A wavelet-based dynamically adaptive dynamical core

results and perspectives

Nicholas Kevlahan

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

PDEs on the Sphere, Montréal, 29 April - 3 May 2019
Collaborators

- **Thomas Dubos**
  Laboratoire de Météorologie Dynamique
  École Polytechnique, France

- **Matthias Aechtner**
  Former PhD student (contributed to shallow water code)
Development of adaptive 3D-hydrostatic dynamical core

1. Shallow water equations on the plane using TRiSK discretization

(Credit: NASA Apollo 17 mission)
2. Shallow water equations on the sphere using TRiSK discretization (Icosahedral C-grid)

(Credit: NASA Apollo 17 mission)
Development of adaptive 3D-hydrostatic dynamical core

Volume penalization for coastline boundary conditions in ocean models

(Credit: NASA Apollo 17 mission)
Development of adaptive 3D-hydrostatic dynamical core

(Credit: NASA Apollo 17 mission)

3D hydrostatic extension using DYNAMICO approach, horizontal adaptivity
Wavelet adaptivity

\[ u(x) = \sum_{k \in K^0} u_k^0 \phi_k^0(x) + \sum_{j=0}^{+\infty} \sum_{k \in L^j} \tilde{u}_k^j \psi_k^j(x) \]
Wavelet adaptivity

\[ u_{\geq}(x) = \sum_{k \in K^0} u_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{k \in L_j} \tilde{u}_k^j \psi_k^j(x) \]

\[ |\tilde{u}_k^j| \geq \varepsilon \]

Function \( u(x) \)

Wavelet locations \( x_k^j \)

\[ \varepsilon = 10^{-3} \]
Wavelet adaptivity

\[ \| u(x) - u_{\geq}(x) \|_\infty = O(\varepsilon) \]

\[ \mathcal{N} = O(\varepsilon^{-1/2N}) \]

\[ \| u(x) - u_{\geq}(x) \|_\infty = O(\mathcal{N}^{-2N}) \]

Function \( u(x) \)

Wavelet locations \( x^j_k \)

\( \varepsilon = 10^{-3} \)
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

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

\[ \tilde{u}_k^j \implies u_k^j \implies \frac{\partial^n u}{\partial x^n}(x_k^j), \quad O(N) \text{ complexity} \]
Dynamically adaptive wavelet method for PDEs

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

\[ \tilde{u}_k^j \implies u_k^j \implies \frac{\partial^n u}{\partial x^n}(x_k^j), \quad O(N) \text{ complexity} \]
Dynamically adaptive wavelet method for PDEs

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

\[ \tilde{u}^j_k \implies u^j_k \implies \frac{\partial^n u}{\partial x^n}(x^j_k), \quad O(N) \text{ complexity} \]
Discrete wavelet transform on the sphere

vertex values $\Rightarrow$ vertex values $\Rightarrow$ vertex values

$\Downarrow$ $\Downarrow$

interpolation errors $\Rightarrow$ interpolation errors

vertex values $\Leftarrow$ vertex values $\Leftarrow$ vertex values

high-pass filter

wavelets

reconstruction

prolongation
2D: TRiSK scheme \textit{(Thuburn et al. 2010)}

Staggered dual grids for mass and vorticity \textit{(Velocity at cell edges)}

Discrete shallow water equations

\[
\begin{align*}
\frac{\partial h_i}{\partial t} &= -\left[\text{div}(F_e)\right]_i \\
\frac{\partial u_e}{\partial t} &= F_e^\perp \hat{q}_e - [\text{grad}(B_i)]_e
\end{align*}
\]

- $F_e = \hat{h}_e u_e$ is thickness flux
- $F_e^\perp$ is perpendicular to $F_e$
Scale commutation properties of differential operators

\[ B_i^0, F_e^0, u_e^0 \quad \xrightarrow{\text{grad}^0, \text{div}^0, \text{curl}^0} \quad \text{grad } B_i^0, \text{div } F_e^0, \text{curl } u_e^0 \]

\[ B_i^1, F_e^1, u_e^1 \quad \xrightarrow{\text{grad}^1, \text{div}^1, \text{curl}^1} \quad \text{grad } B_i^1, \text{div } F_e^1, \text{curl } u_e^1 \]

Commutation diagram
Scale commutation properties of differential operators

Commutation relations

\[ R_h^j \circ \text{div}^{j+1} = \text{div}^j \circ R_F^j \quad \text{conserve mass} \]

\[ \text{curl}^j \circ R_u^j = R_\zeta^j \circ \text{curl}^{j+1} \quad \text{conserve circulation} \]

\[ \text{grad}^j \circ R_B^j = R_u^j \circ \text{grad}^{j+1} \quad \text{no spurious vorticity} \]
Scale commutation properties of differential operators

Commutation relations

\[ R_h^j \circ \text{div}^{j+1} = \text{div}^j \circ R_F^j \quad \text{conserve mass} \]
\[ \text{curl}^j \circ R_u^j = R_\zeta^j \circ \text{curl}^{j+1} \quad \text{conserve circulation} \]
\[ \text{grad}^j \circ R_B^j = R_u^j \circ \text{grad}^{j+1} \quad \text{no spurious vorticity} \]

Adaptive overlay on any flux-based method
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 \textit{(Dubos et al 2015)}

- Multilayer hydrostatic shallow water equations
  \textit{(compressible/incompressible)}
3D: DYNAMICO equations *(Dubos et al 2015)*

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

\[
\begin{align*}
\frac{\partial m_{ik}}{\partial t} + \delta_i U_k &= 0, \\
\frac{\partial \Theta_{ik}}{\partial t} + \delta_i (\theta_{ek}^* U_k) &= 0 \\
\frac{\partial v_{ek}}{\partial t} + \delta_e B_k + \theta_{ek}^* \delta_e \pi_k + (q_k U_k)^\perp_e &= 0
\end{align*}
\]
Multiscale icosahedral grid resolution

<table>
<thead>
<tr>
<th>$J$</th>
<th>$N$</th>
<th>$\Delta x$ [deg]</th>
<th>$\Delta x$ [km]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2,562</td>
<td>4</td>
<td>480</td>
</tr>
<tr>
<td>5</td>
<td>10,242</td>
<td>2</td>
<td>240</td>
</tr>
<tr>
<td>6</td>
<td>40,962</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>163,842</td>
<td>1/2</td>
<td>56</td>
</tr>
<tr>
<td>8</td>
<td>655,362</td>
<td>1/4</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>2,621,442</td>
<td>1/8</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>10,485,762</td>
<td>1/16</td>
<td>7</td>
</tr>
</tbody>
</table>
Multiscale icosahedral grid resolution

<table>
<thead>
<tr>
<th>$J$</th>
<th>$N$</th>
<th>$\Delta x$ [deg]</th>
<th>$\Delta x$ [km]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2562</td>
<td>4</td>
<td>480</td>
</tr>
<tr>
<td>5</td>
<td>10242</td>
<td>2</td>
<td>240</td>
</tr>
<tr>
<td>6</td>
<td>40962</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>163842</td>
<td>1/2</td>
<td>56</td>
</tr>
<tr>
<td>8</td>
<td>655362</td>
<td>1/4</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>2621442</td>
<td>1/8</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>10485762</td>
<td>1/16</td>
<td>7</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$J$</th>
<th>$N$</th>
<th>$\Delta x$ [deg]</th>
<th>$\Delta x$ [km]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2,562</td>
<td>4</td>
<td>480</td>
</tr>
<tr>
<td>5</td>
<td>10,242</td>
<td>2</td>
<td>240</td>
</tr>
<tr>
<td>6</td>
<td>40,962</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>163,842</td>
<td>1/2</td>
<td>56</td>
</tr>
<tr>
<td>8</td>
<td>655,362</td>
<td>1/4</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>2,621,442</td>
<td>1/8</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>10,485,762</td>
<td>1/16</td>
<td>7</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>$J$</th>
<th>$N$</th>
<th>$\Delta x$ [deg]</th>
<th>$\Delta x$ [km]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2562</td>
<td>4</td>
<td>480</td>
</tr>
<tr>
<td>5</td>
<td>10242</td>
<td>2</td>
<td>240</td>
</tr>
<tr>
<td>6</td>
<td>40962</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>163842</td>
<td>1/2</td>
<td>56</td>
</tr>
<tr>
<td>8</td>
<td>655362</td>
<td>1/4</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>2621442</td>
<td>1/8</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>10485762</td>
<td>1/16</td>
<td>7</td>
</tr>
</tbody>
</table>

- Optimize **coarsest** grid, e.g. $J = 5$ (Xu 2006; Heikes & Randall 1995)
- Finer grids by recursive edge-bisection, e.g. $j = 6, 7, 8, \ldots$
- Local **adaptive** grid scale $j$ controlled by error tolerance $\varepsilon$
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 $m_{ik}, \theta_{ik}, u_{ik}$ conservatively *(either every $\Delta t$ or every $10\Delta t$)*
Arbitrary Lagrangian Eulerian vertical coordinates (ALE)

- Initialize hybrid $\sigma - P$ pressure-based grid
- Remap $m_{ik}, \theta_{ik}, u_{ik}$ conservatively (either every $\Delta t$ 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 \( m_{ik}, \theta_{ik}, u_{ik} \) **conservatively** (*either every \( \Delta t \) or every \( 10\Delta t \))
- **Target grid is initial grid**  
  (*could be optimized at each remapping for \( r-\text{adaptivity} \))
- 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 $m_{ik}, \theta_{ik}, u_{ik}$ conservatively *(either every $\Delta t$ 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 $m_{ik}, \theta_{ik}, u_{ik}$ conservatively (either every $\Delta t$ 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 \cite{Shchepetkin2001,EngwirdaKelly2016}
- Limiter: none, monotone, WENO

- At least piecewise linear required for Held–Suarez test case
- \textit{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**
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
Mountain-induced Rossby wave \textit{(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 \textit{(DCMIP 2012 case 4)}

26 vertical levels, results at 867 hPa, hyperdiffusion

\[ \nu_{\text{scalar}} = 5.3 \times 10^{12}, \nu_{\text{div}} = 1.0 \times 10^{14}, \nu_{\text{curl}} = 1.1 \times 10^{13} \]
Baroclinic instability of jet stream  \textit{(DCMIP 2012 case 4)}

Grid compression as $J$ increases \textit{(}J_{\min} = 5, \text{ no diffusion, adapt on trend)}
Baroclinic instability of jet stream  (DCMIP 2012 case 4)

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

Day 9

Adapt on variables

Adapt on trend
Baroclinic instability of jet stream  (DCMIP 2012 case 4)

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

Day 20
Adapt on variables

Adapt on trend

Nicholas Kevlahan (McMaster University)  Wavelet-based dynamical core  Reading September 2018  20 / 26
Baroclinic instability of jet stream (DCMIP 2012 case 4)

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

Nicholas Kevlahan (McMaster University)
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

Nicholas Kevlahan (McMaster University)  Wavelet-based dynamical core  Reading September 2018 25 / 26
Conclusions

- DYNAMICO-based 3D hydrostatic model
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 (e.g. ocean circulation, turbulence generation, tsunamis)
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 ($r_{\text{adaptivity}}$) and possibly de-activating some vertical layers (dormant layers)

Simple physics applied to planets (Saturn, exoplanets)

Ocean model (e.g. ocean circulation, turbulence generation, tsunamis) → Laurent Debreu’s talk!

Nicholas Kevlahan, McMaster University
Conclusions

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

Ongoing work

- Adapt vertical grid by optimizing target grid and possibly de-activating some vertical layers (dormant layers)

Simple physics applied to planets (Saturn, exoplanets)

Ocean model (e.g. ocean circulation, turbulence generation, tsunamis) → Laurent Debreu’s talk!
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 $(r - \text{adaptivity})$ and possibly de-activating some vertical layers (dormant layers)

Simple physics applied to planets (Saturn, exoplanets)

Ocean model (e.g. ocean circulation, turbulence generation, tsunamis)

→ Laurent Debreu’s talk!

Nicholas Kevlahan (McMaster University)

Wavelet-based dynamical core
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

Ongoing work:
- Adapt vertical grid by optimizing target grid ($r - \text{adaptivity}$) and possibly de-activating some vertical layers (dormant layers)
- Simple physics applied to planets (Saturn, exoplanets)
- Ocean model (e.g. ocean circulation, turbulence generation, tsunamis) → Laurent Debreu's talk!
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 \((r-\text{adaptivity})\) and possibly de-activating some vertical layers \((\text{dormant layers})\)
- Simple physics applied to planets \((Saturn, \text{exoplanets})\)
- Ocean model \((\text{e.g. ocean circulation, turbulence generation, tsunamis})\)
  \(\rightarrow\) Laurent Debreu’s talk!