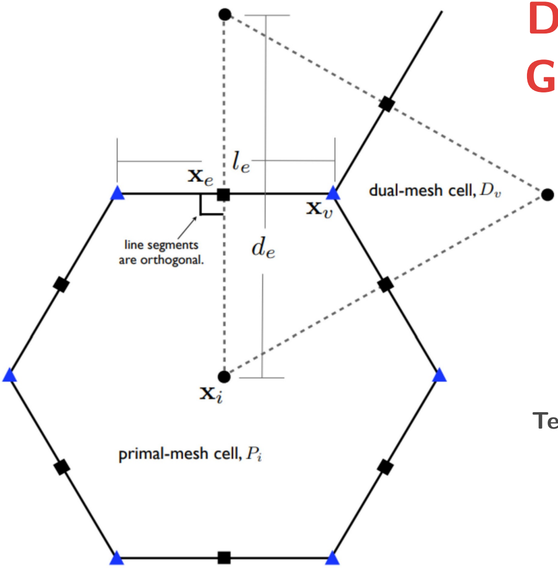
Optimal Primal-Dual Meshes & Discretisations for Geophysical Fluids



DARREN ENGWIRDA

Theoretical Division
Los Alamos National Laboratory
Luminary Cloud Inc

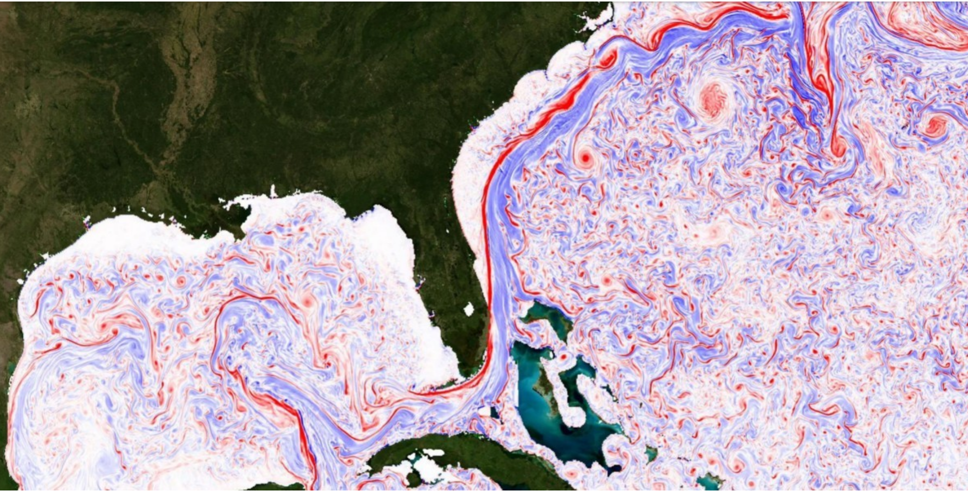
dengwirda@luminarycloud.com

Tetrahedron VII — October, 2023

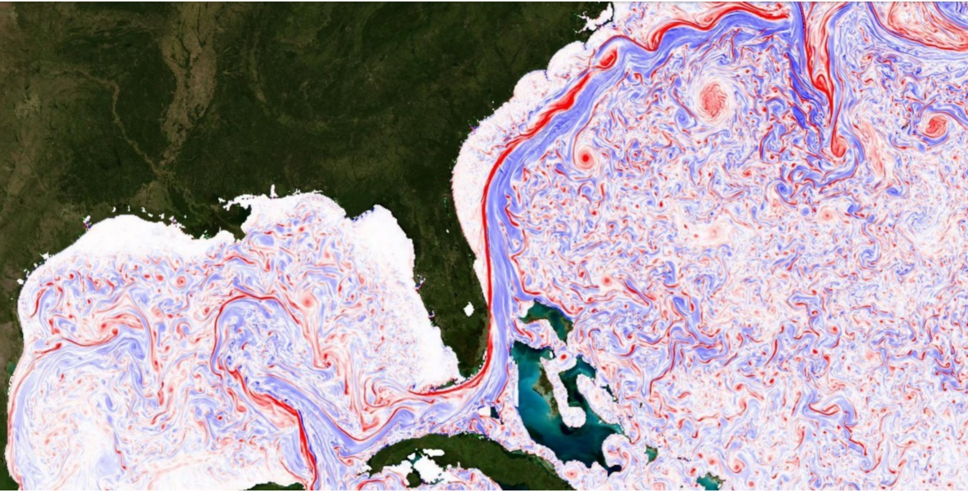
U.S. DEPARTMENT OF ENERGY

Los Alamos NATIONAL LABORATORY

Big whorls have little whorls
That feed on their velocity,
And little whorls have lesser whorls
And so on to viscosity — Lewis Richardson, 1920

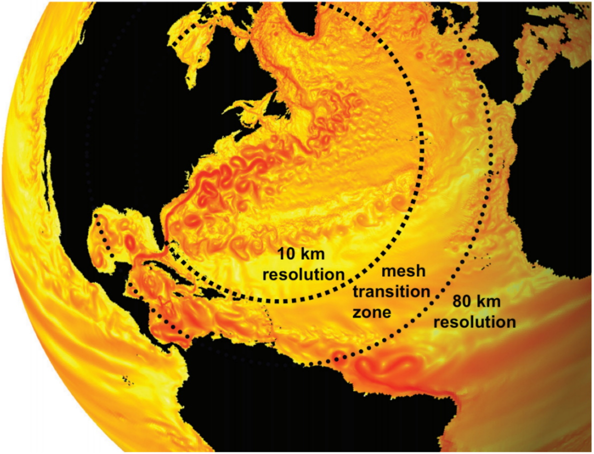


Geophysical flows are characterised by **multiscale turbulence** —
a 'cascade' of eddies at smaller and smaller length scales
(and larger and larger nonlinearity)



Multiscale geophysical flows

Resolving 'everything' at very high-resolution is too computationally expensive — **unstructured meshes used to perform scale-selective eddy resolving simulations**

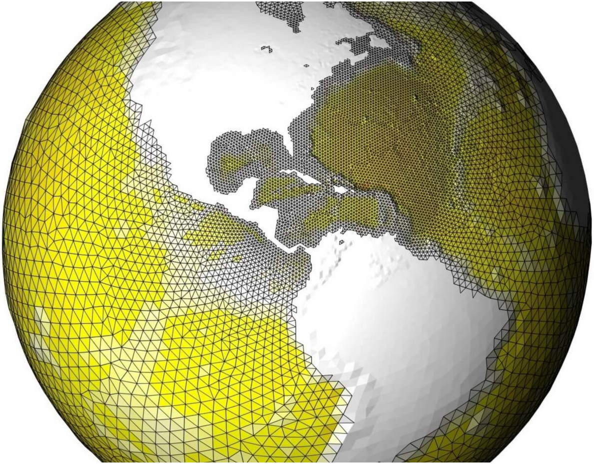


**North Atlantic Eddy Dynamics: MPAS project, Petersen et al, 2015.

Turbulent ocean dynamics resolved by US-DOE's Model for Prediction Across Scales (MPAS-O).

Multiscale geophysical flows

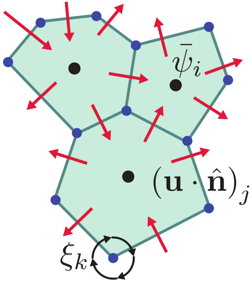
Resolving 'everything' at very high-resolution is too computationally expensive — **unstructured meshes used to perform scale-selective eddy resolving simulations**



**North Atlantic Eddy Dynamics: MPAS project, Petersen et al, 2015.

Requires solution of hard meshing problem: grids must be **centroidal**, **well-centred**, **orthogonal** and **smoothly graded**.

‘Mimetic’ Discetisation Schemes



Solve ‘rotating’ Naiver-Stokes system:

$$\frac{\partial \mathbf{u}}{\partial t} + qh\mathbf{u}^{\perp} = -\frac{1}{\rho_0}\nabla p - \nabla\frac{1}{2}\|\mathbf{u}^2\| + \nabla\cdot(\mathbf{K_u}\nabla\mathbf{u}),$$

$$\frac{\partial h}{\partial t} + \nabla\cdot(hu) = 0,$$

$$\frac{dp}{dz} = -g\rho, \quad \rho = f(\psi, p),$$

$$\frac{\partial \psi}{\partial t} + \nabla\cdot(u\psi) = \nabla\cdot(\mathbf{K}_\psi\nabla\psi)$$

$\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ velocity, $q = \frac{\xi+f}{h}$ potential vorticity, $h = h(\mathbf{x}, t)$ fluid thickness, ψ conserved tracers.

(Orthogonal) staggered unstructured scheme **conserves energy**, **vorticity**, **enstrophy**, **mass**.

Use of **structure-preserving** (mimetic) schemes important wrt. long time-scale dynamics...

...but requires **‘near perfect’** unstructured meshes.

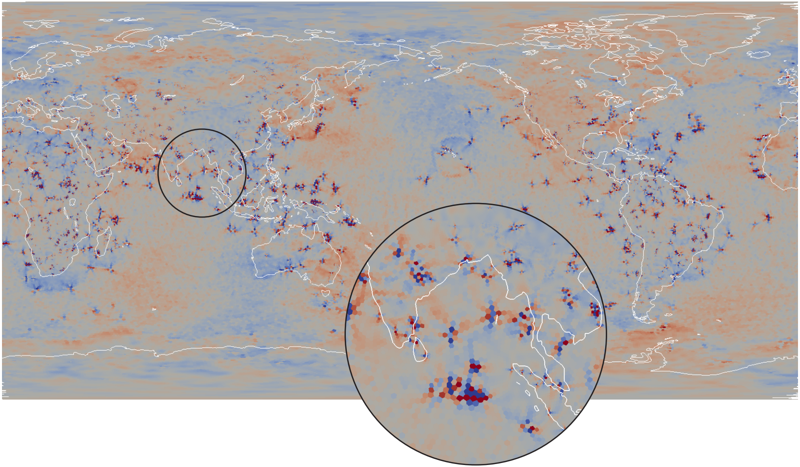
‘Mimetic’ Discetisation Schemes

Why/what are mimetic / structure-preserving discretisations?

- Numerical schemes that ‘mimic’ the continuous properties of the PDEs — discrete operators satisfy continuous vector calculus identities.
- Conserving mass is ‘easy’ — conserving higher-order moments of the flow is hard — discrete Hamiltonian’s, sympectic approaches...
- For geophysical flows, conserve energy (kinetic + potential) and enstropy (the square of potential vorticity) is important.
- Ensures that PDE solution sits on correct manifold in phase space — prevent unphysical equilibria.
- Important for integration of systems over very long time-scales: climate models require O(1000yr) simulations...

What can go wrong...?

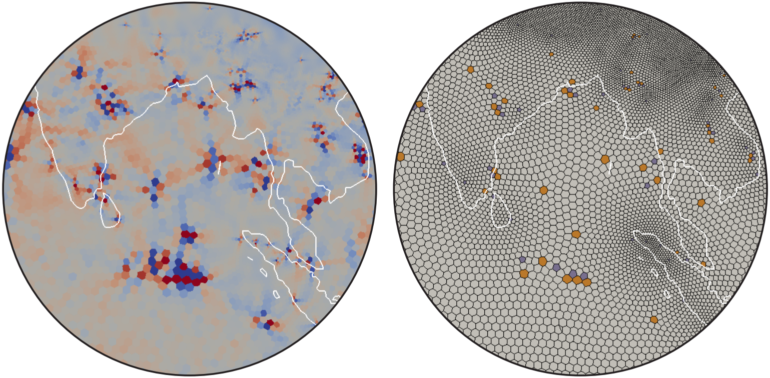
Error in fluid height; Centroidal Voronoi-type multiscale mesh:



Significant **grid-scale imprint** a manifestation of poor convergence in the L^{∞}-norm: mimetic numerics very sensitive to mesh!

What can go wrong...?

Meshes are heavily optimised CVT structures:

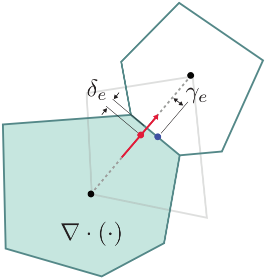


Must require **something more** than standard Delaunay-Voronoi meshes + numerics can offer...

Source of Discretisation Error

'Defect' in primal-dual staggering leads to errors in $\nabla \cdot (\cdot)$, $\nabla(\cdot)$, $\nabla \times (\cdot)$:

- Offsets from edge centroids to primal-dual bisectors δ_e, γ_e.
- Offsets from primal vertices to dual centroids γ_f.
- Offsets from dual vertices to primal centroids δ_f.



$$\int_{d_i} \nabla \cdot (\mathbf{u}\,\psi)\, \mathrm{d}A = \oint_{\partial d_i} (\mathbf{u} \cdot \hat{\mathbf{n}})\,\psi\, \mathrm{d}s \qquad (1)$$

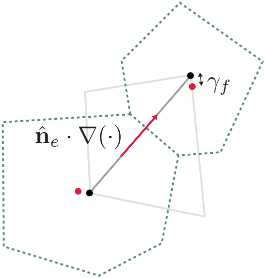
$$\simeq \sum_{e=1}^{n} \int_e (\mathbf{u} \cdot \hat{\mathbf{n}})_e\, \psi_e\, \mathrm{d}l \qquad (2)$$

Only 2nd-order accurate if $\delta_e = 0$, $\gamma_e = 0$.

Source of Discretisation Error

'Defect' in primal-dual staggering leads to errors in $\nabla \cdot (\cdot)$, $\nabla(\cdot)$, $\nabla \times (\cdot)$:

- Offsets from edge centroids to primal-dual bisectors δ_e, γ_e.
- Offsets from primal vertices to dual centroids γ_f.
- Offsets from dual vertices to primal centroids δ_f.



normal component: $\hat{\mathbf{n}}_e \cdot \nabla(\cdot)$ (3)

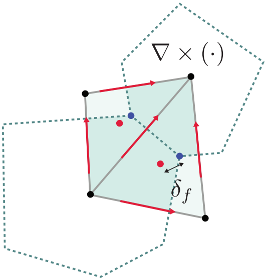
$$\hat{\mathbf{n}}_e \cdot \nabla\Phi \simeq l_e^{-1} (\Phi_2 - \Phi_1) \quad (4)$$

Only 2nd-order accurate if $\gamma_f = 0$.

Source of Discretisation Error

'Defect' in primal-dual staggering leads to errors in $\nabla \cdot (\cdot)$, $\nabla(\cdot)$, $\nabla \times (\cdot)$:

- Offsets from edge centroids to primal-dual bisectors δ_e, γ_e.
- Offsets from primal vertices to dual centroids γ_f.
- Offsets from dual vertices to primal centroids δ_f.



$$|\tau_k|\,\bar{\xi}_k = \int_{\tau_k} \nabla \times \mathbf{u}\ \mathrm{d}A = \oint_{\partial\tau_k} (\mathbf{u} \cdot \hat{\mathbf{t}})\ \mathrm{d}s \tag{5}$$

$$\simeq \sum_{e=1}^{3} \int_e (\mathbf{u} \cdot \hat{\mathbf{t}})_e\ \mathrm{d}l\,. \tag{6}$$

Only 2nd-order accurate if $\delta_f = 0$.

Source of Discretisation Error

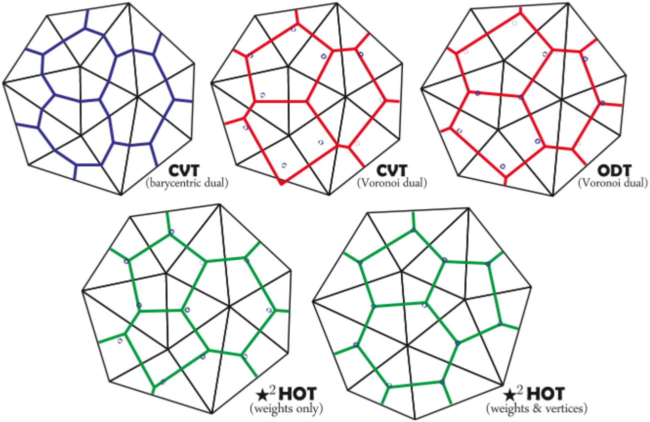
Unstructured mimetic discretisations require 'near perfect' primal-dual meshes to achieve ≥ 1-order accuracy:

$$\delta_e \to 0,\ \gamma_e \to 0,\ \gamma_f \to 0,\ \delta_f \to 0.$$

Not a property of Delaunay-Voronoi pairs (except when resolution is uniform)...

Using ‘Generalised’ Primal-Duals

Laguerre-Power Tessellations (‘**weighted**’ Delaunay-Voronoi pairs) allow better primal-dual meshes to be built — **HOT** (Hodge Optimised Tessellations):



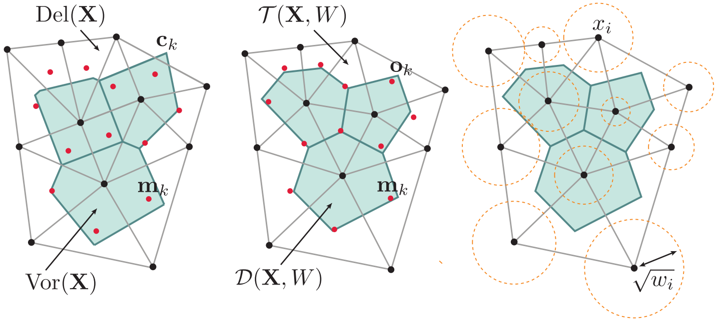
[1] de Goes, Memari, Mullen and Desbrun: *Weighted triangulations for geometry processing*, ACM TOG (2014).

[2] Mullen, Memari, de Goes, Desbrun: *HOT: Hodge-optimised triangulations*, ACM TOG (2011).

Using 'Generalised' Primal-Duals

Introduce distribution of vertex weights w_i to adjust shape of dual cells relative to primal to improve 'shape' + 'staggering' of primal-dual cells.

Power cells formed considering 'weighted distances': $\pi(\mathbf{x}, \mathbf{x}_i) = \mathrm{dist}(\mathbf{x}, \mathbf{x}_i)^2 - w_i$.



Enable generation of 'optimal' (orthogonal) primal-duals that are **more centroidal and well-centred** than Voronoi tessellations.

Using 'Generalised' Primal-Duals

The construction of **generalised dual grids** hinges on the expression for the position of **dual vertices**.

'Lift' problem onto a higher-dimensional space; $\mathbb{R}^d \rightarrow \mathbb{R}^{d+1}$:

$$\{x_i,\, y_i\} \rightarrow \{x_i,\, y_i,\, w_i\} \quad \text{with} \quad w_i \in \mathbb{R}^+ \tag{7}$$

Given a primal simplex, the associated (weighted) dual vertex is given by:

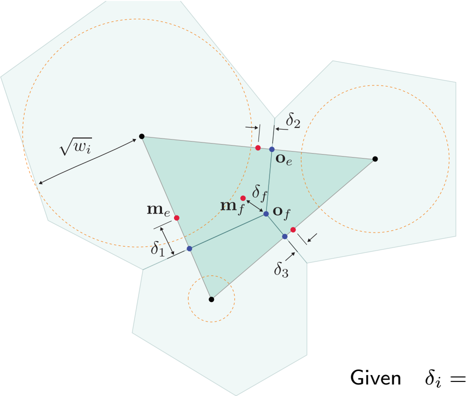
$$\begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \tag{8}$$

$$\underbrace{\frac{1}{2}\begin{bmatrix} (x_2 - x_1)^2 + (y_2 - y_1)^2 - (w_2 - w_1) \\ (x_3 - x_1)^2 + (y_3 - y_1)^2 - (w_3 - w_1) \end{bmatrix}}_{\text{'weighted 'RHS includes a new } \nabla(w) \text{ dependence}} \tag{9}$$

The idea is to choose weights such that the centres (i.e. 'dual' vertices) $\mathbf{c}_i = [x_c,\, y_c]$ are **positioned 'optimally'**.

Optimising 'Laguerre-Power' Meshes

Optimise weights to minimise 'defect' in staggering between primal and dual cells — **offsets associated with discretisation error in $\nabla \cdot (\cdot)$, $\nabla(\cdot)$, $\nabla \times (\cdot)$:**

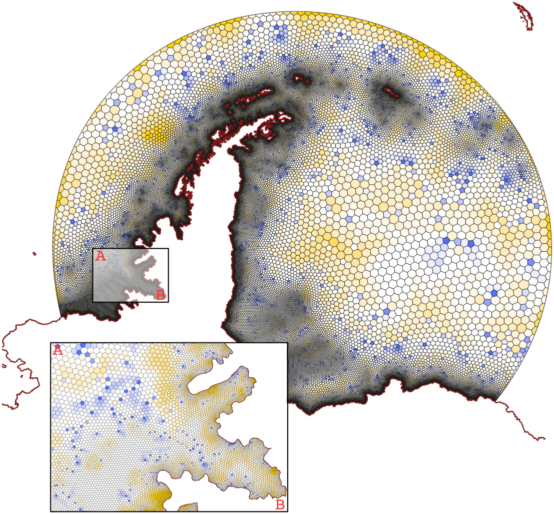


$$\text{Given} \quad \delta_i = \|\mathbf{o}_i - \mathbf{m}_i\|, \tag{10}$$

$$\mathcal{Q}_i^{\mathcal{D}}(\mathbf{X}, W) = \frac{1}{2}\underbrace{\left(1 - \left(\frac{\delta_f}{\bar{l}_f}\right)^2\right)}_{\text{'defect' at triangle}} + \frac{1}{2}\underbrace{\left(\frac{1}{3}\sum_{e=1}^{3} 1 - \left(\frac{\delta_e}{l_e}\right)^2\right)}_{\text{mean 'defect' at edges}}. \tag{11}$$

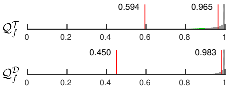
Optimising 'Laguerre-Power' Meshes

Primal-dual optimisation scheme improves shape and regularity of mesh — **centroidal**, **orthogonal**, **well-centred** characteristics.



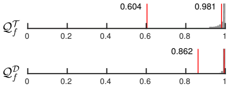
Not optimised:

$(|\tau|_{\text{uncentred}} = 6400)$



Optimised:

$(|\tau|_{\text{uncentred}} = 9)$



Blue: $w_i \ll 0$

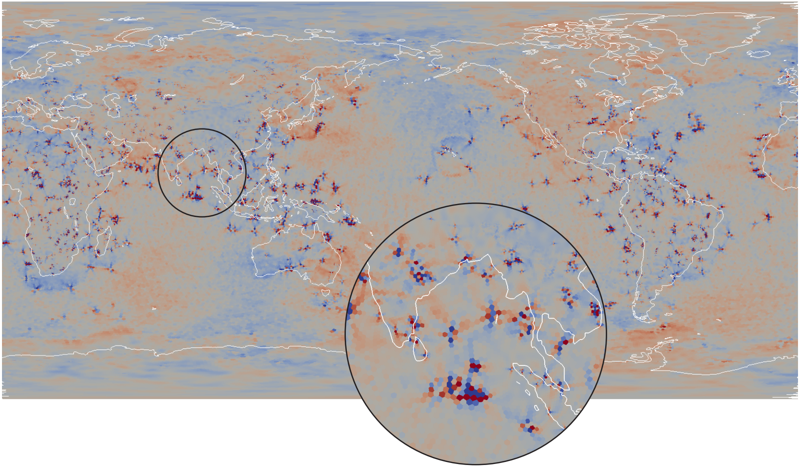
Orange: $w_i \gg 0$

Weights adjust to valence: < 6: -ve weights, > 6: $+$ve weights, $= 6$: no weights.

Optimising 'Laguerre-Power' Meshes

Primal-dual optimisation scheme resolves issues with convergence in L^∞.

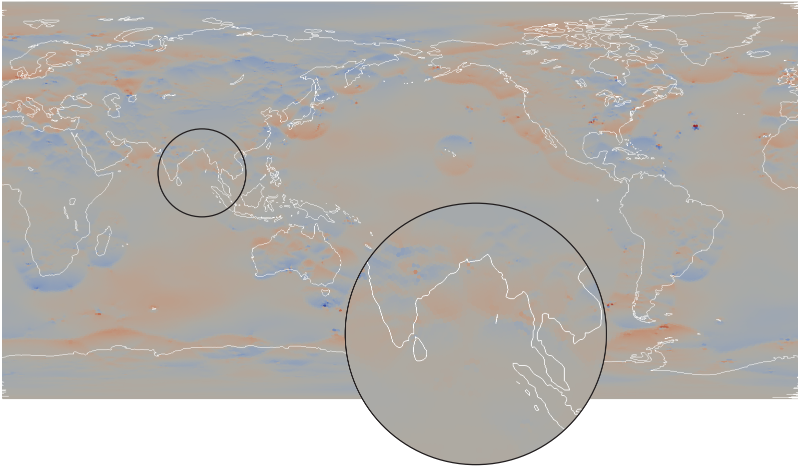
Voronoi-type mesh; error in fluid height:



Optimising ‘Laguerre-Power’ Meshes

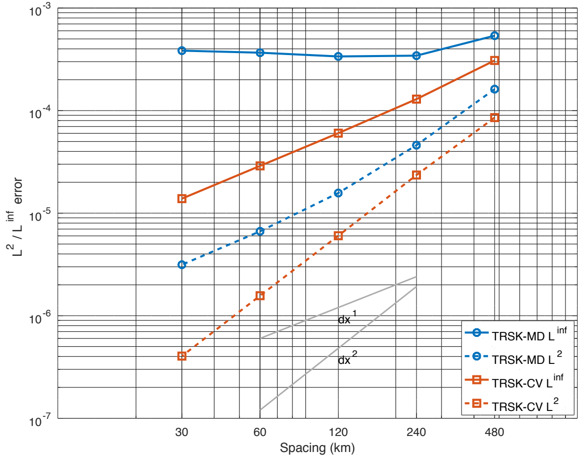
Primal-dual optimisation scheme resolves issues with convergence in L^∞.

Power-type mesh; error in fluid height:



Optimising 'Laguerre-Power' Meshes

Improves L^∞ convergence from 0th- to 1st-order, and L^2 convergence from 1st-order to 2nd-order.



Better resolve turbulent flows!

How to Optimise the Primal-Dual?

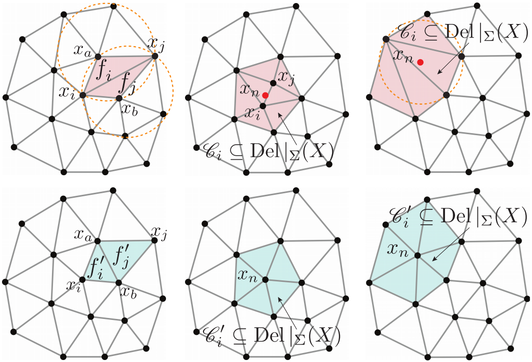
Find a mesh $\mathcal{T}(\mathbf{x})$ that minimises a quality/energy-metric $\mathcal{Q}(\mathbf{x}, \mathcal{T})$

$$\min \mathcal{Q}(\mathbf{x}, \mathcal{T}) \tag{12}$$

Solve by combining standard gradient descent to update positional DoF

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \alpha_i^n \, \eta_i \nabla_i Q(\mathbf{x}^k)\,. \tag{13}$$

with topological operators to enhance structure



How to Optimise the Primal-Dual?

Leads to a reliable 'worst-first' hill-climbing optimisation scheme:

function MeshOpt($\mathcal{T}(\mathbf{x})$)

  while (not optimised enough)

    Build worst-first ordering for positional DoF.

    Apply steepest-descent updates for $\mathbf{x}$:

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta_i^m \cdot \frac{\partial}{\partial \mathbf{x}_i}\left[\mathcal{Q}_j(\mathbf{x}, \mathcal{T})\right]. \qquad (14)$$

    Accept $\mathbf{x}_i^{n+1}$ iff $Q(\mathbf{x}^k)$ improving.

    Collapse/split cells to improve quality.

    Update topology of $\mathcal{T}$ to recover orthogonality.

  end while

Typically successful in practice, **but slow to converge** (wrt. iterations and runtime).

**Engwirda (2018): Generalised primal-dual grids for unstructured co-volume schemes.

Learn from Machine Learning

Machine learning community uses 'momentum' optimisation methods to train neural networks (i.e. large unstructured mesh-like systems) — **are such approaches useful for meshing?**

'Momentum' gradient descent (MGD[1]): add a 'gradient-buffer' $\mathbf{g}^k$ maintaining an average of previous steepest descent directions

$$\mathbf{g}_i^k = \beta\, \mathbf{g}_i^{k-1} + (1-\beta)\, \eta_i \nabla_i Q(\mathbf{x}^k)\,, \tag{15}$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \alpha_i^n\, \mathbf{g}_i^k\,, \tag{16}$$

where $\beta \in [0, \frac{1}{2}]$ is a 'momentum' bias that helps scheme 'burst-through' shallow local minima.

(Experimentally) $\beta = \frac{1}{3}$ **improves convergence of mesh optimisation scheme by around factor of 1.5**.

[1] Nesterov (1983): A method for solving the convex programming problem with convergence rate $O(1/k^2)$

Learn from Machine Learning

Machine learning community uses 'momentum' optimisation methods to train neural networks (i.e. large unstructured mesh-like systems) — **are such approaches useful here?**

'Quasi-hyperbolic' momentum (QHM[1]): add 'linear-discount' ζ to gradient buffer

$$\mathbf{g}_i^k = \underbrace{\beta\,\mathbf{g}_i^{k-1} + (1-\beta)\,\eta_i \nabla_i Q(\mathbf{x}^k)}_{\text{'momentum' recurrence}}\,, \tag{17}$$

$$\zeta_i^n = \alpha_i^n\,\zeta\,, \tag{18}$$

$$\mathbf{g}_i^* = \underbrace{\zeta_i^n\,\mathbf{g}_i^k + (1-\zeta_i^n)\,\eta_i \nabla_i Q(\mathbf{x}^k)}_{\text{linear 'discount'}}\,, \tag{19}$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \alpha_i^n\,\mathbf{g}_i^*\,. \tag{20}$$

(Experimentally) use of a larger $\beta \leftarrow 0.495$, $\zeta \leftarrow 0.825$ in QHM **further improves convergence of mesh optimisation scheme by another factor of 2**.

QHM developed/used by Facebook for training of large ML models.

[1] **Ma & Yarats (2018): Quasi-hyperbolic momentum and Adam for deep learning.

Los Alamos National Laboratory

Learn from Machine Learning

Improved Quasi-Hyperbolic Momentum (QHM) optimisation:

```
function MeshOpt(𝒯(x))
  while (not optimised enough)
    Build worst-first ordering for positional DoF.
    Apply QHM updates for x:
```

$$\mathbf{g}_i^k = \beta\,\mathbf{g}_i^{k-1} + (1-\beta)\,\eta_i \nabla_i Q(\mathbf{x}^k) \tag{21}$$

$$\mathbf{g}_i^* = \zeta_i^n\,\mathbf{g}_i^k + (1-\zeta_i^n)\,\eta_i \nabla_i Q(\mathbf{x}^k) \tag{22}$$

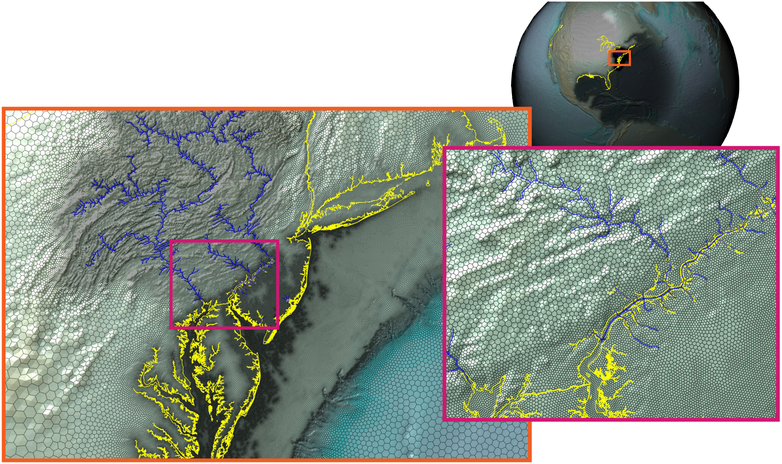
$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta_i^m \cdot g_i^*\,. \tag{23}$$

```
    Accept x_i^{n+1} iff Q(x^k) improving.
    Collapse/split cells to improve quality.
    Update topology of 𝒯 to recover orthogonality.
  end while
```

Quite simple modification to standard gradient descent that **improves convergence by factor of 2–3**. Only $O(1)$ extra work, $O(1)$ extra space per vertex.

Results: Primal-dual Meshes for Earth System Models

Build very high-quality primal-dual meshes — **centroidal**, **well-centred**, **orthogonal**, **smooth-grading**, **boundary conforming**...



**Embedded mid-Atlantic coastal-zone: ICoM project, Engwirda et al, 2020.

Results: Primal-dual Meshes for Earth System Models

Consider problems of varying complexity — all include strong variation in resolution + complex geometrical boundaries:

GREAT LAKES: 100K CELLS.
COASTAL OCEAN: 400K CELLS.
GLOBAL CLIMATE: 2.5M CELLS.
HURRICANE INUNDATION: 25M CELLS.

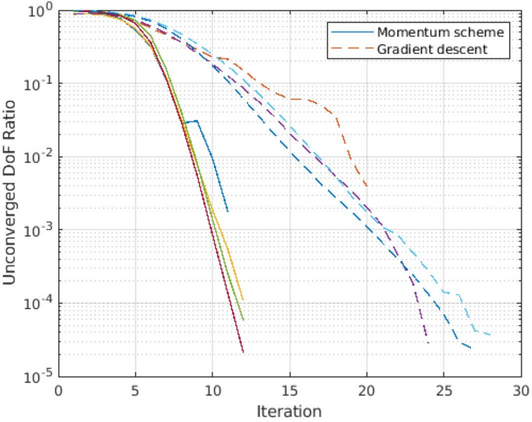
JIGSAW[1] library to build initial primal meshes (Frontal-Delaunay); optimise with standard gradient descent vs momentum scheme (QHM).

Optimisation terminates iff:

- All cells well-centred (non-obtuse).
- No new topological updates.
- DoFs approximately converged: $\Delta Q_i \leq 1 \times 10^{-5}$.

[1] www.github.com/dengwirda/jigsaw

Performance of standard gradient descent (dashed lines) vs momentum scheme (QHM; solid lines)



QHM improves convergence rate by ≥ 2 on all cases, irrespective of problem complexity.

Summary & Outlook

Use Laguerre-Power meshes instead of Delaunay-Voronoi pairs!

- **Meshing and solving are different sides of the same coin, not parts of different currencies**.
- Improved geometry of primal-dual staggering reduces discretisation error in mimetic scheme.
- Extend HOT mesh paradigm to optimise Laguerre-Power primal-dual to maximise accuracy of discretisation.
- Leverage Quasi-Hyperbolic Momentum scheme from ML community to build optimal Laguerre-Power pairs efficiently.

Thanks!