Time stepping schemes for atmospheric modelling

Numerical methods for weather prediction training course
April 2026

Michail Diamantakis
michail.diamantakis@ecmwf.int

Outline of this lecture

- Equation formulations and time stepping
- Conservative flux-form versus non-conservative formulations
- Eulerian versus semi-Lagrangian
- Time stepping schemes commonly used by different weather and climate models

Two commonly used equation formulations in operational NWP models

Hydrostatic approximation

- Atmosphere is approximately in hydrostatic equilibrium
- In hydrostatic vertical motion is diagnosed from continuity equation
- Filters the very fast sound waves ⇒ no stability problems associated with very large acoustic CFL numbers in the vertical

Non-hydrostatic (NH) equation model

- Most accurate description of the atmosphere.
- More expensive equation set with often more complex and computationally demanding numerical algorithms: better use when needed i.e. high resolutions where dynamics begin to resolve convection explicitly

What motions time-stepping should resolve?

- **Rossby waves** must be **accurately resolved and transported** – essential **for good weather predictions**
- **Accurate gravity wave** representation is **essential for realistic momentum transport and large-scale flow evolution**
- **Fast acoustic waves** carry little energy - **not important for weather** but their implications must be considered: **they may limit severely timestep of numerical schemes**
  - ☐ To avoid such timestep restrictions, ideally an **unconditionally stable numerical scheme** is needed which may dissipate acoustic waves but does not dissipate other meteorologically important waves
- Using very high-order time-stepping schemes is not very practical:
  - ☐ Uncertainties from model components (e.g. parametrizations) tend to dominate and mask any gains in temporal accuracy

Scalability: an important requirement

- **Moore's law** (CPU performance doubling every ~ 18 months) **no longer holds** but advances in HPC architectures continue to improve time-to-solution on modern supercomputers
- **NWP solvers must scale well on exascale machines** and **run efficiently on heterogenous architectures** with accelerators (GPU-CPU)
- **Grids**: regular **lat/lon** are **not suited for high resolution global modelling**:
  - **Explicit time-stepping**: meridian convergence at poles ⇒ extremely high resolution => **CFL limit results in tiny time-step and high cost**
  - Grid anisotropy near poles may lead to **poor convergence** of elliptic solvers in implicit solvers + **high communication cost**
- **Global spectral transform models at high resolution do not scale well mainly due to the high communication cost of transpositions**

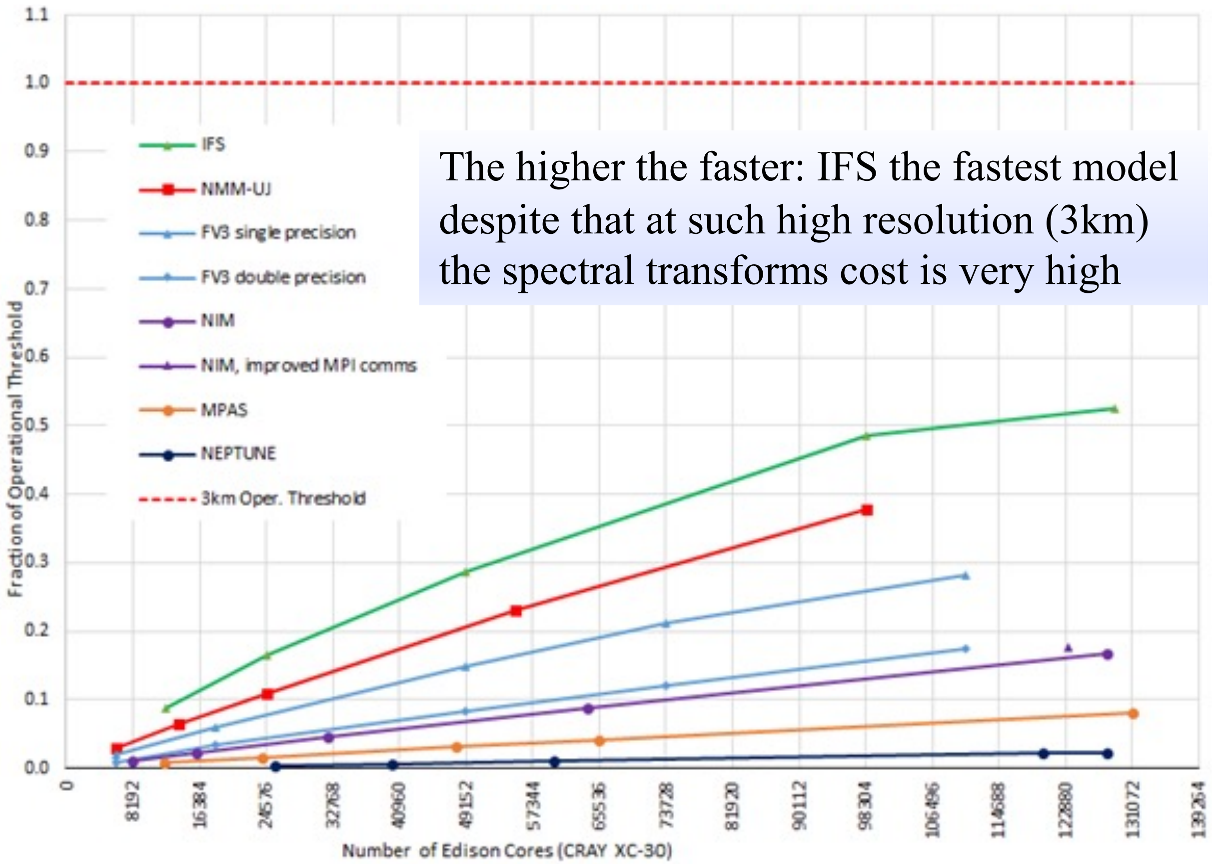
Common time stepping schemes in NWP

- Schemes currently used in atmospheric modelling
  - Semi-Lagrangian, semi-implicit: unconditionally stable ⇨ large timesteps used for efficiency (they are not designed for conservation)
  - Eulerian - they can conserve but they are conditionally stable:
    - ➔ Flux-form explicit Eulerian transport with semi-implicit time-stepping for fast forcing term integration
    - ➔ Split-explicit Eulerian time schemes
    - ➔ HEVI: Horizontally Explicit / Vertically Implicit
    - ➔ IMEX: implicit-explicit Runge-Kutta schemes

SISL method and efficiency

Semi-Lagrangian, semi-implicit time integration is still very efficient: IFS is one of the fastest (the fastest?) global models

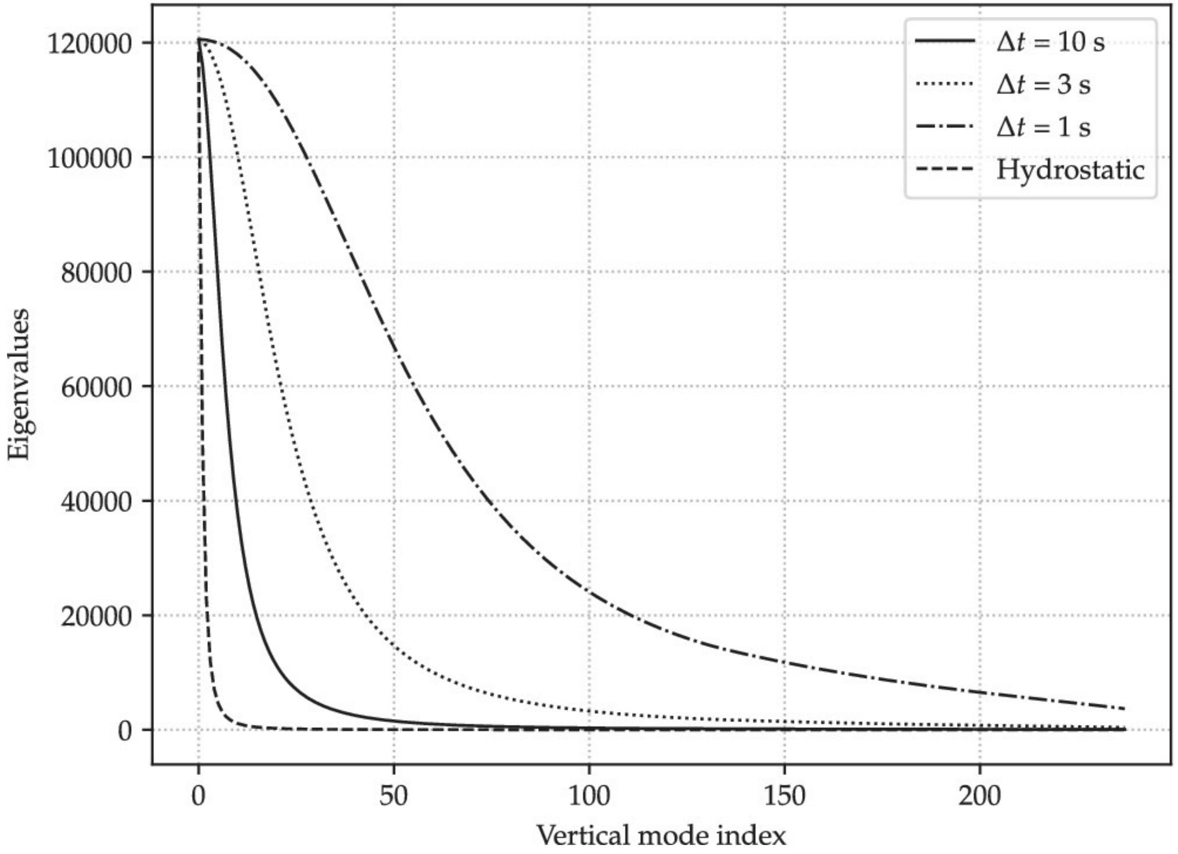
Model	Resolution / config	Time step	SYPD	Nodes	Source
IFS	~5 km global (**hydrostatic**)	~240 s	~1.0	$O(10^3-10^4)$	Neumann et al. (2019)
ICON	~5 km global (**NH**)	~45 s	~0.15–0.2	$O(10^3–10^4)$	Neumann et al. (2019)



The higher the faster: IFS the fastest model despite that at such high resolution (3km) the spectral transforms cost is very high

- *An example of IFS computational performance at approximately 3km resolution on a dry baroclinic wave case with tracers, adapted from Michalakes et al, NGGPS AVEC report, **2015***
- *Different “candidate US global models” were compared to IFS*

A criticism in using very long time steps in non-hydrostatic dynamics



Long time steps inhibit NH models to reach their potential in accurately representing vertical gravity wave propagation

Figure and conclusions from Burgot, Auger, Benard, QJRMS 2021 (100m resolution)

"improvements of the vertical wave propagation (especially gravity waves) sought during the implementation of an NH model in favour of an H model, are fully satisfied when the time step is small"

Pros and Cons of Eulerian methods

Conservation properties a key strength:

- **Local and global mass conservation in a flux-form model with Eulerian time stepping is ensured by using finite-volume or conservative high-order spatial discretizations**

Computational efficiency:

- In **explicit Eulerian** conservative transport schemes the **CFL must be < 1** for stability (unlike SISL which can run at CFL>1)
- **Fully explicit methods**: highly scalable, simple to implement but very short timestep for stability
- **Combination** of an **explicit** scheme for advection with **semi-implicit** time stepping for fast processes **allows longer timesteps** than a purely explicit approach. Scalability then depends on elliptic solver type and its implementation (Müller & Scheichl QJRMS 2013)
- Overall because Eulerian schemes use local operators they have much reduced communication costs (and scale better) than spectral SISL schemes

Flux formulation of equations

"Flux" or conservative "form" governs evolution of densities (air + tracer)

$$\phi = \frac{\rho_\phi}{\rho}$$

Density of air, tracer: ρ, ρ_φ, mixing or specific ratio ϕ

Continuity equation (air density)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0$$

Continuity equation (tracer density)

$$\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho\phi \vec{V}) = 0 \Rightarrow \frac{\partial}{\partial t} \int_V \rho\phi dV = - \int_{\partial V} \rho\phi \overrightarrow{V} \cdot \vec{n} dS$$

- Change of mass in volume V = net flux through boundaries ∂V
- Numerical discretizations with the same property can be derived and conserve locally and globally

Non-conservative form using Lagrangian description

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{V} \cdot \nabla$$

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \vec{V}$$

$$\frac{D\phi}{Dt} = 0$$

IFS uses the above form

- Density, tracer mixing ratio are carried along the flow
- Very efficient, facilitates Lagrangian type advection schemes
- Analytically tries to conserve the value of ϕ (ratio) along a trajectory - it doesn't ensure that its mass in a region remains the same
- In addition, numerical schemes are "vulnerable" to interpolation errors
- Cannot track what enters/leaves a grid cell; grid-cells do not exist (just grid-points)
- Conservation is not possible without additional numerical "fixes"

1D Eulerian advection scheme: flux-form vs non-conservative

Advection of a tracer with **density Φ** and **mixing ratio m_ϕ** in flux conservative form:

$$\frac{\partial \Phi}{\partial t} + \frac{\partial (u\Phi)}{\partial x} = 0, \quad \Phi = \rho m_\phi$$

ρ: air density

Finite difference forward in-time discretization (conservative form):

$$\Phi_j^{n+1} = \Phi_j^n + \frac{\Delta t}{\Delta x}\left[F^n_{j+1/2} - F^n_{j-1/2}\right]$$

- For a numerical flux such as $F_j = (u\Phi)_{j-1/2}$ the above scheme is conservative:

  $$\sum_{j=1}^{N} \Phi_j^{n+1}\Delta x = \sum_{j=1}^{N} \Phi_j^n \Delta x$$

  (assuming suitable BCs)

- If the discretization is TVD it remains stable

Equivalent non-conservative form of the advection equation for a tracer with **mixing ratio m_ϕ**:

$$\frac{\partial m_\phi}{\partial t} + u\frac{\partial m_\phi}{\partial x} = 0$$

Finite difference discretization (upwinding):

$$(m_\phi)_j^{n+1} = (m_\phi)_j^n + \frac{\Delta t\, u_j^n}{\Delta x}\left[(m_\phi)_j^n - (m_\phi)_{j-1}^n\right]$$

$\Downarrow$ Assume constant density ρ of background air and compute total mass at two consecutive timesteps

$$\sum_{j=1}^{N} (m_\phi)_j^{n+1}\rho\Delta x_j \neq \sum_{j=1}^{N} (m_\phi)_j^{n}\rho\Delta x_j$$

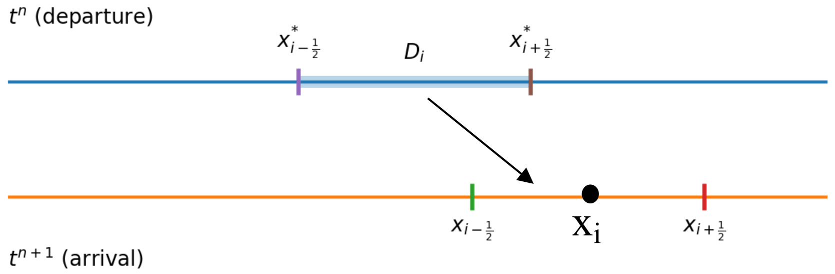
if resolution or velocity varies

ECMWF

Flux-Form semi-Lagrangian (FFSL) methods

- FFSL = SL trajectories + finite-volume conservation: applies SL trajectories to the flux-form conservation law (large time steps, stability + mass conservation)
- Sketch of an FFSL algorithm
  - **Find backtracked cell interfaces** (not departure points) using **SL trajectories**
  - **Reconstruct tracer mass** inside each **grid cell** (continuous sub-grid representation):

    ⇨ **Piecewise Parabolic Method (PPM)** often used (conservative, high-order)
  - **Integrate reconstructed mass over the departure volume** (upstream control volume)
  - **Assign to the arrival cell average** (conservative remapping)



$$\overline{\rho\phi}_i^{\,n+1} = \frac{1}{\Delta x_i}\int_{D_i} (\rho\phi)^n(x)\, dx,$$

where ρ is the air density, ϕ the tracer mixing ratio and $(\rho\phi)(x)$ the reconstructed tracer density

FFSL integral is computed splitting the departure volume into sub-intervals and integrating each local reconstruction

A simple test model for integrating fast processes: 1d gravity wave equations

Linearised shallow water equations:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial \phi}{\partial x} = 0 \\ \dfrac{\partial \phi}{\partial t} + \Phi \dfrac{\partial u}{\partial x} = 0 \end{cases}$$

Fluid mean depth

$$\Phi = gH$$
$$\phi = gh$$

Perturbation from mean depth

$\partial/\partial t$ on first equation and eliminate ϕ to obtain the familiar equation of a 1-dimensional wave:

$$\frac{\partial^2 u}{\partial t^2} - \Phi \frac{\partial^2 u}{\partial x^2} = 0$$

propagating with speed: $c \equiv \dfrac{\omega}{k} = \pm\sqrt{\Phi} = \pm\sqrt{gH}$

Explicit Leapfrog time stepping on 1D GW eqn

Three-time-level explicit Leapfrog scheme

Leapfrog on a general problem: $\frac{d\psi}{dt} = f(\psi) \Rightarrow \psi^{n+1} = \psi^{n-1} + 2\Delta t\ f(\psi^n)$

Leapfrog on 1D GW equations:

$$\begin{cases} u_j^{n+1} = u_j^{n-1} - 2\Delta t \dfrac{\phi_{j+1}^n - \phi_{j-1}^n}{2\Delta x} \\ \phi_j^{n+1} = \phi_j^{n-1} - 2\Delta t\ \Phi \dfrac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \end{cases}$$

Note Paul Williams (U of Reading) & associates work on high order leapfrog and high order filters for leapfrog

Neutral (no damping) + 2nd order BUT phase + dispersion errors + computational mode

Von Neuman stability: $\Delta t \le \frac{\Delta x}{\sqrt{\Phi}} \approx \frac{\Delta x}{300}$

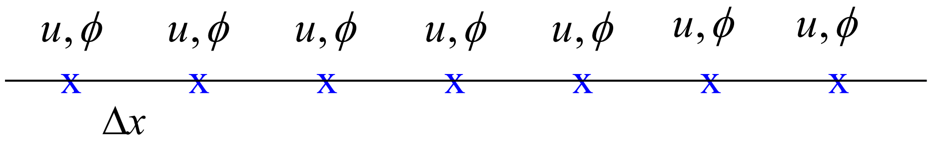
Solution is a combination of a physical and a computational mode which can be damped by use of a time filter, e.g. Asselin filter (but damps energy in long integrations)

$$\psi^n \leftarrow \psi^n + \gamma\left(\psi^{n-1} - 2\psi^n + \psi^{n+1}\right),\ \gamma > 0,\ \psi = u,\varphi$$

Typical value for global models: $\gamma = 0.06$

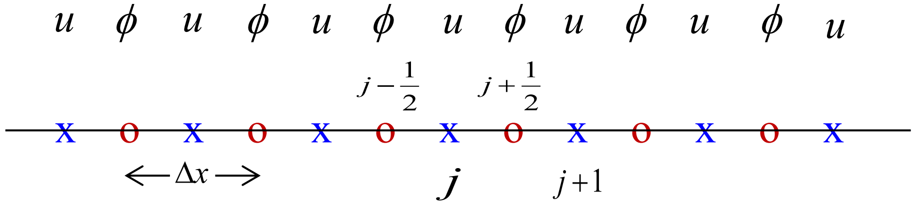
Staggering variables to improve accuracy

- The prognostic variables can be
  - On the same location on the grid, i.e. co-located



$$\begin{cases}\Delta_t u_j + \dfrac{\phi_{j+1}-\phi_{j-1}}{2\Delta x} = 0\\[2ex] \Delta_t \phi_j + \Phi\dfrac{u_{j+1}-u_{j-1}}{2\Delta x} = 0\end{cases}$$

  - In between (half way) each other, i.e. staggered

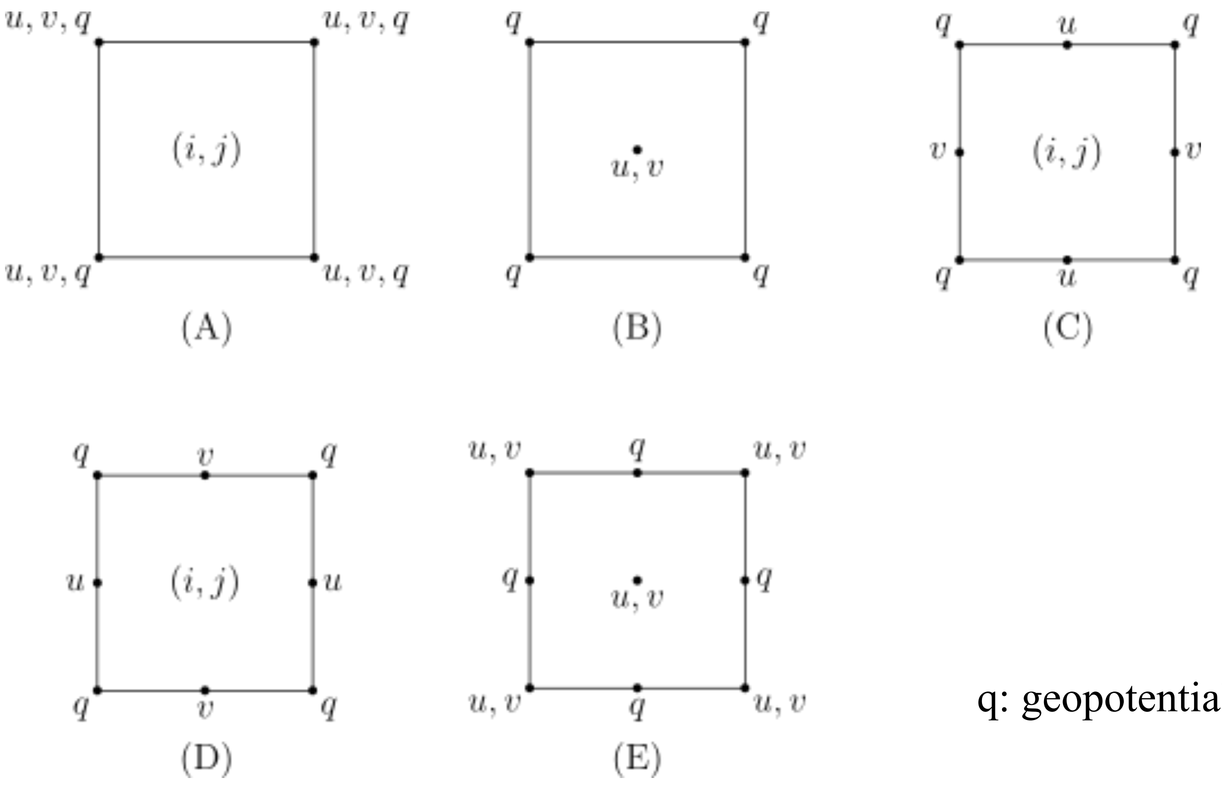


$$\begin{cases}\Delta_t u_j + \dfrac{\phi_{j+1/2}-\phi_{j-1/2}}{\Delta x} = 0\\[2ex] \Delta_t \phi_{j+1/2} + \Phi\dfrac{u_{j+1}-u_{j}}{\Delta x} = 0\end{cases}$$

➔ Improved accuracy + dispersion properties

➔ On explicit techniques staggering results into a more restrictive timestep e.g.: $\dfrac{c\Delta t_{\max}}{\Delta x/2} < 1$ instead of $\dfrac{c\Delta t_{\max}}{\Delta x} < 1$

Different Arakawa horizontally staggered grids



"A benefit of C-grid is that it captures well the propagation of inertio-gravity waves and hence the process of geostrophic adjustment"
Arakawa and Lamb 1977

q: geopotential or pressure

Fig source: Wikipedia By Rpnl ocn - Own work, CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=47077493

Vertical grid staggering

Lorenz staggering

- Good for energy conservation
- Presence of a computational mode

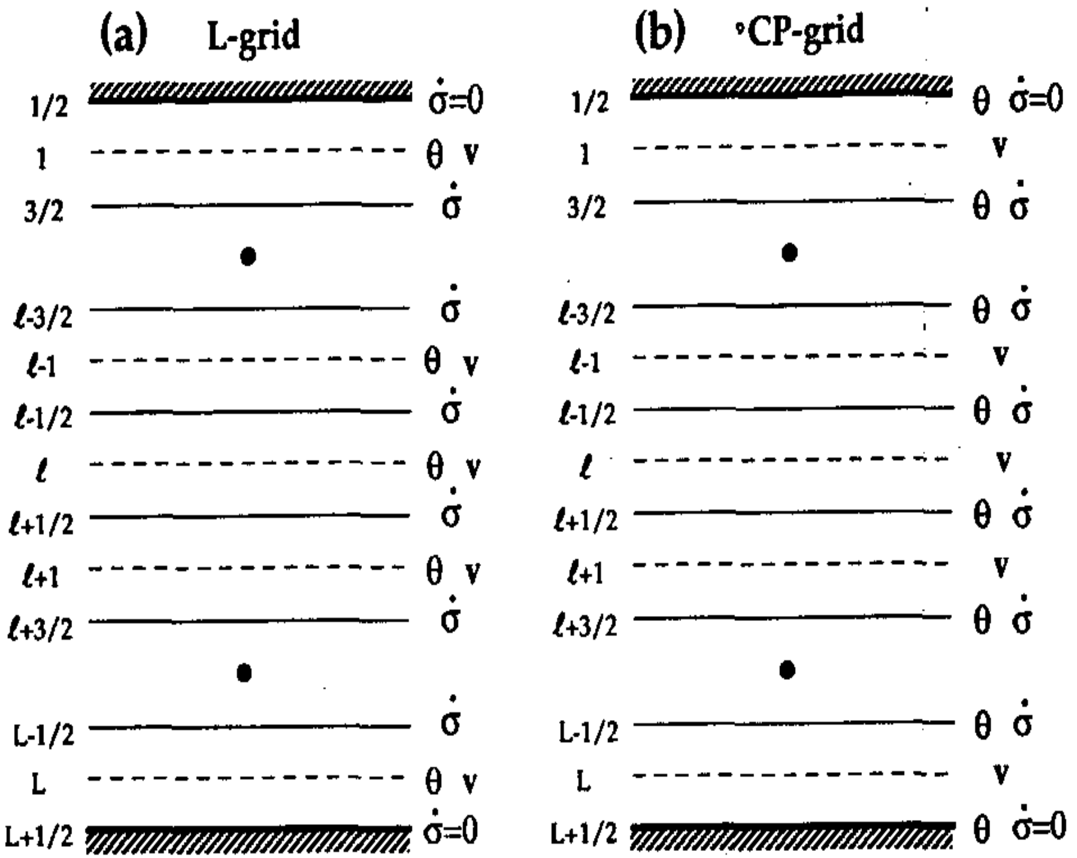


FIG. 1. An illustration of (a) the Lorenz grid and (b) the Charney–Phillips grid for a σ coordinate.

Picture from Arakawa and Konor MWR, 1996, vol 124, 511-

Charney-Philips staggering

- No computational mode
- Conservation of PV

ECMWF IFS:

- No staggering at all

Why is that acceptable?

- High order (spectral transform) horizontal discretization
- High order (finite-element) vertical discretization

Enhancing stability: forward-backward integration

- **Forward-backward scheme:** a predictor-corrector type scheme

The predictor and the corrector are applied on separate equations.

$$\begin{cases} \phi_j^{n+1} = \phi_j^n - \dfrac{\Phi\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) \\ u_j^{n+1} = u_j^n - \dfrac{\Delta t}{2\Delta x}(\phi_{j+1}^{n+1} - \phi_{j-1}^{n+1}) \end{cases}$$

forward

backward (pseudo-implicit)

$$\Delta t \le \frac{2\Delta x}{\sqrt{\Phi}} \approx \frac{\Delta x}{150}$$

Fwd-Bwd versus Leapfrog:

1. Wider stability region allowing twice as big timestep compared with leapfrog
2. Neutral (no damping)
3. Two-time-level scheme ⇒ no computational mode
4. Not as accurate being 1st order

Runge-Kutta RK3 scheme

- Runge-Kutta (Wicker & Skamarock, MWR 2002) RK3 scheme:
  - three-stage, two-time-level (2nd order) scheme from the RK family

$$Solve: \quad \frac{dY}{dt} = f(Y)$$

$$\begin{aligned} Y^{*} &= Y^{n} + \frac{\Delta t}{3} f(Y^{n}) \\ Y^{**} &= Y^{n} + \frac{\Delta t}{2} f(Y^{*}) \\ Y^{n+1} &= Y^{n} + \Delta t\, f(Y^{**}) \end{aligned}$$

Applied on 1d-GW eqn:

$$Y = \begin{pmatrix} u \\ \phi \end{pmatrix}, \quad f(Y) = \begin{pmatrix} -\dfrac{\partial \phi}{\partial x} \\ -\Phi \dfrac{\partial u}{\partial x} \end{pmatrix}$$

3rd or 4th order FD scheme for estimating derivatives

Compared with leapfrog almost doubles (1.62) Δt when 3rd order spatial discretization used

In WRF this is used in combination with time-splitting …

Splitting the time integration: the motivation

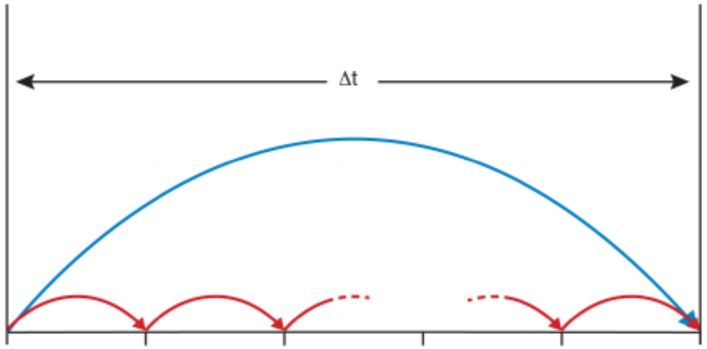
- In an atmospheric model fast and slow wave motions co-exist
  - Splitting exploits the multi-time-scale nature of the governing equations
- Explicit techniques are only conditionally stable which imposes use of very small timesteps for fast processes
  - Let Δt be the longest permissible timestep for integrating stably the slow process => Δt will be too long for stable integration of the fast process
- A practical solution is to split the integration:
  - ➔ integrate slow process with "long" Δt
  - ➔ integrate fast process with a fraction of it i.e. Δt/n

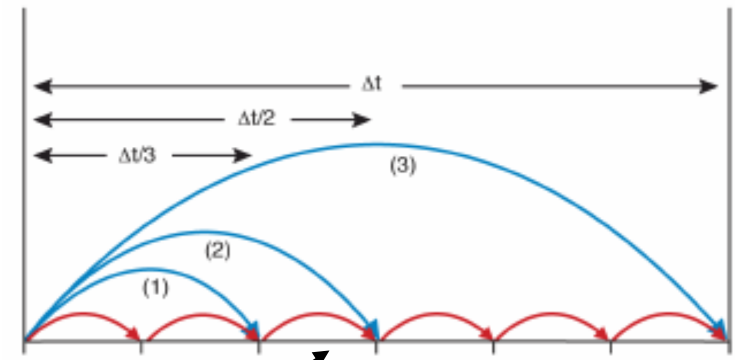
Split-explicit example in a diagram

Split-explicit Euler

Split-explicit RK3

Runge-Kutta internal step i: each approximates solution at t+c_i Δt where, c_i=1/3,1/2,1 the RK3 coefficient





sub-step fast process only (with timestep Δt/n)

(Diagram from S.J. Lock, ECMWF Seminar proceedings 2013, HEVI time-stepping for NWP and climate models)

Split-explicit forward Euler integration

$$\frac{\partial \psi}{\partial t} = F(\psi) + S(\psi)$$

Fast forcing term   Slow forcing term

Integrate forward n_s - times with $\Delta\tau = \Delta t / n_s$ from t to t + Δt :

$$\psi^{t+m\Delta\tau} = \psi^{t+(m-1)\Delta\tau} + \Delta\tau\, F\left(\psi^{t+(m-1)\Delta\tau}\right) + \Delta\tau\, S\left(\psi^{t}\right), \quad m = 1,2,\ldots n_s$$

Fast term updated   Slow term kept constant (stored)

Equivalent to :

$$\psi^{t+\Delta t} = \psi^{t} + \Delta\tau \sum_{m=1}^{n_s} \left[F\left(\psi^{t+(m-1)\Delta\tau}\right) + S\left(\psi^{t}\right) \right] = \overbrace{\psi^{t} + \Delta t\, S\left(\psi^{t}\right)}^{\text{fw Euler with big step}} + \Delta\tau \sum_{m=1}^{n_s} F\left(\psi^{t+(m-1)\Delta\tau}\right),$$

It is an efficient approach :

Store : $R\left(\psi^{t}\right) = \Delta\tau\, S\left(\psi^{t}\right)$

Integrate : $\psi^{t+m\Delta\tau} = \psi^{t+(m-1)\Delta\tau} + \Delta\tau\, F\left(\psi^{t+(m-1)\Delta\tau}\right) + R\left(\psi^{t}\right), \quad m = 1,2,\ldots n_s$

$1*S + n_s*F$ evaluations

versus FW-Euler

$n_s*S + n_s*F$ evaluations

Split-explicit RK3 integration

$$\frac{\partial \psi}{\partial t} = F(\psi) + S(\psi)$$

Step 1: integrate from t to $t+\Delta t/3$ with $\Delta\tau = \Delta t / n_s$:

$$\psi^{t+m\Delta\tau} = \psi^{t+(m-1)\Delta\tau} + \Delta\tau\, F\left(\psi^{t+(m-1)\Delta\tau}\right) + \Delta\tau\, S\left(\psi^{t}\right), \quad m = 1,2,\ldots n_s/3$$

Step 2: integrate from t to $t+\Delta t/2$ with $\Delta\tau = \Delta t / n_s$:

$$S\left(\psi^{*}\right) = S\left(\psi^{t+\Delta t/3}\right), \qquad \psi^{*} \equiv \psi^{t+\Delta t/3} : \text{ final result from stage 1}$$

$$\psi^{t+m\Delta\tau} = \psi^{t+(m-1)\Delta\tau} + \Delta\tau\, F\left(\psi^{t+(m-1)\Delta\tau}\right) + \Delta\tau\, S\left(\psi^{*}\right), \quad m = 1,2,\ldots n_s/2$$

Step 3: integrate from t to $t+\Delta t$ with $\Delta\tau = \Delta t / n_s$:

$$S\left(\psi^{**}\right) = S\left(\psi^{t+\Delta t/2}\right), \qquad \psi^{**} \equiv \psi^{t+\Delta t/2} : \text{ final result from stage 2}$$

$$\psi^{t+m\Delta\tau} = \psi^{t+(m-1)\Delta\tau} + \Delta\tau\, F\left(\psi^{t+(m-1)\Delta\tau}\right) + \Delta\tau\, S\left(\psi^{**}\right), \quad m = 1,2,\ldots n_s$$

S term is evaluated only once per internal RK step and added at each sub-cycle

Drawbacks of the split-explicit approach

- In deep global models O(100km) there is no much benefit from split-explicit approach in the horizontal
  - Stratospheric polar jet velocities are not very far from speed of sound ⇨ advective CFL number is close to acoustic CFL number
    - ➔ All processes are fast and therefore horizontal splitting will not bring significant efficiency benefit
  - Splitting needs damping for stabilization
- Other than split-explicit methods:
  - Horizontally Explicit Vertically Implicit (HEVI)
  - Implicit Explicit (IMEX) RK (unconditionally stable implicit scheme for fast processes and cheap explicit for slow)

HEVI schemes

Vertical acoustic CFL in NH models is much larger than horizontal because of the much thinner vertical layers: explicit time-stepping requires very small timesteps

$$e.g. \quad \Delta t_{\max} < \frac{\Delta z}{c} = \frac{10m}{300m/s} \approx 0.03s$$

A solution: Horizontally Explicit, Vertically Implicit schemes

- Explicit in the horizontal scheme (horizontal CFL << than vertical)
- Unconditionally stable implicit scheme for the vertical to deal with high acoustic CFL numbers

Some HEVI / split-explicit models

- **ICON (DWD Germany): global NWP, LAM weather and climate unified model**
  - Forward-backward explicit time-stepping (no splitting) in the horizontal + vertically semi-implicit (Crank-Nicolson)
- **EU-COSMO: former DWD operational NH LAM**
  - Split-explicit RK3 in the horizontal + semi-implicit in the vertical
- **NICAM: cloud resolving NH global model (Japan)**
  - Split-explicit forward-backward in the horizontal + implicit in vertical
- **WRF, MPAS (USA): LAM, Global research & operational**
  - Split-explicit RK3 + semi-implicit in the vertical

Some useful theoretical properties

- **A-stability: unconditionally stability for damping & oscillatory linear problems $\frac{dy}{dt} = \lambda y, \quad \lambda = \beta + i\omega, \ \beta < 0$ and consequently for linear constant coefficient systems**
  - Explicit methods cannot be A-stable (stability functions are polynomials rather than rational functions)
- **L-Stability: A-stable + rapid decay for stiff problems at long timesteps i.e. $\lim_{q\Delta t \to \infty} \frac{y^{t+\Delta t}}{y^{t}} = 0$ for the above linear equation**
- **Strong Stability Preserving: SSP is a desirable property for a hyperbolic PDE $u_t = -f(u)_x$**
  - A scheme is SSP if it preserves the Total Variation Diminishing (TVD) property under a suitable time step restriction

$$TV(u^{n+1}) \leq TV(u^{n}), \quad TV(u) = \sum_{j=1}^{N} \left| u_{j+1} - u_{j} \right|$$

IMEX: Blending explicit with implicit

$$\frac{d\mathbf{y}}{dt} = \underbrace{\mathbf{s}(t,\mathbf{y})}_{\text{slow process}} + \underbrace{\mathbf{f}(t,\mathbf{y})}_{\text{fast process}}$$

$$\begin{array}{c|c} c & \tilde{A} \\ \hline & \tilde{b} \end{array} = \begin{array}{c|ccc} c_1 & \tilde{\alpha}_{11} & \cdots & \tilde{\alpha}_{1\nu} \\ \vdots & \vdots & & \vdots \\ c_\nu & \tilde{\alpha}_{\nu 1} & \cdots & \tilde{\alpha}_{\nu\nu} \\ \hline & \tilde{b}_1 & \cdots & \tilde{b}_\nu \end{array} \qquad \begin{array}{c|c} c & A \\ \hline & b \end{array} = \begin{array}{c|ccc} c_1 & \alpha_{11} & \cdots & \alpha_{1\nu} \\ \vdots & \vdots & & \vdots \\ c_\nu & \alpha_{\nu 1} & \cdots & \alpha_{\nu\nu} \\ \hline & b_1 & \cdots & b_\nu \end{array}$$

$\tilde{\alpha}_{ij} = 0 \quad \forall j \geq i$ (explicit) $\qquad \alpha_{ij} = 0$ for $j > i$ (diagonally impl)

Compute RK stages $\mathbf{Y}^{(i)} \approx y(t_n + c_i\Delta t)$, $i = 1, \ldots, \nu$ and then new solution $\mathbf{y}^{n+1}$:

$$\mathbf{Y}^{(i)} = \mathbf{y}^n + \Delta t \sum_{j=1}^{i-1} \tilde{\alpha}_{ij}\mathbf{s}(t^n + c_j\Delta t, \mathbf{Y}^{(j)}) + \Delta t \sum_{j=1}^{i} \alpha_{ij}\mathbf{f}(t^n + c_j\Delta t, \mathbf{Y}^{(j)})$$

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t \sum_{j=1}^{\nu} \tilde{b}_j\mathbf{s}(t^n + c_j\Delta t, \mathbf{Y}^{(j)}) + \Delta t \sum_{j=1}^{\nu} b_j\mathbf{f}(t^n + c_j\Delta t, \mathbf{Y}^{(j)})$$

Example of IMEX – ARK2(2,3,2)

- **Giraldo et al SIAM J.Sci.Comp., 2013 option in NUMA NH US Navy model**

$$
\begin{array}{c|ccc}
0 & 0 & & \\
2-\sqrt{2} & 2-\sqrt{2} & 0 & \\
1 & 1-\alpha_{32} & \alpha_{32} & 0 \\
\hline
 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 1-\frac{1}{\sqrt{2}}
\end{array}
\qquad
\begin{array}{c|ccc}
0 & 0 & & \\
2-\sqrt{2} & 1-\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} & \\
1 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \\
\hline
 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 1-\frac{1}{\sqrt{2}}
\end{array}
$$

$$a_{32} = \frac{1}{6}\left(3+2\sqrt{2}\right)$$

- 2nd order + L-Stable, overall very accurate and stable (Weller et al JCP 2013)

A more recent technique: exponential integrators

PDE system is split to a linear and nonlinear part:

$$\frac{dU}{dt} = F(U), \quad F(U) = J\,U + N\left(U\right)$$

J: Jacobian, N(U): the nonlinear residual.

Exponential Integrators:

- Stable with long timesteps
- Accurate with fast dynamics
- They reduce unphysical oscillations

Multiplying with an integrating factor e^{-Jt}:

$$\frac{dU}{dt} = J\,U + N\left(U\right) \Rightarrow U(t_n + \Delta t) = e^{\Delta t J}\,U(t_n) + \int_0^{\Delta t} e^{(\Delta t - \tau)J}\,N\left(U\left(t_n + \tau\right)\right)d\tau$$

Clancy et al, Tellus 2013: use of exponential integration methods in atmospheric models

- The matrix exponential-vector product can be computed using truncated Taylor expansions or Krylov techniques (Niesen and Wright ACM TOMS 38(3), 2012)
- The integral can be computed using numerical quadrature formulae

$$U_{n+1} = e^{hJ_n}U_n + h\sum_{i=1}^{s} \boldsymbol{b_i}\,N\big(U(t_n + c_i h)\big), \quad h \equiv \Delta t$$

$$U(t_n + c_i h) = e^{c_i h J_n}U_n + h\sum_{j=1}^{s} \boldsymbol{a_{ij}}\,N\left(U\big(t_n + c_j h\big)\right)$$

Coefficients **$\boldsymbol{a}$, b are matrix functions of $\boldsymbol{hJ_n}$** and c_i are scalars corresponding the quadrature nodes [0,1]. Essentially a Runge-Kutta time-stepping scheme where its coefficients satisfy the RK order conditions.

Further reading: Luan et al, JCP Vol 376, Jan 2019, p 817-837

Overview

There are many choices of numerical techniques

What to choose depends on the problem you solve (mathematical formulation, resolution, domain) and the computer architecture you apply your algorithm

Nowadays mainly due to hardware requirements and interest in developing very high resolution systems there is considerable research & development activity in scalable compact stencil Eulerian techniques which are also suited for developing dynamical cores with formal conservation properties e.g. PMAP model

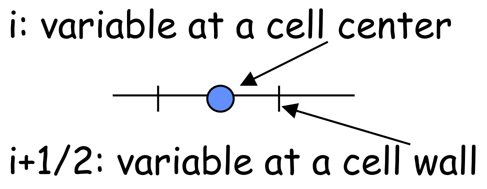
Some references (alphabetically by author's surname)

- J. Coiffier book: Fundamentals of Numerical Weather Prediction (2011)
- Dale Durran's book: "Numerical methods for Wave Equations in Geophysical Fluid Dynamics"
- Lauritzen et al book: Numerical Techniques for Global Atmospheric Models, Springer 2011
- Mengaldo et al, Archives of Comp. Meth. in Eng. (2018): Current and Emerging Time-Integration Strategies in Global NWP
- Wicker & Skamarock (MWR 2001): "Time-Splitting Methods for Elastic Models Using Forward Time Schemes"

MPDATA: 2nd order positive definite conservative advection

Smolarkiewicz & Margolin (1998) MPDATA in finite difference form:

Upstream approximation of flux equation: $\frac{\partial \Psi}{\partial t} = -\frac{\partial}{\partial x}(u\Psi),$


i: variable at a cell center

i+1/2: variable at a cell wall

$$\Psi_i^{n+1} = \Psi_i^n - \left[F\left(\Psi_i^n, \Psi_{i+1}^n, U_{i+1/2}\right) - F\left(\Psi_{i-1}^n, \Psi_i^n, U_{i-1/2}\right)\right],$$

$$F(\Psi_L, \Psi_R, U) \equiv [U]^+\Psi_L + [U]^-\Psi_R \ , \quad U \equiv \frac{u\Delta t}{\Delta x} \ \text{(localCourant Number)}$$

$$[U]^+ \equiv 0.5(U+|U|), \quad [U]^- \equiv 0.5(U-|U|).$$

MPDATA steps

- ♦ Compute 1st order upstream approximation $\Psi_i^{(1)}$ rom above formula
- ♦ Subtract from $\Psi_i^{(1)}$ estimate of error to obtain 2nd order accuracy

$$\Psi_i^{(2)} = \Psi_i^{(1)} - \left[F\left(\Psi_i^{(1)}, \Psi_{i+1}^{(1)}, V_{i+1/2}^{(1)}\right) - F\left(\Psi_{i-1}^{(1)}, \Psi_i^{(1)}, V_{i-1/2}^{(1)}\right)\right]$$

where

$$V_{i+1/2}^{(1)} \equiv (|U| - U^2)\frac{\Psi_{i+1}^{(1)} - \Psi_i^{(1)}}{\Psi_{i+1}^{(1)} + \Psi_i^{(1)}} \equiv (|U| - U^2)A_{i+1/2}^{(1)}$$

Pseudo-velocity