Towards a wavelet-based dynamically adaptive climate model

Nicholas Kevlahan

Department of Mathematics and Statistics   visiting Laboratoire Jean Kuntzmann (Grenoble)

Rencontres mathématiques de Rouen – juin 2019

Collaborators

- Thomas Dubos
  Laboratoire de Météorologie Dynamique
  École Polytechnique, France
- Matthias Aechtner
  Former PhD student (contributed to shallow water code)

Geophysical flows

(Credit: NASA Apollo 17 mission)

- Atmosphere and ocean dynamics
- Long distance wave propagation *(tsunamis)*
- Numerical weather prediction
- Climate modelling

Geophysical flows: need for new numerical methods

- Climate and weather models under-resolved
- Need discrete conservation of mass and potential vorticity and other mimetic properties
- Computational grid should adapt to achieve specified error tolerance, or resolve features of interest
- Spherical grid should avoid singularities *(near poles)*

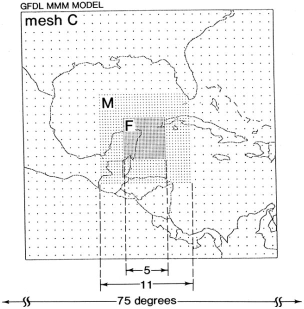
Adaptivity in numerical models

- Adaptivity changes resolution to guarantee uniform error, or focus on regions of interest
- Optimal use of computational resources

Adaptivity in numerical models

- Adaptivity changes resolution to guarantee uniform error, or focus on regions of interest
- Optimal use of computational resources

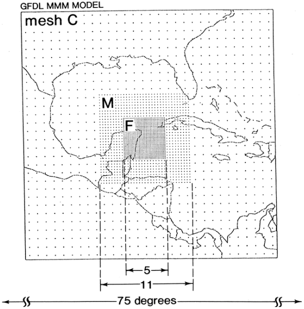
fixed uniform nested grids. . .



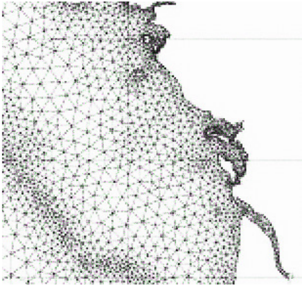
Adaptivity in numerical models

- Adaptivity changes resolution to guarantee uniform error, or focus on regions of interest
- Optimal use of computational resources

fixed uniform nested grids...



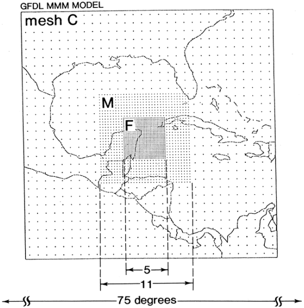
unstructured grids...



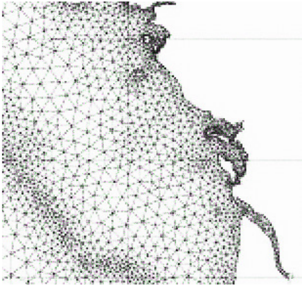
Adaptivity in numerical models

- Adaptivity changes resolution to guarantee uniform error, or focus on regions of interest
- Optimal use of computational resources

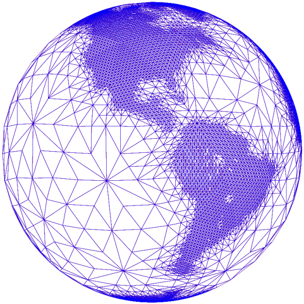
fixed uniform nested grids...



unstructured grids...



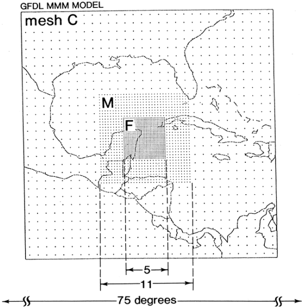
structured grids...



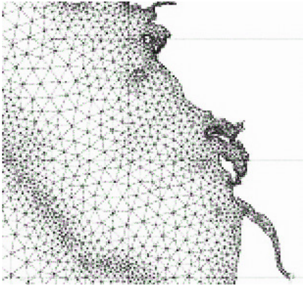
Adaptivity in numerical models

- Adaptivity changes resolution to guarantee uniform error, or focus on regions of interest
- Optimal use of computational resources

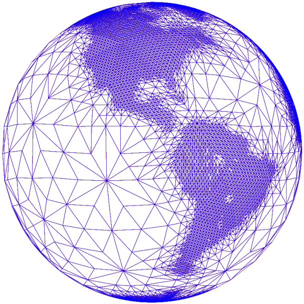
fixed uniform nested grids. . .



unstructured grids. . .



structured grids. . .



. . . or **Wavelets!**

Development of adaptive 3D-hydrostatic dynamical core

(Credit: NASA Apollo 17 mission)

1. Shallow water equations on the plane using TRiSK discretization

Development of adaptive 3D-hydrostatic dynamical core

(Credit: NASA Apollo 17 mission)

2. Shallow water equations on the sphere using TRiSK discretization *(Icosahedral C-grid)*

Development of adaptive 3D-hydrostatic dynamical core

(Credit: NASA Apollo 17 mission)

3. Volume penalization for coastline boundary conditions in ocean models

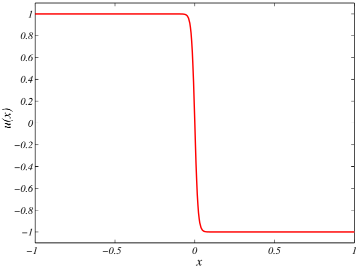
Development of adaptive 3D-hydrostatic dynamical core

(Credit: NASA Apollo 17 mission)

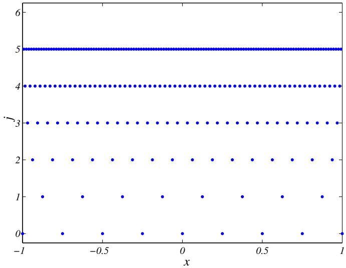
4. 3D hydrostatic extension using DYNAMICO approach, horizontal adaptivity

Wavelet adaptivity

$$u(x) = \sum_{k \in \mathcal{K}^0} u_k^0 \phi_k^0(x) + \sum_{j=0}^{+\infty} \sum_{k \in \mathcal{L}^j} \tilde{u}_k^j \psi_k^j(x)$$



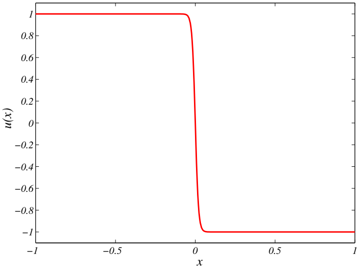
Function $u(x)$



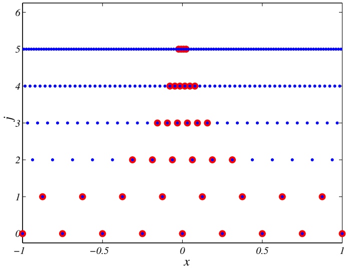
Wavelet locations x_k^j

Wavelet adaptivity

$$u_{\geq}(x) = \sum_{k\in\mathcal{K}^0} u_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{\substack{k\in\mathcal{L}^j \\ |\tilde{u}_k^j|\geq\varepsilon}} \tilde{u}_k^j \psi_k^j(x)$$



Function $u(x)$

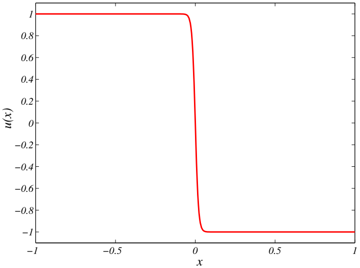


Wavelet locations x_k^j

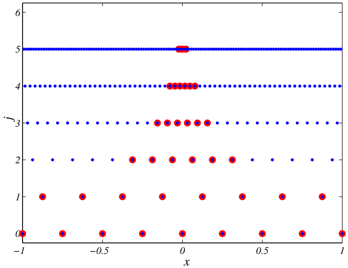
$\varepsilon = 10^{-3}$

Wavelet adaptivity

$$\begin{aligned}
||u(x) - u_{\geq}(x)||_\infty &= O(\varepsilon) \\
\mathcal{N} &= O(\varepsilon^{-1/2N}) \\
||u(x) - u_{\geq}(x)||_\infty &= O(\mathcal{N}^{-2N})
\end{aligned}$$



Function $u(x)$



Wavelet locations x_k^j

$\varepsilon = 10^{-3}$

Dynamically adaptive wavelet method for PDEs

$$F\left(\frac{\partial u}{\partial t}, \frac{\partial^n u}{\partial x^n}, x, t\right) = 0$$

Dynamically adaptive wavelet method for PDEs

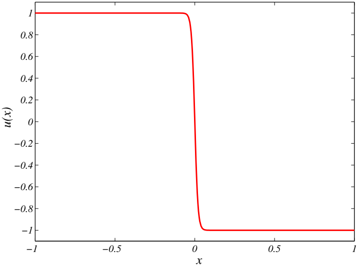
$$F\left(\frac{\partial u}{\partial t}, \frac{\partial^n u}{\partial x^n}, x, t\right) = 0$$

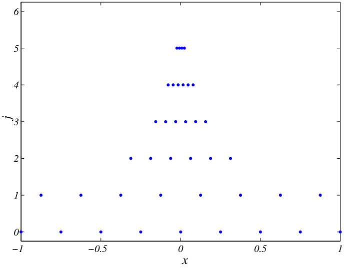
$$\tilde{u}_k^j \Longrightarrow u_k^j \Longrightarrow \frac{\partial^n u}{\partial x^n}(x_k^j), \quad O(\mathcal{N}) \text{ complexity}$$

Dynamically adaptive wavelet method for PDEs

$$F\left(\frac{\partial u}{\partial t}, \frac{\partial^n u}{\partial x^n}, x, t\right) = 0$$

$$\tilde{u}_k^j \Longrightarrow u_k^j \Longrightarrow \frac{\partial^n u}{\partial x^n}(x_k^j), \quad O(\mathcal{N}) \text{ complexity}$$

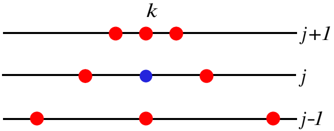




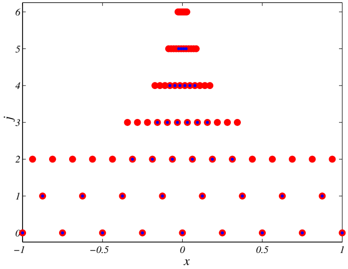
Dynamically adaptive wavelet method for PDEs

$$F\left(\frac{\partial u}{\partial t}, \frac{\partial^n u}{\partial x^n}, x, t\right) = 0$$

$$\tilde{u}_k^j \Longrightarrow u_k^j \Longrightarrow \frac{\partial^n u}{\partial x^n}(x_k^j), \quad O(\mathcal{N}) \text{ complexity}$$



Adjacent zone



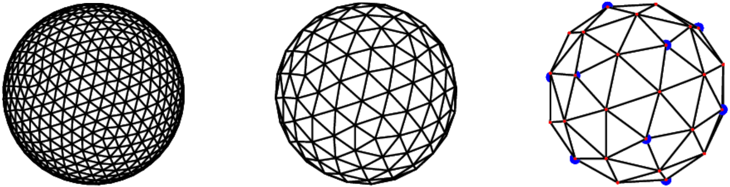
Dynamically adaptive wavelet method for PDEs

(Burgers equation)

Dynamically adaptive wavelet method for PDEs

(Propagating front)

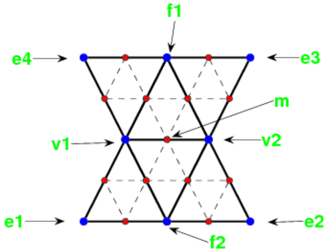
Discrete wavelet transform on the sphere



vertex values	$\Rightarrow$	**vertex values**	$\Rightarrow$	**vertex values**	***restriction***
$\Downarrow$		$\Downarrow$			***high-pass filter***
interpolation errors		**interpolation errors**			***wavelets***

interpolation errors		**interpolation errors**			***wavelets***
$\Downarrow$		$\Downarrow$			***reconstruction***
vertex values	$\Leftarrow$	**vertex values**	$\Leftarrow$	**vertex values**	***prolongation***

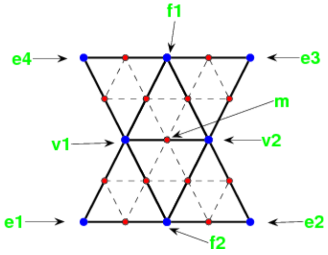
Wavelet transform on sphere *(Schröder & Sweldens 1995)*



Predict values at fine vertices m by *interpolation* using stencil of coarse vertices k

Lift values at vertices k to conserve properties (*mean*) in smooth approximation

Wavelet transform on sphere *(Schröder & Sweldens 1995)*



Analysis: $j+1 \to j$

$$\text{(high pass)} \quad \tilde{u}^j_m = u^{j+1}_m - \sum_{k\in\mathcal{K}_m} \tilde{s}^j_{k,m} u^{j+1}_k$$

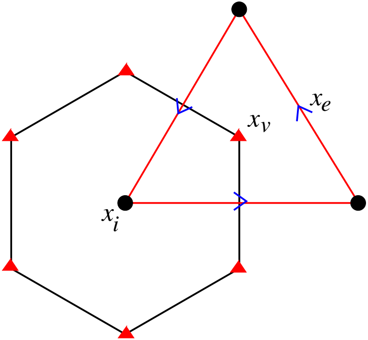
$$\text{(restrict)} \quad u^j_k = u^{j+1}_k + \sum_{m\in\mathcal{M}_k} s^j_{k,m} \tilde{u}^j_m$$

Synthesis: $j \to j+1$

$$\text{(prolong)} \quad u^{j+1}_k = u^j_k - \sum_{m\in\mathcal{M}_k} s^j_{k,m} \tilde{u}^j_m$$

$$\text{(reconstruct)} \quad u^{j+1}_m = \tilde{u}^j_m + \sum_{k\in\mathcal{K}_m} \tilde{s}^j_{k,m} u^{j+1}_k$$

2D: TRiSK scheme *(Thuburn et al. 2010)*



Staggered dual grids for mass and vorticity
(Velocity at cell edges)

Discrete shallow water equations

$$
\begin{aligned}
\frac{\partial h_i}{\partial t} &= -[\mathrm{div}(F_e)]_i \\
\frac{\partial \mathbf{u}_e}{\partial t} &= F_e^{\perp} q_e - [\mathrm{grad}(B_i)]_e
\end{aligned}
$$

- $F_e = h_e u_e$ is thickness flux
- $F_e^{\perp}$ is perpendicular to F_e

Scale commutation properties of differential operators

$$\begin{tikzcd}
B_i^0,\ F_e^0,\ u_e^0 \arrow[r, "\mathrm{grad}^0,\ \mathrm{div}^0,\ \mathrm{curl}^0"] & \mathrm{grad}\ B_i^0,\ \mathrm{div}\ F_e^0,\ \mathrm{curl}\ u_e^0 \\
B_i^1,\ F_e^1,\ u_e^1 \arrow[u, "R"'] \arrow[r, "\mathrm{grad}^1,\ \mathrm{div}^1,\ \mathrm{curl}^1"] & \mathrm{grad}\ B_i^1,\ \mathrm{div}\ F_e^1,\ \mathrm{curl}\ u_e^1 \arrow[u, "R"']
\end{tikzcd}$$

Commutation diagram

Scale commutation properties of differential operators

Commutation relations

$$
\begin{aligned}
R_h^j \circ \mathsf{div}^{j+1} &= \mathsf{div}^j \circ R_F^j && \textit{conserve mass} \\
\mathsf{curl}^j \circ R_{\mathbf{u}}^j &= R_\zeta^j \circ \mathsf{curl}^{j+1} && \textit{conserve circulation} \\
\mathsf{grad}^j \circ R_B^j &= R_{\mathbf{u}}^j \circ \mathsf{grad}^{j+1} && \textit{no spurious vorticity}
\end{aligned}
$$

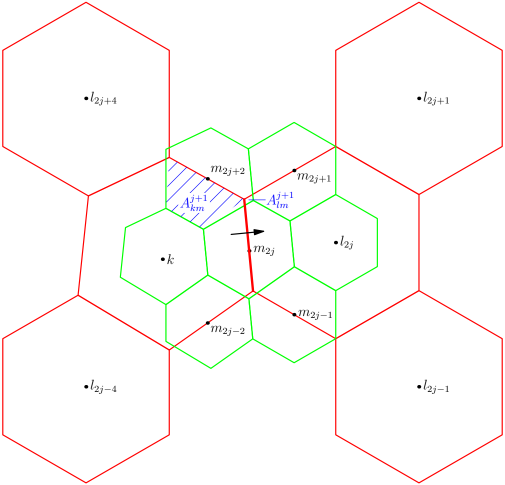
Scale commutation properties of differential operators

Commutation relations

$$
\begin{aligned}
R_h^j \circ \mathsf{div}^{j+1} &= \mathsf{div}^j \circ R_F^j && \textit{conserve mass} \\
\mathsf{curl}^j \circ R_{\mathbf{u}}^j &= R_\zeta^j \circ \mathsf{curl}^{j+1} && \textit{conserve circulation} \\
\mathsf{grad}^j \circ R_B^j &= R_{\mathbf{u}}^j \circ \mathsf{grad}^{j+1} && \textit{no spurious vorticity}
\end{aligned}
$$

Adaptive overlay on any flux-based method

Extension to icosahedral C-grid on sphere: flux restriction



Fine and coarse scale cells to calculate flux restriction through coarse edge indicated by arrow. A_{km}^{j+1} and A_{lm}^{j+1} are partial areas.

3D: DYNAMICO equations *(Dubos et al 2015)*

- Multilayer hydrostatic shallow water equations *(compressible/incompressible)*

3D: DYNAMICO equations *(Dubos et al 2015)*

- Multilayer hydrostatic shallow water equations *(compressible/incompressible)*
- Derive equations of motion from discrete Hamiltonians

3D: DYNAMICO equations *(Dubos et al 2015)*

- Multilayer hydrostatic shallow water equations *(compressible/incompressible)*
- Derive equations of motion from discrete Hamiltonians
- TRiSK for horizontal discretization

3D: DYNAMICO equations *(Dubos et al 2015)*

- Multilayer hydrostatic shallow water equations *(compressible/incompressible)*
- Derive equations of motion from discrete Hamiltonians
- TRiSK for horizontal discretization
- Conserves energy *(or enstrophy)* and mass

3D: DYNAMICO equations *(Dubos et al 2015)*

- Multilayer hydrostatic shallow water equations *(compressible/incompressible)*
- Derive equations of motion from discrete Hamiltonians
- TRiSK for horizontal discretization
- Conserves energy *(or enstrophy)* and mass

$$\frac{\partial m_{ik}}{\partial t} + \delta_i U_k = 0, \qquad \frac{\partial \Theta_{ik}}{\partial t} + \delta_i \left(\theta^*_{ek} U_k\right) = 0$$

$$\frac{\partial v_{ek}}{\partial t} + \delta_e B_k + \theta^*_{ek} \delta_e \pi_k + \left(q_k U_k\right)^{\perp}_e = 0$$

Multiscale icosahedral grid resolution

J	N	$\overline{\Delta x}$ $[deg]$	$\overline{\Delta x}$ $[km]$
4	2 562	4	480
5	10 242	2	240
6	40 962	1	120
7	163 842	$1/2$	56
8	655 362	$1/4$	28
9	2 621 442	$1/8$	14
10	10 485 762	$1/16$	7

Multiscale icosahedral grid resolution

J	N	$\overline{\Delta x}$ $[deg]$	$\overline{\Delta x}$ $[km]$
4	2 562	4	480
5	10 242	2	240
6	40 962	1	120
7	163 842	1/2	56
8	655 362	1/4	28
9	2 621 442	1/8	14
10	10 485 762	1/16	7

- Optimize coarsest grid, e.g. $J = 5$ *(Xu 2006; Heikes & Randall 1995)*

Multiscale icosahedral grid resolution

J	N	$\overline{\Delta x}$ $[deg]$	$\overline{\Delta x}$ $[km]$
4	2 562	4	480
5	10 242	2	240
6	40 962	1	120
7	163 842	$1/2$	56
8	655 362	$1/4$	28
9	2 621 442	$1/8$	14
10	10 485 762	$1/16$	7

- Optimize coarsest grid, e.g. $J = 5$ *(Xu 2006; Heikes & Randall 1995)*
- Finer grids by recursive edge-bisection, e.g. $j = 6, 7, 8, \ldots$

Multiscale icosahedral grid resolution

J	N	$\overline{\Delta x}$ $[deg]$	$\overline{\Delta x}$ $[km]$
4	2 562	4	480
5	10 242	2	240
6	40 962	1	120
7	163 842	1/2	56
8	655 362	1/4	28
9	2 621 442	1/8	14
10	10 485 762	1/16	7

- Optimize coarsest grid, e.g. $J = 5$ *(Xu 2006; Heikes & Randall 1995)*
- Finer grids by recursive edge-bisection, e.g. $j = 6, 7, 8, \dots$
- Local adaptive grid scale j controlled by error tolerance ε

Arbitrary Lagrangian Eulerian vertical coordinates (ALE)

- Initialize hybrid $\sigma - P$ pressure-based grid

Arbitrary Lagrangian Eulerian vertical coordinates (ALE)

- Initialize hybrid $\sigma - P$ pressure-based grid
- Remap $m_{ik}, \theta_{ik}, v_{ek}$ conservatively *(either every Δt or every $10\Delta t$)*

Arbitrary Lagrangian Eulerian vertical coordinates (ALE)

- Initialize hybrid $\sigma - P$ pressure-based grid
- Remap $m_{ik}, \theta_{ik}, v_{ek}$ conservatively *(either every Δt or every $10\Delta t$)*
- Target grid is initial grid
  (could be optimized at each remapping for $r-$adaptivity)

Arbitrary Lagrangian Eulerian vertical coordinates (ALE)

- Initialize hybrid $\sigma - P$ pressure-based grid
- Remap $m_{ik}, \theta_{ik}, v_{ek}$ conservatively *(either every Δt or every $10\Delta t$)*
- Target grid is initial grid
  (could be optimized at each remapping for $r-$adaptivity)
- Tried various piecewise remapping schemes: continuous, linear, parabolic, quartic *(Shchepetkin 2001; Engwirda & Kelly 2016)*

Arbitrary Lagrangian Eulerian vertical coordinates (ALE)

- Initialize hybrid $\sigma - P$ pressure-based grid
- Remap $m_{ik}, \theta_{ik}, v_{ek}$ conservatively *(either every Δt or every $10\Delta t$)*
- Target grid is initial grid
  (could be optimized at each remapping for $r-$adaptivity)
- Tried various piecewise remapping schemes: continuous, linear, parabolic, quartic *(Shchepetkin 2001; Engwirda & Kelly 2016)*
- Limiter: none, monotone, WENO

Arbitrary Lagrangian Eulerian vertical coordinates (ALE)

- Initialize hybrid $\sigma - P$ pressure-based grid
- Remap $m_{ik}, \theta_{ik}, v_{ek}$ conservatively *(either every Δt or every $10\Delta t$)*
- Target grid is initial grid
  (could be optimized at each remapping for $r-$adaptivity)
- Tried various piecewise remapping schemes: continuous, linear, parabolic, quartic *(Shchepetkin 2001; Engwirda & Kelly 2016)*
- Limiter: none, monotone, WENO

- *At least piecewise linear required for Held–Suarez test case*
- *Piecewise constant sufficient for mountain induced Rossby wave and baroclinic instability*

Grid adaptation strategy at each time step

- In each vertical layer remove nodes/edges with normalized wavelet coefficients $< \varepsilon$ *(estimate norms or calculate dynamically)*

Grid adaptation strategy at each time step

- In each vertical layer remove nodes/edges with normalized wavelet coefficients $< \varepsilon$ *(estimate norms or calculate dynamically)*
- Add nearest neighbours in space and scale to allow grid refinement

Grid adaptation strategy at each time step

- In each vertical layer remove nodes/edges with normalized wavelet coefficients $< \varepsilon$ *(estimate norms or calculate dynamically)*
- Add nearest neighbours in space and scale to allow grid refinement
- Add additional nodes/edges required to construct wavelets and TRiSK operators

Grid adaptation strategy at each time step

- In each vertical layer remove nodes/edges with normalized wavelet coefficients $< \varepsilon$ *(estimate norms or calculate dynamically)*
- Add nearest neighbours in space and scale to allow grid refinement
- Add additional nodes/edges required to construct wavelets and TRiSK operators
- Adapted horizontal grid is union of adapted grids in all vertical layers

Grid adaptation strategy at each time step

- In each vertical layer remove nodes/edges with normalized wavelet coefficients $< \varepsilon$ *(estimate norms or calculate dynamically)*
- Add nearest neighbours in space and scale to allow grid refinement
- Add additional nodes/edges required to construct wavelets and TRiSK operators
- Adapted horizontal grid is union of adapted grids in all vertical layers
- Equivalent to propagating tendency error for constant coefficient linear equations

Grid adaptation strategy at each time step

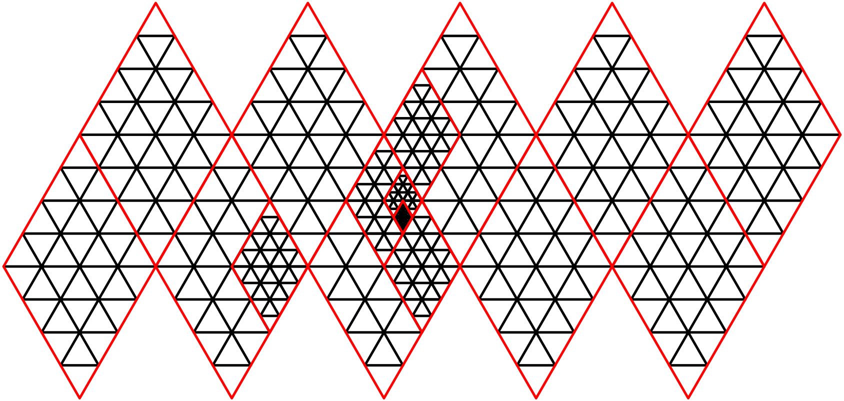
- In each vertical layer remove nodes/edges with normalized wavelet coefficients $< \varepsilon$ *(estimate norms or calculate dynamically)*
- Add nearest neighbours in space and scale to allow grid refinement
- Add additional nodes/edges required to construct wavelets and TRiSK operators
- Adapted horizontal grid is union of adapted grids in all vertical layers
- Equivalent to propagating tendency error for constant coefficient linear equations
- Simple, good results in practice

Grid adaptation strategy at each time step

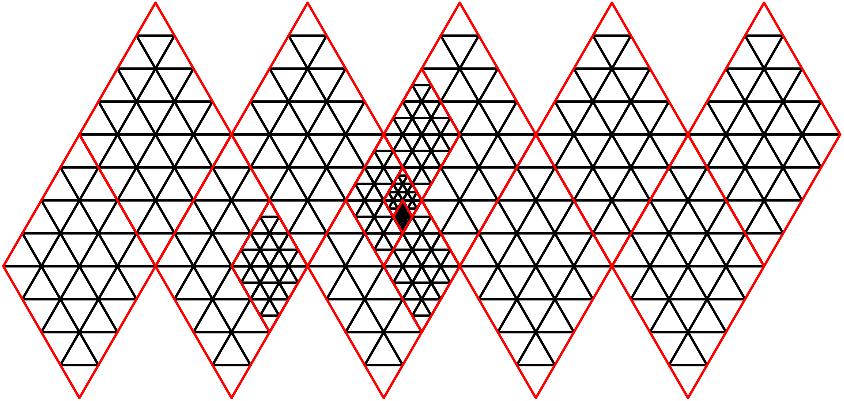
- In each vertical layer remove nodes/edges with normalized wavelet coefficients $< \varepsilon$ *(estimate norms or calculate dynamically)*
- Add nearest neighbours in space and scale to allow grid refinement
- Add additional nodes/edges required to construct wavelets and TRiSK operators
- Adapted horizontal grid is union of adapted grids in all vertical layers
- Equivalent to propagating tendency error for constant coefficient linear equations
- Simple, good results in practice

Generates horizontally adapted grid

Hybrid data structure: irregular tree data structure with regular patches

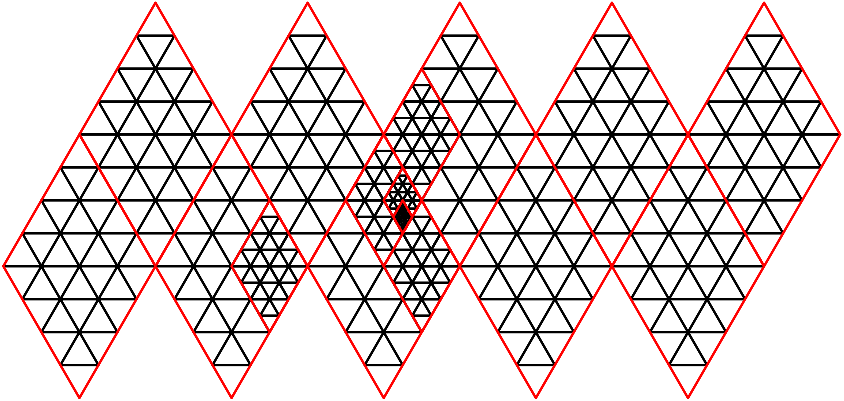


Hybrid data structure: irregular tree data structure with regular patches



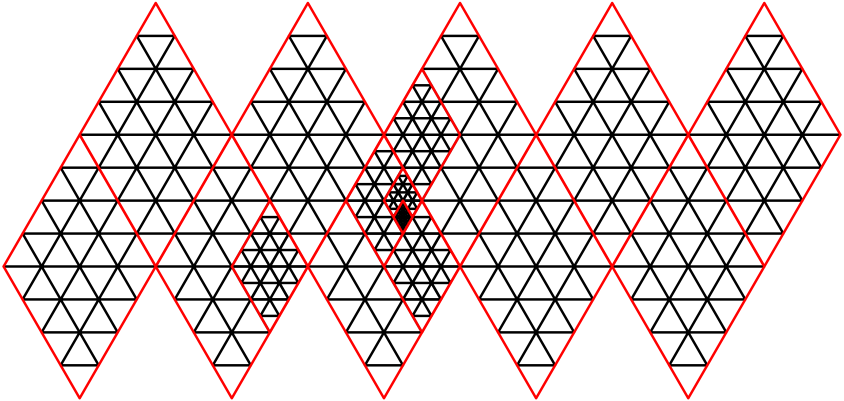
- Icosahedron divided into 10 regular lozenge domains.

Hybrid data structure: irregular tree data structure with regular patches



- Icosahedron divided into 10 regular lozenge domains.
- Domains refined adaptively into sub-domains.

Hybrid data structure: irregular tree data structure with regular patches

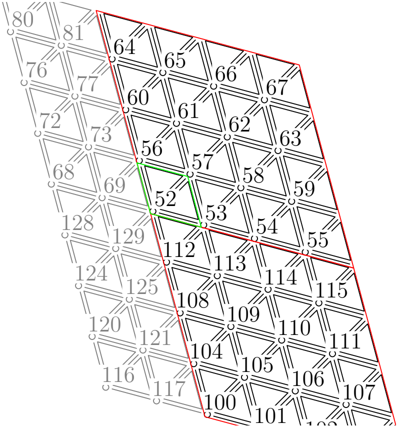


- Icosahedron divided into 10 regular lozenge domains.
- Domains refined adaptively into sub-domains.
- Lowest level locally is regular 4×4 patch.

Parallelization

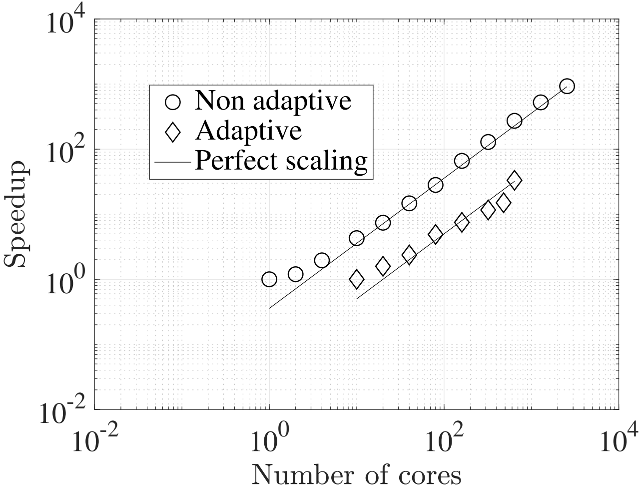
- Parallelization using `mpi`
- Sub-domains distributed to different cores
- Ghost points added, values communicated as necessary for operators
- Hybrid tree–patch data structure
- Communications at each trend computation and at each grid adaptation step
- Where possible communication is non-blocking
- Simple load balancing at each a check point save

Computational grid with ghost cells

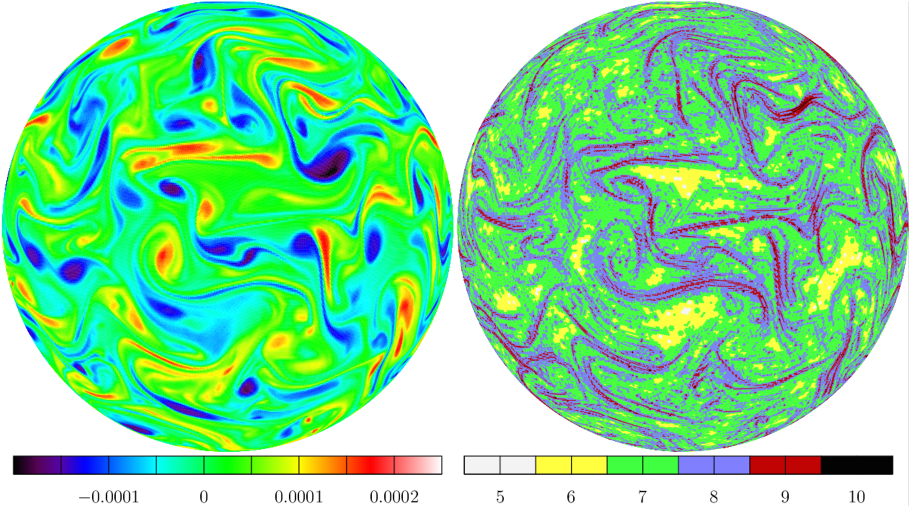


4×4 patch is regular grid of elements. Element is one node, two triangles and three edges. Ghost points added at edges of sub-domain.

Strong parallel scaling



2D shallow water turbulence ($1/16^{\circ}$ max resolution)



Relative vorticity

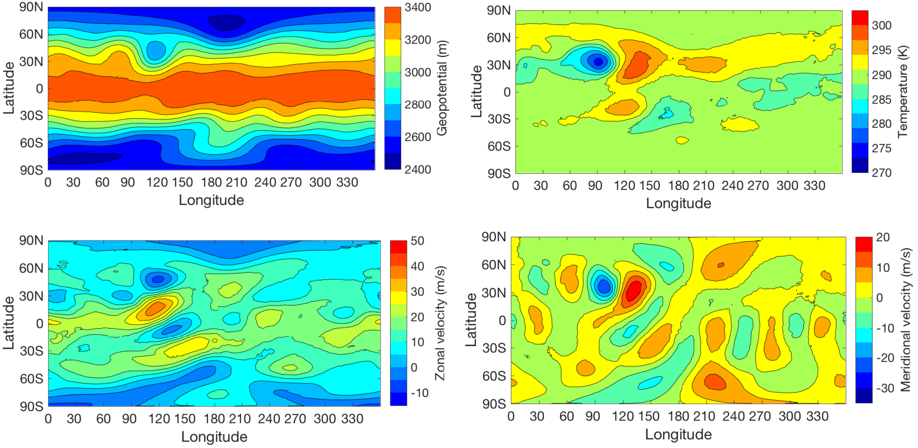
Level

Mountain-induced Rossby wave *(DCMIP 2008 case 5)*

26 vertical levels, results at 700 hPa, no diffusion

Mountain-induced Rossby wave at 25 days *(DCMIP 2008 case 5)*

26 vertical levels, results at 700 hPa, no diffusion



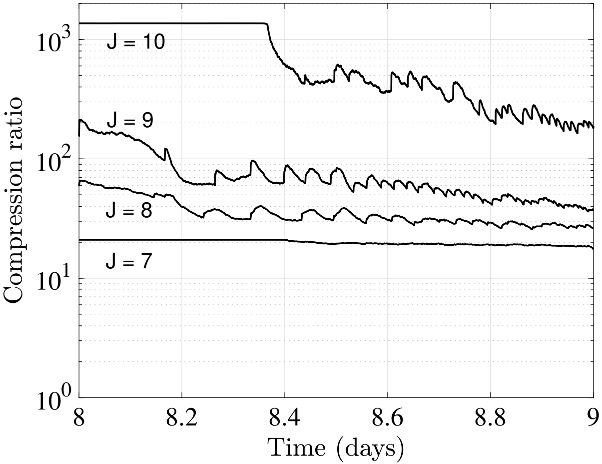
Baroclinic instability of jet stream *(DCMIP 2012 case 4)*

26 vertical levels, results at 867 hPa, hyperdiffusion

$\nu_{scalar} = 5.3 \times 10^{12}, \nu_{div} = 1.0 \times 10^{14}, \nu_{curl} = 1.1 \times 10^{13}$

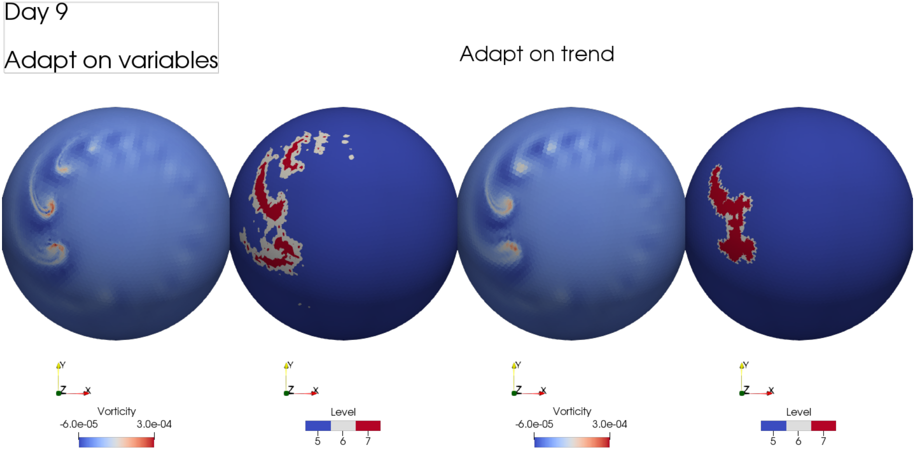
Baroclinic instability of jet stream *(DCMIP 2012 case 4)*

Grid compression as J increases *($J_{min} = 5$, no diffusion, adapt on trend)*



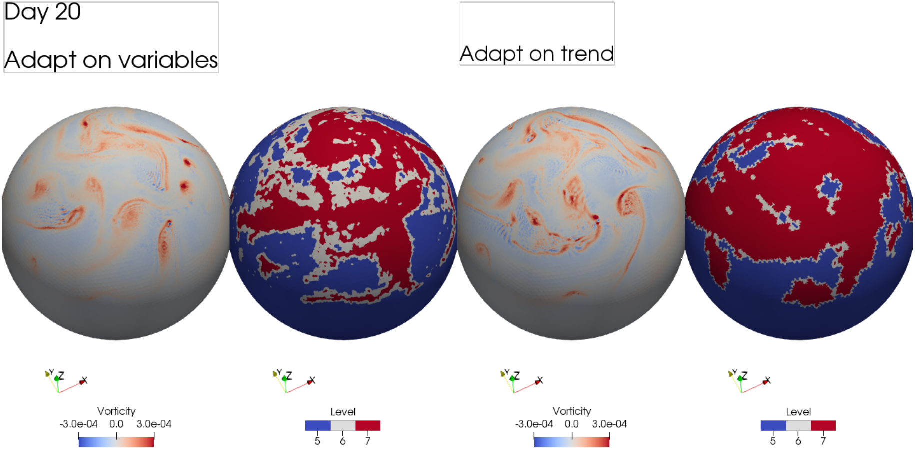
Baroclinic instability of jet stream *(DCMIP 2012 case 4)*

Compare adaptivity *(No diffusion, equal number of active grid points)*



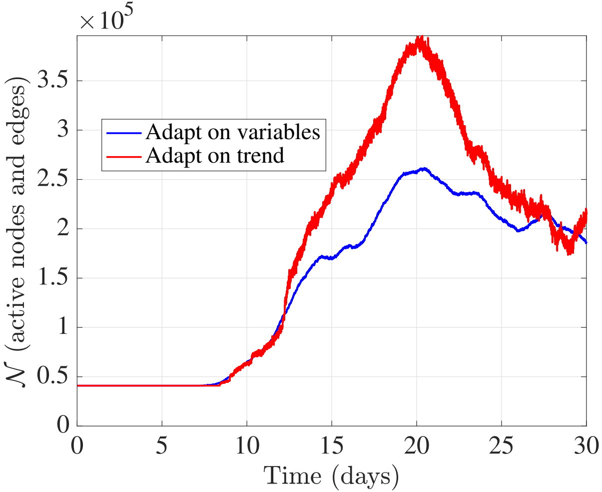
Baroclinic instability of jet stream *(DCMIP 2012 case 4)*

Compare adaptivity *(No diffusion, equal number of active grid points at day 9)*



Baroclinic instability of jet stream *(DCMIP 2012 case 4)*

Compare adaptivity *(No diffusion, equal number of active grid points at day 9)*

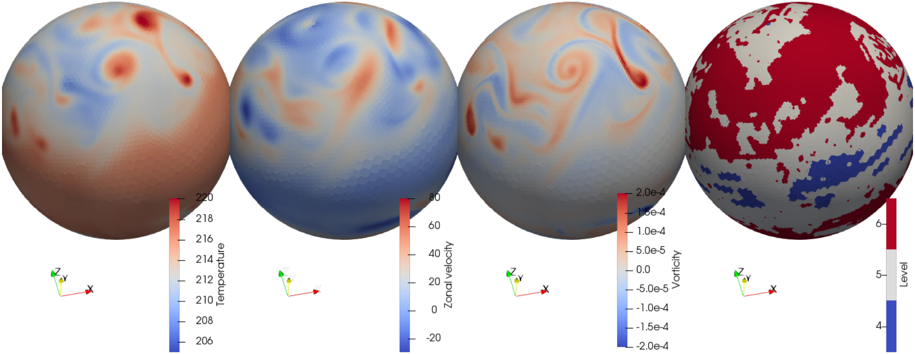


Held & Suarez (1994) $1/4^\circ$ maximum resolution

$\varepsilon = 0.02$, 18 vertical levels, results at 250 hPa, piecewise parabolic remapping

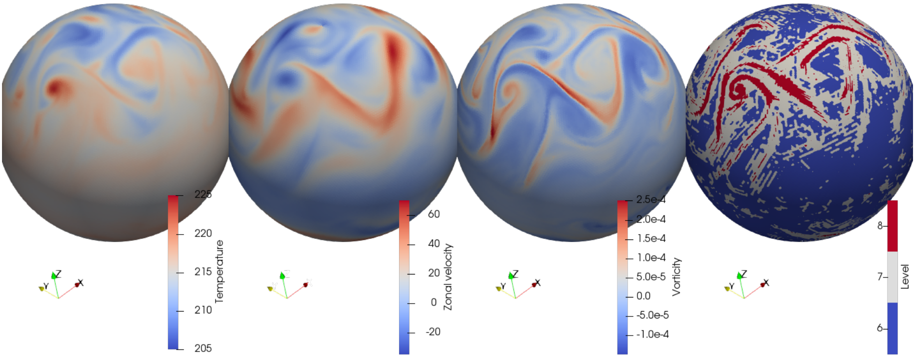
Held & Suarez (1994) low resolution 1° run

$\varepsilon = 0.04$ *(18 vertical levels, results at 250 hPa, piecewise parabolic remapping)*



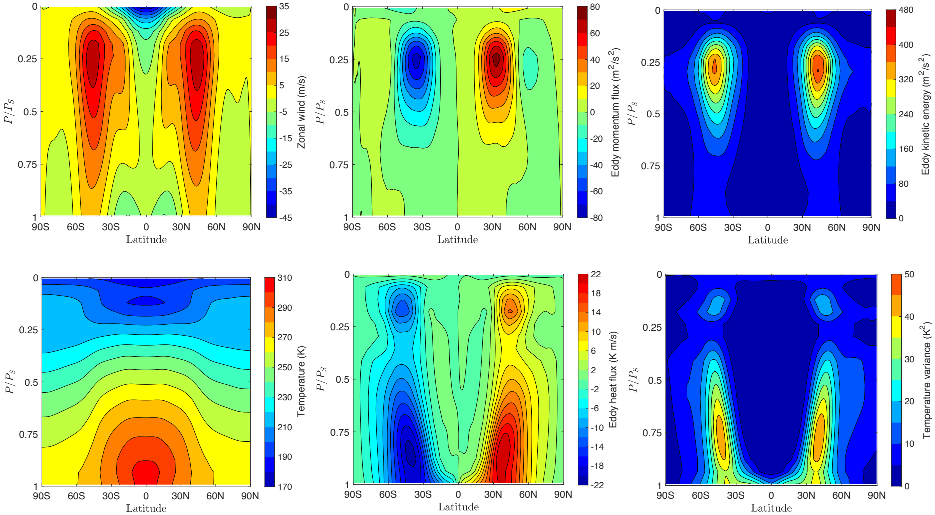
Held & Suarez (1994) high resolution $1/4^\circ$ run

$\varepsilon = 0.02$ *(18 vertical levels, results at 250 hPa, piecewise parabolic remapping)*



Held & Suarez (1994) zonal statistics

High resolution: $J = 6, 7, 8$, $\varepsilon = 0.02$, piecewise parabolic remapping



Conclusions

- DYNAMICO-based 3D hydrostatic model

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping
- Multiscale representation

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping
- Multiscale representation
- Dynamic adaptivity controlled by local estimate of interpolation error

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping
- Multiscale representation
- Dynamic adaptivity controlled by local estimate of interpolation error
- Vertically uniform, horizontally adapted grid

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping
- Multiscale representation
- Dynamic adaptivity controlled by local estimate of interpolation error
- Vertically uniform, horizontally adapted grid
- Adaptivity overlay on TRiSK discretization or other flux-based schemes

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping
- Multiscale representation
- Dynamic adaptivity controlled by local estimate of interpolation error
- Vertically uniform, horizontally adapted grid
- Adaptivity overlay on TRiSK discretization or other flux-based schemes
- Efficient parallelization using `mpi`, dynamic load balancing, hybrid data structure

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping
- Multiscale representation
- Dynamic adaptivity controlled by local estimate of interpolation error
- Vertically uniform, horizontally adapted grid
- Adaptivity overlay on TRiSK discretization or other flux-based schemes
- Efficient parallelization using `mpi`, dynamic load balancing, hybrid data structure
- Large grid compression achieved at high resolutions

Conclusions

- DYNAMICO-based 3D hydrostatic model
- Lagrangian vertical coordinate (ALE) with conservative remapping
- Multiscale representation
- Dynamic adaptivity controlled by local estimate of interpolation error
- Vertically uniform, horizontally adapted grid
- Adaptivity overlay on TRiSK discretization or other flux-based schemes
- Efficient parallelization using `mpi`, dynamic load balancing, hybrid data structure
- Large grid compression achieved at high resolutions

Ongoing work

- Adapt vertical grid by optimizing target grid *($r-$adaptivity)* and possibly de-activating some vertical layers *(dormant layers)*
- Simple physics applied to planets *(Saturn, exoplanets)*
- Ocean model *(ocean circulation, turbulence generation, tsunamis)*