Organization and technical aspects of LMD model developments

E. Millour (LMD)
and the LMDZ development teams

Japanese-French worshop
May 11th 2015, Kobe

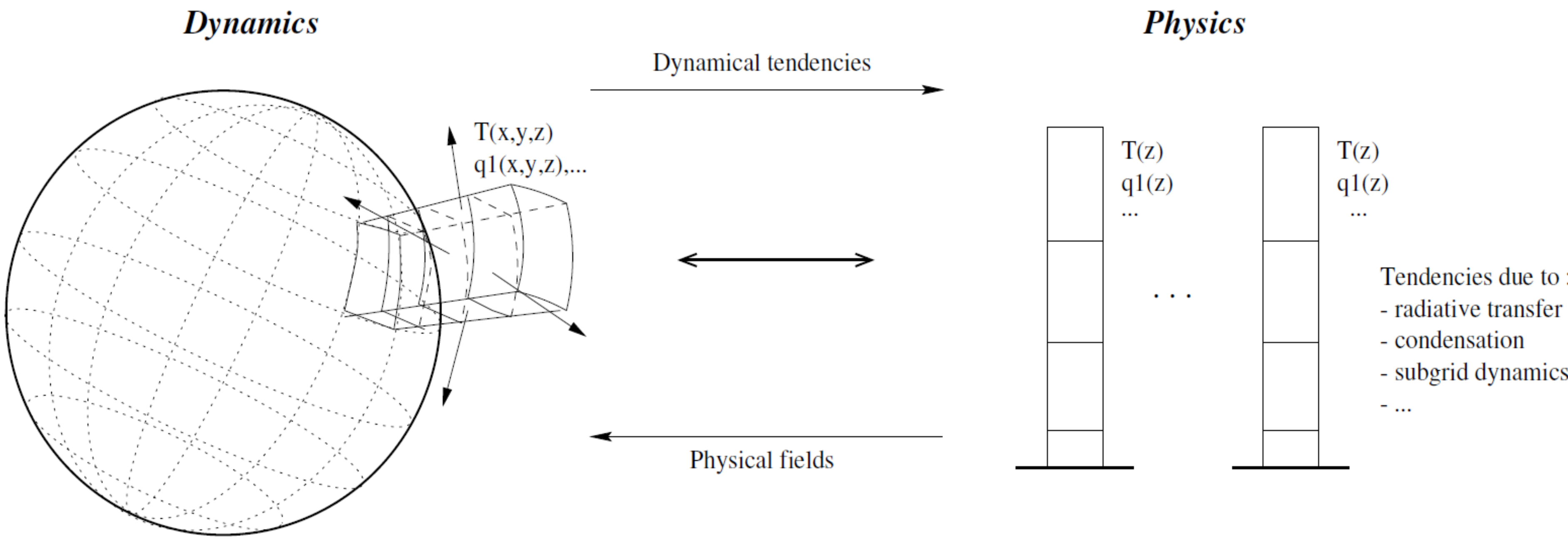
Overview of LMDZ

- **LMDZ**: the Global Circulation Model (GCM) developed at LMD (and co-workers).

- Versions of the LMDZ code are used for modeling the Earth (e.g. LMDZ5 used in the IPCC exercise) and planets (LMDZ-MARS, LMDZ-VENUS, etc.).
  We try (want!) to not reinvent the wheel each time…
  In practice, over the years, the codes were first developed separately (Venus from LMDZ4 Earth code, Mars code from LMDZ3 code, Generic model from Mars model, etc.). We now strive for a more unified way of developing our codes.

- Code designed and maintained to be used in serial on « small » computers as well as in parallel (mixed MPI/OpenMP) on supercomputers.

Grids in LMDZ



Separation between physics and dynamics:

- "dynamics": solving the GFD equations on the sphere; usually with the assumption of a hydrostatic balance and thin layer approximation. Valid for most terrestrial planets.
- "physics": (planet-specific) local processes, local to individual atmospheric columns.

Horizontal grids in LMDZ

Grid dimensions specified when compiling LMDZ:

makelmdz_fcm -d iimxjjmxllm

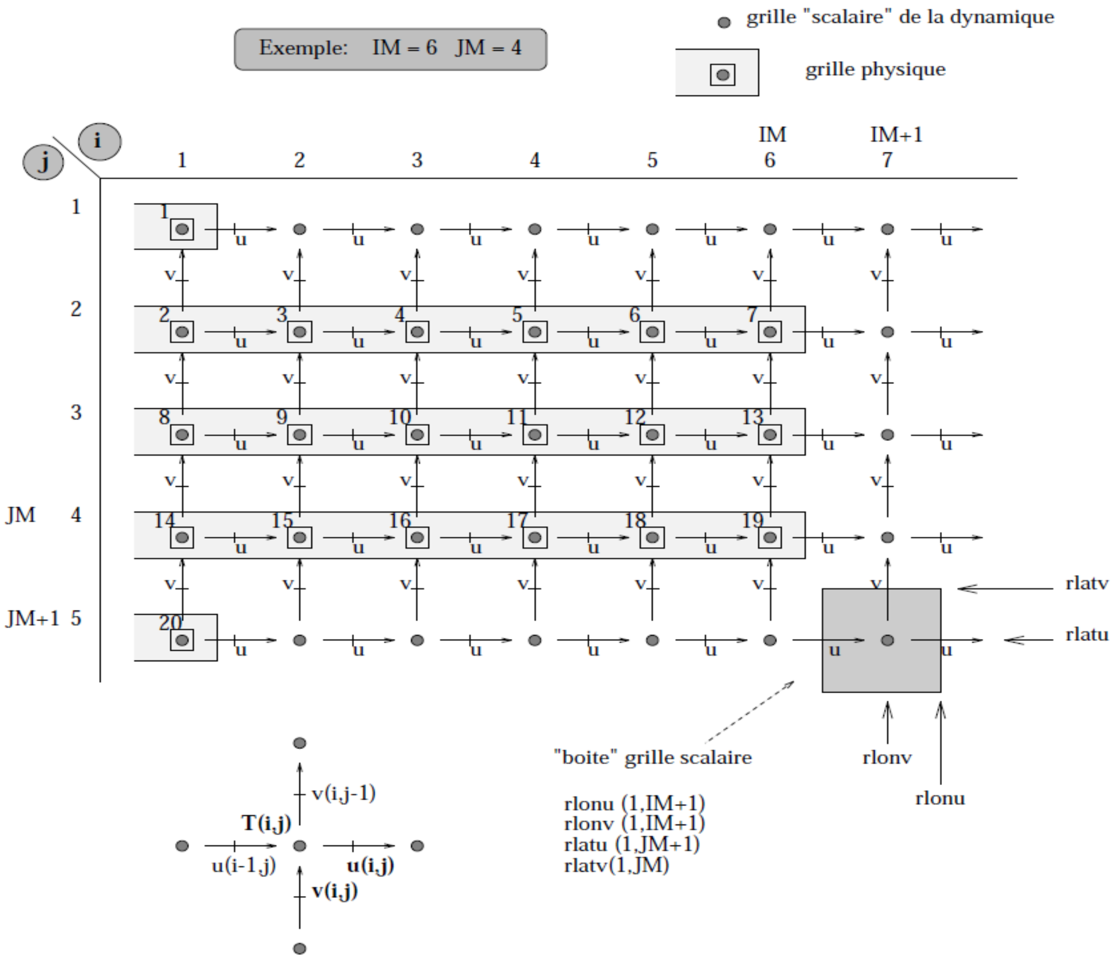
In the dynamics:

- Staggered grids, u, v and scalars (temperature, tracers) are on different meshes
- Global lonxlat grids with redundant grid points

- at the poles
- in longitude

In the physics:

- Colocated variables
- No global lonxlat horizontal grid, columns are labelled **using a single index** (from North Pole to South Pole)



Vertical discretization in LMDZ

- Model levels are hybrid sigma-pressure levels:

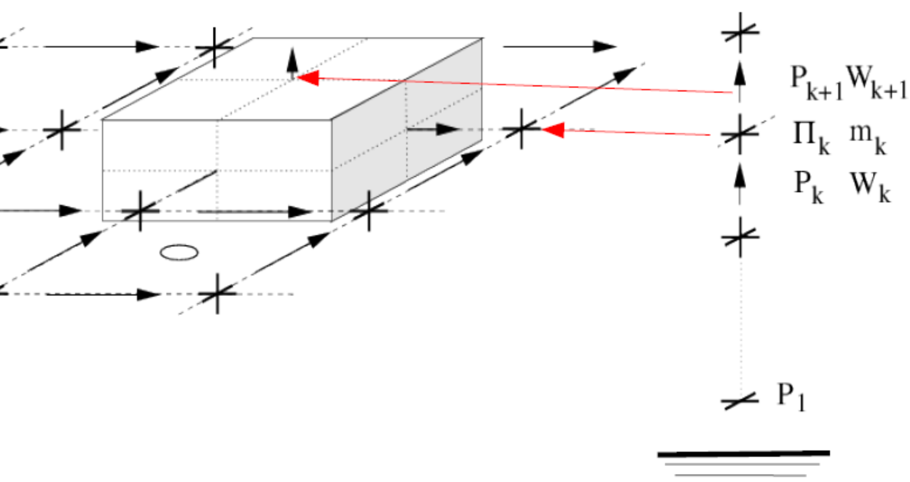
  P(level,time) = **ap**(level) + **bp**(level) . **Ps** (time)

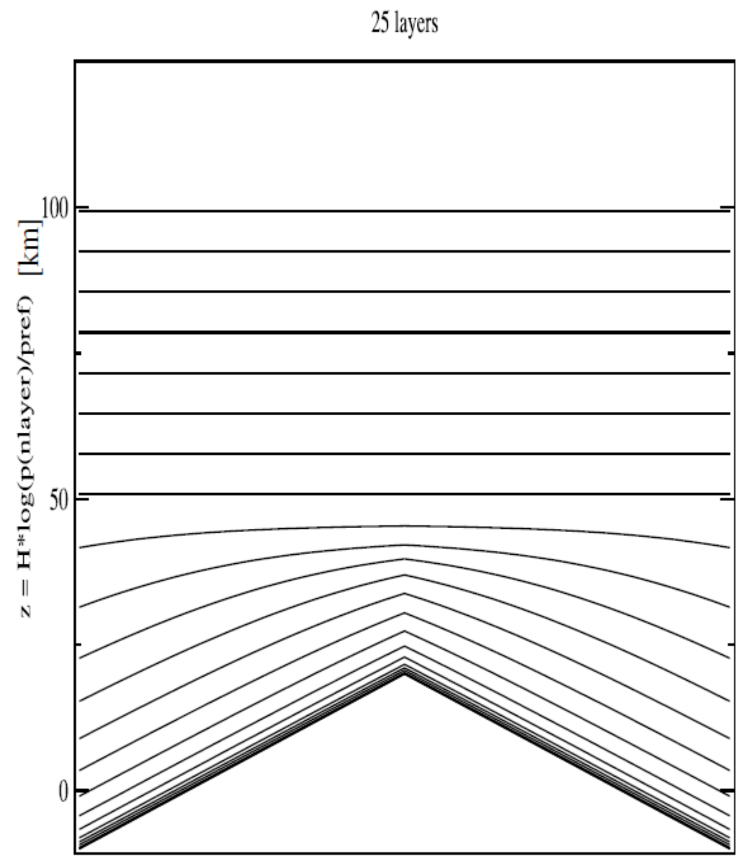
  hybrid coordinates **ap**(k) and **bp**(k) are fixed for a given model run

  Surface pressure **Ps**(t) varies during the run

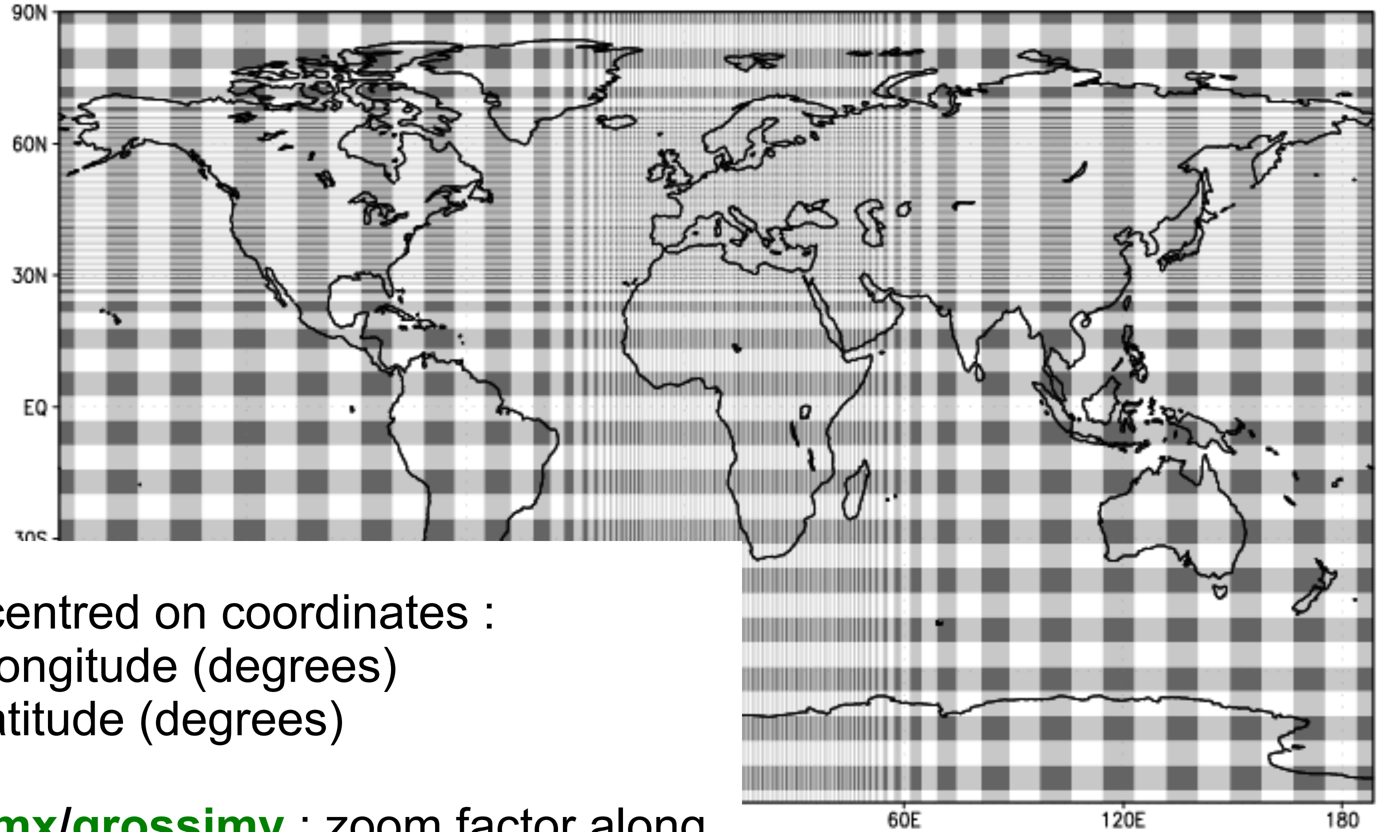
- Near the surface ap ~ 0 => bp(k) ~ P/Ps

- At high altitudes , bp ~ 0 => ap(k) ~ P





LMDZ, Z for Zoom

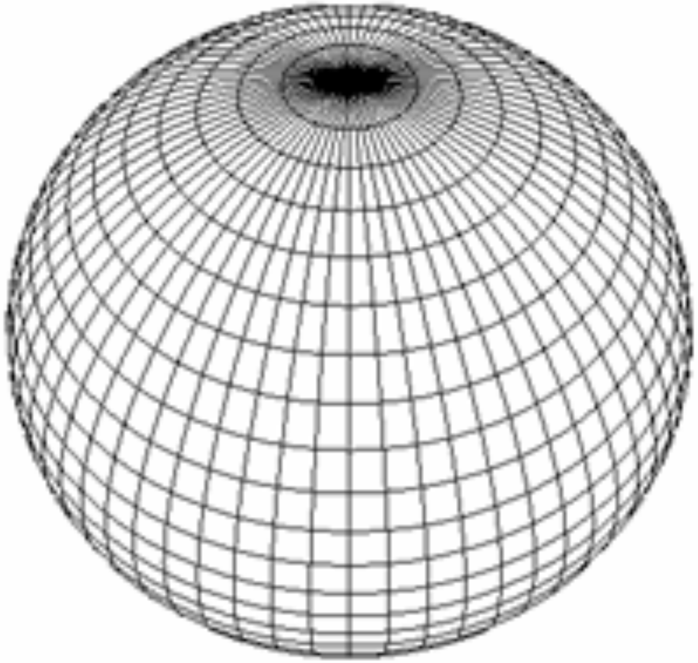


Zoom centred on coordinates :
clon : longitude (degrees)
clat : latitude (degrees)

grossimx/**grossimy** : zoom factor along x/y directions (i.e. lon/lat)
Computed as the ratio of the smallest mesh (I.e. in the zoom), compared to the mesh size for a global regular grid with the same total number of points.

dzoomx/**dzoomy** : fraction of the grid containing the zoomed area:
dzoom*360° by
dzoomy*180°

Longitudinal polar filter



- A lon-lat grid implies that the meshes tighten dramatically as the pole is approached.
- CFL conditions there would dictate using an extremely small time step for our explicit (leapfrog-Matsuno) time marching scheme.

- Longitudinal (Fourier) filtering, removing high spatial frequencies, is used to enforce that resolved features are at the level of those at ~60°

Energy spectra and lateral dissipation

- Observations (Nastrom & Gage 1985, Lindborg 1999) collected over length scales from a few to thousands of km display a characteristic energy cascade (from Skamarock, 2004).

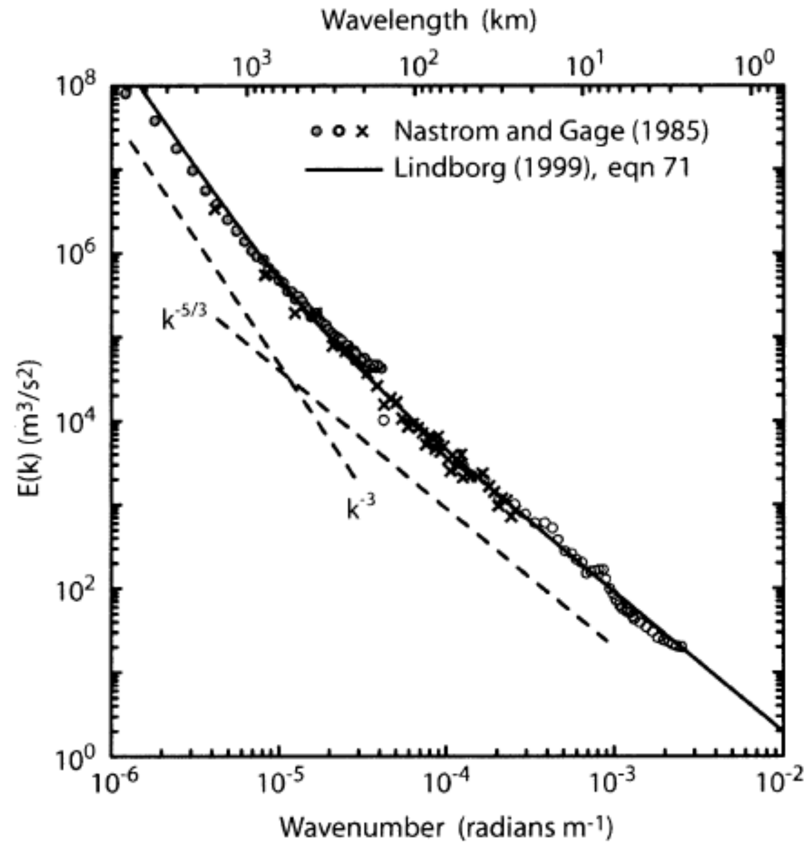
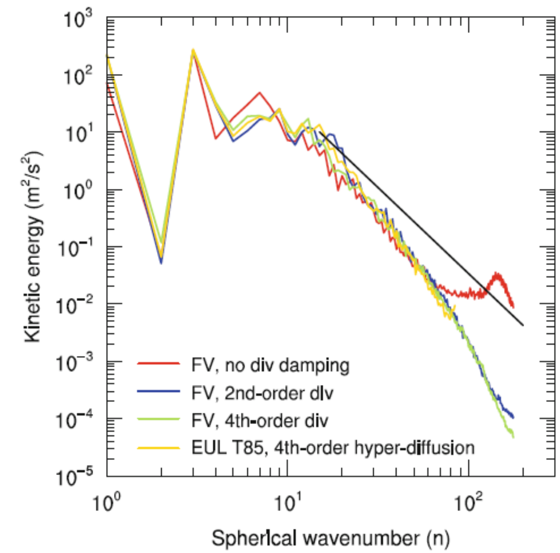


FIG. 1. Nastrom and Gage (1985) spectrum derived from the GASP aircraft observations (symbols) and the Lindborg (1999) functional fit to the MOZAIC aircraft observations.

- In order to fulfil the observed energy cascade from resolved scales to unresolved scales in GCMs, a dissipation term is added:

$$Dissip(\psi) = \frac{(-1)^{q+1}}{\tau}\nabla^{2q}\psi$$

Lateral dissipation in GCMs as a tool to pin the energy cascade



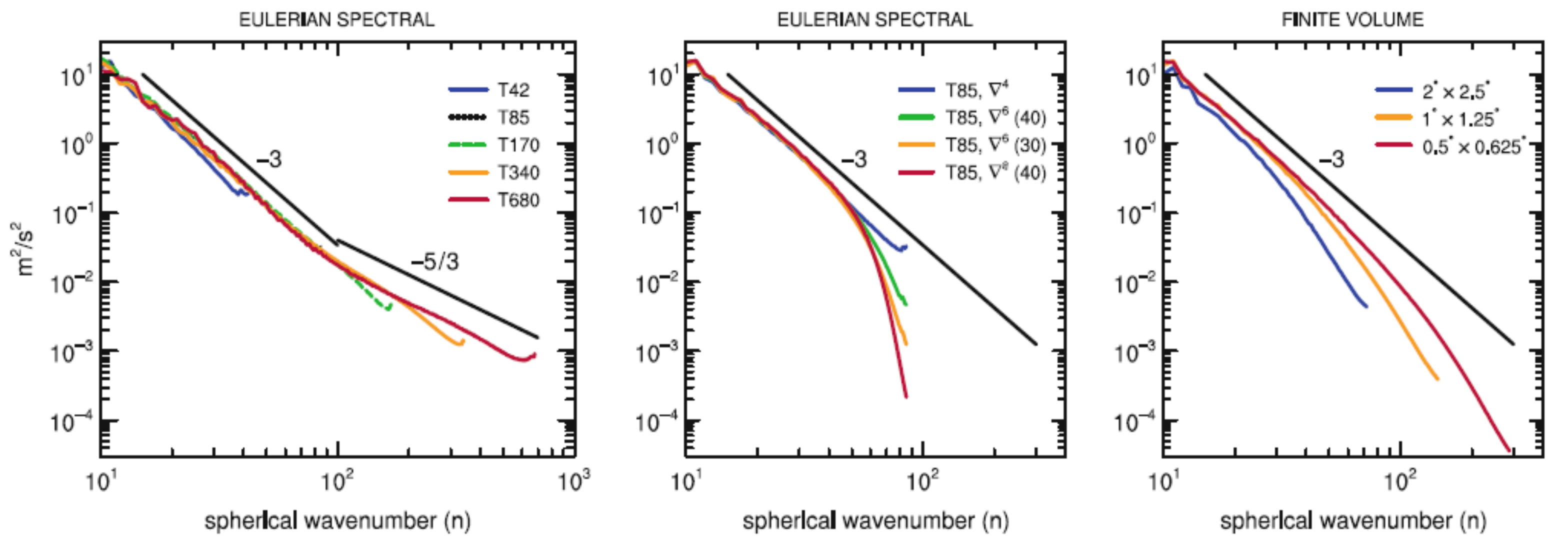


Fig. 13.4 250 hPa kinetic energy spectra as a function of the spherical wavenumber (n) in aqua-planet simulations from (*left*) CAM Eulerian spectral dynamical core with ∇^4 diffusion for different resolutions, (*center*) T85L26 Eulerian spectral dynamical with ∇^4, ∇^6 and ∇^8 diffusion, and (*right*) CAM Finite Volume (FV) dynamical core for different *lat* × *lon* resolutions in degrees and 26 levels

Controlling dissipation in LMDZ

- Parameters :

tetagdiv: dissipation time scale (s) for smallest wavelength for u,v (grad.div component)

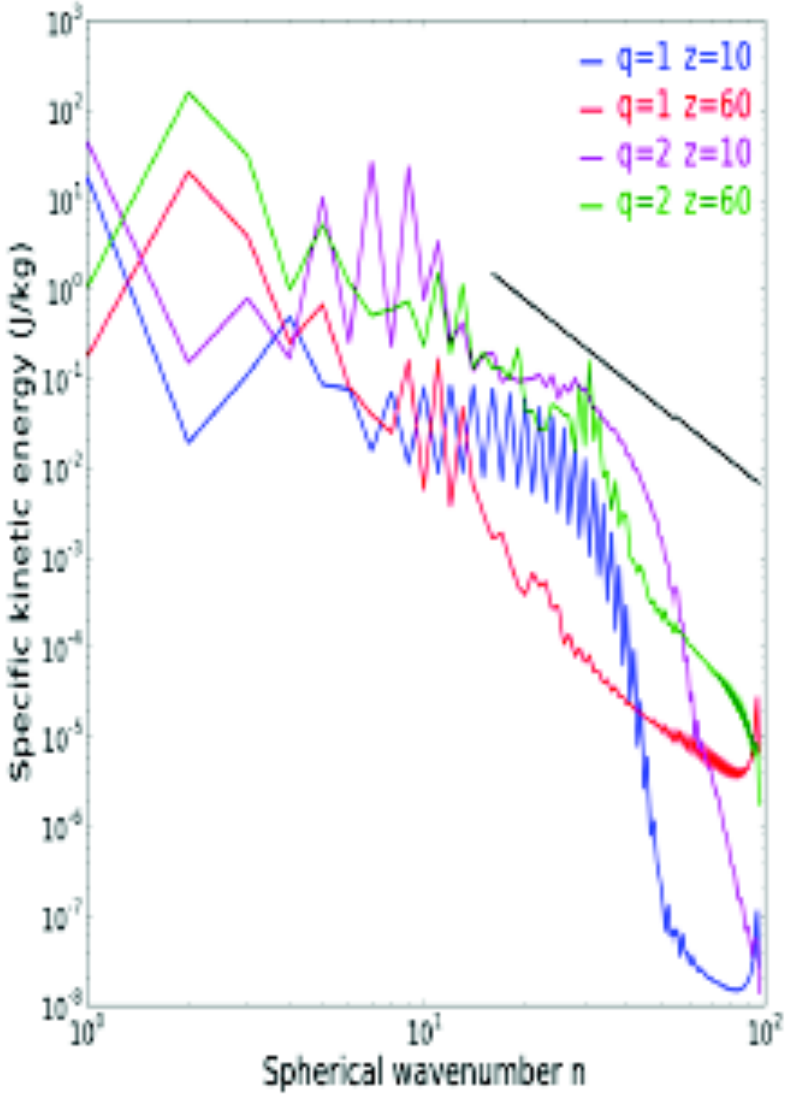
tetagrot: dissipation time scale (s) for smallest wavelength for u,v (grad.rot component)

tetatemp: dissipation time scale (s) for smallest wavelength for potential temperature (div.grad)

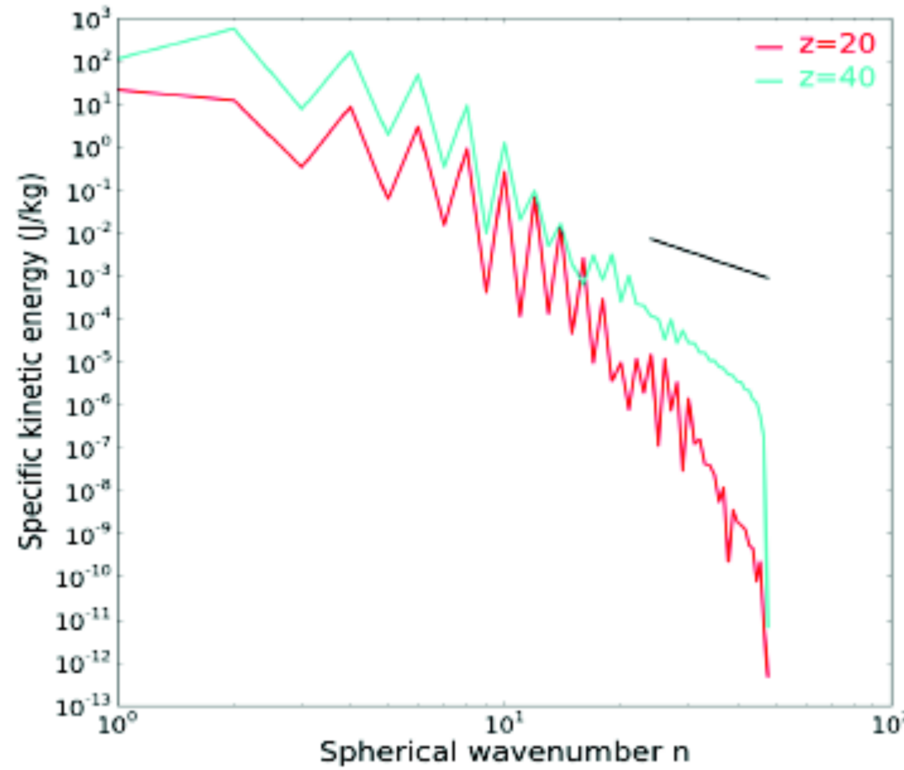
optimal teta values depend on horizontal resolution

- Moreover there is a multiplicative factor for the dissipation coefficient, which can be controlled by the user.

Some spectra
(from LMDZ/Saturn)



Energy spectra for simulations with 1 and 2 diffusion order.



Energy spectra at two different altitude levels z in the northern summer on a 64x48x64 grid. τ value is 3s. Spectra ranges up to $n = 48$.

The sponge layer

- In addition to lateral dissipation, it is necessary to damp vertically propagating waves (non-physically reflected downward from model top).
- The sponge layer is limited to topmost layers (usually 4) and added during the dissipation step.
- Sponge modes and parameters:

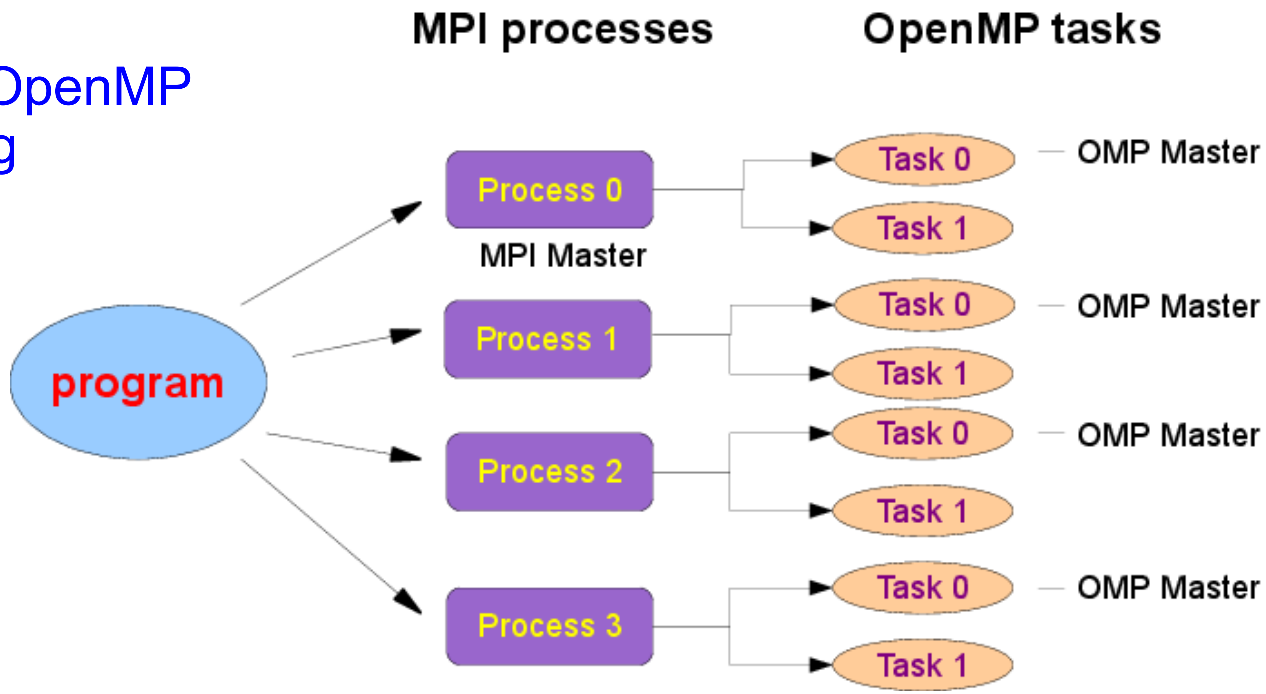
Type: Can act on zonal wind, meridional wind and/or temperature. Relaxation can be towards zero or mean zonal values.

Intensity: User-provided relaxation characteristic time scale (at topmost layer; relaxation intensity decreases along successive descending layers).

Parallelism in LMDZ

A few words on what is done

Hybrid MPI/OpenMP programming



Different parallelization approaches in the dynamics and physics

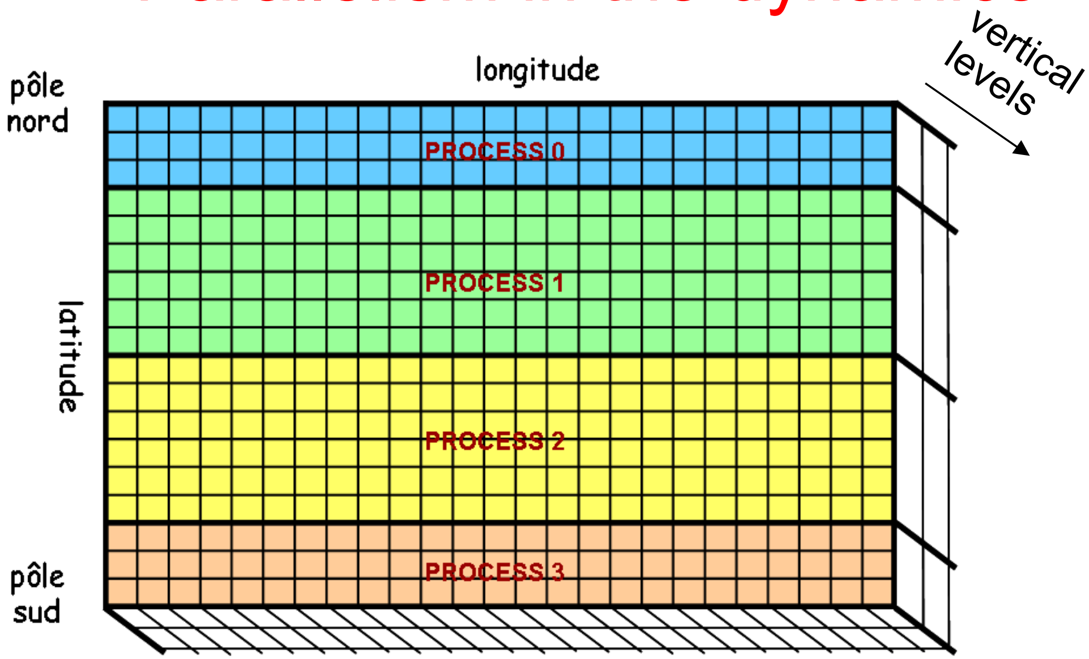
- In the dynamics

  Short time steps; many interactions between neighbouring meshes, and therefore numerous cases of data exchange and synchronizations. The subtler part of the parallelism in the code..

- In the physics

  Longer time steps; **no interaction** between neighbouring columns of the atmosphere.

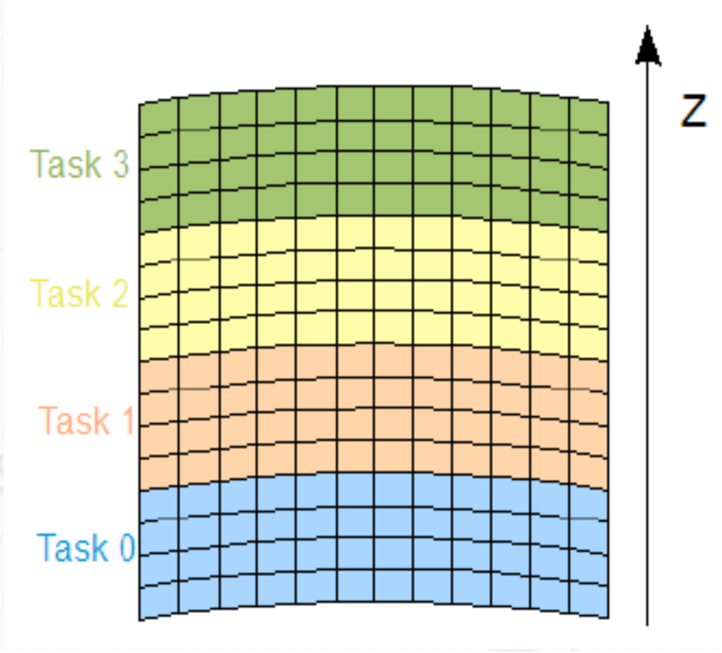
Parallelism in the dynamics



=> MPI tiling

- Tiling is by bands of latitude.
- A minimum of 3 latitude bands per MPI process is mandatory.
- But the work load is not the same for all latitudes (essentially because of the polar filter).
- We have an option to dynamically optimize (during the run) the band distribution of processes. But this scheme is only used to tune the right distribution, which is then kept fixed.

Parallelism in the dynamics



=> OpenMP split

- The split is done along the vertical levels only (the outermost loop in most computations).
- An indicate size of blocs to assign to each thread can be specified at run time.
- In practice, target chunks of 4 or 5 vertical levels for each OpenMP task (an optimal compromise, but which may depend on the machine on which the code is run).

Parallelism in the physics

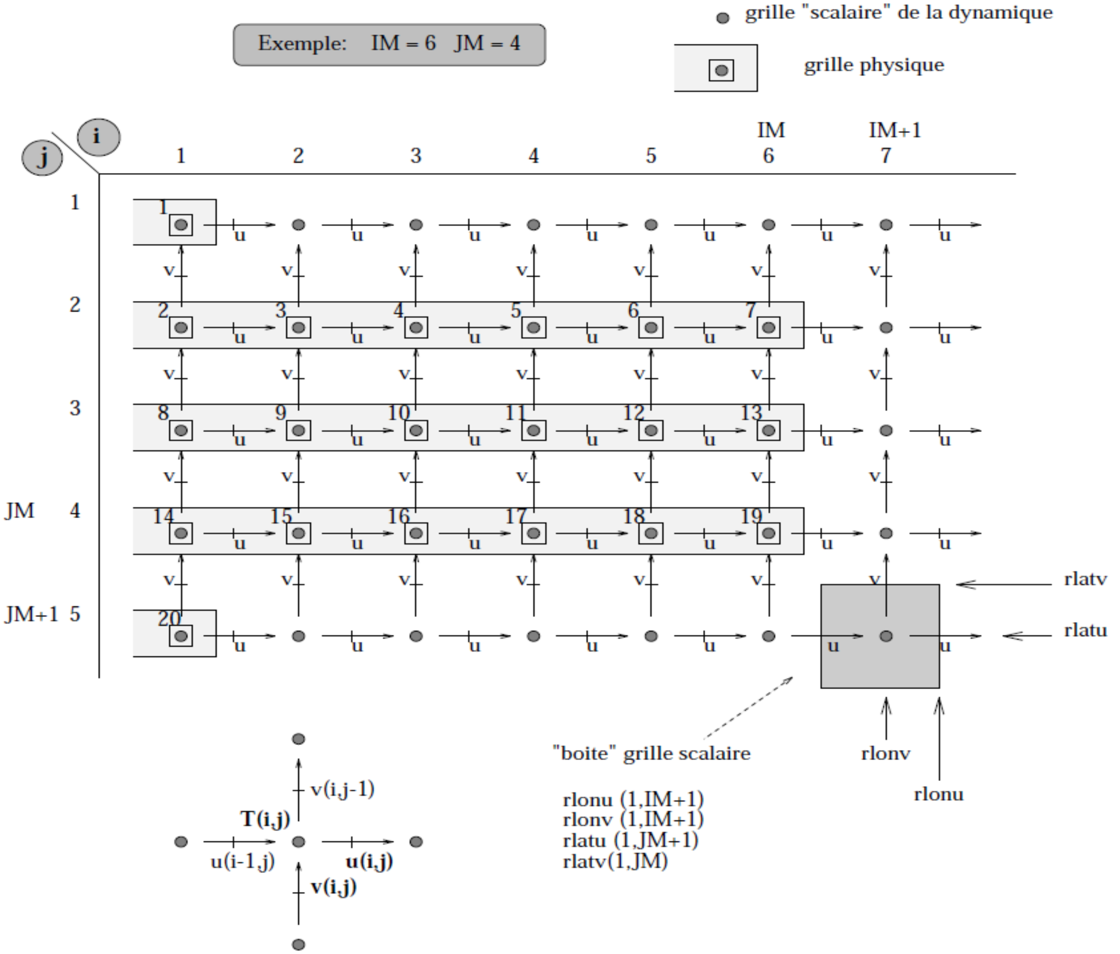
- The physics handles physical phenomena which interact within a single atmospheric column: radiation, convection boundary layer, etc.

- Individual columns of atmosphere **do not interact** with one another.

- The parallelization strategy is to distribute the colums of atmosphere over all cores.

- The physics grid : klon_glo geographic points over klev vertical levels.

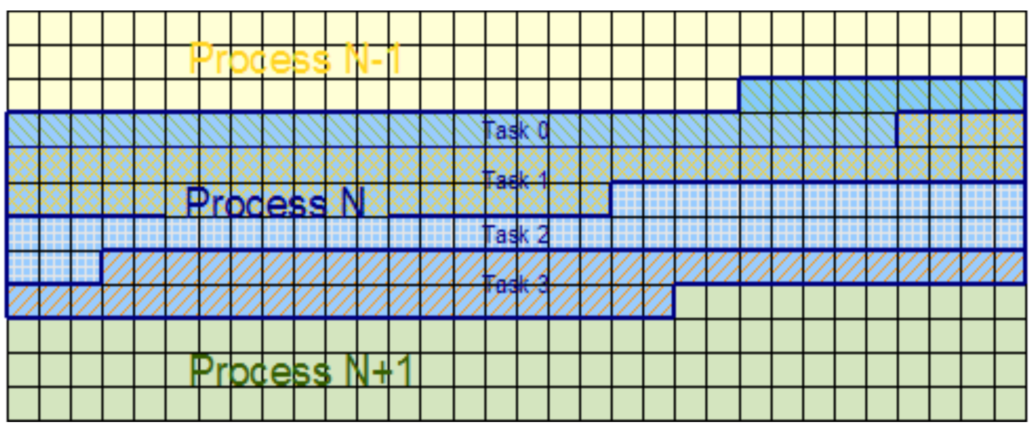
First node (1) => North pole, last node (klon_glo) => South pole.



Parallelism in the physics

- The columns from the global domain are first distributed among the MPI processes.
  - The global domain : klon_glo columns of atmosphere

- The columns of each MPI domain are assigned to the OpenMP tasks assigned to the that process :
  - In each MPI domain : klon_mpi columns : Σ klon_mpi = klon_glo
  - In each OpenMP domain : klon_omp columns : Σ klon_omp = klon_mpi



Code organization and management

It's all about making everyone's life easier
(i.e. being efficiently lazy):

•Using the svn (subversion) tool to maintain codes and share updates.

•Enforcing (as much as possible) the separation between dynamics and physics in models.

•Sharing input/output library/routines across models and post-processing tools

•Sharing dynamical cores between models (at code level, not just in principle).

Code organization and management

Illustrative example

- Codes are in the same svn repository, in separate directories:
  DOC, LMDZ.GENERIC, LMDZ.MARS, LMDZ.VENUS,
  LMDZ.TITAN, LMDZ.COMMON

- Source code is split in directories, e.g. for LMDZ.GENERIC:
  dyn3d , dynlonlat_phylonlat, filtrez, grid, misc, phystd,
  Where the dynlonlat_phylonlat directory contains the interface
  between dynamics and physics (1D dynamics in phy***)

- LMDZ.COMMON stores « only » dynamical cores:
  dyn3d, dyn3d_common, dyn3dpar, dynlonlat_phylonlat, filtrez,
  grid, misc, phymars, phystd, phytitan, phyvenus
  Where phy*** are just <u>links</u> to LMDZ.*** directories

- Again, the idea is to set up an infrastructure that makes it easy to use a given physics package with a (i.e. any) given dynamical core.

Upcoming improvements

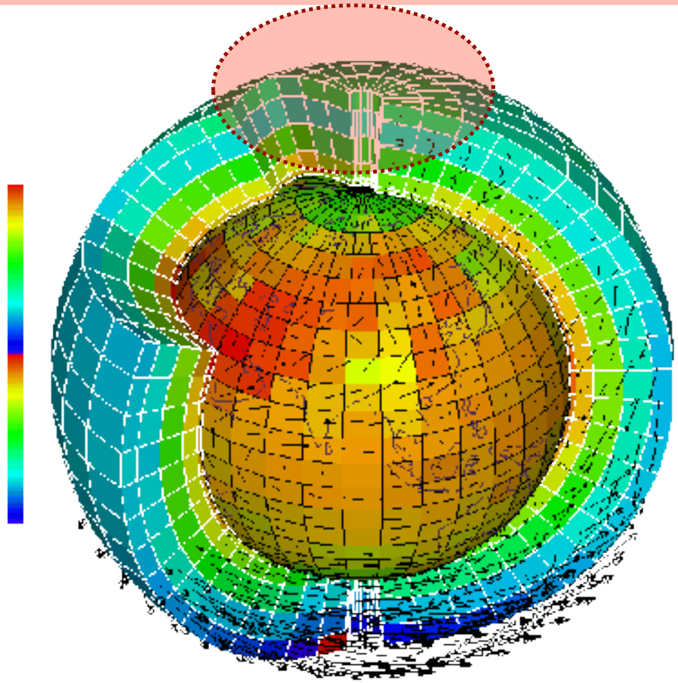
•A new dynamical core (Dynamico).

•Revisiting the equations we solve (i.e. extention to deep atmospheres, going non-hydrostatic, …), and how we solve them.

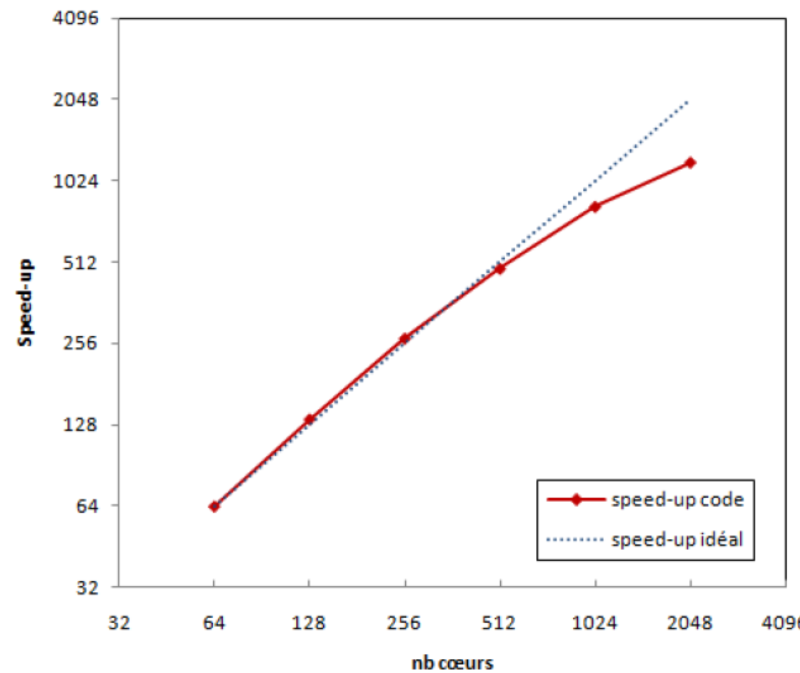
•Being able to run on multicore (>10000 cores) supercomputers.

=> All that, thanks to very hard work by T. Dubos (LMD), M. Tort (LMD), Y. Meurdesoif (LSCE), and co-workers.

The pole problem : FFT filters around the pole for stability
=> global dependency (Williamson, 2007)

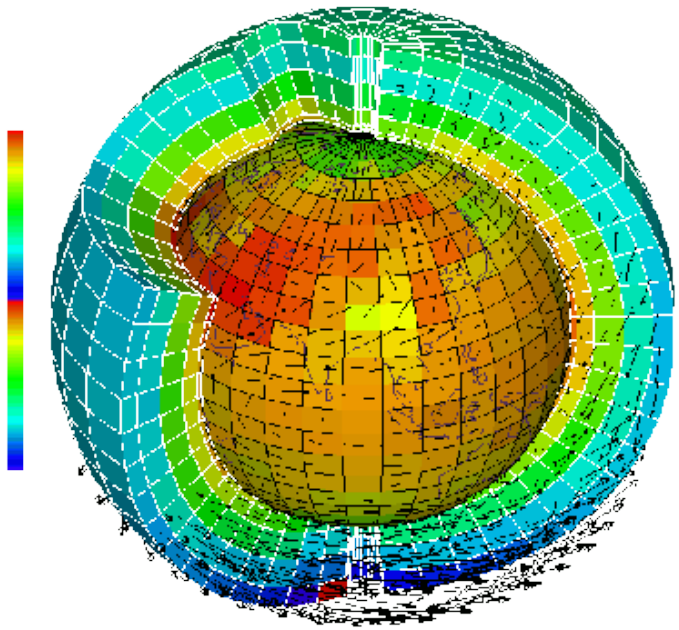


LMD-Z lon-lat core

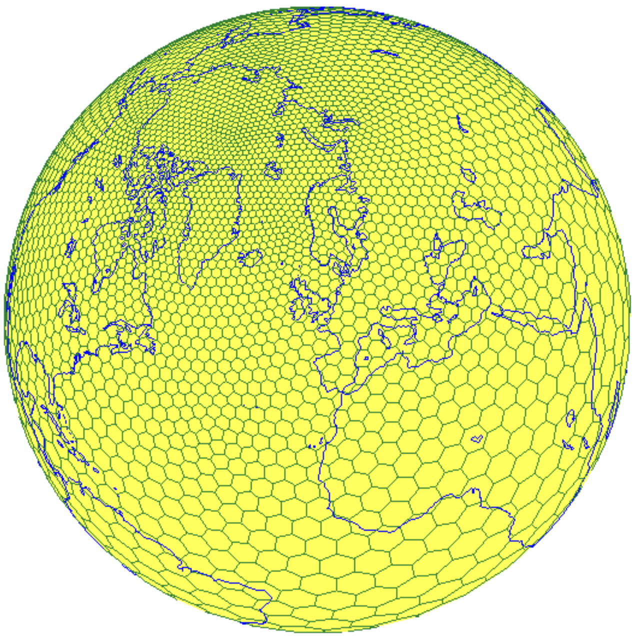


Moreover it is quite unnecessary to compute the physics on the small meshes in the polar regions.

Y. Meurdesoif (2010, 1/4 degree)



LMDZ lon-lat core



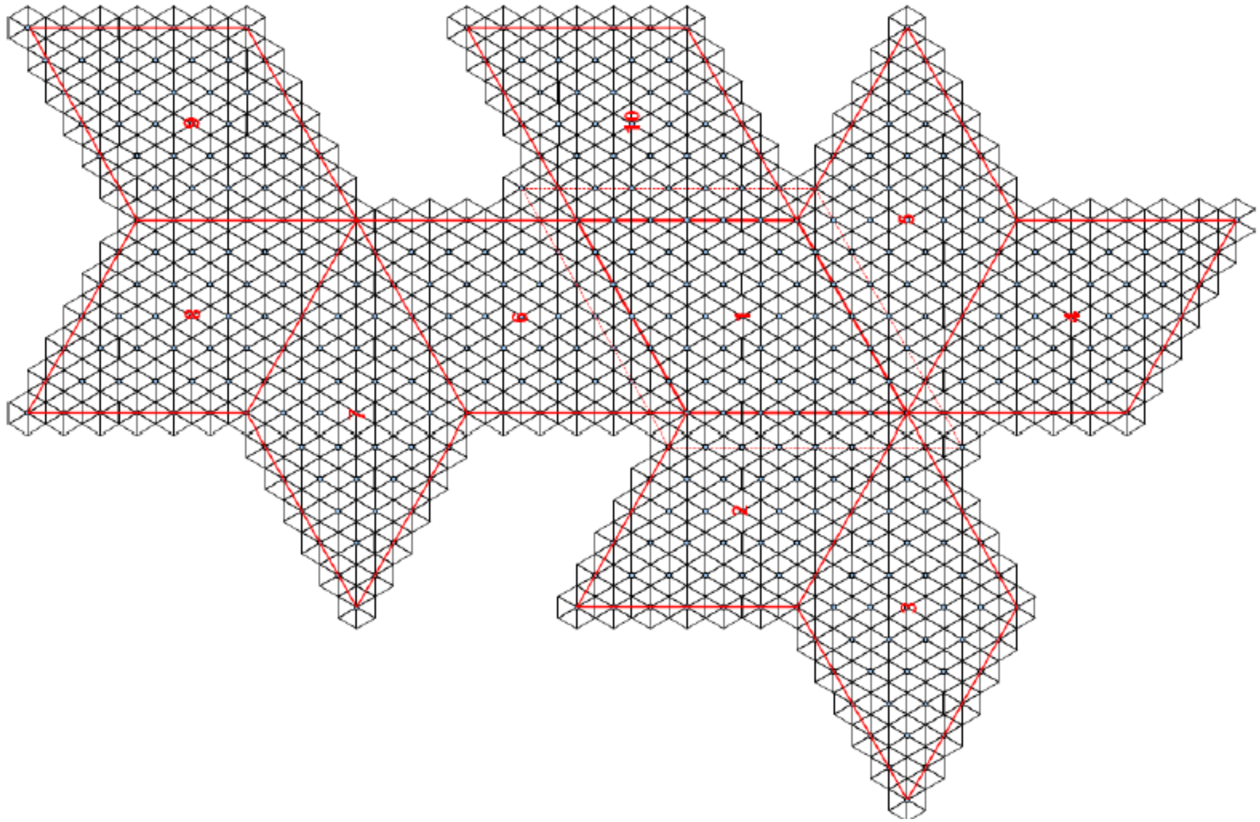
Dynamico icosaedral core

Enstrophy-conserving finite differences on lon-lat mesh (Sadourny, 1975)

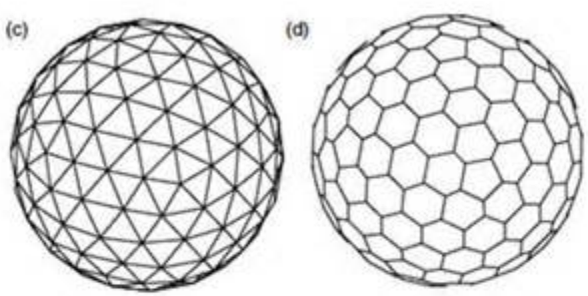
Positive definite finite-volume transport (Hourdin & Armengaud, 1999)

?

The icosahedral grid

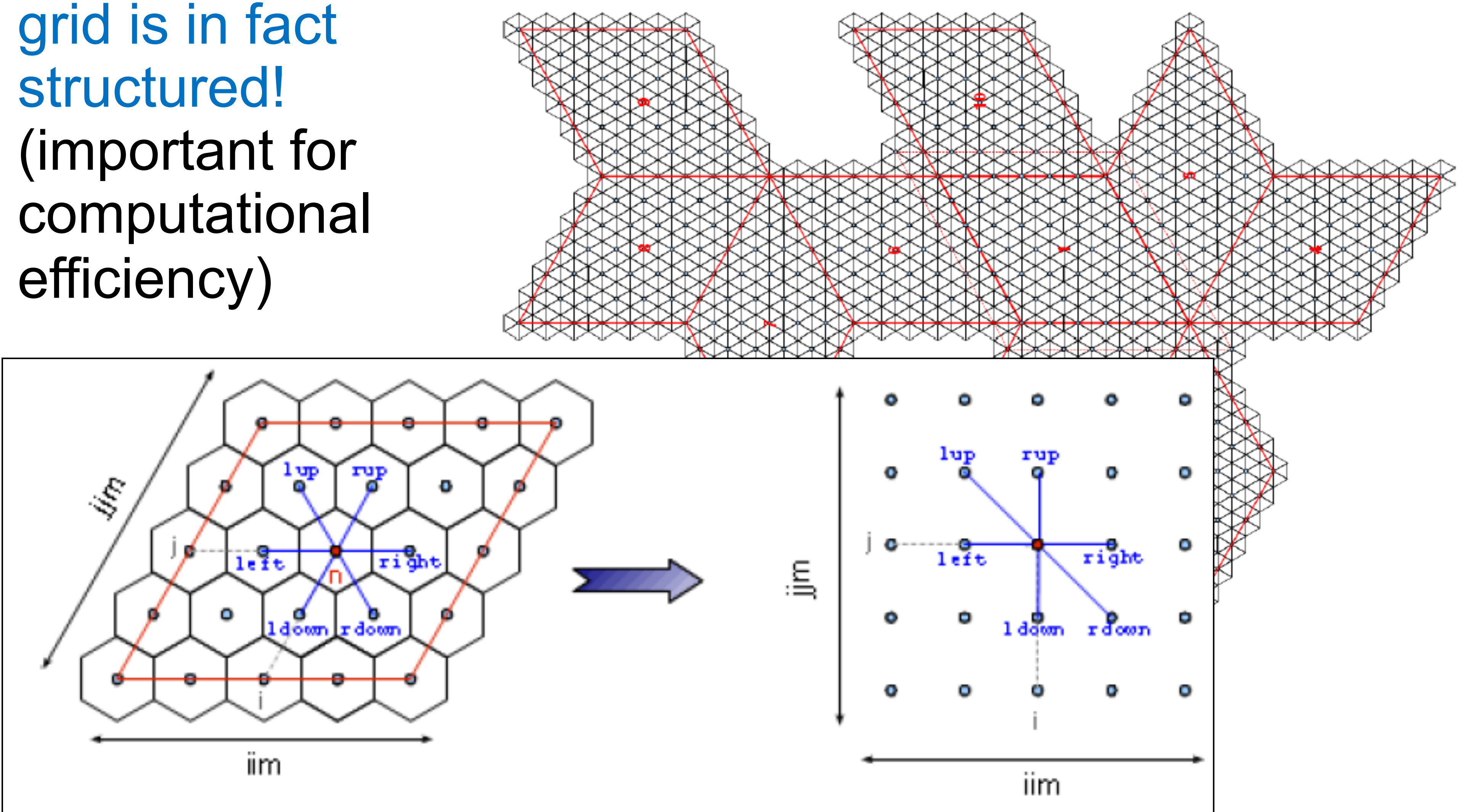


Subdividing the 20 faces yield a triangular or (almost) hexagonal mesh.



The icosahedral grid

The icosahedral grid is in fact structured!
(important for computational efficiency)



Hamiltonian formulation in generalized vertical coordinates
(Dubos & Tort, MWR 2014)

$$\mu\partial_\eta \frac{\delta \mathcal{H}}{\delta \mu} = (\partial_\eta v_i)\frac{\delta \mathcal{H}}{\delta v_i} - \partial_i\left(v_3 \frac{\delta \mathcal{H}}{\delta v_i}\right) + (\partial_\eta \Phi)\frac{\delta \mathcal{H}}{\delta \Phi} - V_3\partial_\eta \frac{\delta \mathcal{H}}{\delta V_3} - \Theta\partial_\eta \frac{\delta \mathcal{H}}{\delta \Theta}$$

$$\begin{aligned}
\partial_t \mu + \partial_i \frac{\delta \mathcal{H}}{\delta v_i} + \partial_\eta(\mu\dot{\eta}) &= 0, \\
\partial_t \Theta + \partial_i\left(\theta\frac{\delta \mathcal{H}}{\delta v_i}\right) + \partial_\eta(\Theta\dot{\eta}) &= 0, \\
\partial_t v_i + (\partial_\eta v_i - \partial_i v_3)\dot{\eta} + \frac{\partial_j v_i - \partial_i v_j}{\mu}\frac{\delta \mathcal{H}}{\delta v_j} + \partial_i\left(\frac{\delta \mathcal{H}}{\delta \mu} + \dot{\eta} v_3\right) + \theta\partial_i\left(\frac{\delta \mathcal{H}}{\delta \Theta}\right) &= 0, \\
\partial_t V_3 + \partial_\eta(V_3\dot{\eta}) + \frac{\delta \mathcal{H}}{\delta \Phi} &= 0, \\
\partial_t \Phi + \dot{\eta}\partial_\eta \Phi - \frac{\delta \mathcal{H}}{\delta V_3} &= 0.
\end{aligned}$$

Integration by parts
+ invariance w.r.t. vertical remapping
=> conservation of energy

Isentropic / Isopycnal	*Mass-based*	*z-based*
$\dot{\eta} = 0$	Diagnosed from horizontal mass flux	$\partial_t \Phi = 0$

Hamiltonian formulation in generalized vertical coordinates

(Dubos & Tort, MWR 2014)

$$\mu\partial_\eta \frac{\delta\mathcal{H}}{\delta\mu} = (\partial_\eta v_i)\frac{\delta\mathcal{H}}{\delta v_i} - \partial_i\left(v_3\frac{\delta\mathcal{H}}{\delta v_i}\right) + (\partial_\eta\Phi)\frac{\delta\mathcal{H}}{\delta\Phi} - V_3\partial_\eta\frac{\delta\mathcal{H}}{\delta V_3} - \Theta\partial_\eta\frac{\delta\mathcal{H}}{\delta\Theta}$$

$$\begin{aligned}
\partial_t\mu + \partial_i\frac{\delta\mathcal{H}}{\delta v_i} + \partial_\eta(\mu\dot\eta) &= 0,\\
\partial_t\Theta + \partial_i\left(\theta\frac{\delta\mathcal{H}}{\delta v_i}\right) + \partial_\eta(\Theta\dot\eta) &= 0,\\
\partial_t v_i + (\partial_\eta v_i - \cancel{\partial_i v_3})\,\dot\eta + \frac{\partial_j v_i - \partial_i v_j}{\mu}\frac{\delta\mathcal{H}}{\delta v_j} + \partial_i\left(\frac{\delta\mathcal{H}}{\delta\mu} + \cancel{\dot\eta v_3}\right) + \theta\partial_i\left(\frac{\delta\mathcal{H}}{\delta\Theta}\right) &= 0,\\
\cancel{\partial_t V_3} + \partial_\eta(\cancel{V_3\dot\eta}) + \frac{\delta\mathcal{H}}{\delta\Phi} &= 0,\\
\partial_t\Phi + \dot\eta\partial_\eta\Phi - \cancel{\frac{\delta\mathcal{H}}{\delta V_3}} &= 0.
\end{aligned}$$

Hydrostatic $\boxed{v_3 = \dfrac{\partial L}{\partial u^3} = 0}$

Isentropic / Isopycnal

$\dot\eta = 0$

Mass-based

Diagnosed from horizontal mass flux

z-based

$\partial_t\Phi = 0$

density, entropy, velocity

Least action principle
(Tort & Dubos, 2014)

$$\delta \int L(\mathbf{x}, \rho, s, \mathbf{u})\mathrm{d}m\mathrm{d}t = 0$$

L=kinetic+Coriolis
-internal-potential

$$(x, y, z) \to (\xi^1, \xi^2, \xi^3) \qquad \mathbf{u}$$

$$u^i = D\xi^i / Dt$$

contravariant relative velocity

$$D/Dt = \partial_t + u^i \partial_i$$

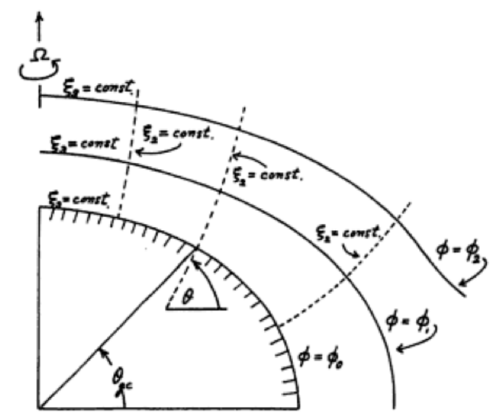
$$v_i = \partial L / \partial u^i$$

covariant absolute velocity

$$v_i = G_{ij} u^j + R_j$$

Euler-Lagrange equations
Tort & Dubos, JAS 2014

$$\frac{Dv_i}{Dt} + \frac{\partial L}{\partial \xi^i} = \frac{1}{\mu} \partial_i \left(\mu^2 \frac{\partial L}{\partial \mu} \right)$$



Surfaces of constant ξ_2 and constant ξ_3

Hamiltonian formulation
Dubos & Tort, MWR 2014

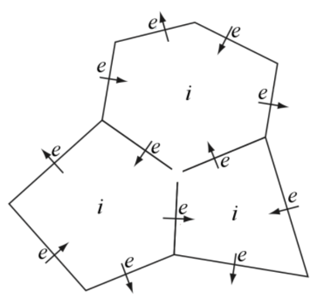
Curl form

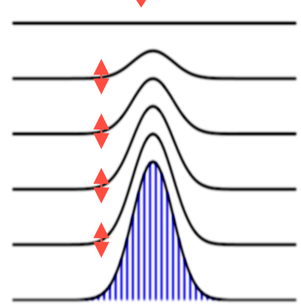
$$\frac{\partial v_i}{\partial t} = \dots$$

Flux form

$$\frac{\partial}{\partial t}(\mu v_i) = \dots$$

Discretize !
TRiSK alg.

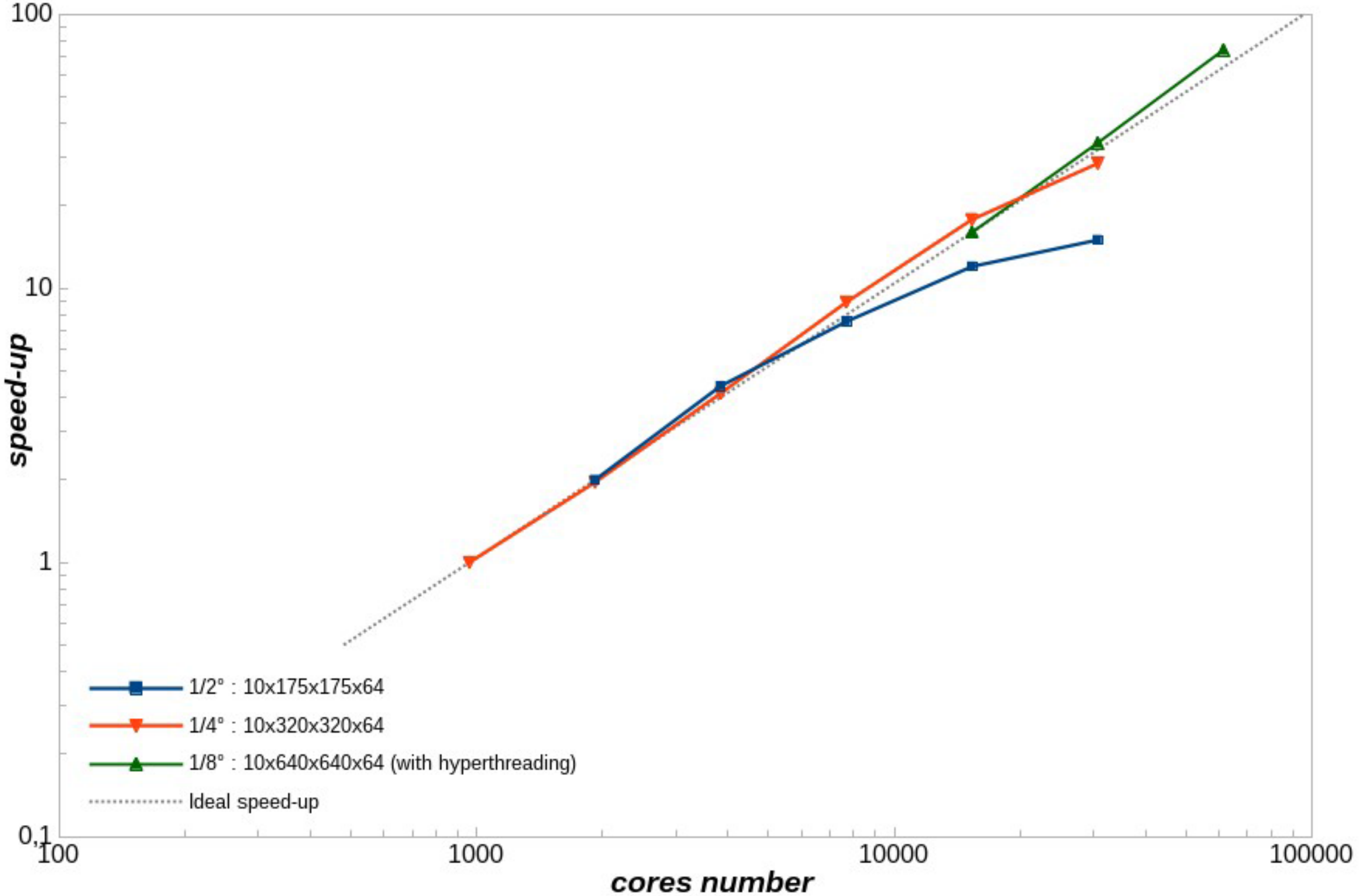




Eulerian coordinates in planetary frame

Generalized vertical coordinate

Scaling on massively multicores



- Very efficient scaling (tested up to 30000 cores) using Dynamico/Saturn prototype.

Miscellaneous concluding thoughts

- Still some work to do to finalize in practice the physics/dynamics separation. But with the arrival of a new dynamical core (and also with enabling using WRF mesoscale dynamics), it is mandatory to do it fully and well.

- Likewise there is a new inputs/outputs library under development at IPSL, XIOS, which is very efficient even in multicore environments. We plan to integrate it shortly.

- We already know that we need to work further to unify our post processing tools. This is a target for the near future.

- We also need to work on documentation...