New computing architectures :

an opportunity to move towards more composable models ?

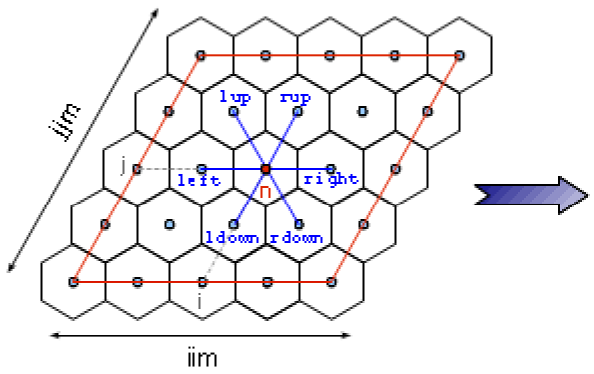
Thomas Dubos

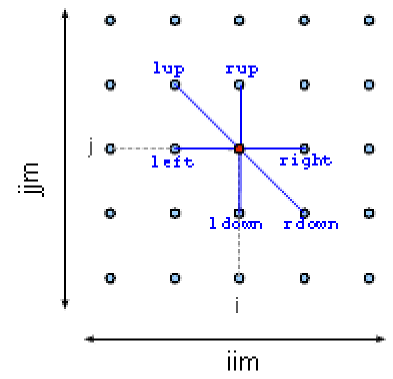
LMD / IPSL – Ecole Polytechnique / IP Paris

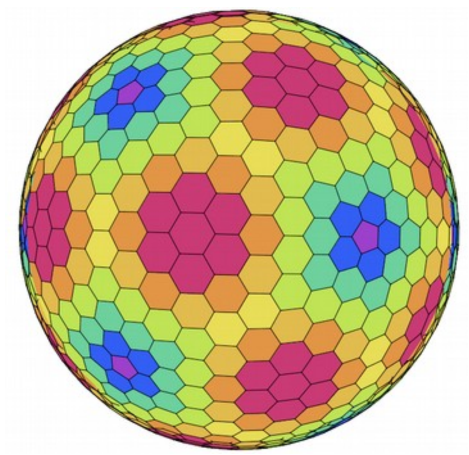
- Steps towards GPU computing with DYNAMICO and LMDZ
  - DYNAMICO
  - Simple dry physics
  - LMDZ

- From refactoring to composability
  - Low-hanging fruit
  - Higher-hanging fruits

DYNAMICO (Dynamical Core on Icosahedron)



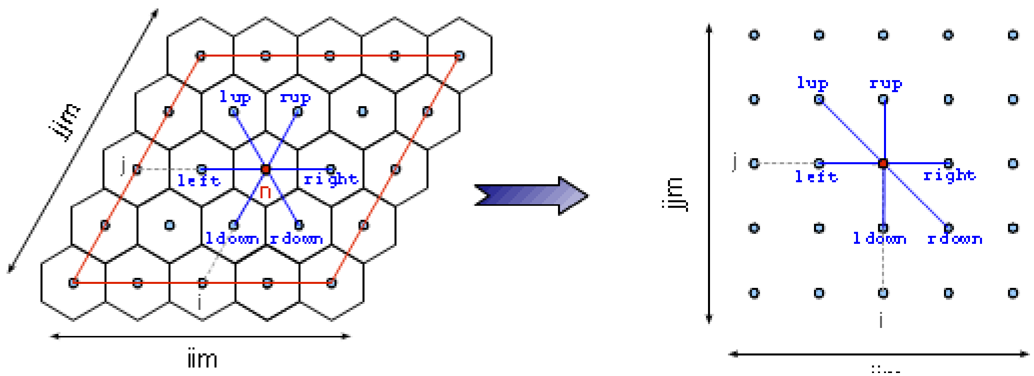




- **Direct access to neighbours via constant offsets**
- No special case for pentagons (handled by metrics)
- Vertical direction in outer loops

```
DO ij=ij_begin,ij_end
   ! convm = -div(mass flux), sign convention as in Ringler et al. 2012, eq. 21
   convm(ij,l)= -1./Ai(ij)*(ne_right*hflux(ij+u_right,l)   +  &
        ne_rup*hflux(ij+u_rup,l)        +  &
        ne_lup*hflux(ij+u_lup,l)        +  &
        ne_left*hflux(ij+u_left,l)      +  &
        ne_ldown*hflux(ij+u_ldown,l)    +  &
        ne_rdown*hflux(ij+u_rdown,l))
   ! dtheta_rhodz =  -div(flux.theta)
   dtheta_rhodz(ij,l)=-1./Ai(ij)*(ne_right*Ftheta(ij+u_right)    +  &
        ne_rup*Ftheta(ij+u_rup)         +  &
        ne_lup*Ftheta(ij+u_lup)         +  &
        ne_left*Ftheta(ij+u_left)       +  &
        ne_ldown*Ftheta(ij+u_ldown)     +  &
        ne_rdown*Ftheta(ij+u_rdown))
ENDDO
```

DYNAMICO (Dynamical Core on Icosahedron)




```
!$acc parallel loop collapse(2)
DO l = ll_begin, ll_end
  DO ij = ij_begin_ext,ij_end_ext
    uu_right = 0.5*(rhodz(ij,l)+rhodz(ij+t_right,l))*u(ij+u_right,l)
    uu_right = uu_right*le_de(ij+u_right)
    hflux(ij+u_right,l) = uu_right
    uu_lup = 0.5*(rhodz(ij,l)+rhodz(ij+t_lup,l))*u(ij+u_lup,l)
    uu_lup = uu_lup *le_de(ij+u_lup)
    hflux(ij+u_lup,l) = uu_lup
    uu_ldown = 0.5*(rhodz(ij,l)+rhodz(ij+t_ldown,l))*u(ij+u_ldown,l)
    uu_ldown = uu_ldown*le_de(ij+u_ldown)
    hflux(ij+u_ldown,l) = uu_ldown
  END DO
END DO
```

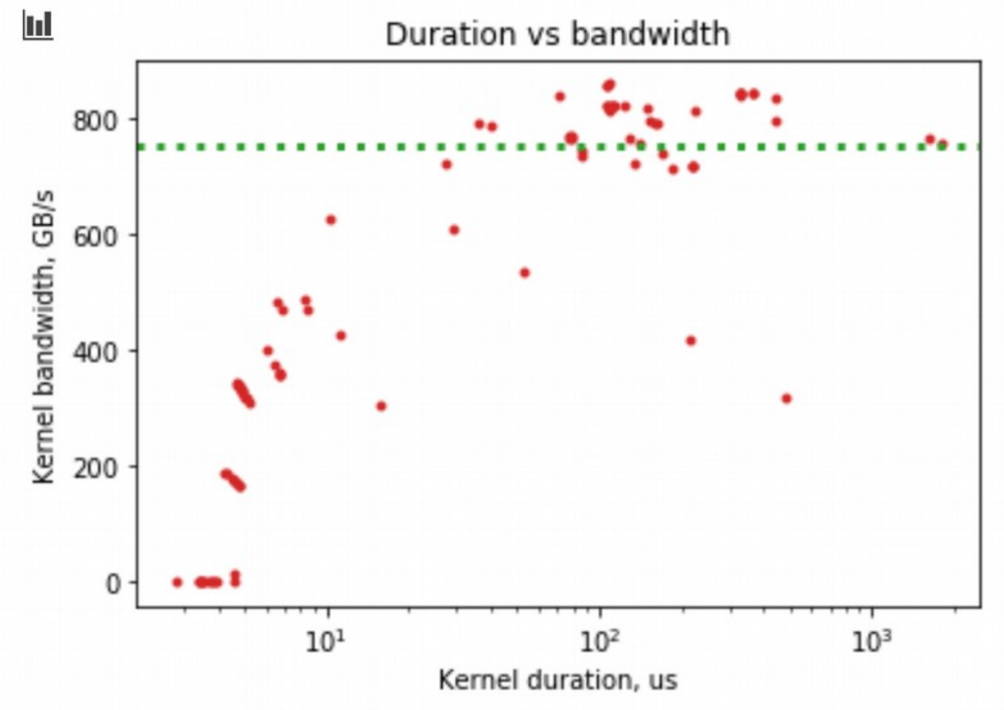
- Manual GPU port via **OpenACC directives**

Y. Meurdesoif + IDRIS/HPE

		1°	1/2°	1/4°
	nbp=40	nbp=80	nbp=160	nbp=320
1 Nœud CPU (40 procs MPI)	67,7 s	282,2 s	1193,0 s	5102,5 s
1 GPU (1 proc MPI)	26,6 s (2,5)	77,3 s (3,6)	298,8 s (4,0)	N/A
2 GPUs (2 procs MPI)	14,7 s (4,6)	40,3 s (7,0)	151,6 s (7,9)	648,5 s (7,9)
4 GPUs (4 procs MPI)	11,5 s (5,9)	23,9 s (11,8)	74,9 s (15,9)	304,3 s (16,8)

Simplified dry physics

- From PhD thesis of F. Hourdin
    - SW radiation : weak absorption
    - LW radiation : short absorption
    - Down-gradient turbulent fluxes
    - Bulk formulae
    - Heat diffusion in 11-layer soil
    - Dry ajustment
    - Implicit time stepping for turbulence with coupling to surface and radiation
- **99 % science** : non-scientific tasks (MPI, I/O, namelists, …) outsourced to host model via F2003 function pointers (callbacks/plugins)
- 3000 LOC
- Interfaced with LMDZ and DYNAMICO
- Manual port to OpenACC during 2021 Hackathon at IDRIS



	phys		
ngrid (per GPU)	CPU ms/step	GPU ms/step	speedup
4000	15,65	2,70	5,80
16000	69,00	7,89	8,74
64000	314,83	26,52	11,87
256000	1 709,30	100,54	17,00

L. Fairhead, E. Millour + … + IDRIS / Nvidia

LMDZ physics

DYNAMICO

- 100 % in-house
- ~20 kernels : purely computational, well-defined inputs and outputs
- ~3000 LOC to port
- Very regular computation and memory access
- Few, long-term developers

LMDZ physics

- In-house code + imported code (ECRad)
- No systematic separation between computational and non-computational tasks
- Inputs and outputs may be arguments or in modules
- 150 000+ LOC but how many to port ?
- Computation and memory access may be irregular (convection)
- Many developers, few long-time
- Community code serving to experiment new modelling ideas (paramerizations)

Current plan to GPU-enable LMDZ (started in 2022) = two-step approach

- Refactoring (mostly by domain scientists)
  - Regard sub-sets of routines pertaining to one parameterization as ultimately autonomous
  - Separate init, compute, diagnostics, I/O …
  - Clarify inputs and outputs of computational routines
- Manual (current) or automatic (future?) insertion of directves (mostly by comp. Sci.)

Lesson : a key effort towards exascale is to isolate the computational parts of the code and refactor them into a sufficiently simple and regular style, making manual or automatic insertion of directives doable.

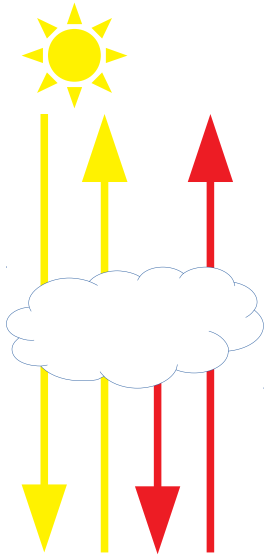
How much more effort would it require to make our models truly composable ?

- *Composed of modules which exist in several « implementations » (equivalent or not)*
- *Because minimal inputs/outputs and hypotheses have been clearly defined (interface/contract)*
- *While maintaining a well-defined notion of internal consistency*

Composability would allow/facilitate :

- Explore **« what if » worlds**
- **Relax implicit/explicit limitations**
- Switch between different parameterizations of the same process => **structural uncertainity**

Composability : radiative transfer



Radiative heating rate

Radiative energy flux

$$\rho c_p \delta T = \delta Q = -\frac{\partial F_{rad}}{\partial z}$$

- Many atmospheric models include an externally-developed radiative transfer code (e.g. RRTM, EcRad)
- Possible because of ***consensus*** or ***de facto standard*** on inputs (profile of temperature & pressure, cloudiness …) and outputs (radiative fluxes)
- Especially, deciding that outputs are ***fluxes*** ensures conservation of energy

Towards composability : thermodynamics

Systematic approach to thermodynamic consistency (Ooyama, 1990 ; Bannon, 2003) : thermodynamic functions derive from a single ***thermodynamic potential,*** function of **canonical state variables**

Example : dry air as ideal perfect gas

Enthalpy $$\mathrm{d}H = T\mathrm{d}S + V\mathrm{d}p$$

$$h(p,s) = h_{00} - C_p T_0 \left(\frac{p}{p_0}\right)^{R/C_p} \exp\frac{s-s_0}{C_p}$$

$$T(p,s) = \frac{\partial h}{\partial s} = T_0 \left(\frac{p}{p_0}\right)^{R/C_p} \exp\frac{s-s_0}{C_p}$$

$$\mathrm{d}h = C_p \mathrm{d}T$$

$$\theta = T(p_0,s) = T_0 \exp\frac{s-s_0}{C_p}$$

$$\theta = T\left(\frac{p}{p_0}\right)^{R/C_p}$$

$$\alpha(p,s) = \frac{\partial h}{\partial p} = \frac{R}{C_p}\frac{h-(h_0-C_pT_0)}{p}$$

$$p = \rho R T$$

Towards composability : thermodynamics

- **Dynamics** do not explicitly care about the equation of state or even which conservative variable is used. All that it needs is a few thermodynamic functions :

$$h(p,\theta,q) \qquad \rho^{-1} = \partial h/\partial p \qquad \pi = \partial h/\partial\theta \qquad \mu = \partial h/\partial q$$

$$\frac{1}{\rho}\nabla p = \nabla\left(h - \theta\pi - \mu q\right) + \theta\nabla\pi + q\nabla\mu$$

Considering these relationships as provided by a « plugin » module would allow :

- Departures from the ideal perfect gas (Lebonnois, 2010)
- Relax hard-coded restrictions, e.g. temperature-independent latent heats
- « What if » worlds : what if water vapor had the same molar mass as dry air (Yang et al. 2021)

Towards composability : thermodynamics

- Similarly with common inputs and outputs of **turbulent closures :**

$$N^2 = \rho g \left(\frac{\partial^2 h}{\partial p \partial \theta} \frac{\partial \theta}{\partial z} + \frac{\partial^2 h}{\partial p \partial q} \frac{\partial q}{\partial z} \right) \Rightarrow \; Ri_g$$

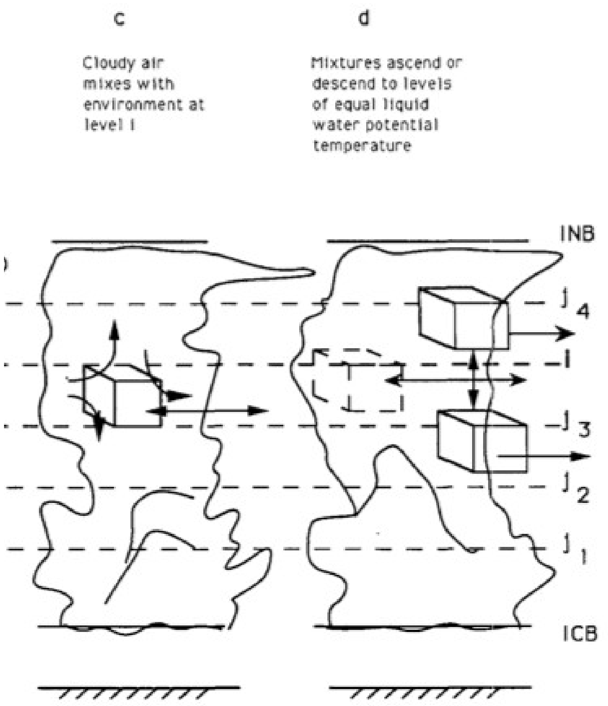
$$\overline{b'w'} \simeq \rho g \left(\frac{\partial^2 h}{\partial p \partial \theta} \overline{\theta' w'} + \frac{\partial^2 h}{\partial p \partial q} \overline{q' w'} \right)$$

Ocean models use the Thermodynamic Equation of Seawater (TEOS-10, Feistel 2008).

How about Thermodynamic Equation(s) for (Moist) Air ?

Towards composability : convection ?

- Parameterization of convection (shallow/deep) is notoriously difficult
- Many different approaches
- Coupled to many processes : microphysics, radiation …
- Non-local, possibly stochastic, …
- Affects momentum, temperature, moisture but also all tracers



Emanuel, 1991

Towards composability : convection ?

- Many different approaches, but also some similarities (i.e. mass-flux schemes)
- No obvious unifying structure
  - profiles of entrainment/detrainment
  - transilience matrix => conservation of mass

$$\delta\rho(z) = \int \left(\delta\rho(z' \to z) - \delta\rho(z \to z')\right) \mathrm{d}z'$$

$$\Rightarrow \quad \delta\left(q(z)\rho(z)\right) = \int \left(q(z')\delta\rho(z' \to z) - q(z)\delta\rho(z \to z')\right) \mathrm{d}z'$$

↑ *kg stuff / kg air*

New computing architectures :

an opportunity to move towards more composable models ?

- Preparing legacy codes for exascale requires significant refactoring
- *Not* only computational science : extensive refactoring requires understanding of physical contents
- The goal of combining physics-based and ML-based components also creates a strong incentive for modularity / composability
- Opinion : do not stop at minimal effort, push for composability

- Composability is not a new idea ; however it requires a clarification of the physical constraints / hypotheses that restrict or not the « decoupling » of internal blocks
- Dynamics : picture is quite clear now, at least in theory
- Physics :
    - little fundamental work on such questions (Polcher et al, 1998, Catry et al., 2007), no consensus
    - low-hanging fruits : thermodynamics, turbulence ?
    - appetite to address hard problems (convection, …) ?