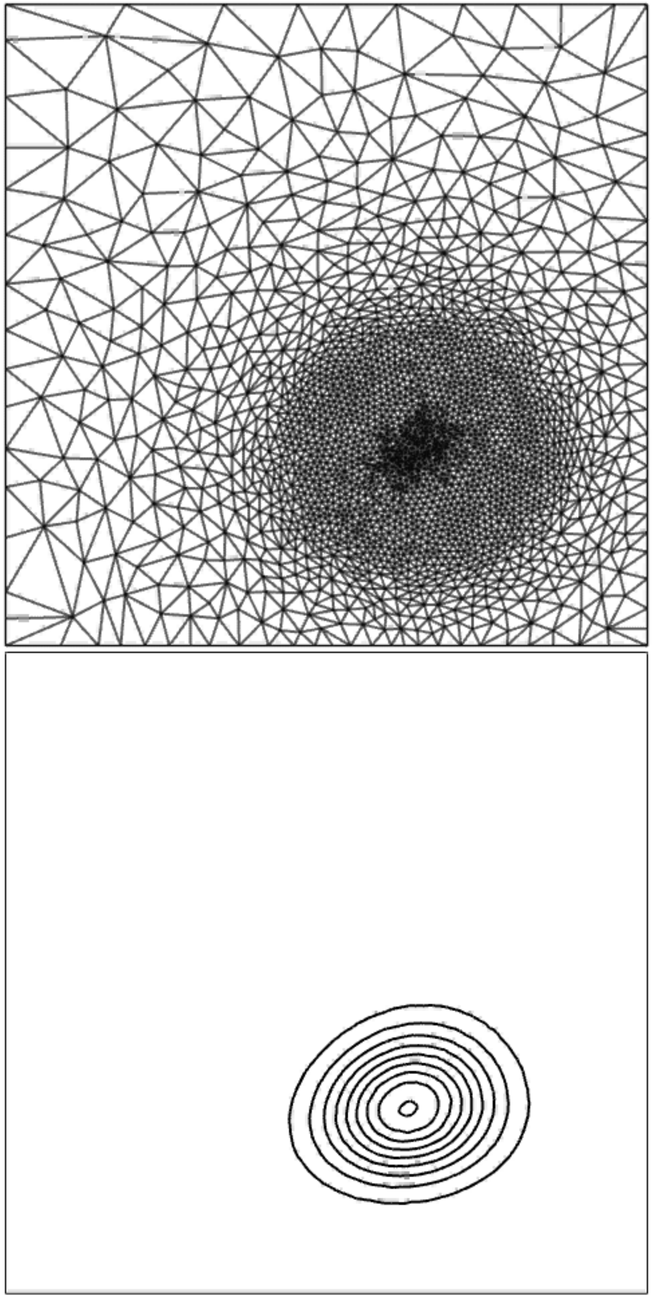




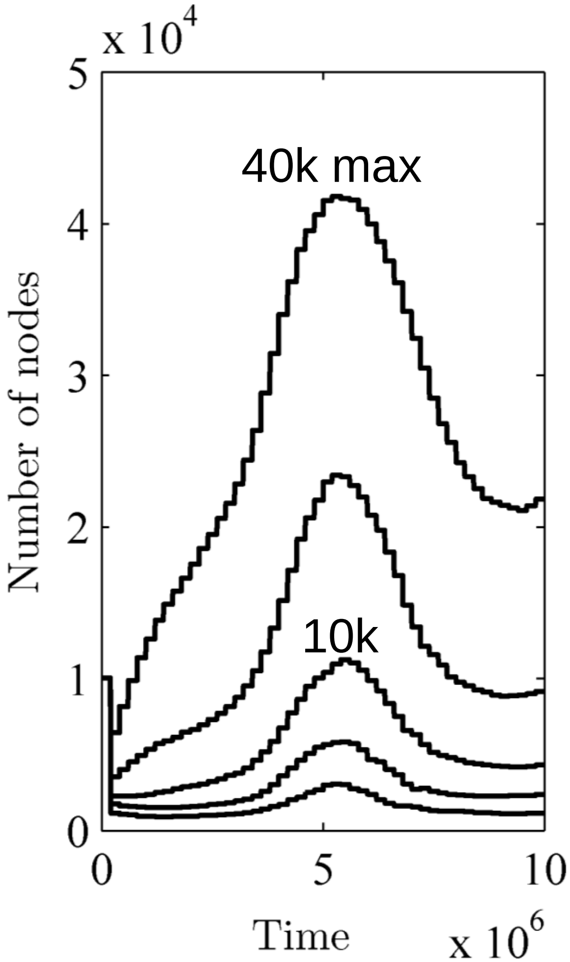


Approx. two orders of magnitude improvement in the error/cost relationship between uniform and <u>anisotropic</u> adaptive refinement, only one order for <u>isotropic</u>

Advecting a tracer field through the boundary layer

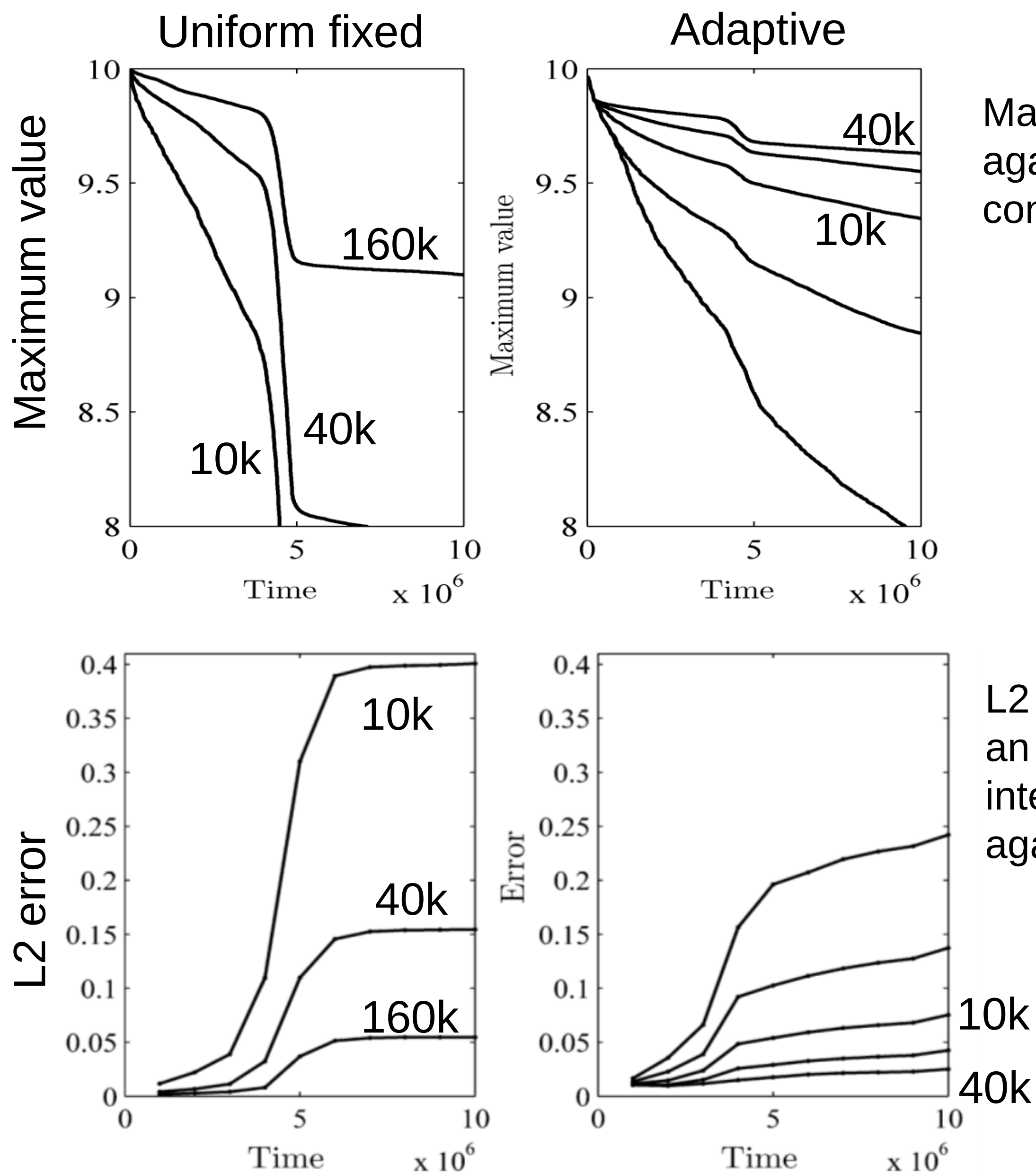


Initial Gaussian tracer field advected with the velocity field from the Stommel gyre – stretching through the boundary layer makes this a tough problem for the advection method.



Number of nodes against time shown right with 5 different tracer error weights.

Comparisons between uniform fixed and adaptive simulations

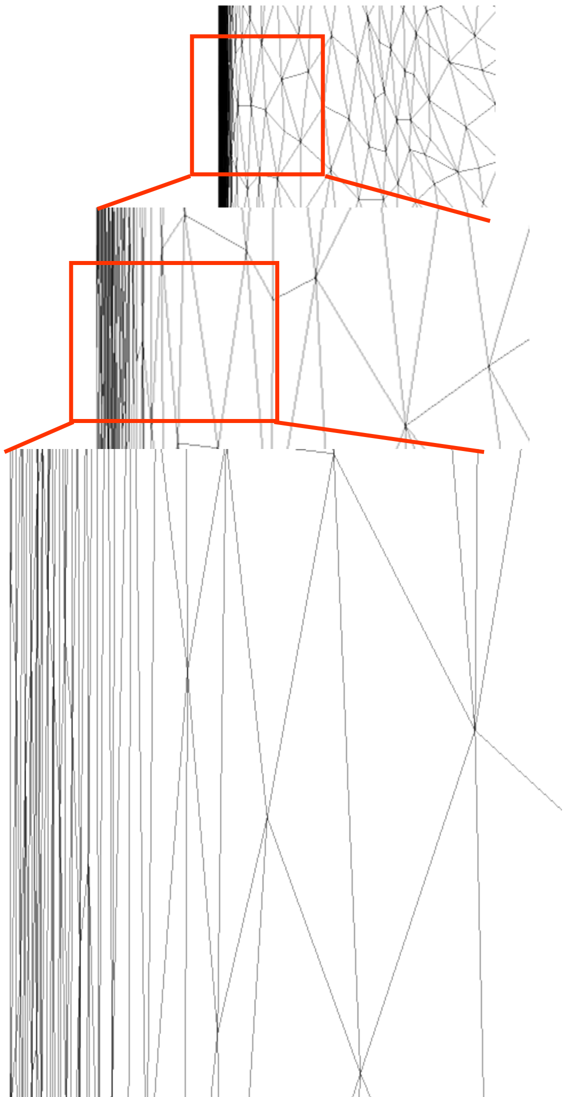


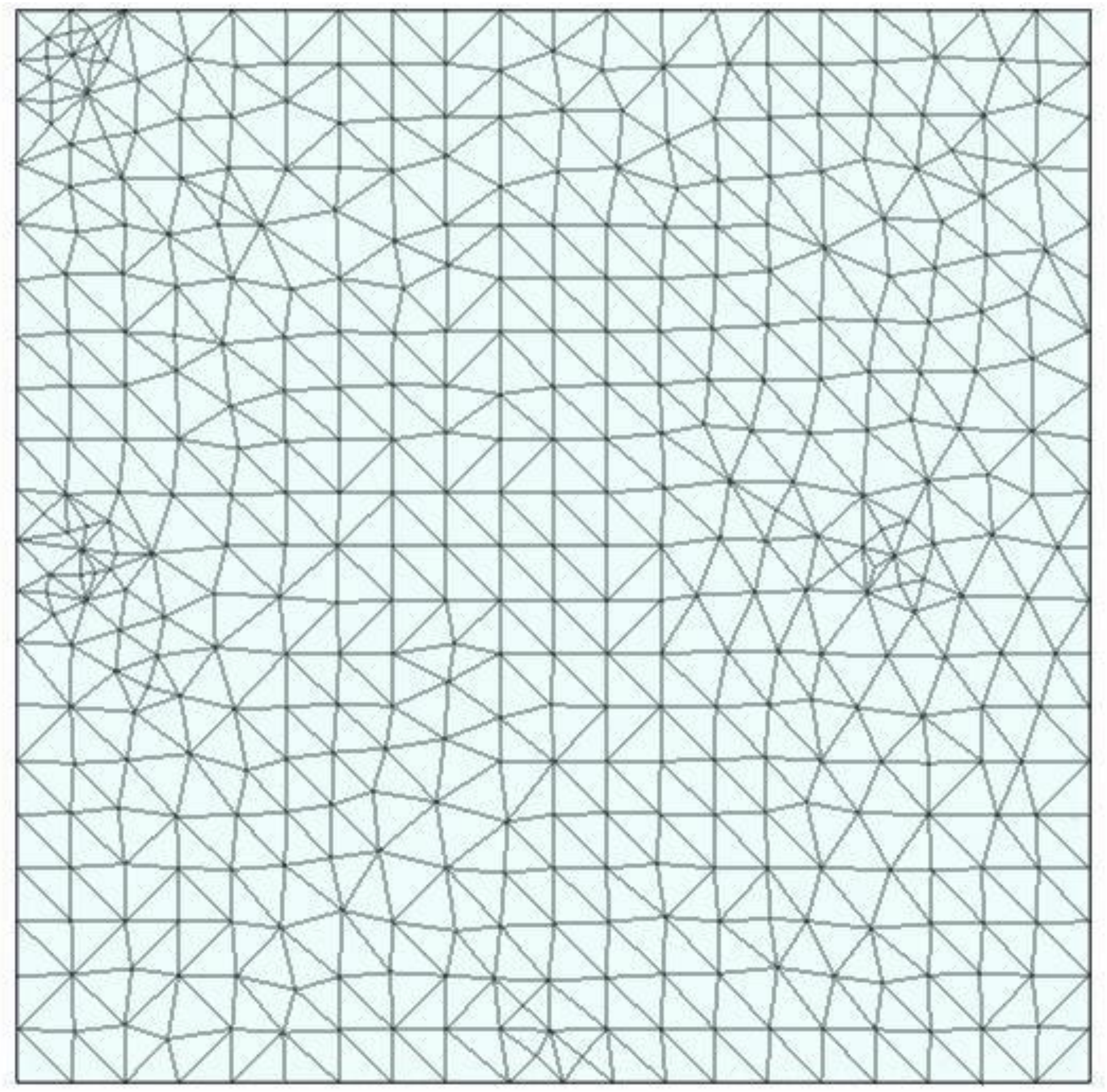
Maximum tracer concentration against time – should remain constant at 10.

L2 norm of error compared to an exact solution obtained by integrating back particle paths, against time.

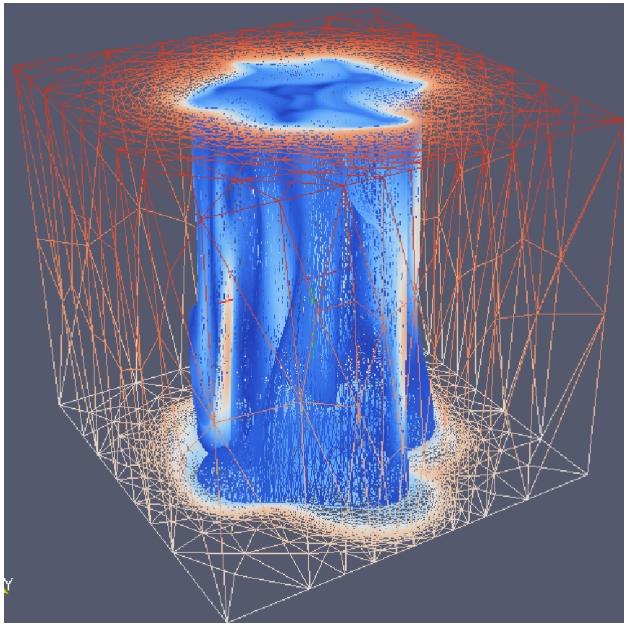
Conclusion: comparable quality for an order of magnitude fewer nodes

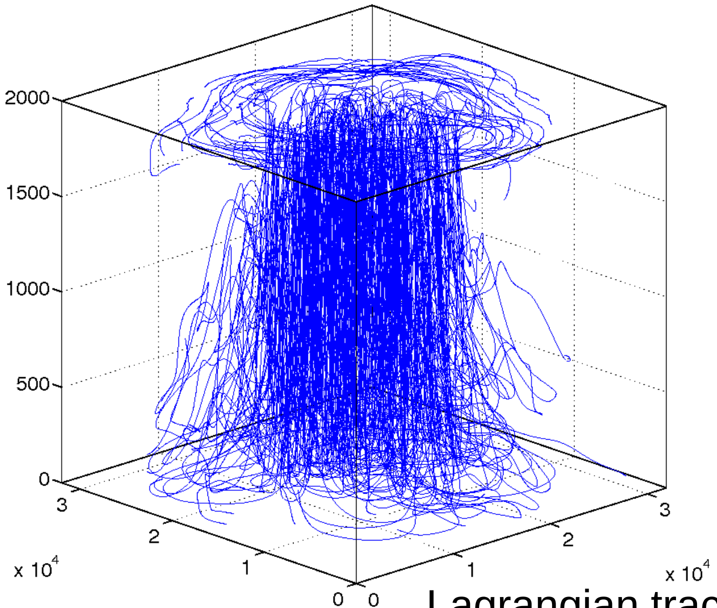
Barotropic wind driven gyre: western boundary current and eddies resolved with highly anisotropic resolution

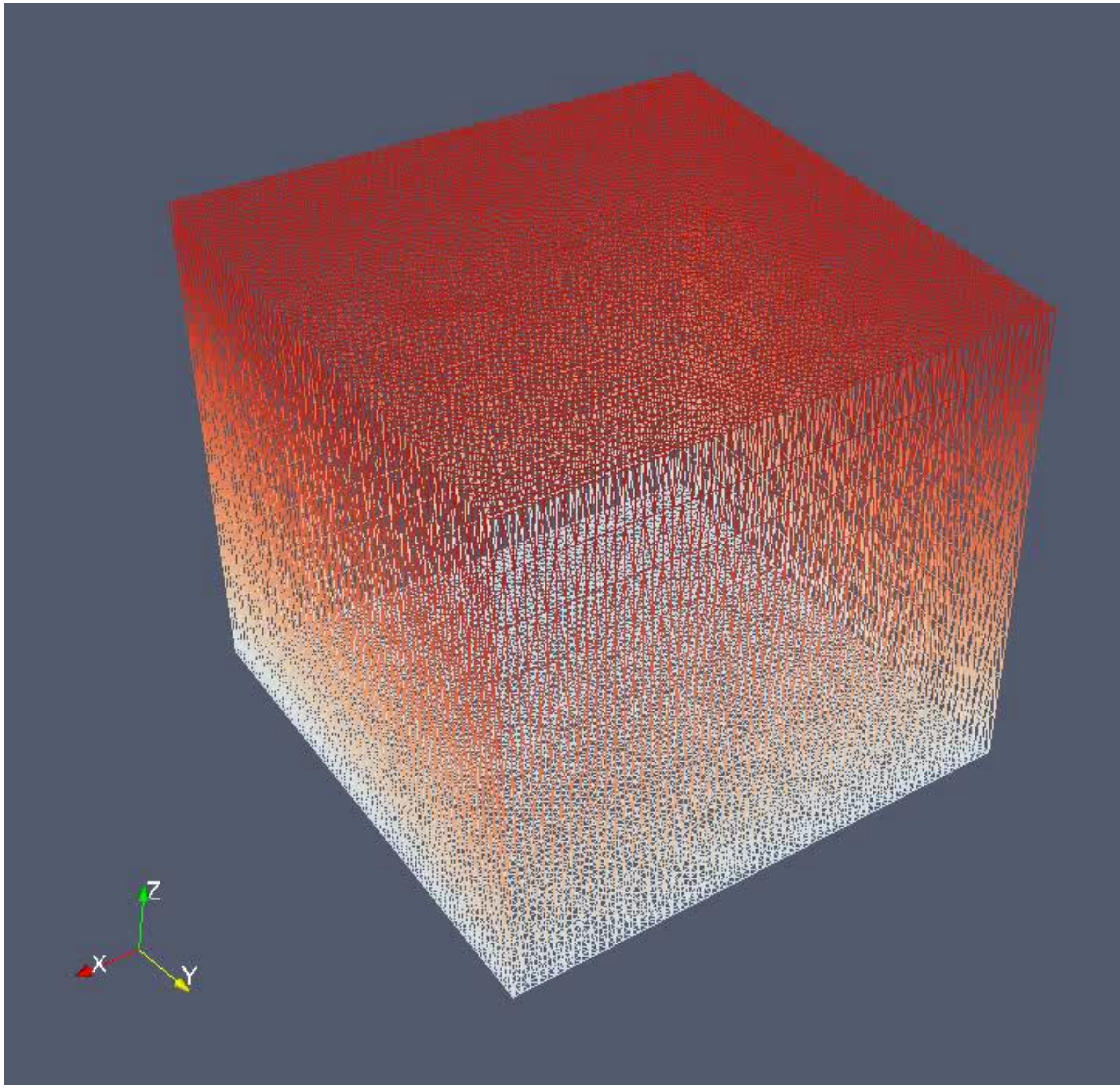




Now consider 3D with buoyancy and rotation: surface forced convection

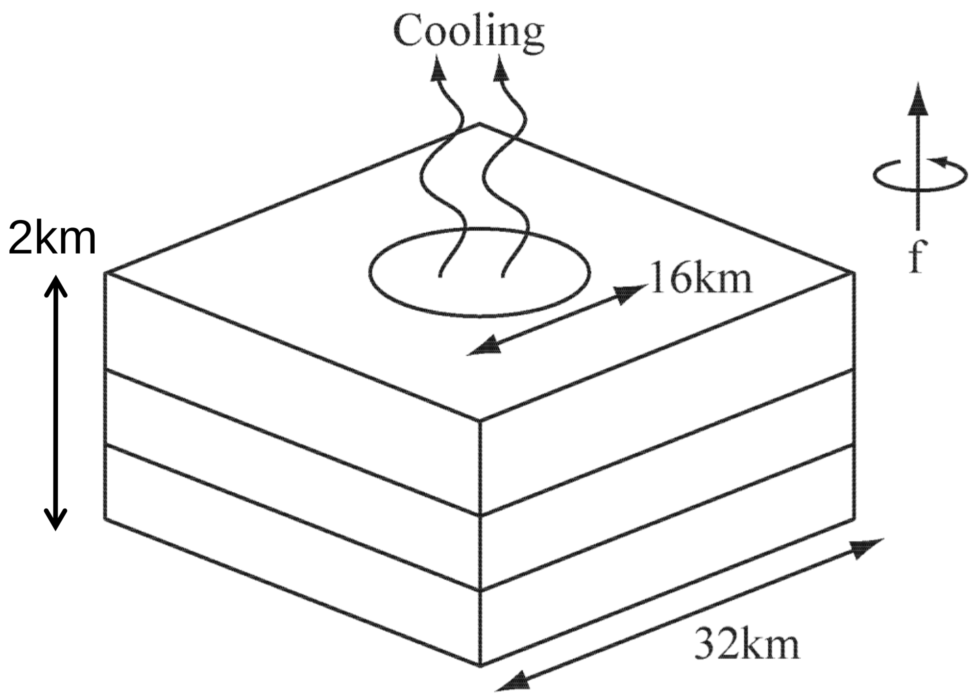


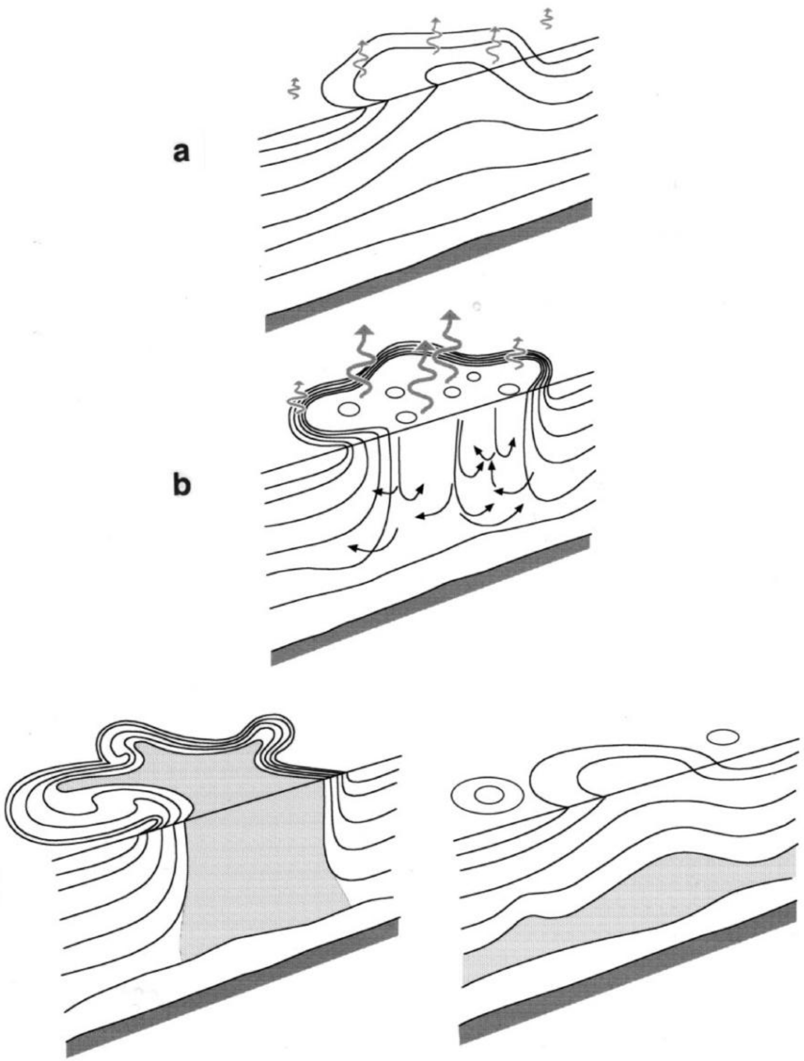




Lagrangian tracers/detectors

But that was not a particularly high aspect ratio

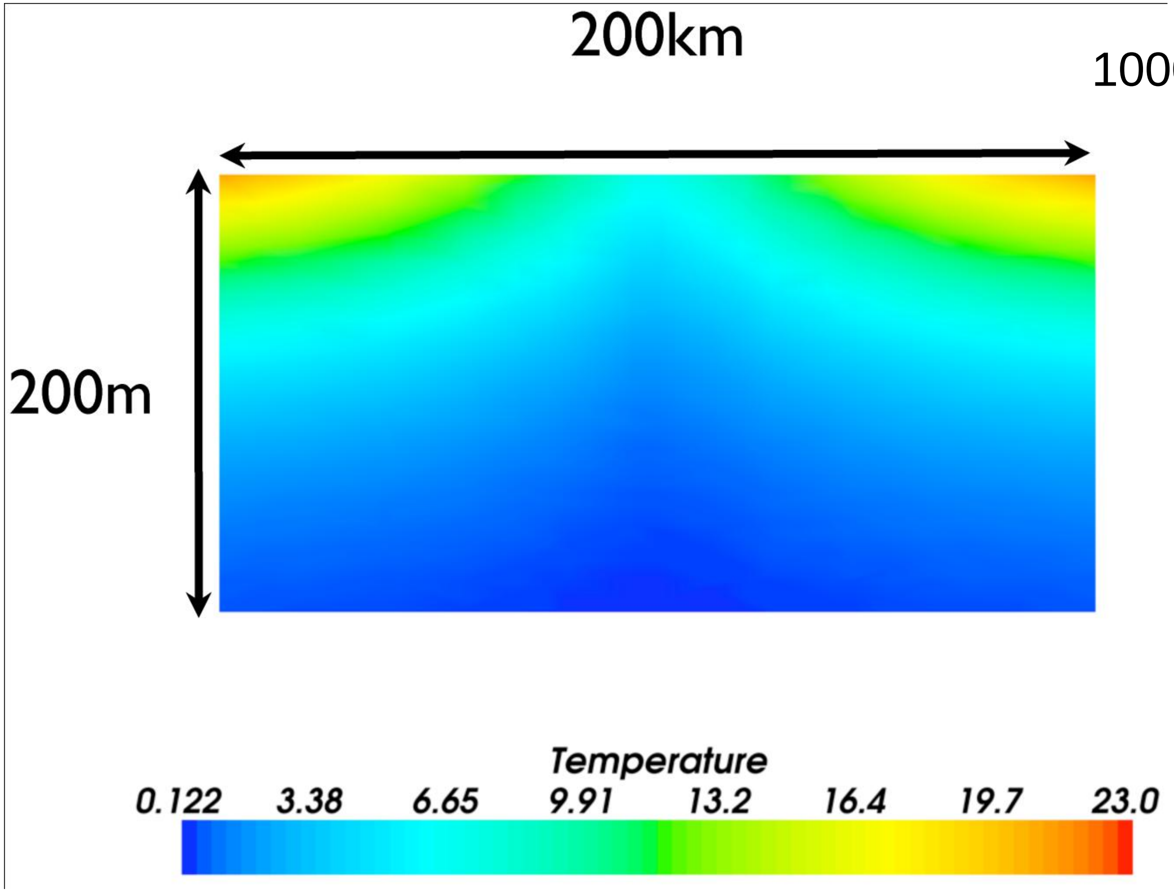




What if we want to study the whole event's life-cycle, including (a) preconditioning by basin scale flow and (c) restratification over a much larger area such as the Labrador Sea?

Marshall and Schott, 1999:
Open ocean deep convection: observations, models and theory, Rev of Geophysics, 37, 1, 1-64

High aspect ratio restratification – a smooth problem so should be relatively easy to deal with numerically!

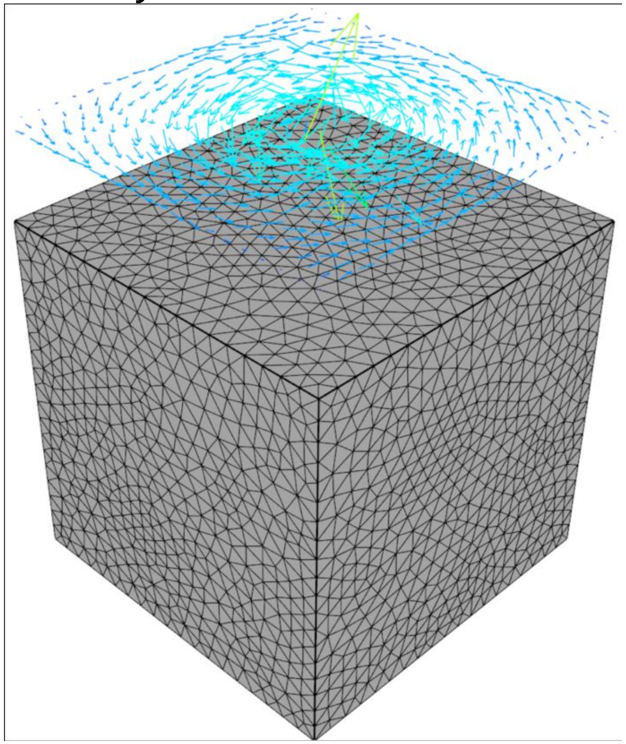


1000:1 aspect ratio

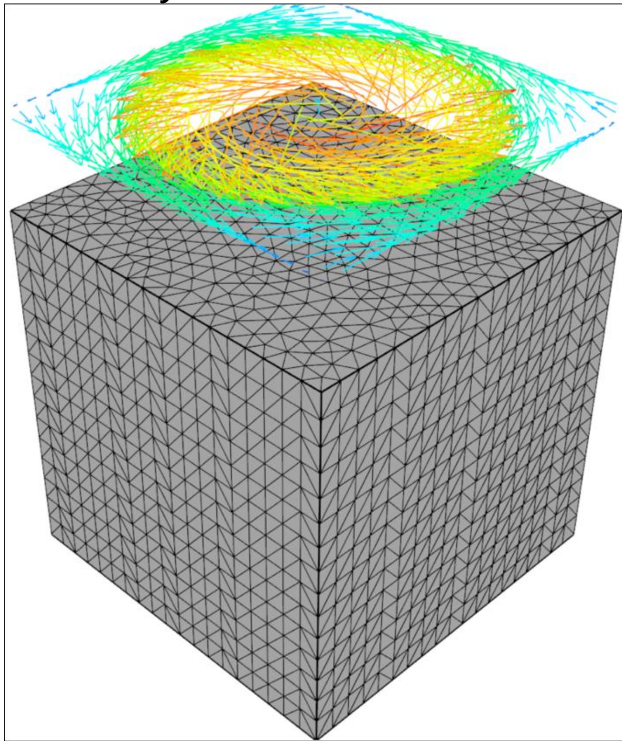
Set up taken from: H. Avlesen et al. A convergence study of two ocean models applied to a density driven flow. Int. J. Num. Meth. Fluids, 36(6): 639-657, 2001.

Initial balanced flow (fix temperature and evolve u and p to steady state)

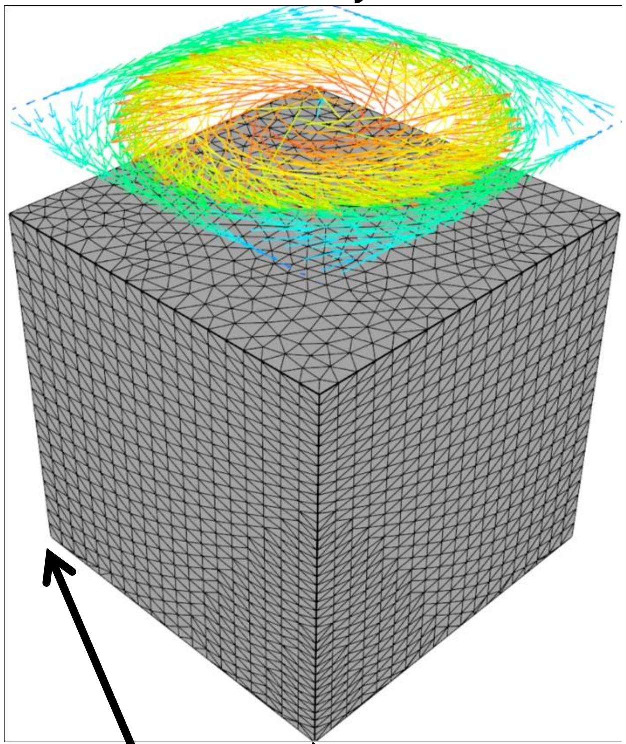
Fully unstructured in 3D



Layered in vertical



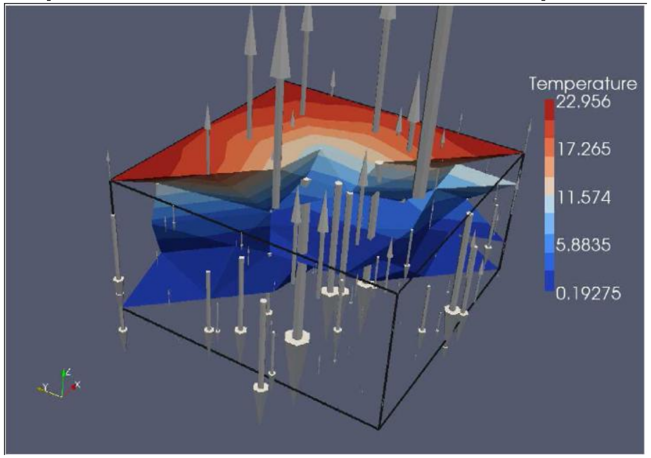
Columns, non-layered



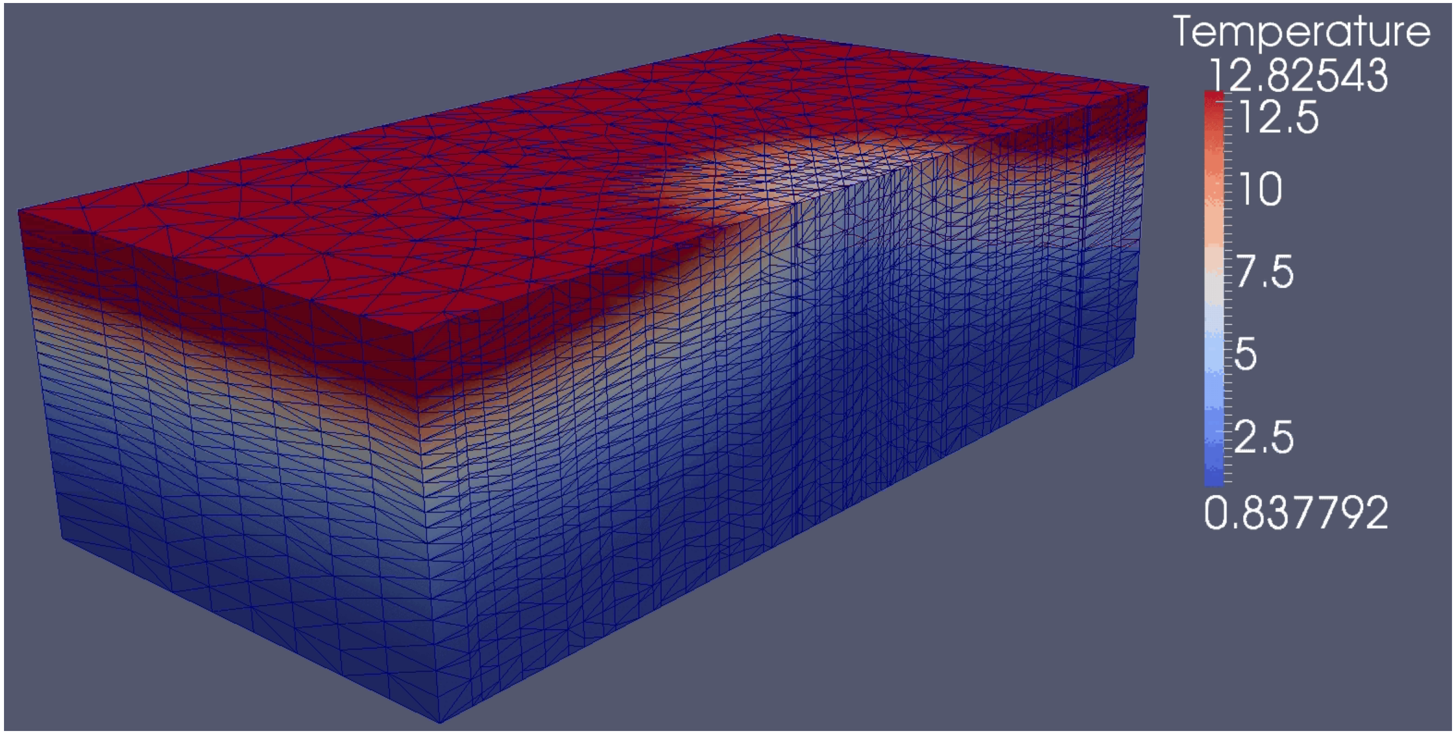
20 elements
in vertical

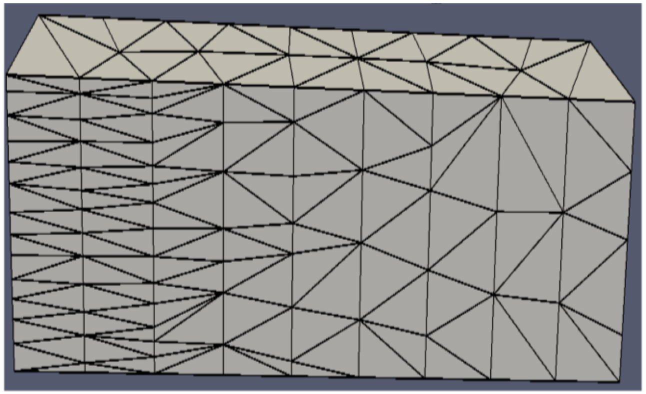
40 elements
in vertical

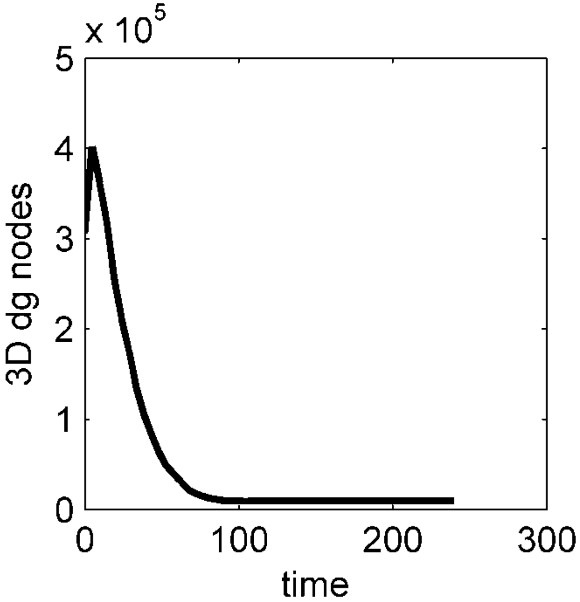
Spurious noise, all element pairs

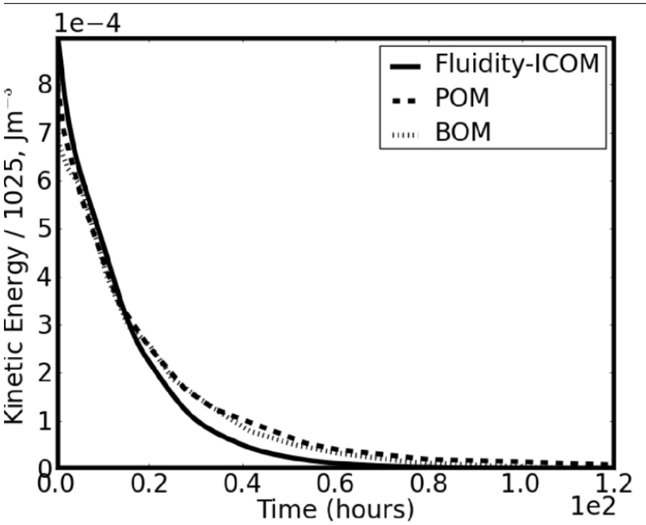


Spin-down experiment with new dynamic 2+1D adaptivity

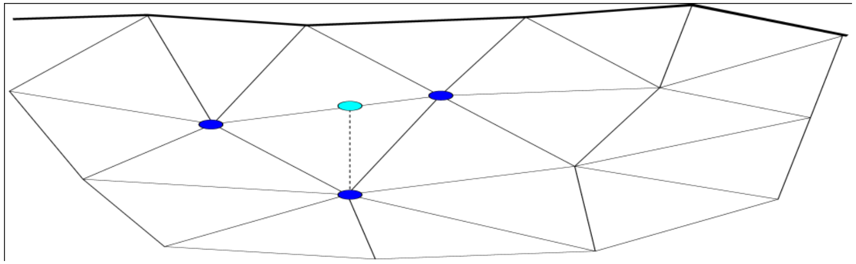








What's the problem?



- Consider the vertical momentum equation in a non-hydrostatic model:

$$\frac{Dw}{Dt} + \boxed{\frac{\partial p}{\partial z} = -\rho g}$$ hydrostatic balance

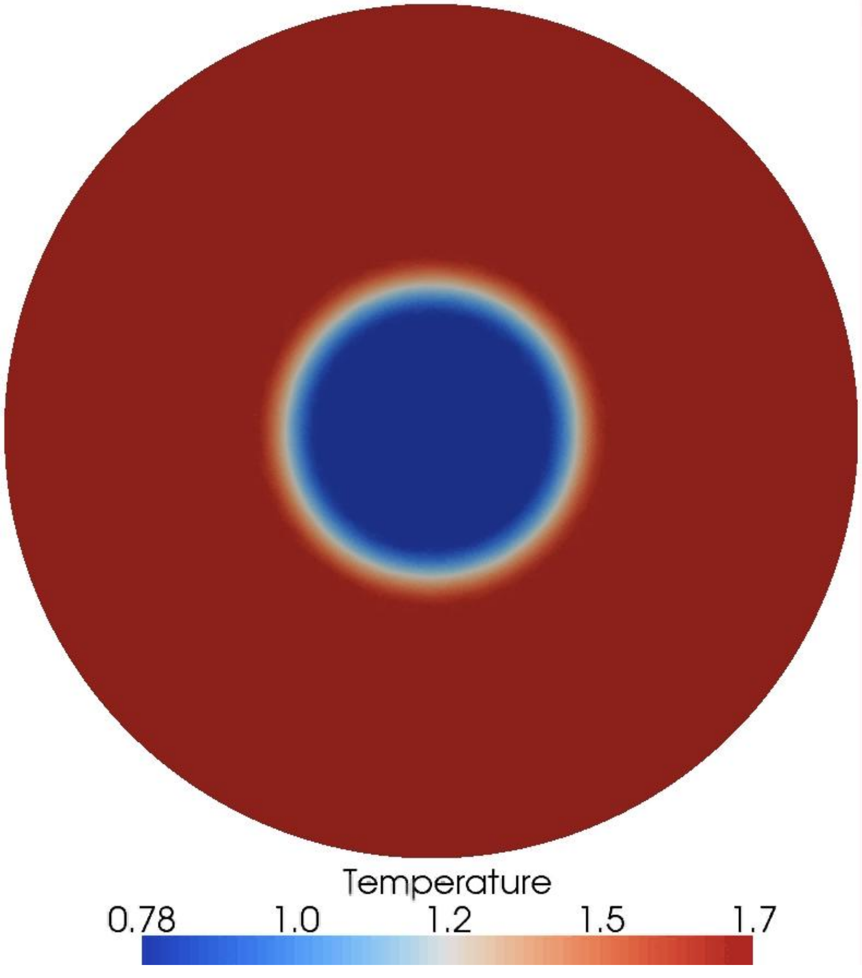
- Assuming a linear p, the truncation error in the horizontal part of the pressure leads to an error in the vertical pressure gradient of approximately:

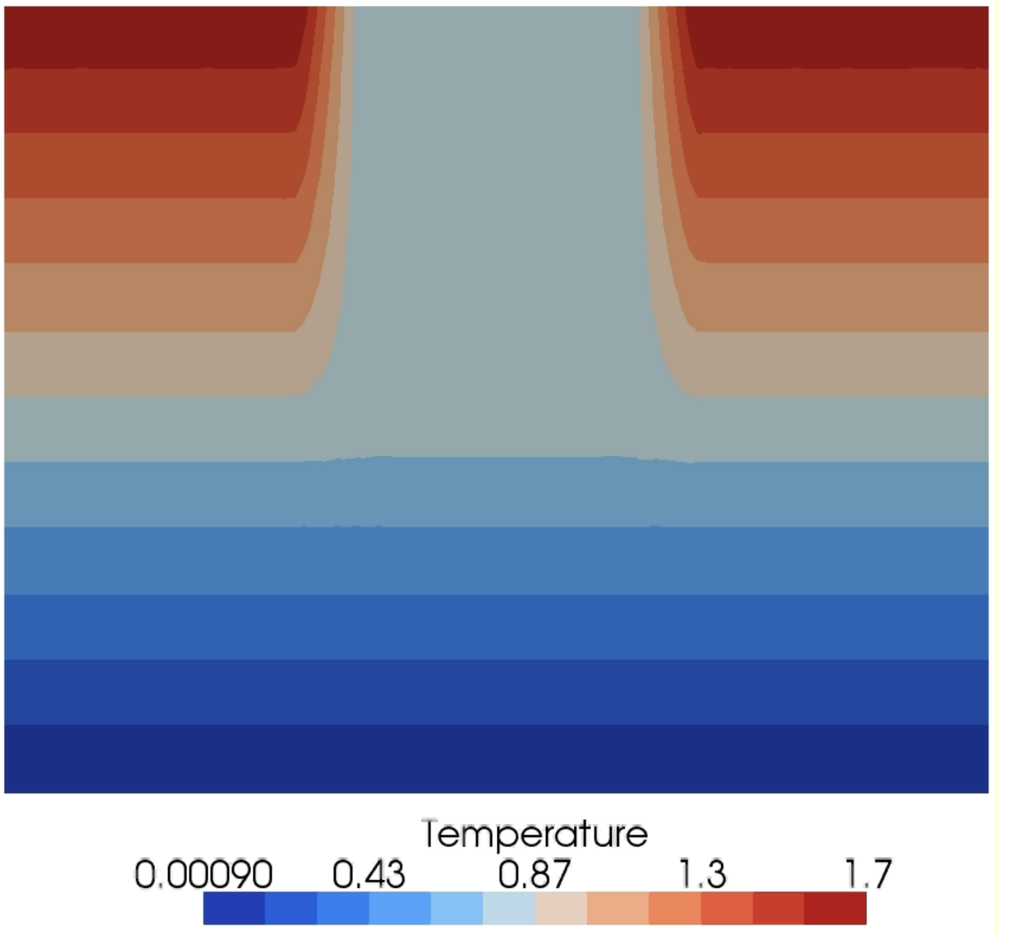
$$\frac{\partial^2 p}{\partial^2 x}\frac{\Delta x^2}{\Delta z} \approx \frac{\rho f U}{L}\frac{\Delta x^2}{\Delta z} \approx \rho f W \frac{L}{H}\frac{\Delta x}{\Delta z}\frac{\Delta x}{L}$$

- As the aspect ratio (L/H) becomes large, to keep this truncation error small compared to W, this indicates a constraint on the horizontal mesh spacing
- Currently working to understand this constraint for idealised problems
- For meshes with nodes appearing in columns vertically this truncation error is not present

A more challenging restratification problem (fixed mesh)

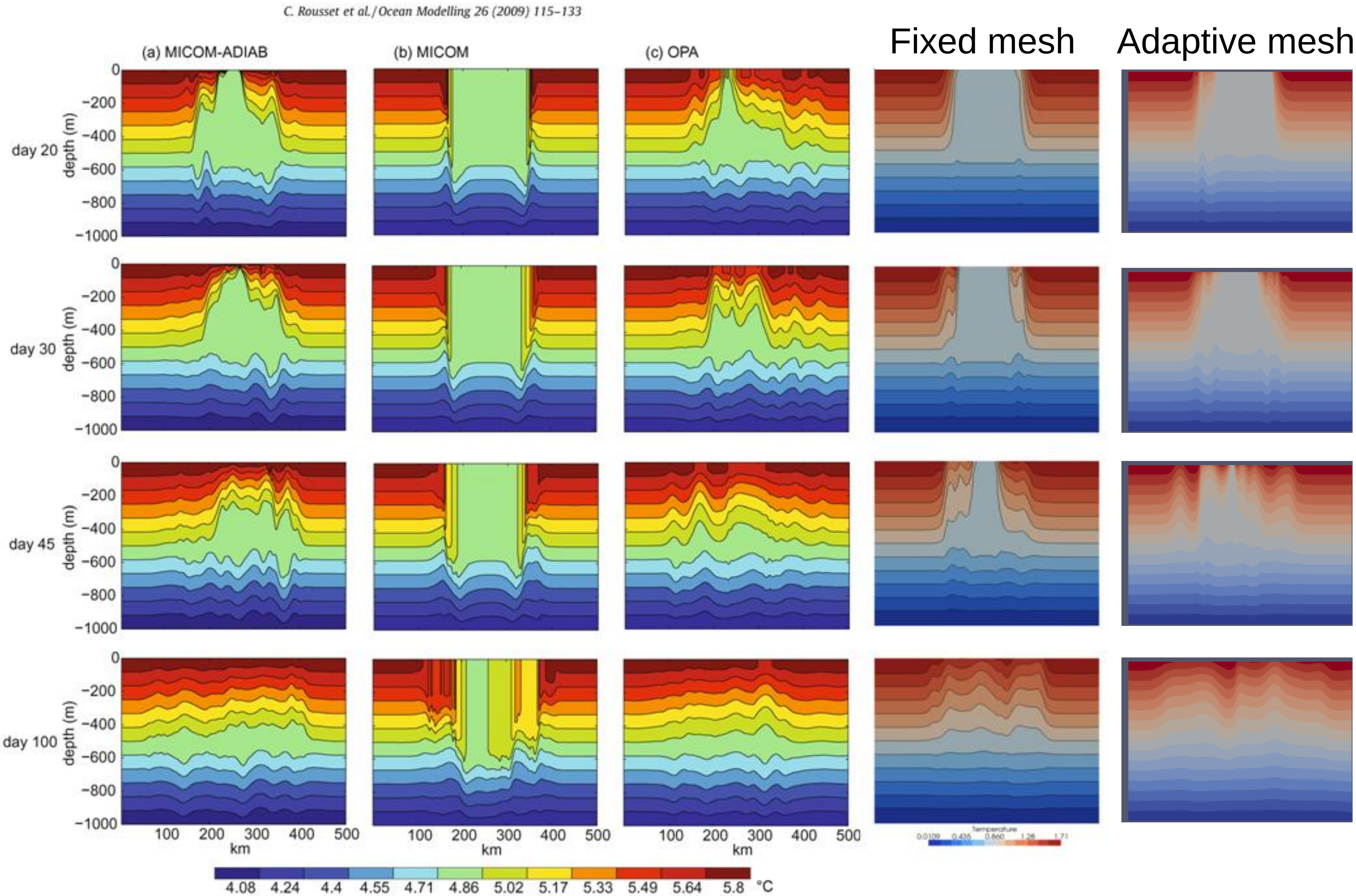
Real aspect ratio 500:1





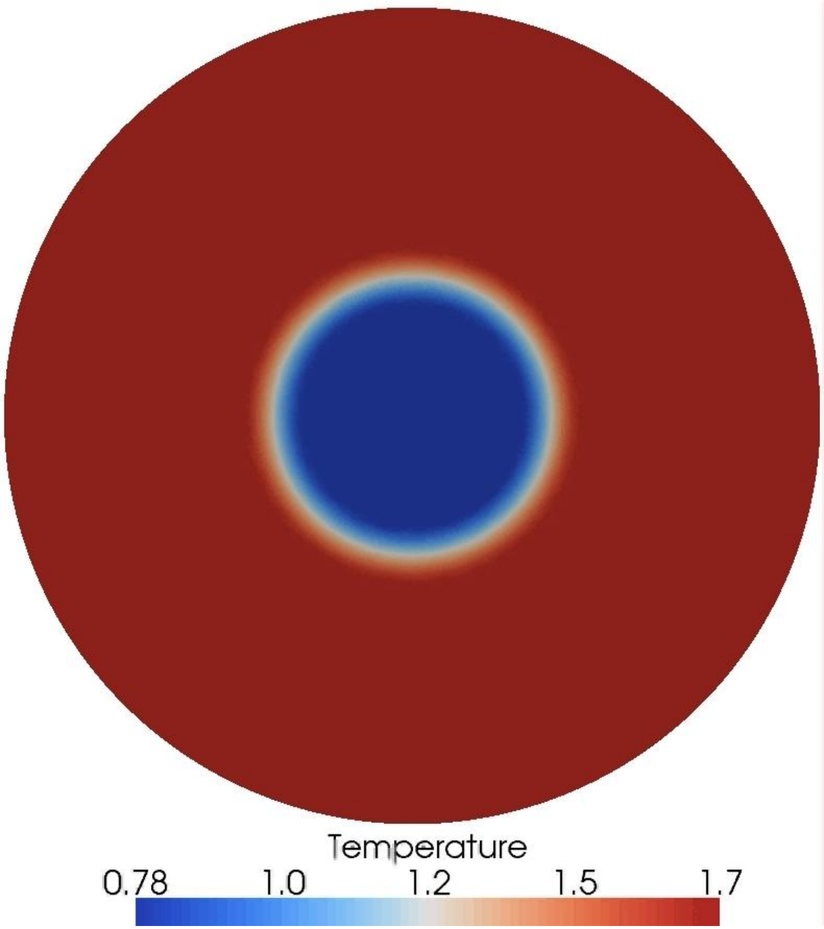
Set up taken from: Rousset et al. A multi-model study of the restratification phase in an idealized convection basin. Ocean Modelling, 26(3-4): 115-133, 2009.

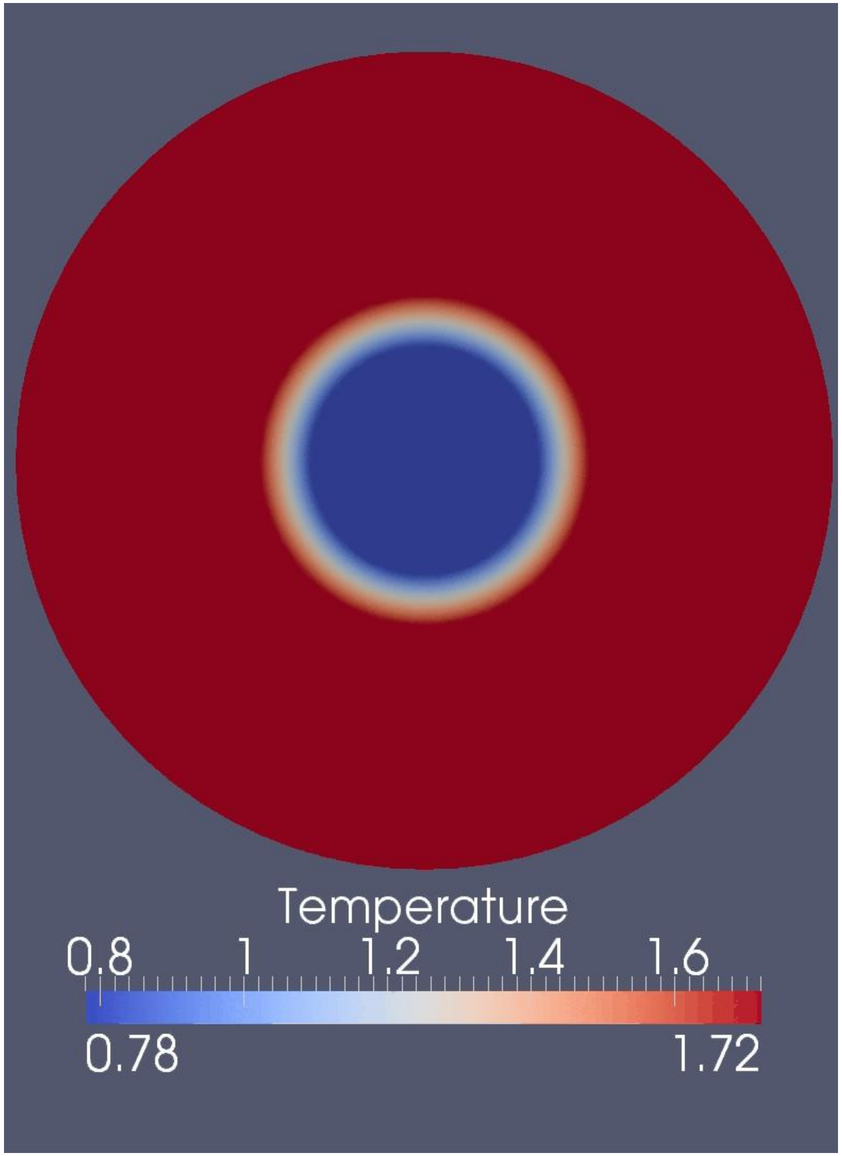
Use of an adaptive mesh appears to be getting the timings of the restratification slightly closer to the reference solution



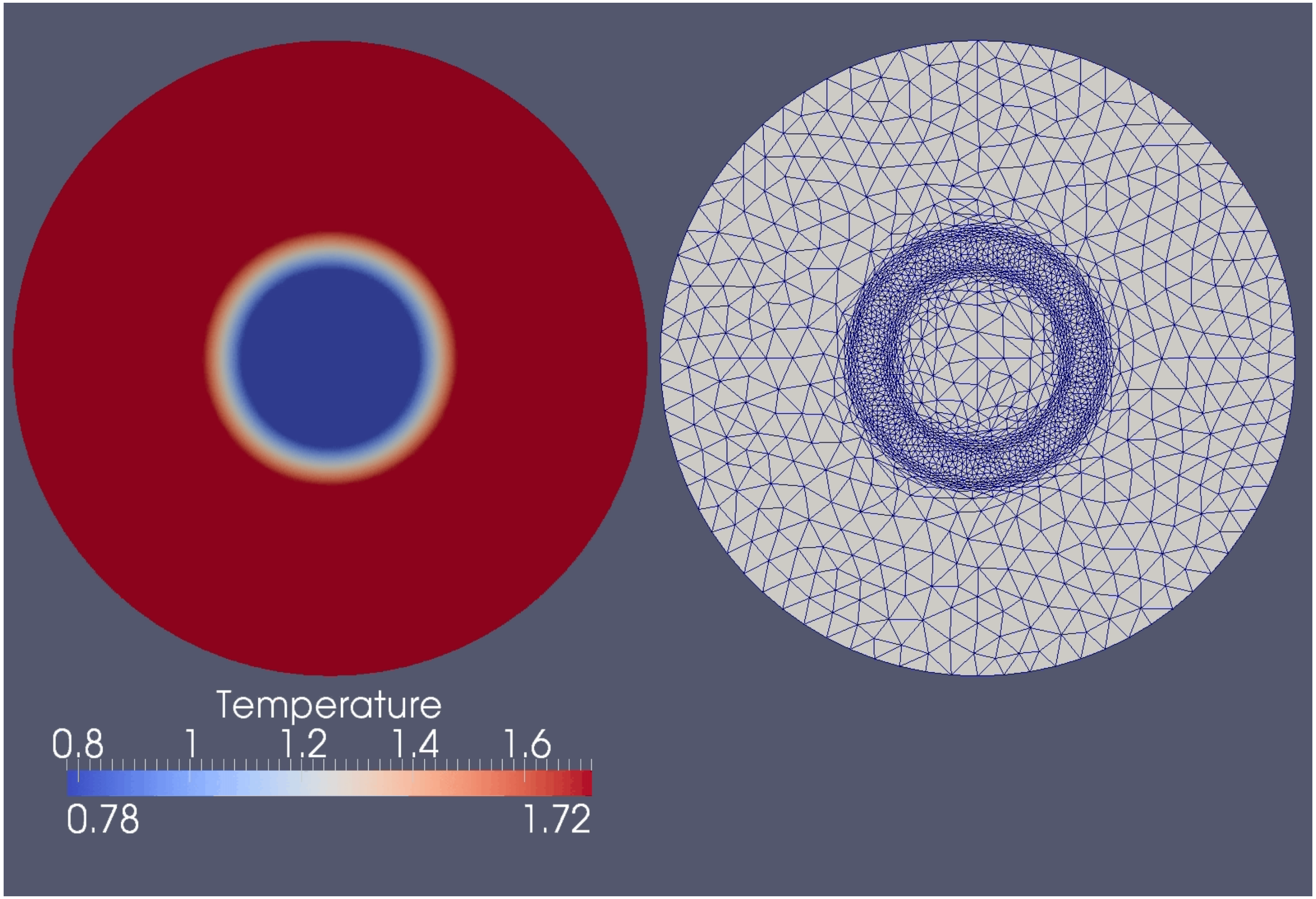
Rousset et al. 2009

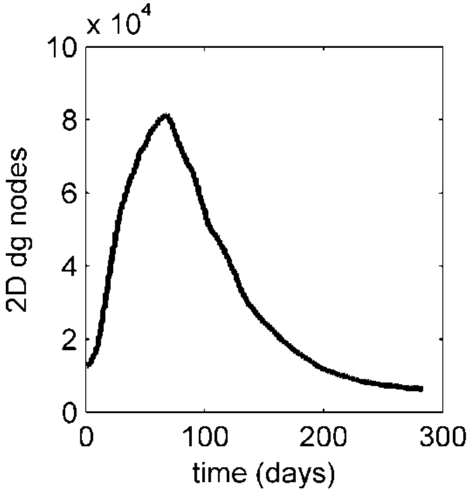
Compare results from fixed and adaptive runs



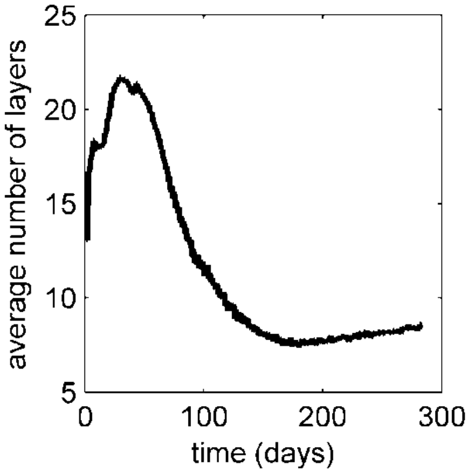


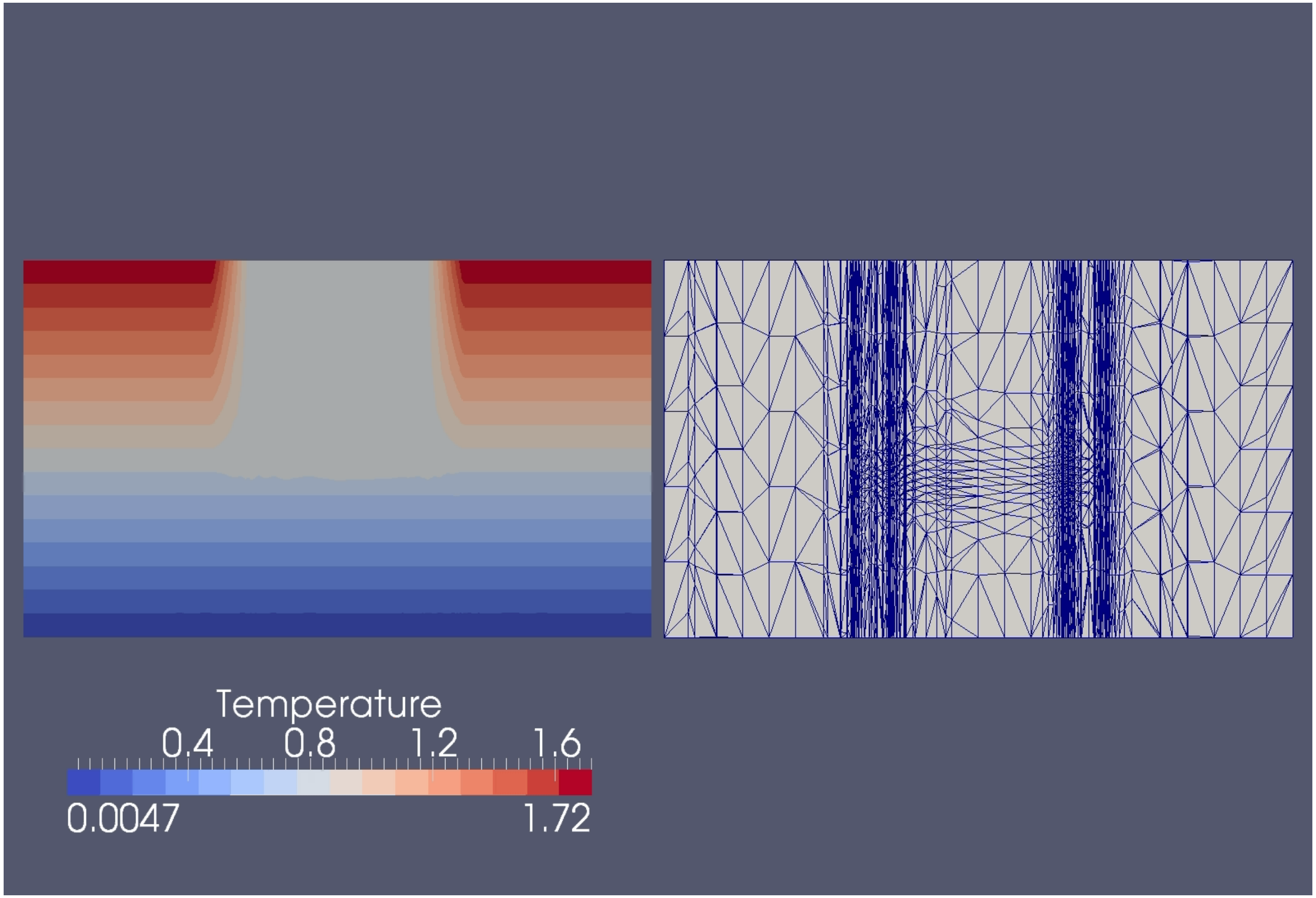
The surface T field and the 2D adapted surface mesh





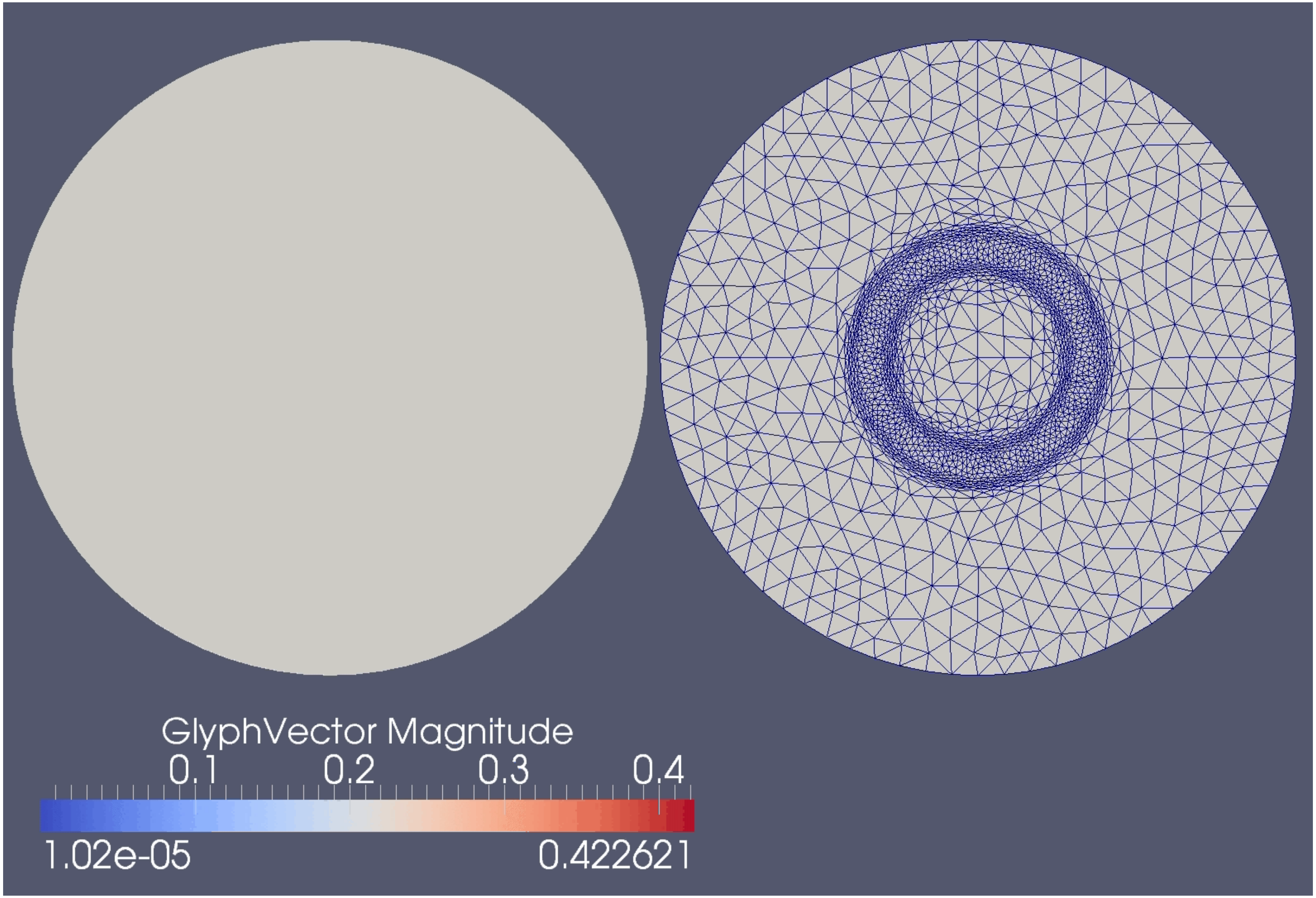
The vertical mesh (a slice through an unstructured mesh always looks a little unpleasant!)



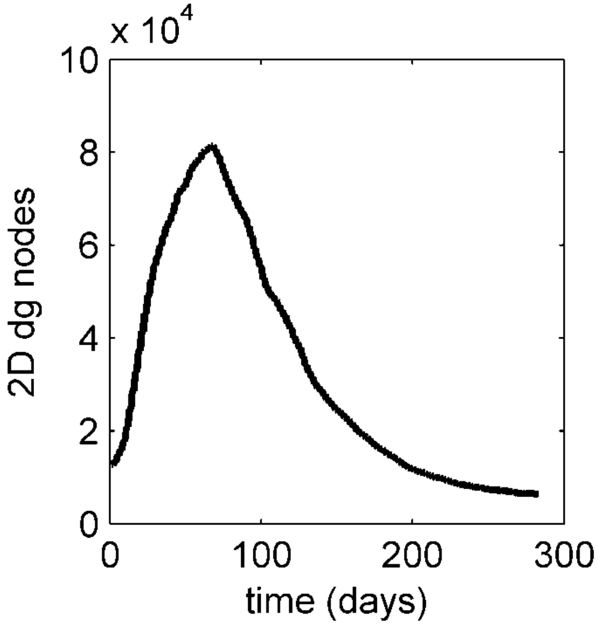


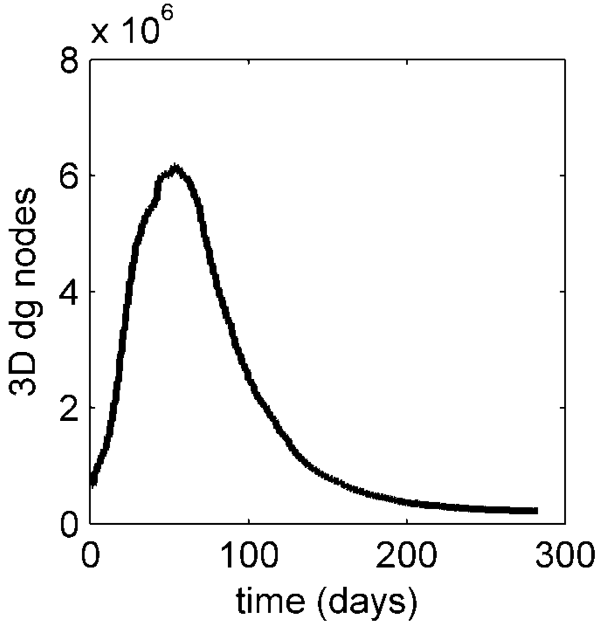
Adapt surface in 2D (using a metric 'collapsed' from 3D) and then in 1D down each column of nodes

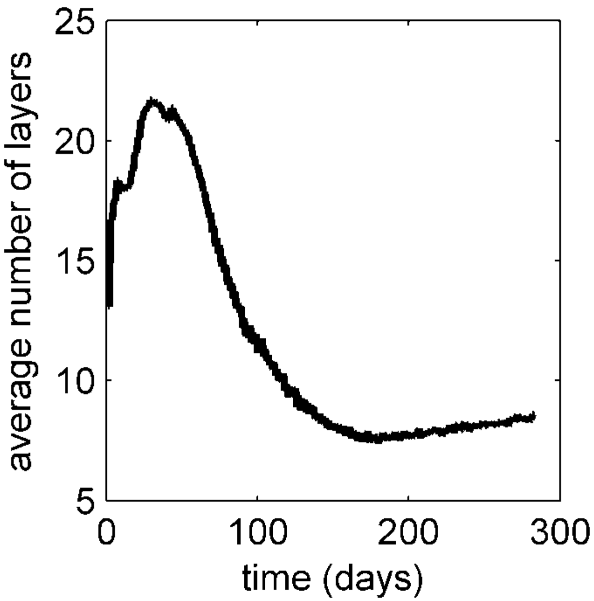
The surface velocity field (NB. metric used here based only on T (0.005 weight))



DG d.o.f. counts for the adaptive restratification problem







- The resolution of the 3D mesh peaks a little earlier that the surface 2D mesh
- Indicating that more refinement is occurring in the vertical earlier in the simulation

<u>Timings</u>

- Fixed: np 64, 136 hours, dt=7200, dx= 2.5km, 15 layers (10@50m 5@100m) 16.2M vel d.o.f.s
- Adaptive: np 8, 231 hours, dt=3600, max/min dx,dz = 2km/20km, 20m/200m, max 6M vel dofs

- 64*136 = 8704; 8*231*0.5 = 924 -- very rough indication of factor 10 reduction in CPU costs

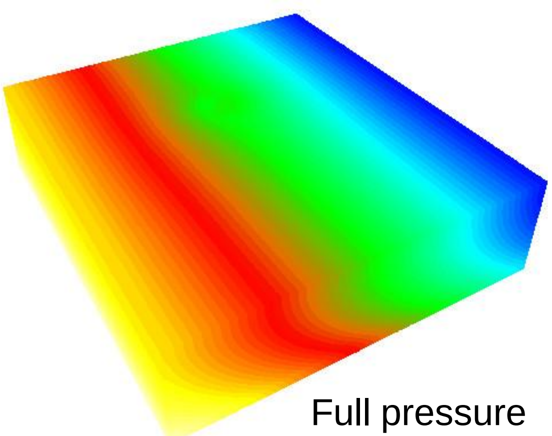
More on the balance problem

Standard CFD FE pairs (e.g. P1P1, P2P1) perform poorly when trying to represent large scale balanced states, i.e. when the horizontal and vertical pressure gradient closely balances Coriolis and buoyancy terms

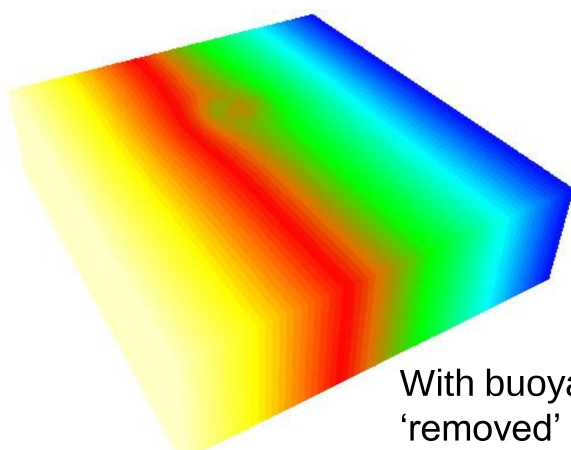
Balanced pressure solver solution: Helmholtz decomposition of Coriolis + buoyancy terms, solve for the dominant balancing pressure (take the divergence and solve elliptic problem)

$$2\Omega \times \mathbf{u} + \rho g \mathbf{k} = \nabla \times \mathbf{A} + \nabla p_b$$

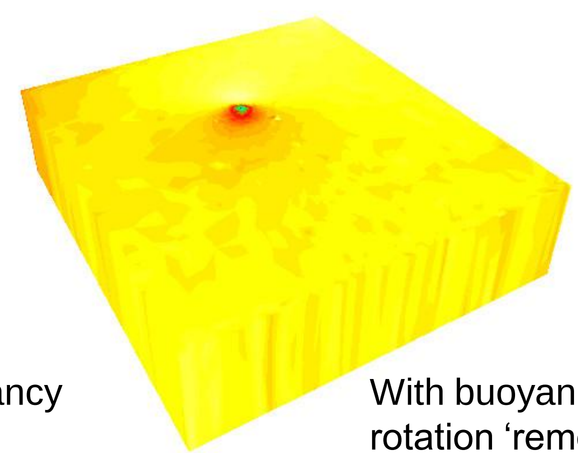
$C^T M^{-1} C_b p_b$ is then included as one of the ‘other terms’ in the pressure projection and has the effect of cancelling out the dynamically unimportant part of the Coriolis and buoyancy terms. Free to use a higher-order representation of the balanced pressure without violating LBB.



Full pressure

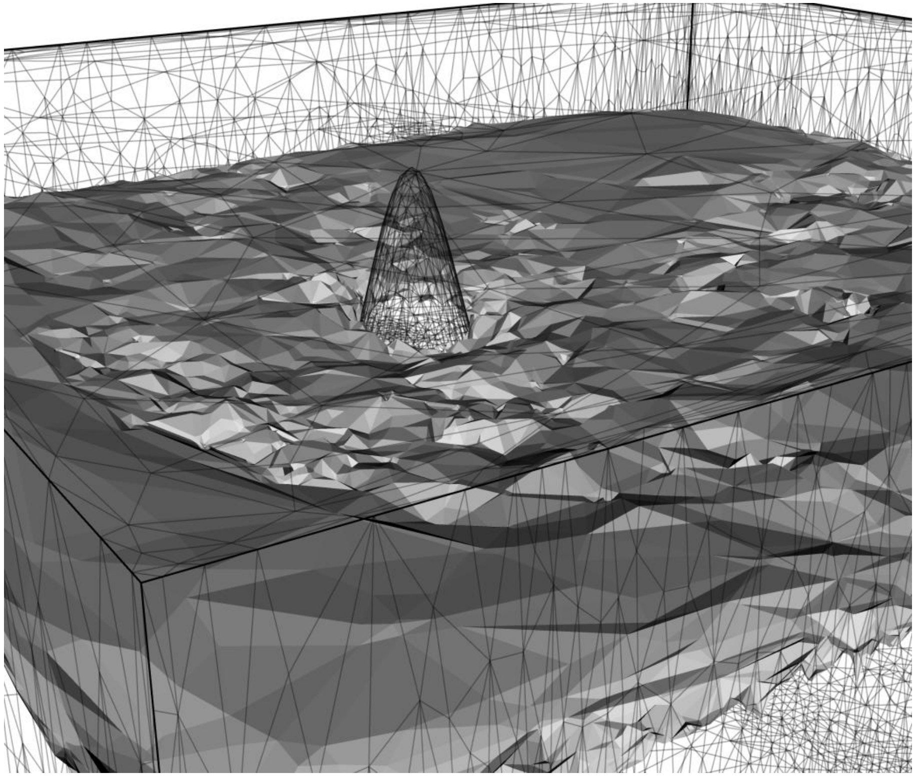


With buoyancy ‘removed’



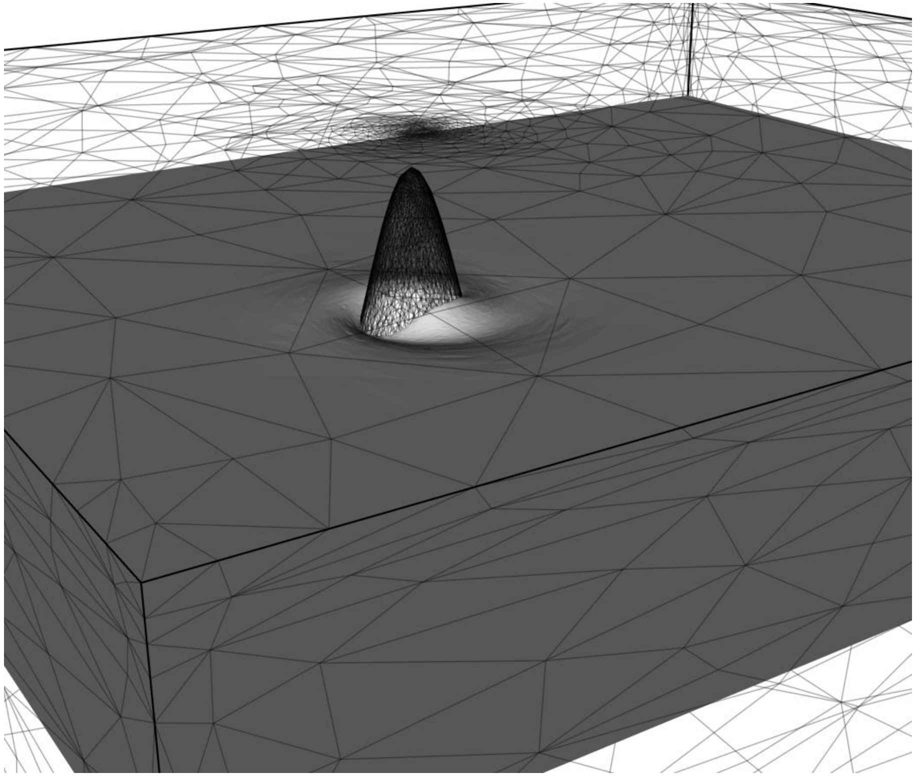
With buoyancy and rotation ‘removed’

Stratified flow past a Gaussian seamount



Stabilised P1P1 with no balanced solver

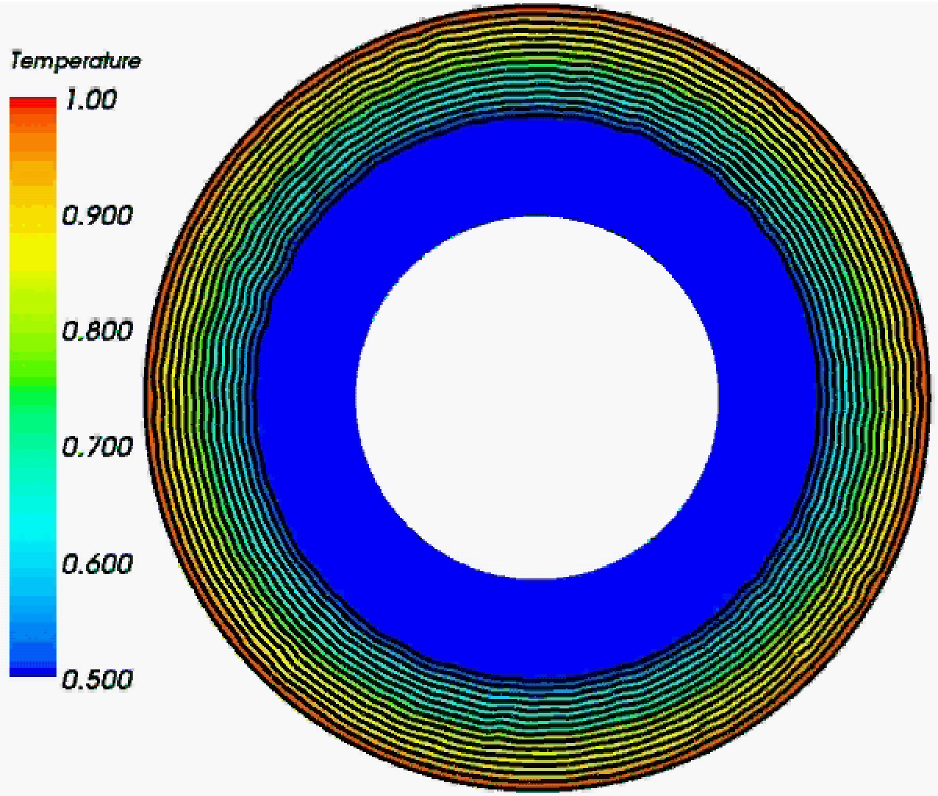
Any noise will have disastrous results on mesh adaptivity – bad result at high cost!

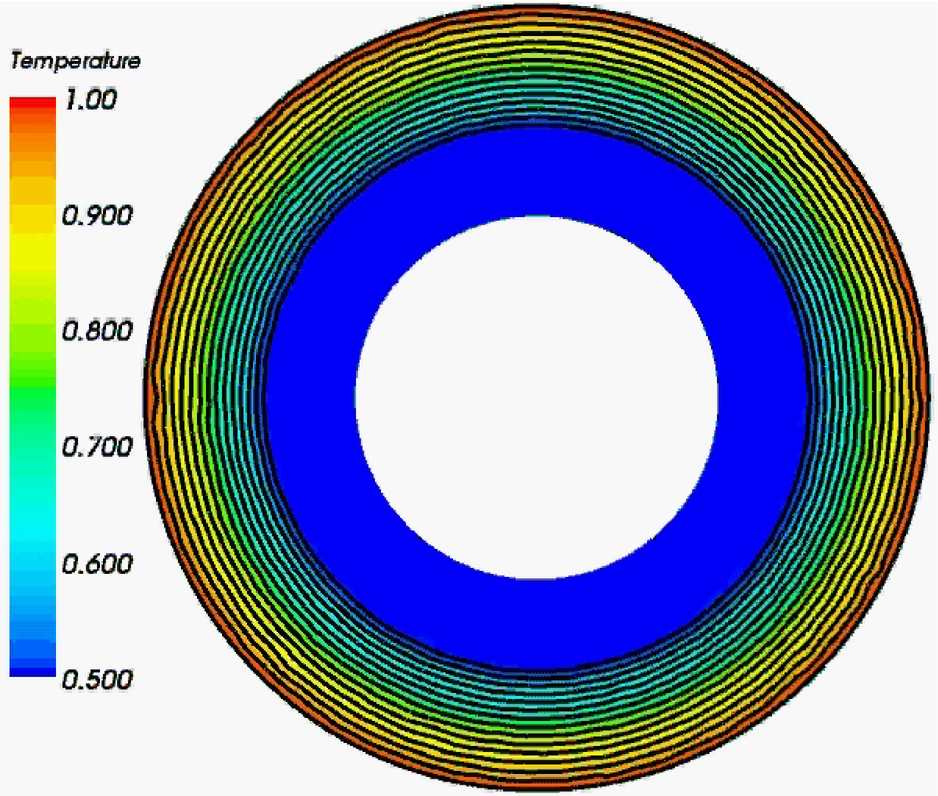


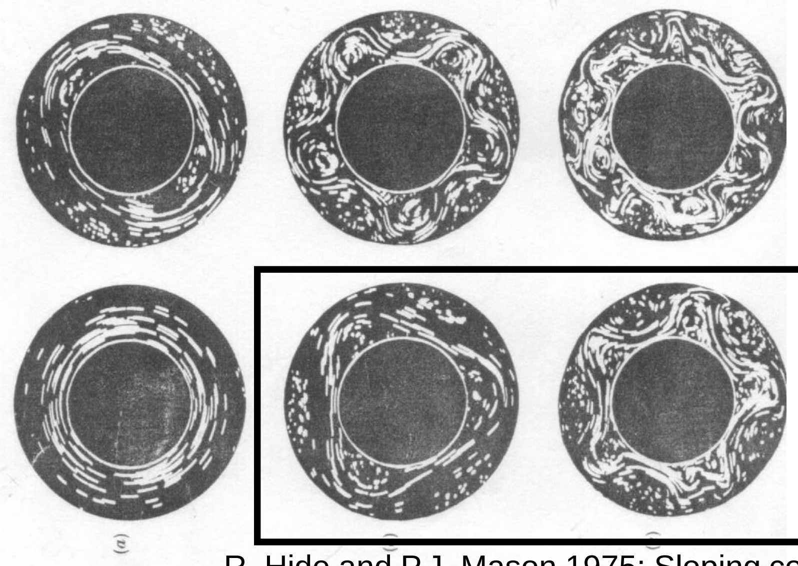
Using the hydrostatic & geostrophic balanced pressure solver, note that the mesh is now not refining to capture noise in the dynamics

Validation against Laboratory experiments

The 3D differentially heated rotating annulus at two rotation rates. The bounds of the normalised temperature have been limited to aid visualisation at the end of the movie



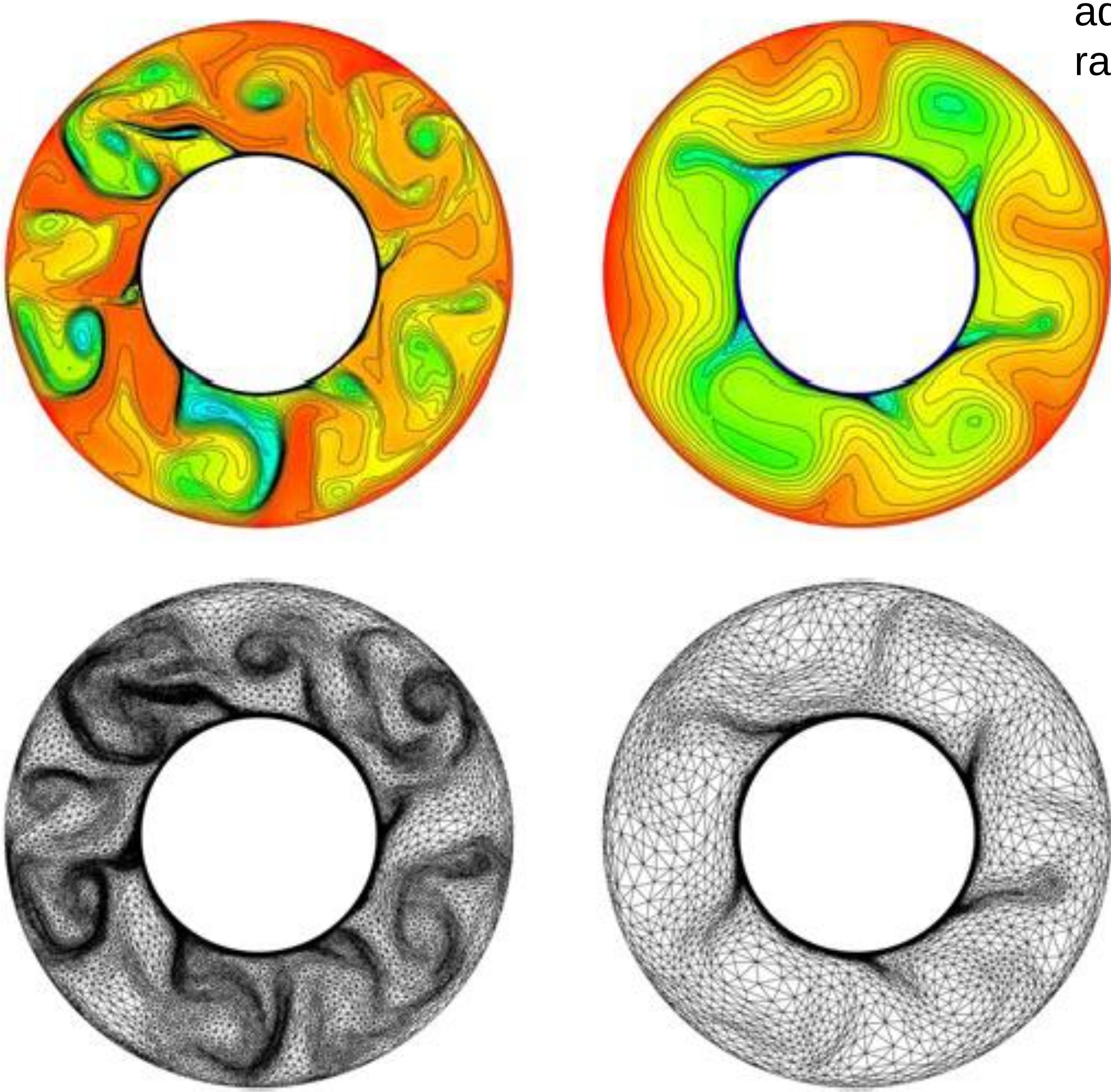


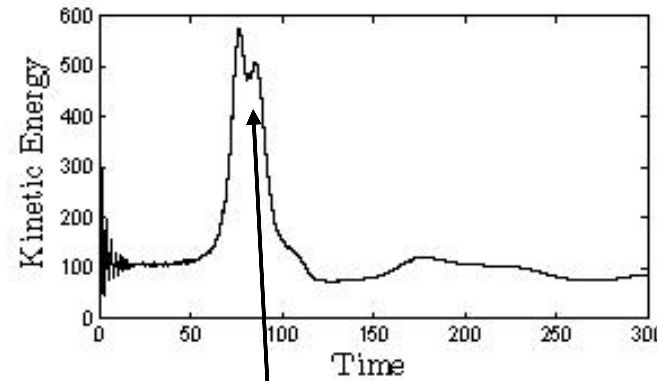


R. Hide and P.J. Mason 1975: Sloping convection in a rotating fluid. Adv. Phys. 24, 47-100

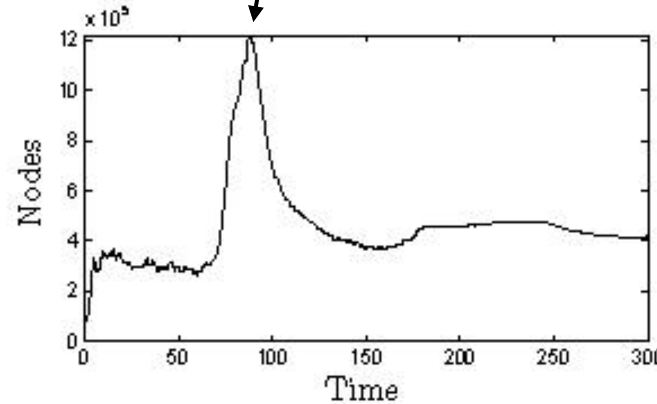
Two snapshots in time at the faster rotation rate

Recent quantitative comparisons with heat transport data from lab runs shows that it is hard to beat an optimised fixed mesh at lower rotation rates, but adaptivity is needed for higher rotation rates (Maddison, 2010)

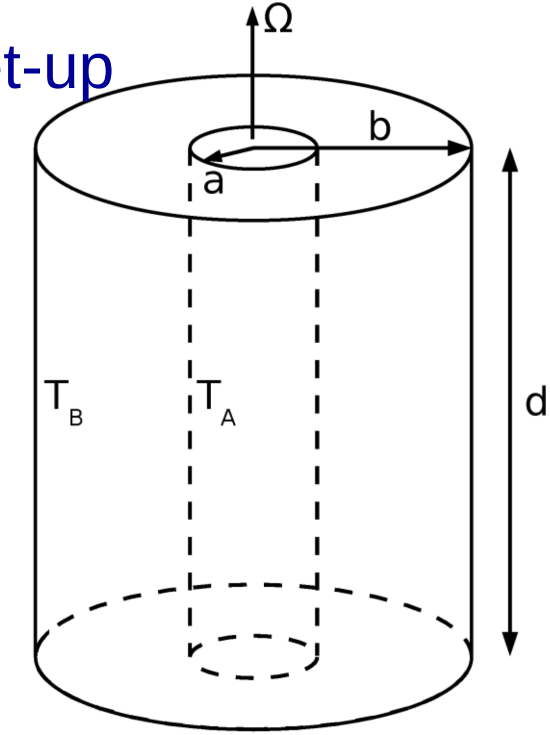




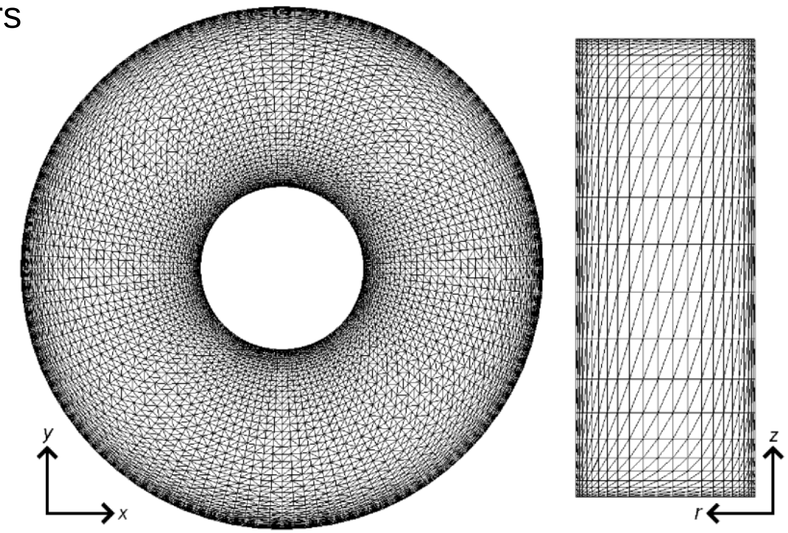
Here the maximum computational cost corresponds to the time of maximum kinetic energy



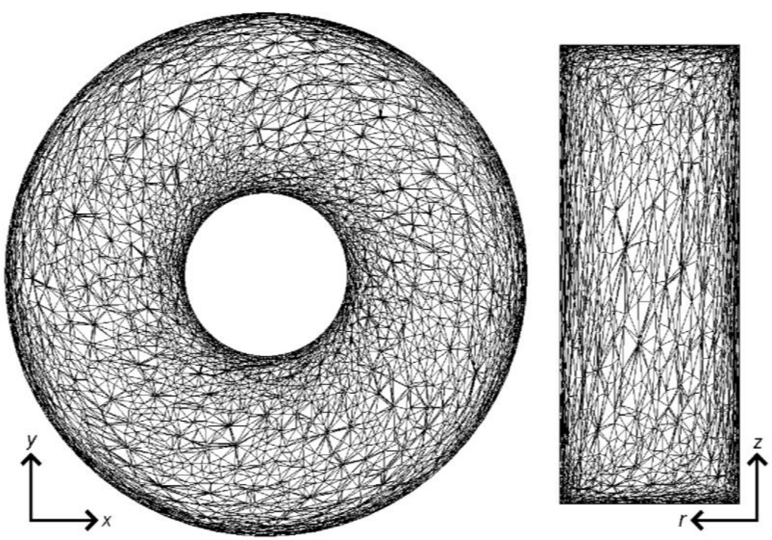
Rotating annulus: quantitative test problem set-up



Stretched structured mesh used by the finite difference MORALS code – shown to reproduce experiments well with this telescoping into boundary layers



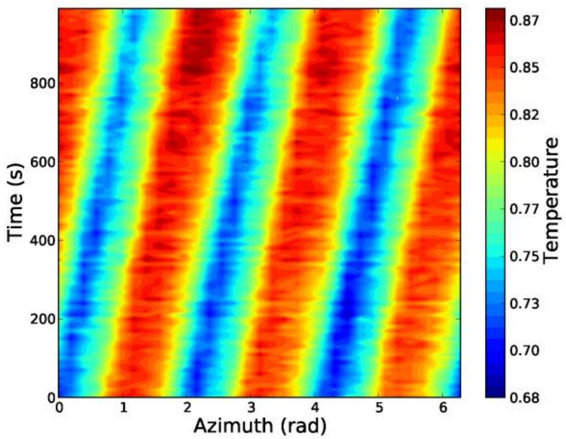
The above mesh spacing is used to derive a metric which a fully unstructured tet mesh is generated



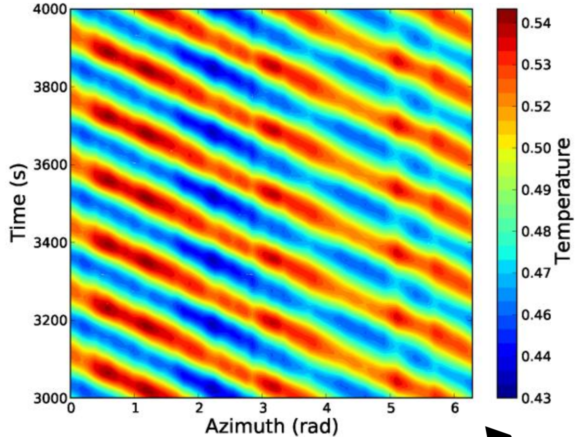
Parameter	Symbol	Value
Minor radius	a	2.5 cm
Major radius	b	8.0 cm
Depth	d	14.0 cm
Kinematic viscosity	ν	2.1×10^{-2} cm^2 s^{-2}
Thermal diffusivity	κ	1.3×10^{-3} cm^2 s^{-2}
Volumetric expansion coefficient	α	3.3×10^{-4} K^{-1}
Gravitational acceleration	g	981 cm s^{-2}
Inner wall temperature	T_A	18 °C
Outer wall temperature	T_B	22 °C
Rate of rotation	Ω	1 rad s^{-1}

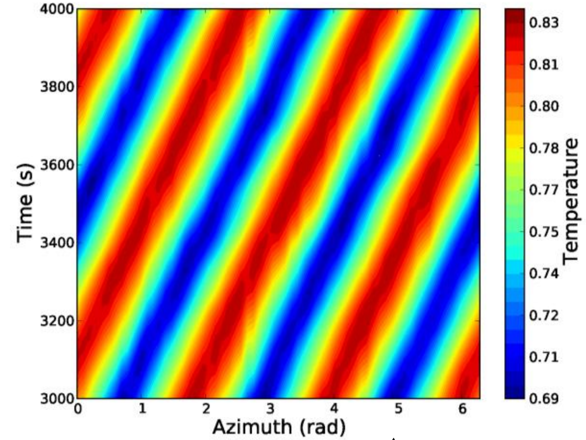
Comparisons with data: thermocouple at mid-height mid-radius and system heat transport

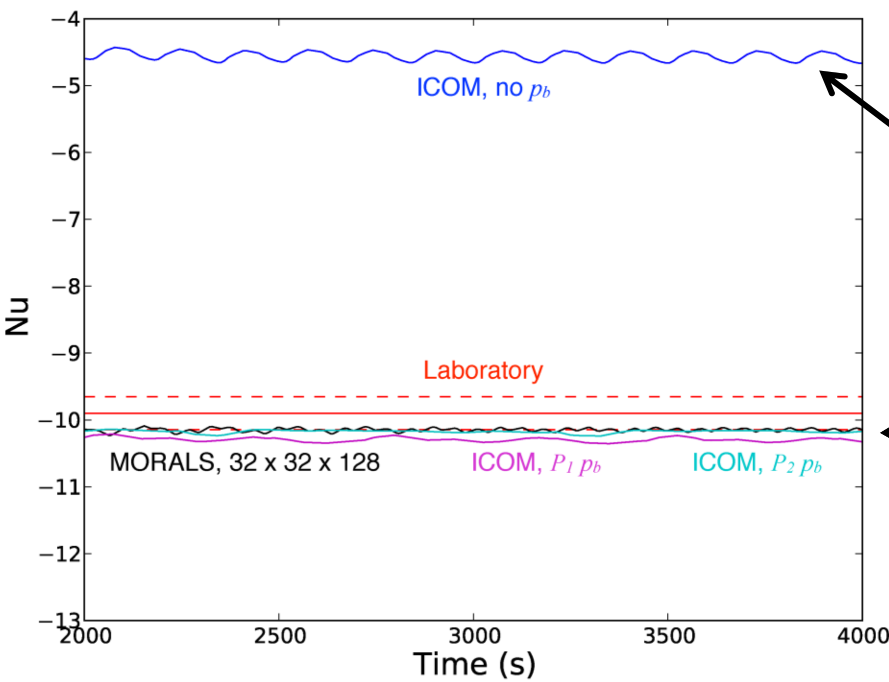
Maddison et al., 2011, doi:10.1016/j.ocemod.2011.04.009



Lab data







The standard discretisation gets the answer completely wrong!

With balanced pressure solver things much better P1 similar to P2

A free surface method suitable for fully unstructured non-hydrostatic models (implicit, mass conservative)

- The kinematic free surface boundary condition can be written (in arbitrary coordinate frame) as

$$\frac{\partial \eta}{\partial t} = \frac{n \cdot u}{n \cdot n_g}$$

- Assuming the Boussinesq approximation, and that we have subtracted out the ‘hydrostatic mode’ associated with ρ_0 then on the free surface we have

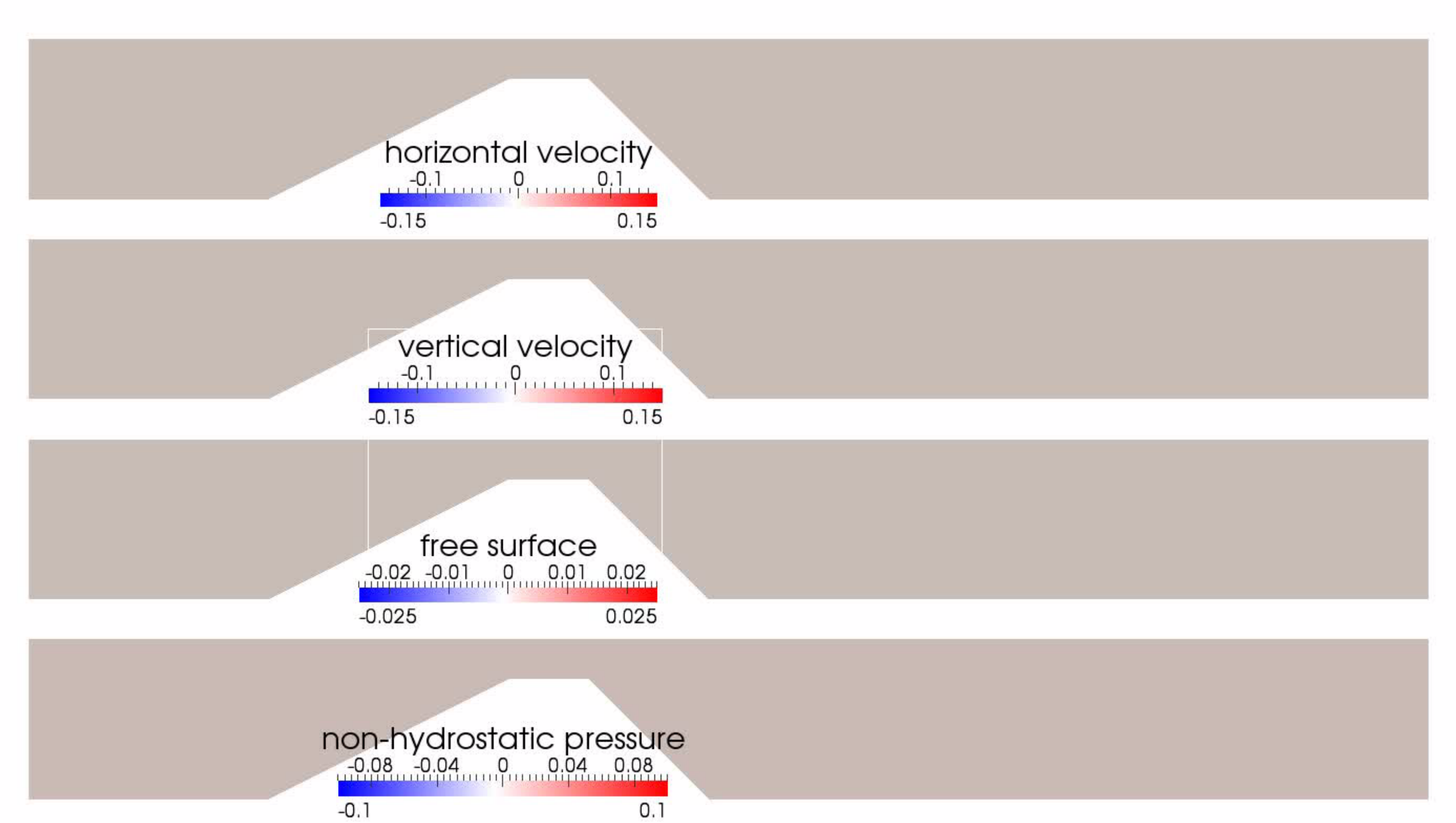
$$p = \rho_0 g \eta \implies \frac{\partial p}{\partial t} = \rho_0 g \frac{n \cdot u}{n \cdot n_g} \quad \text{on} \quad \Gamma_{\text{fs}}$$

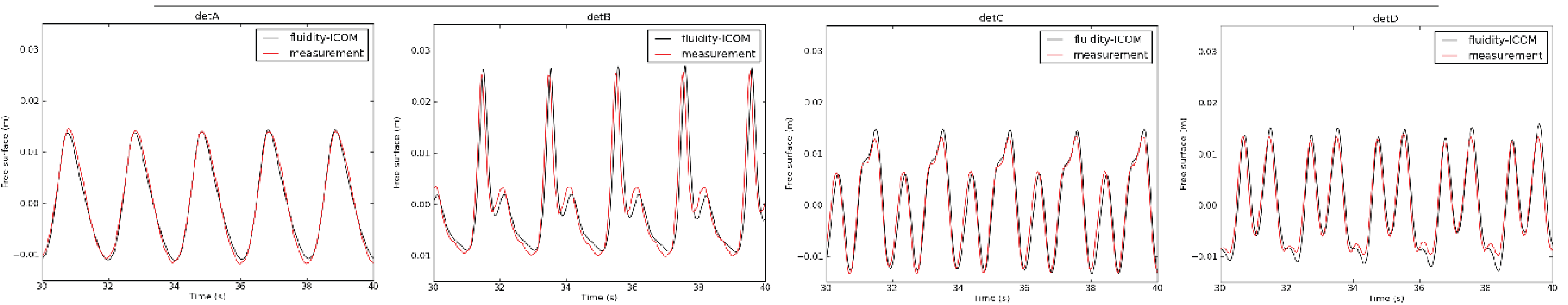
- Now the weak form of continuity looks like

$$\begin{aligned} 0 = \int_\Omega \psi \nabla \cdot u &= -\int_\Omega \nabla \psi \cdot u + \int_{\Gamma_b} \cancelto{0}{\psi n \cdot u} + \int_{\Gamma_{\text{fs}}} \psi n \cdot u \\ &= -\int_\Omega \nabla \psi \cdot u + \int_{\Gamma_{\text{fs}}} \psi \frac{n \cdot n_g}{\rho_0 g} \frac{\partial p}{\partial t} \end{aligned}$$

- Upon discretisation in space and time the final term becomes one of the ‘other terms’ mentioned in the pressure projection method
- Solution gives us a pressure for which the value at the free surface is related to η, we can choose to move the mesh based on this, but need to re-compute mass matrices etc

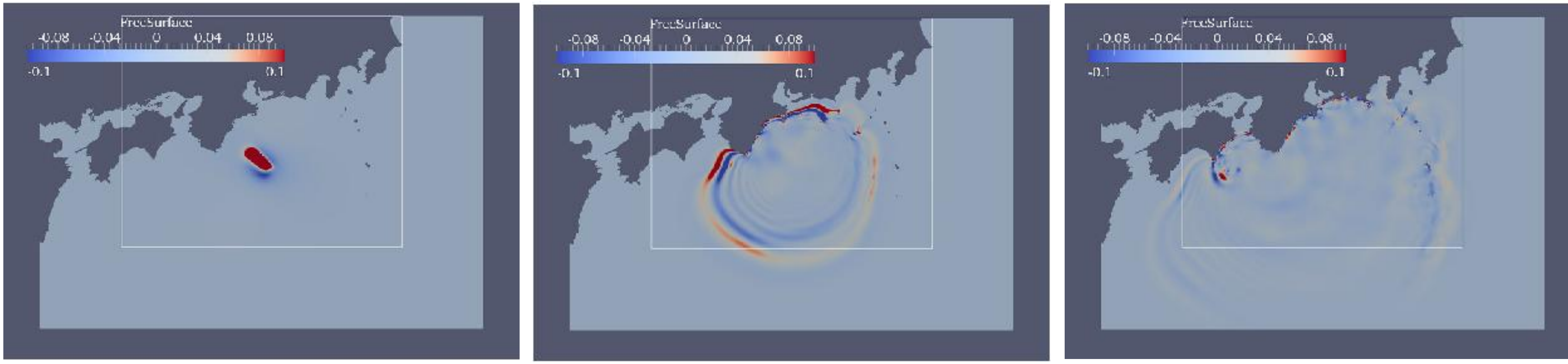
Small scale test problem – Beji-Battjes, comparison to lab experiments, Kramer et al., 2011



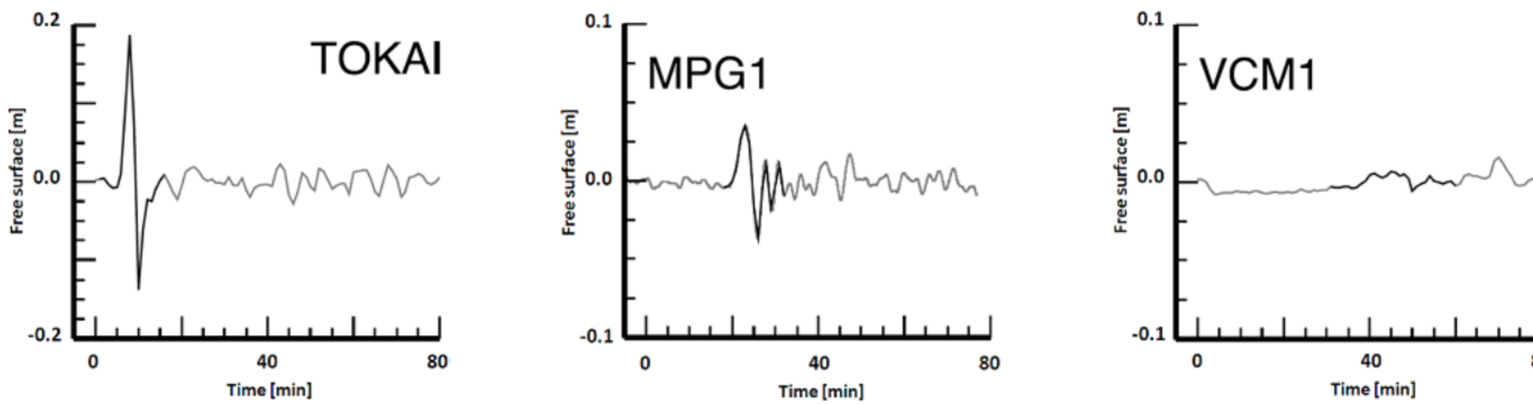


Tsunami (2004 off Kii peninsula) simulation

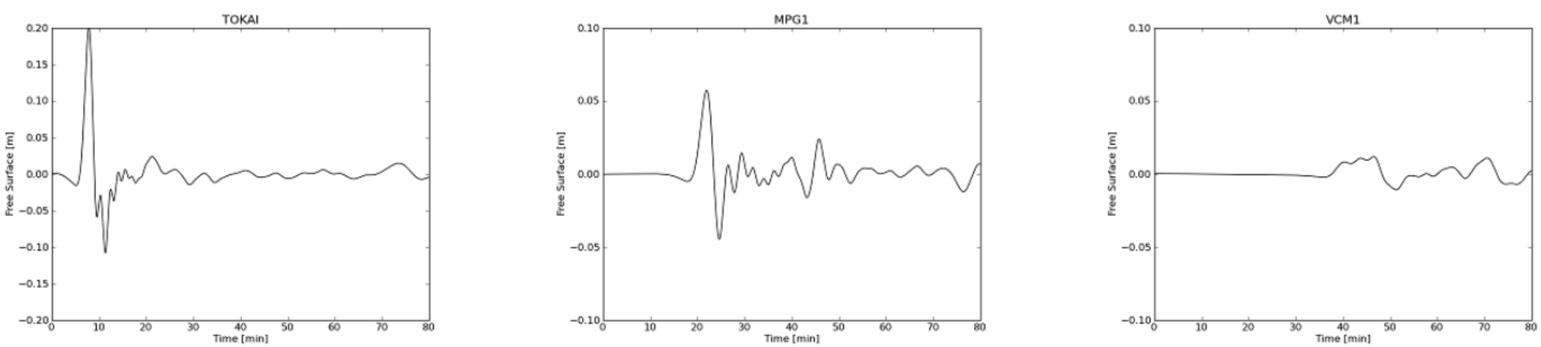
Study conducted by Dr Yusuke Oishi (Fujitsu Laboratories of Europe)



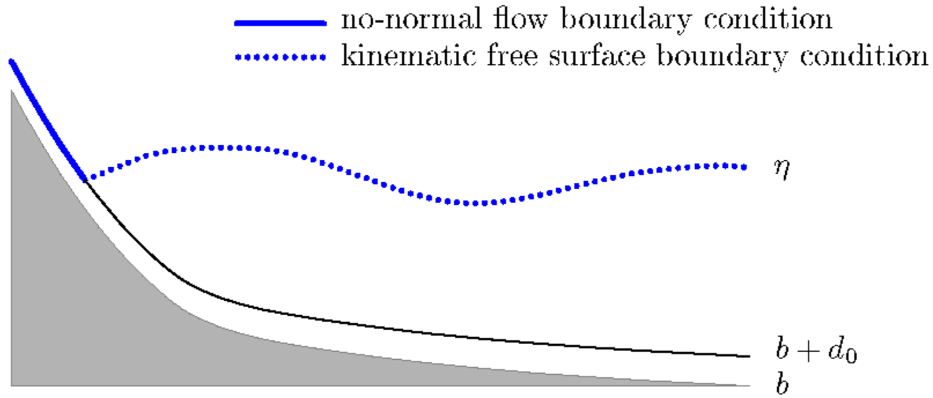
Observations



Numerical



Simple extension to deal with inundation



- Change the boundary condition applied via continuity if the water column depth falls below some small tolerance d_0 :

$$\eta = \max\left(\frac{p}{\rho_0 g}, b + d_0\right) \implies \frac{n \cdot n_g}{\rho_0 g}\frac{\partial}{\partial t}\max\left(\frac{p}{\rho_0 g}, b + d_0\right) = n \cdot u$$

- Discretise this in space and time and this leads to an alternate form of the ‘other terms’ added previously to incorporate a free surface pressure boundary condition in the pressure projection

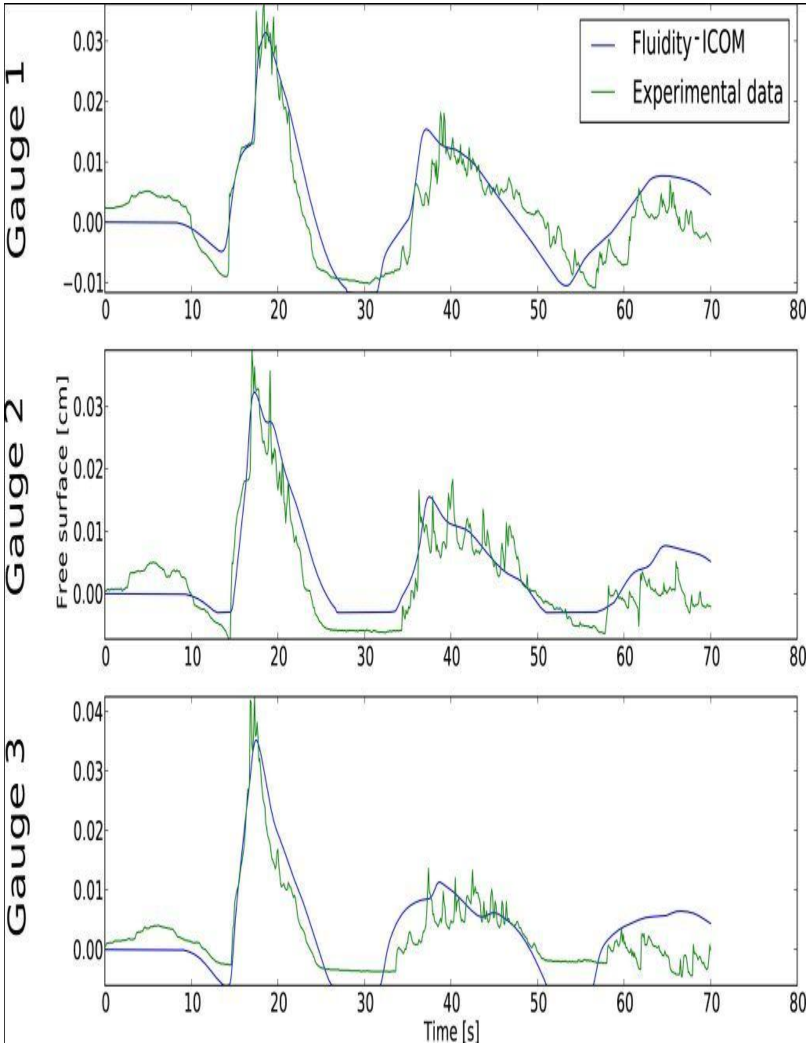
- Apply depth dependent bottom drag to ‘kill off’ motion in the remaining thin layer

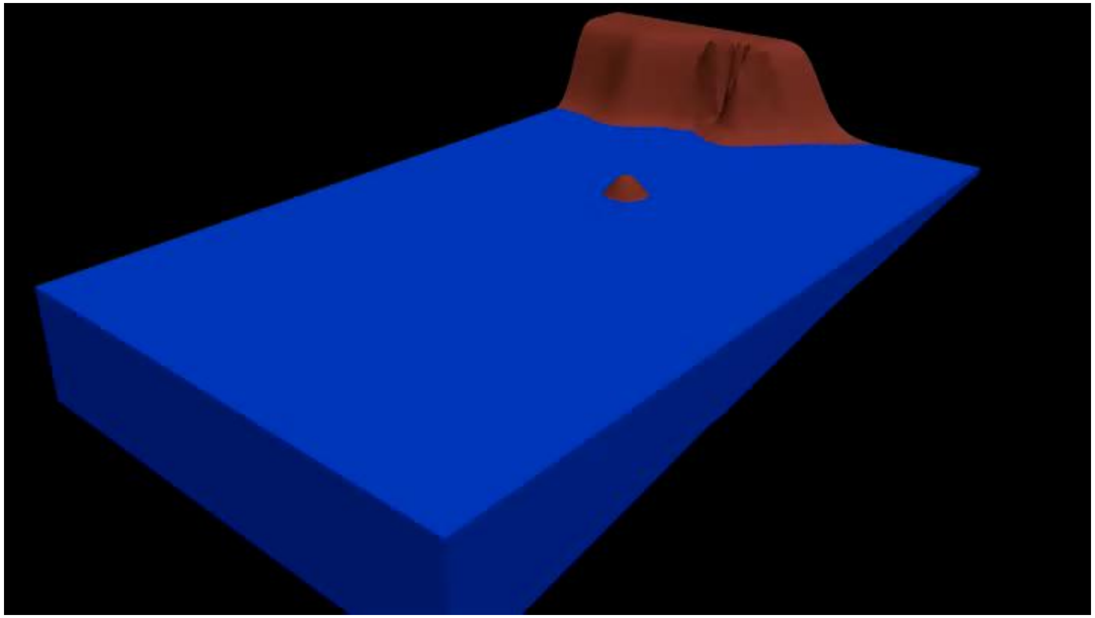
- Retains the implicit, mass-conservative, and ability to work with arbitrary mesh properties

- See Funke et al., submitted, 2011 for further details and validation

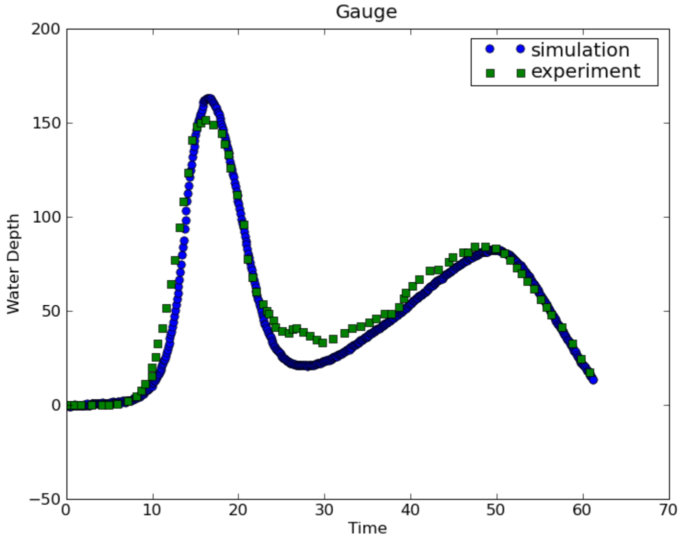
Hokkaido-Nansei-Oki tsunami (1993); benchmark lab data for inundation at Monai Valley, Okushiri Island, Japan

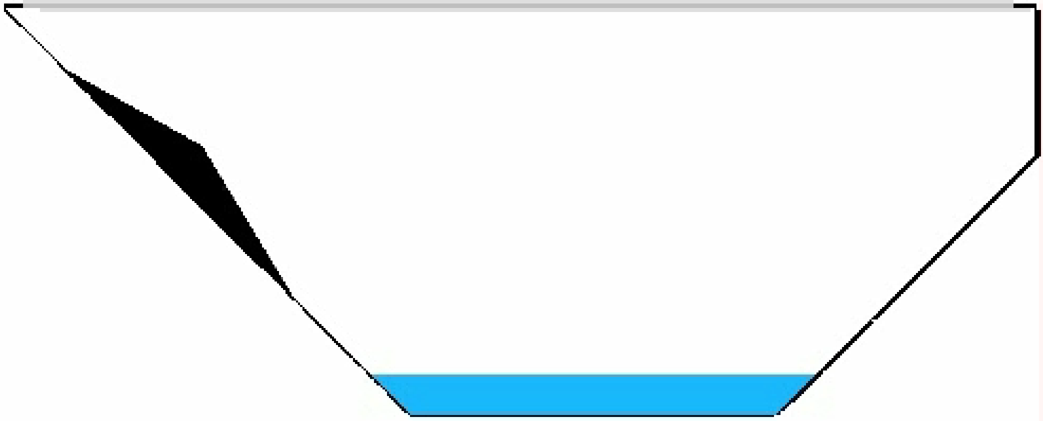
http://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory_MonaiValley/index.html;   Funke et al., 2011

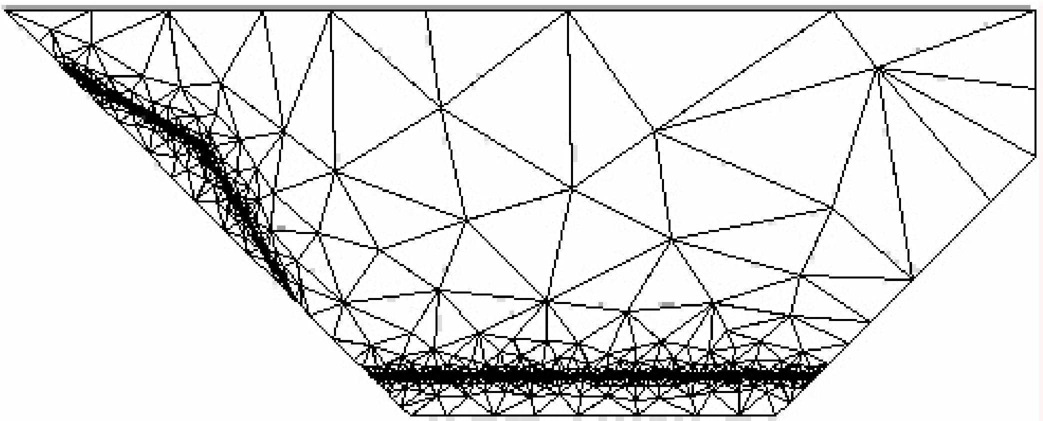


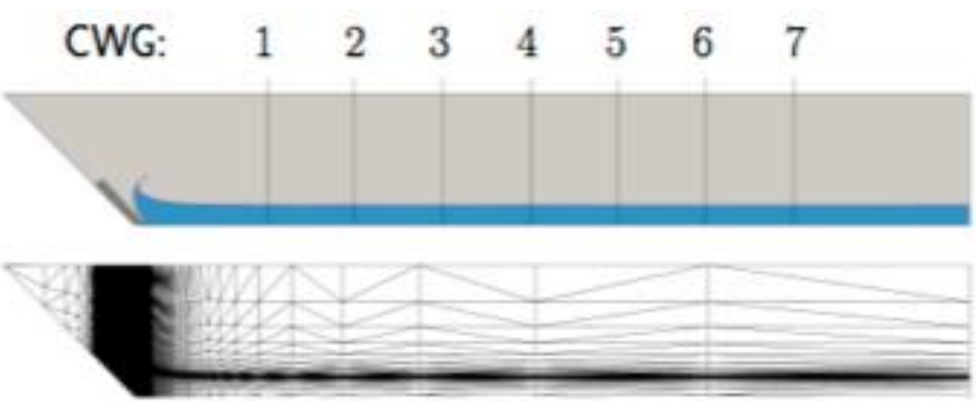


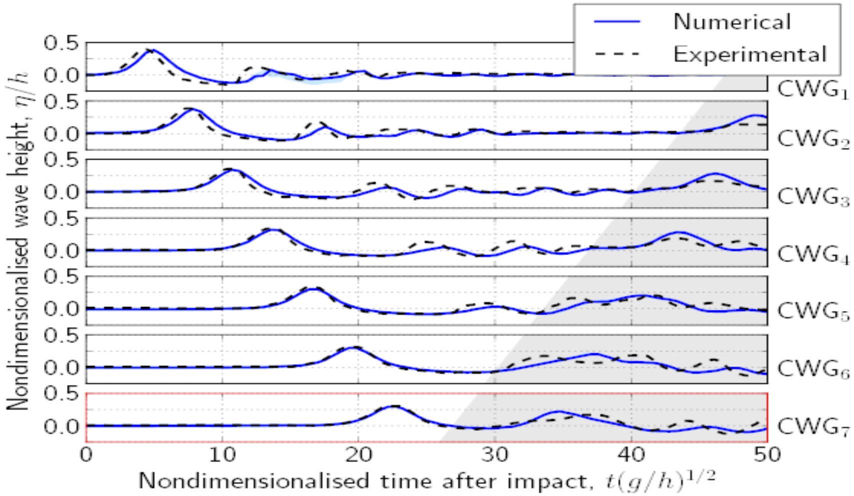
Landslide generated waves simulated with a multi-material approach (Wilson et al., 2010)



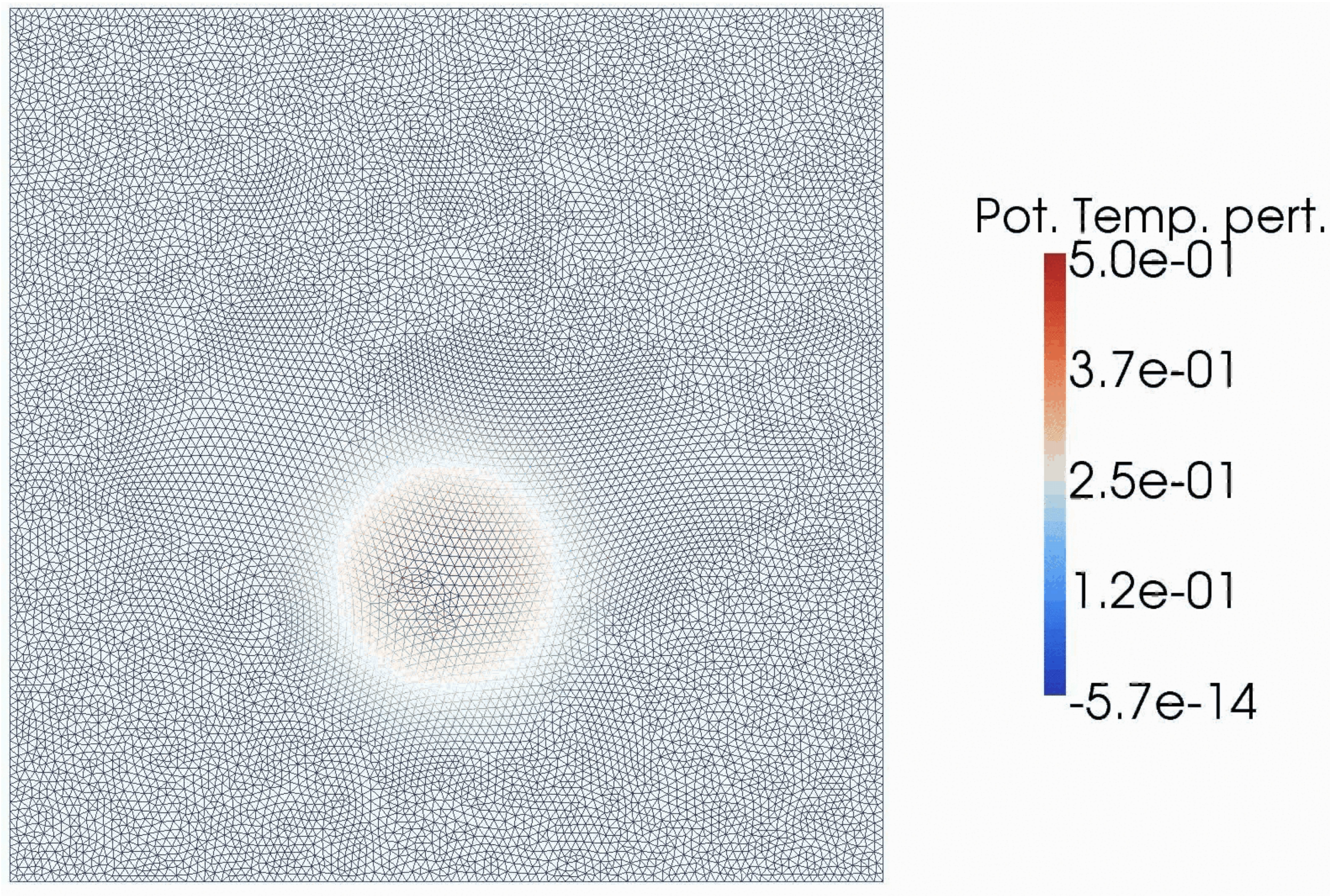




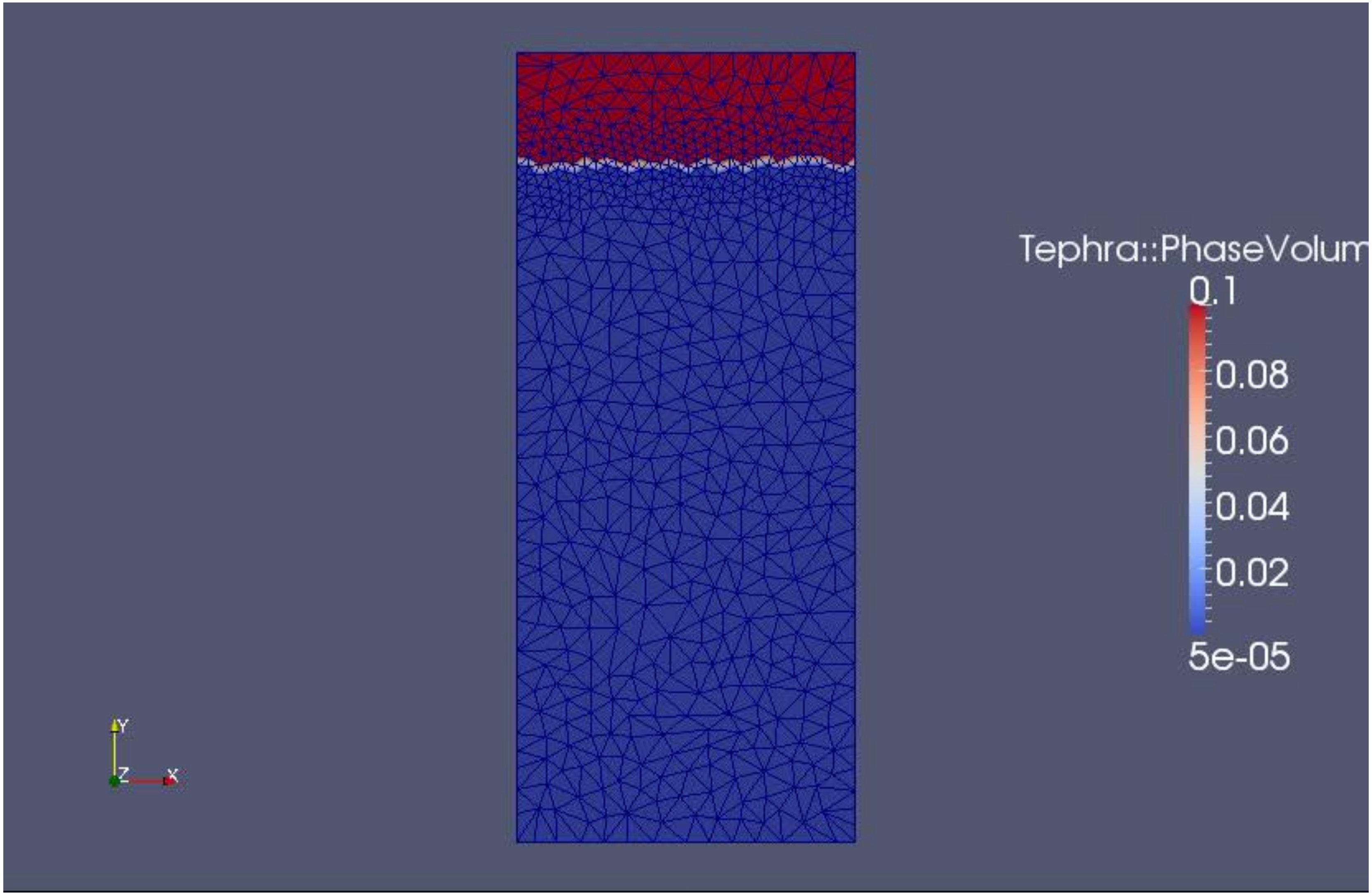




Compressible Navier-Stokes: warm rising bubble example
(Nelson et al., 2011)



A multi-phase simulation of tephra settling in the ocean (Jacobs et al.)



Summary

- Demonstrated CFD and GFD simulations using adaptive anisotropic unstructured meshes

- Robustness and flexibility over mesh, discretisation methods, equation sets and code is key – multi-disciplinary progress, skills/ideas exchange, funding!

- Robustness of the underlying numerical method is obviously always key, but feedbacks with the ability of the model to resolve spurious grid scale noise an ‘interesting’ property

- The principal hurdle now is probably computational cost

- Of course in addition to testing the inclusion of more real-world configurations, processes, forcings etc

- Active research areas not discussed: multi-physics (coupling/OASIS4, sea ice, ice shelves, biology, sediment, solids), adjoints (DWR errors), reduced order models, hybrid massively parallel optimisation, GPUs, code generation, geodynamics (Stokes), air pollution, tidal power, wave breaking defences, ...

Finally

- If you would like the code: http://launchpad.net/fluidity (Open Source – LGPL license)
- 4 month (22Aug-21Dec 2012) Isaac Newton Institute Programme on Multiscale Numerics for the Atmosphere & Ocean (http://www.newton.ac.uk/programmes/AMM/)
- If you would like an invitation let me know (m.d.piggott@imperial.ac.uk)

Isaac Newton Institute for Mathematical Sciences

Multiscale Numerics for the Atmosphere and Ocean

22 August - 21 December 2012

Organisers: Dr D Ham *(Imperial College London)*, Dr M Piggott *(Imperial College London)*, Dr T Ringler *(Los Alamos)*, Dr H Weller *(Reading)* and Dr N Wood *(Met Office)*

Programme Theme

Numerical models of the atmosphere and ocean have proved to be immensely valuable forecasting tools for short time-scale weather and longer time-scale seasonal and climate prediction. As the decades pass, these models have been improving due to increased computing power, improved modelling of the dynamics, improved parametrisation of sub-grid scale processes and improved use of observations. These modelling improvements may be slowing and further large increases in computing power will almost certainly emerge from heterogenous computing architectures configued in even more massively parallel machines. If we are unable to exploit these new opportunities in high-performance computing, our current models and codes risk becoming obsolete.

This programme will bring together leading developers of ocean and atmosphere models with numericists and computer scientists to explore radical new formulations which will address the limitations of current models and enable the most effective use of the computing platforms of the future.

Particular themes will include:

Adaptive simulation techniques

Adaptive meshes
Criteria for refinement
Mesh movement vs mesh refinement vs locally increasing polynomial order
Data assimilation and inverse problems on adaptive meshes

Numerical techniques

Closures accurate over a wide range of resolutions
Discretisations suited to the atmosphere and ocean
Solution of equation sets appropriate for the local mesh resolution
Preservation of balance, conservation, monotonicity, accuracy and high curvature under adaptation
Spurious wave reflection and refraction from mesh inhomogeneity

Computing techniques

Algorithims with sufficient computational efficiency and parallelism
Mapping numerical schemes to emerging massively parallel computer architectures