Is there a need for local horizontal discretizations?

Daan Degrauwe, Steven Caluwaerts, Piet Termonia

RMI Belgium
Introduction

- What’s wrong with a spectral model?
- What can local methods bring us, and at what price?
- Conclusions and prospects
In fact, our goal is pretty simple: develop the best possible atmospheric model. (and have fun while doing so)
In fact, our goal is pretty simple: develop the best possible atmospheric model. (and have fun while doing so)

...but what does best mean?
Compromises are necessary (e.g. accuracy vs. cpu time)
In fact, our goal is pretty simple: develop the best possible atmospheric model. (and have fun while doing so)

... but what does best mean?
Compromises are necessary (e.g. accuracy vs. cpu time)

Currently, our model uses a spectral horizontal discretization. Are we confronting the limitations of spectral methods? What trade-off is made by local methods?
In fact, our goal is pretty simple: develop the best possible atmospheric model. (and have fun while doing so)

...but what does best mean?

Compromises are necessary (e.g. accuracy vs. cpu time)

Currently, our model uses a spectral horizontal discretization. Are we confronting the limitations of spectral methods? What trade-off is made by local methods?

Spoiler alert: no definitive answers will be given in this presentation!
From the *accuracy* point of view, spectral methods are unsurpassable: their order of accuracy is infinite!

(Limited tests indicate that) even over steep slopes, the accuracy of spectral methods remains unchallenged.

Moreover, the calculation of derivatives and solving the Helmholtz equation are trivial. This allows for (semi-)implicit time-stepping and large timesteps. So our spectral dynamics are also quite *efficient*. 
From the *accuracy* point of view, spectral methods are unsurpassable: their order of accuracy is infinite!

(Limited tests indicate that) even over steep slopes, the accuracy of spectral methods remains unchallenged.

Moreover, the calculation of derivatives and solving the Helmholtz equation are trivial. This allows for (semi-)implicit time-stepping and large timesteps. So our spectral dynamics are also quite *efficient*.

... but they require spectral transforms (FFT or Legendre transform for the global). These are nonlocal, i.e. they require domain-wide communication. This makes their use problematic on massively parallel machines.

(another disadvantage of a spectral model is the requirement of a homogeneous reference state for the semi-implicit time-stepping)

But at what point do the costs no longer justify the accuracy?

– to answer this question, we must closely investigate the alternatives.
When considering alternatives for the spectral horizontal discretization, we try to keep as much as possible of the model intact:

- only way to make a clean comparison
- limited development cost (no need to modify physics, ...)

So for the time being, we stick to a semi-implicit time discretization and a semi-Lagrangian advection scheme.
When considering alternatives for the spectral horizontal discretization, we try to keep as much as possible of the model intact:

- only way to make a clean comparison
- limited development cost (no need to modify physics, . . . )

So for the time being, we stick to a semi-implicit time discretization and a semi-Lagrangian advection scheme.

Finite-difference discretizations are considered on the following grids:

- **A-grid**
  - wind speed not staggered
- **C-grid**
  - wind speed staggered
- **Z-grid**
  - vorticity/divergence not staggered

These discretizations are tested with a 1D shallow water toy model.
It is well known (Mesinger & Arakawa, 1976) that the dispersion relation of gravity waves on an A-grid is problematic (negative group velocity at the shortest scales).

The C-grid doesn’t have this problem, but the staggering makes semi-Lagrangian advection 3 times more expensive.

Pierre Bénard has shown (cfr. Piet’s presentation of last year) that in certain cases, the short waves behave better on an A-grid than on a C-grid.

Z-grid seems to offer the best of both worlds (at the expense of solving a Poisson equation to retrieve wind from vorticity/divergence), if time-symmetry is respected.
Checking the geopotential behavior for a geostrophic adjustment problem seems to confirm this.
Checking the geopotential behavior for a geostrophic adjustment problem seems to confirm this.
Checking the geopotential behavior for a geostrophic adjustment problem seems to confirm this.

![Geopotential Behavior Graph](image)
...but when watching the result of the $u$-component in the same test, something strange is observed.

Already after a single timestep, the $u$-field turns out to be very noisy!
What’s happening here? We’ll do the analysis for a simpler SWE system without Coriolis terms.

- The (linearized) SWE are a hyperbolic system. The solution is dominated by two waves.
  
  These waves \((w_1, w_2)\) are a combination of the prognostic variables \((u, \phi)\), and can be seen as ‘more fundamental’ solutions since they propagate independently from one another.

- The exact transformation between wave amplitudes and prognostic variables is not wavenumber dependent:

  \[
  \begin{pmatrix}
  u \\
  \phi 
  \end{pmatrix} = \begin{pmatrix}
  1 & -1 \\
  c & c 
  \end{pmatrix} \begin{pmatrix}
  w_1 \\
  w_2 
  \end{pmatrix}
  \quad \text{with } c \sim 100 \text{ m/s}
  \]
In the discrete case, the transformation is determined by the *eigenvectors* of the amplification matrix (while the dispersion relation is determined by the eigenvalues).

For a spectral, A-grid and C-grid discretization, the wavenumber-independence is retained.
In the discrete case, the transformation is determined by the \textit{eigenvectors} of the amplification matrix (while the dispersion relation is determined by the eigenvalues).

For a spectral, A-grid and C-grid discretization, the wavenumber-independence is retained.

But for the Z-grid discretization, the transformation becomes wavenumber-dependent:

\[
\begin{pmatrix}
u \\
\phi
\end{pmatrix} =
\begin{pmatrix}
\sqrt{2 - 2 \cos k\Delta x} & -\sqrt{2 - 2 \cos k\Delta x} \\
\frac{c \sin k\Delta x}{c \sin k\Delta x} & \frac{c \sin k\Delta x}{c \sin k\Delta x}
\end{pmatrix}
\begin{pmatrix}
w_1 \\
w_2
\end{pmatrix}
\]

For the shortest waves \((k\Delta x \to \pi)\), the \(\phi\)-component becomes relatively small:

\[
u \approx \phi \\
\phi \approx \frac{w_1}{w_2}
\]
In the discrete case, the transformation is determined by the \textit{eigenvectors} of the amplification matrix (while the dispersion relation is determined by the eigenvalues).

For a spectral, A-grid and C-grid discretization, the wavenumber-independence is retained.

But for the Z-grid discretization, the transformation becomes wavenumber-dependent:

\[
\begin{pmatrix}
u \\
\phi
\end{pmatrix} = \begin{pmatrix}
\sqrt{2 - 2 \cos k \Delta x} & -\sqrt{2 - 2 \cos k \Delta x} \\
c \sin k \Delta x & c \sin k \Delta x
\end{pmatrix}
\begin{pmatrix}
w_1 \\
w_2
\end{pmatrix}
\]

For the shortest waves \((k \Delta x \to \pi)\), the \(\phi\)-component becomes relatively small:

As a consequence, an initial state without \(u\)-component is decomposed into two waves with non-negligible (but initially opposite) \(u\)-component.
Decomposition of the adjustment test problem initial state:

The two gravity waves propagate in opposite directions, and after a single timestep, this results in a noisy $u$-field.
Conclusions from these tests

- What we’ve seen so far:
  - Every discretization has its strengths and weaknesses.
  - The quality of a discretization is case-dependent (advection vs. adjustment).
  - The discretization effects may be very subtle. Even careful inspection of the dispersion relation (eigenvalues) is no guarantee to have proper behavior.
Conclusions from these tests

- What we’ve seen so far:
  - Every discretization has its strengths and weaknesses.
  - The quality of a discretization is case-dependent (advection vs. adjustment).
  - The discretization effects may be very subtle. Even careful inspection of the dispersion relation (eigenvalues) is no guarantee to have proper behavior.

- To make these conclusions even more relative: how representative is the shallow water toy model for a 3D atmospheric model?
  - diabatic effects triggering shortest waves
  - numerical diffusion filtering shortest waves
Conclusions from these tests

- What we’ve seen so far:
  - Every discretization has its strengths and weaknesses.
  - The quality of a discretization is case-dependent (advection vs. adjustment).
  - The discretization effects may be very subtle. Even careful inspection of the dispersion relation (eigenvalues) is no guarantee to have proper behavior.

- To make these conclusions even more relative: how representative is the shallow water toy model for a 3D atmospheric model?
  - diabatic effects triggering shortest waves
  - numerical diffusion filtering shortest waves

- And that’s not the end yet. The most suitable discretization also depends on the (future!) hardware:
  - efficiency
  - scalability (in fact the main motivation for reviewing the spectral dynamical core)
  - energy consumption
Conclusions from these tests

- What we’ve seen so far:
  - Every discretization has its strengths and weaknesses.
  - The quality of a discretization is case-dependent (advection vs. adjustment).
  - The discretization effects may be very subtle. Even careful inspection of the dispersion relation (eigenvalues) is no guarantee to have proper behavior.

- To make these conclusions even more relative: how representative is the shallow water toy model for a 3D atmospheric model?
  - diabatic effects triggering shortest waves
  - numerical diffusion filtering shortest waves

- And that’s not the end yet. The most suitable discretization also depends on the (future!) hardware:
  - efficiency
  - scalability (in fact the main motivation for reviewing the spectral dynamical core)
  - energy consumption

- So the ‘best’ solution is very much situation dependent.

The only way out of this situation is to aim for a modular code, where different options can be used next to each other.
The ESCAPE project was recently approved for H2020 EU funding.

ECMWF is the coordinating partner; other partners include HIRLAM and ALADIN members, HPC hardware manufacturers, universities and supercomputing centers.

The core of ESCAPE is the identification of fundamental algorithm building blocks ('NWP dwarfs'), e.g.
- spectral transforms
- sparse solvers
- unstructured mesh generation
- advective transport mechanisms
- time-stepping strategies
- ...

Adaptation of NWP dwarfs to hardware accelerators

Benchmarking strategies to gauge code efficiency and energy consumption on heterogeneous hardware

Breakdown of the model in these dwarfs means modularity
Thank you!