A semi-implicit, semi-Lagrangian, p-adaptive discontinuous Galerkin method for the shallow water equations on the sphere

Giovanni Tumolo

ICTP Abdus Salam - Trieste

< gtumolo@ictp.it >

Offenbach, 13 May 2013

Outline

- Motivation and overview
- Model problem: shallow water equations
- Numerical formulation ingredients:
  - semi-implicit semi-Lagrangian time integration
  - discontinuous Galerkin space discretization
  - p-adaptivity strategy
- Numerical validation:
  - Läuter unsteady test
  - McDonald's and Bates cross-polar flow
  - Williamson's test 5
  - Williamson's test 6
- Conclusions and future plans

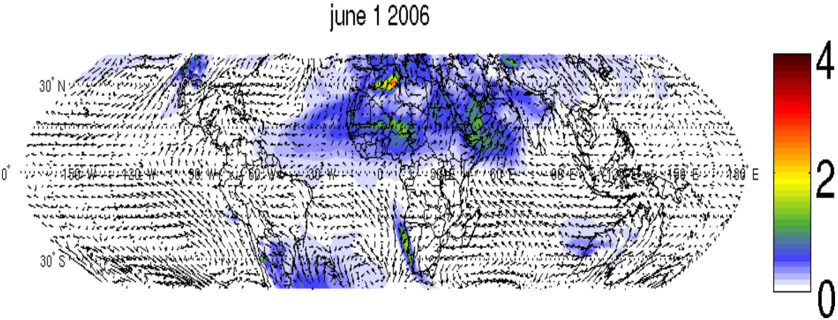
- Joint work with Luca Bonaventura (MOX-Politecnico di Milano)

- with thanks to
  - Filippo Giorgi (ICTP Abdus Salam)
  - Marco Restelli (MPI for Plasma Physics)

- and acknowledgements for funding from
  - The Abdus Salam International Centre for Theoretical Physics
  - The INdAM-GNCS
  - The Italian Ministery of Research and Education

Motivation

- Goal: design a new generation *nonhydrostatic* dynamical core for *regional* climate modelling system RegCM, developed at Abdus Salam ICTP-Trieste, in the Earth System Physics group led by F. Giorgi.



- Discontinuous Galerkin (DG) based models are very appealing, but ...

Overview(I): DG challenging issues

- ... when coupled to explicit time stepping, DG methods are affected by severe stability restrictions as polynomial order increases:

  "The RKDG algorithm is stable provided the following condition holds:

  $$u\frac{\Delta t}{h} < \frac{1}{2p+1}$$

  where p is the polynomial degree; (for the linear case this implies a CFL limit $\frac{1}{3}$)"

  Cockburn-Shu, Math. Comp. 1989

- ... moreover DG requires more degrees of freedom per element than Continuous Galerkin (CG) approach, thus more expensive.

To increase computational efficiency of DG we exploit two ideas:

- coupling DG to SI-SL techniques (no CFL conditions)
- introduction of p-adaptivity (flexible degrees of freedom)

⇒ p-SISLDG (G.Tumolo, L.Bonaventura, M.Restelli, J. Comput. Phys., 2013)

- as first step employed in a simple modelling framework (SWE),
- then to be applied to a fully nonhydrostatic dynamical core for RegCM.

Overview(II). Governing eqs: link btw. SWE and NH vertical slice eqs.

Euler equations (forget Coriolis force for a moment):

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\boldsymbol{u} = 0,$$
$$\frac{D\boldsymbol{u}}{Dt} + \frac{1}{\rho}\nabla p = -g\boldsymbol{k},$$
$$\frac{D\Theta}{Dt} = 0,$$

(being $\frac{D}{Dt}$ the Lagrangian derivative, R the constant of dry air) , can be written using $\Theta = T\left(\frac{p}{p_0}\right)^{-R/c_P}$, $\Pi = \left(\frac{p}{p_0}\right)^{R/c_P}$ as thermodynamic variables:

$$\frac{D\Pi}{Dt} + (\gamma-1)\Pi\nabla\cdot\boldsymbol{u} = 0,$$
$$\frac{D\boldsymbol{u}}{Dt} + c_p\Theta\nabla\Pi = -g\boldsymbol{k},$$
$$\frac{D\Theta}{Dt} = 0.$$

where $\gamma = c_P/c_V$.
Decompose thermodynamic variables in basic state and perturbation:

$$\Pi(x,y,z,t) = \pi^*(z) + \pi(x,y,z,t)$$
$$\Theta(x,y,z,t) = \theta^*(z) + \theta(x,y,z,t)$$

where π^*, θ^* are chosen s.t. $c_P\theta^*\frac{d\pi^*}{dz} = -g$, and consider a vertical slice ($\frac{\partial}{\partial y} = 0$):

$$\frac{D\Pi}{Dt} + (\gamma - 1)\Pi \nabla \cdot \boldsymbol{u} = 0,$$
$$\frac{Du}{Dt} + c_p \Theta \frac{\partial \pi}{\partial x} = 0,$$
$$\frac{Dw}{Dt} + c_p \Theta \frac{\partial \pi}{\partial z} - g\frac{\theta}{\theta *} = 0,$$
$$\frac{D\theta}{Dt} + w\frac{d\theta^*}{dz} = 0.$$

The SISL semi-discretization is:

$$\frac{\Pi^{n+1} - E(t^n, \Delta t)\Pi}{\Delta t} + \alpha(\gamma - 1)\Pi^n \nabla \cdot \boldsymbol{u}^{n+1} + (1-\alpha)(\gamma - 1)E(t^n, \Delta t)[\Pi \nabla \cdot \boldsymbol{u}] = 0,$$
$$\frac{u^{n+1} - E(t^n, \Delta t)u}{\Delta t} + \alpha c_p [E(t^n, \Delta t)\Theta] \frac{\partial \pi}{\partial x}^{n+1} + (1-\alpha) c_P E(t^n, \Delta t)\left[\Theta \frac{\partial \pi}{\partial x}\right] = 0,$$
$$\frac{w^{n+1} - E(t^n, \Delta t)w}{\Delta t} + \alpha c_p [E(t^n, \Delta t)\Theta] \frac{\partial \pi}{\partial z}^{n+1} - \alpha g \frac{\theta^{n+1}}{\theta *} +$$
$$+(1-\alpha) c_P E(t^n, \Delta t)\left[\Theta \frac{\partial \pi}{\partial z}\right] - (1-\alpha) g E(t^n, \Delta t)\left[\frac{\theta}{\theta *}\right] = 0,$$
$$\frac{\theta^{n+1} - E(t^n, \Delta t)\theta}{Dt} + \alpha \frac{d\theta^*}{dz} w^{n+1} + (1-\alpha)E(t^n, \Delta t)\left[\frac{d\theta^*}{dz} w\right] = 0.$$

where:

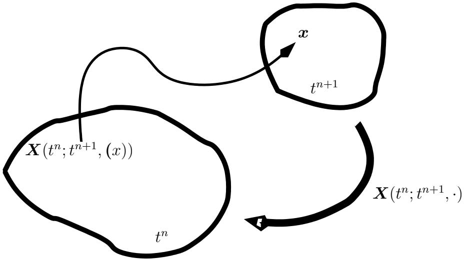
- $G^n = G(\cdot, t^n)$,
- $\alpha \in [0, 1]$ is a fixed implicitness parameter,
- $E(t^n, \Delta t)$ = SL-evolution operator associated to $\boldsymbol{u}^n$: $\left[E(t^n, \Delta t)G\right](\boldsymbol{x}) = G^n(\boldsymbol{x}_D)$.

SL evolution operator on *scalar* valued functions

$$\boldsymbol{x}_D = \boldsymbol{x} - \int_{t^n}^{t^{n+1}} \boldsymbol{u}^n \left(\boldsymbol{X}(t; t^{n+1}, \boldsymbol{x}) \right) dt,$$

where $\boldsymbol{X}(t; t^{n+1}, \boldsymbol{x})$ is the solution of:

$$\begin{cases} \frac{d}{dt}\boldsymbol{X}(t; t^{n+1}, \boldsymbol{x}) = \boldsymbol{u}^n \left(\boldsymbol{X}(t; t^{n+1}, \boldsymbol{x}) \right) \\ \boldsymbol{X}(t^{n+1}; t^{n+1}, \boldsymbol{x}) = \boldsymbol{x} \end{cases} .$$



In practice *two* steps are required to compute $[E(t^n, \Delta t)G](\boldsymbol{x})$:

1. departure point $\boldsymbol{x}_D$ computation (e.g. McGregor, Mon. Wea. Rev.,1993);
2. interpolation of G^n at departure point.

i.e.:

$$\Pi^{n+1} + \alpha\Delta t(\gamma - 1)\Pi^n \nabla \cdot \boldsymbol{u}^{n+1} = E(t^n, \Delta t)\left[\Pi\left(1 - (1-\alpha)\Delta t(\gamma - 1)\nabla \cdot \boldsymbol{u}\right)\right],$$

$$u^{n+1} + \alpha\Delta t c_p [E(t^n, \Delta t)\Theta] \frac{\partial \pi}{\partial x}^{n+1} = E(t^n, \Delta t)\left[u - (1-\alpha)\Delta t c_P \Theta \frac{\partial \pi}{\partial x}\right],$$

$$w^{n+1} + \alpha\Delta t c_p [E(t^n, \Delta t)\Theta] \frac{\partial \pi}{\partial z}^{n+1} - \alpha\Delta t g \frac{\theta^{n+1}}{\theta *} = E(t^n, \Delta t)\left[w - (1-\alpha)\Delta t\left(c_P \Theta \frac{\partial \pi}{\partial z} - g\frac{\theta}{\theta *}\right)\right],$$

$$\theta^{n+1} = -\alpha\Delta t \frac{d\theta^*}{dz} w^{n+1} + E(t^n, \Delta t)\left[\theta - (1-\alpha)\Delta t \frac{d\theta^*}{dz} w\right].$$

Inserting the discretized energy eq. into the discrete vertical momentum eq.
(see e.g. *M. Cullen Q.J.R. Meterol. Soc. 1990, or L. Bonaventura J. Comput. Phys. 2000*):

$$\left(1 + (\alpha\Delta t)^2 \frac{g}{\theta *}\frac{d\theta^*}{dz}\right) w^{n+1} + \alpha\Delta t c_p [E(t^n, \Delta t)\Theta] \frac{\partial \pi}{\partial z}^{n+1} =$$

$$E(t^n, \Delta t)\left[w - (1-\alpha)\Delta t\left(c_P \Theta \frac{\partial \pi}{\partial z} - g\frac{\theta}{\theta *}\right)\right] + \alpha\Delta t \frac{g}{\theta *} E(t^n, \Delta t)\left[\theta - (1-\alpha)\Delta t \frac{d\theta^*}{dz} w\right],$$

$\Longrightarrow$ decoupling of discrete energy eq. from continuity and momentum eqs., which now form a system of three eqs. in three unknowns $\pi^{n+1}, u^{n+1}, w^{n+1}$: after its solution, θ^{n+1} can be recovered from the energy equation (now a diagnostic equation) and Θ field can be reconstructed.

So, the SISL discretization of the nonhydrostatic vertical slice equations is:

$$\pi^{n+1} + \alpha\Delta t(\gamma - 1)\Pi^n \nabla \cdot \boldsymbol{u}_V^{n+1} = -\pi^* + E(t^n, \Delta t)\left[\Pi\left(1 - (1-\alpha)\Delta t(\gamma - 1)\nabla \cdot \boldsymbol{u}_V\right)\right],$$

$$u^{n+1} + \alpha\Delta t c_p [E(t^n, \Delta t)\Theta] \frac{\partial \pi}{\partial x}^{n+1} = E(t^n, \Delta t)\left[u - (1-\alpha)\Delta t c_P \Theta \frac{\partial \pi}{\partial x}\right],$$

$$\left(1 + (\alpha\Delta t)^2 \frac{g}{\theta^*}\frac{d\theta^*}{dz}\right) w^{n+1} + \alpha\Delta t c_p [E(t^n, \Delta t)\Theta] \frac{\partial \pi}{\partial z}^{n+1} =$$

$$E(t^n, \Delta t)\left[w - (1-\alpha)\Delta t\left(c_P\Theta\frac{\partial \pi}{\partial z} - g\frac{\theta}{\theta^*}\right)\right] + \alpha\Delta t \frac{g}{\theta^*} E(t^n, \Delta t)\left[\theta - (1-\alpha)\Delta t \frac{d\theta^*}{dz} w\right],$$

to be compared with SISL semi-discretization of SWE in planar geometry:

$$h^{n+1} + \alpha\Delta t \; h^n \nabla \cdot \boldsymbol{u}_H^{n+1} = E(t^n, \Delta t)\left[h\left(1 - (1-\alpha)\Delta t \; \nabla \cdot \boldsymbol{u}_H\right)\right],$$

$$u^{n+1} + \alpha\Delta t g \frac{\partial h}{\partial x}^{n+1} = -\alpha\Delta t g \frac{\partial b}{\partial x} + E(t^n, \Delta t)\left[u - (1-\alpha)\Delta t g\left(\frac{\partial h}{\partial x} + \frac{\partial b}{\partial x}\right)\right],$$

$$v^{n+1} + \alpha\Delta t g \frac{\partial h}{\partial y}^{n+1} = -\alpha\Delta t g \frac{\partial b}{\partial y} + E(t^n, \Delta t)\left[v - (1-\alpha)\Delta t g\left(\frac{\partial h}{\partial y} + \frac{\partial b}{\partial y}\right)\right],$$

where $\boldsymbol{u}_V = (u, w)^T, \boldsymbol{u}_H = (u, v)^T$.

Conclusion: there is a one to one correspondence btw. SISL dicretized SWE and NH vertical slice eqs.,

$$\pi \longleftrightarrow h,$$
$$u \longleftrightarrow u,$$
$$w \longleftrightarrow v.$$

First step: SWE SISLDG model

SWE in vector form are considered:

$$\frac{Dh}{Dt} + h\nabla \cdot \boldsymbol{u} = 0,$$
$$\frac{D\boldsymbol{u}}{Dt} + g\nabla h + f\hat{\boldsymbol{k}} \times \boldsymbol{u} = -g\nabla b,$$

where h fluid depth, b bathymetry elevation, f Coriolis parameter.

Both continuity *and* momentum equation in advective form (SL approach).

Orthogonal curvilinear coordinates x, y are used:

- on the sphere $x = \lambda$ (lon.), $y = \theta$ (lat.), $m_x = a\cos y$, $m_y = a$, a earth radius,
- on the plane x, y Cartesian and $m_x = m_y = 1$.

$$(dl)^2 = m_x^2 dx^2 + m_y^2 dy^2$$
$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{u}{m_x}\frac{\partial}{\partial x} + \frac{v}{m_y}\frac{\partial}{\partial y},$$
$$u = m_x \frac{Dx}{Dt}, \quad v = m_y \frac{Dy}{Dt}, \quad \boldsymbol{u} = (u, v)^T.$$

SISL time semi-discretization

Now continuity eq. also is discretized with SL approach (new w.r.t. previous SISLDG scheme)

$$\frac{h^{n+1} - E(t^n, \Delta t)h}{\Delta t} = -\alpha\ h^n \nabla \cdot \boldsymbol{u}^{n+1} - (1-\alpha)\ E(t^n, \Delta t)\Big(h \nabla \cdot \boldsymbol{u}\Big)$$

$$\begin{aligned} \frac{\boldsymbol{u}^{n+1} - E(t^n, \Delta t)\boldsymbol{u}}{\Delta t} \quad = \quad & -\alpha \left(g \nabla h^{n+1} + g \nabla b + f \hat{\boldsymbol{k}} \times \boldsymbol{u}^{n+1} \right) \\ & -(1-\alpha)\ E(t^n, \Delta t)\Big(g \nabla h + g \nabla b + f \hat{\boldsymbol{k}} \times \boldsymbol{u} \Big) \end{aligned}$$

The SL-evolution operator on a vector valued function $\boldsymbol{G}(\cdot, t)$ is again:

$$\left[E(t^n, \Delta t) \boldsymbol{G} \right] (\boldsymbol{x}) = \boldsymbol{G}^n(\boldsymbol{x}_D)$$

... but what of components?

SL evolution operator on *vector* valued functions

$$\boldsymbol{G}^n(\boldsymbol{x}_D) = \mathcal{G}_x^n(\boldsymbol{x}_D)\hat{\boldsymbol{i}}(\boldsymbol{x}_D) + \mathcal{G}_y^n(\boldsymbol{x}_D)\hat{\boldsymbol{j}}(\boldsymbol{x}_D) + \mathcal{G}_z^n(\boldsymbol{x}_D)\hat{\boldsymbol{k}}(\boldsymbol{x}_D)$$

In *curved* geometry $\hat{\boldsymbol{i}}(\boldsymbol{x}) \neq \hat{\boldsymbol{i}}(\boldsymbol{x}_D)$, $\hat{\boldsymbol{j}}(\boldsymbol{x}) \neq \hat{\boldsymbol{j}}(\boldsymbol{x}_D)$, $\hat{\boldsymbol{k}}(\boldsymbol{x}) \neq \hat{\boldsymbol{k}}(\boldsymbol{x}_D)$, hence:

$$\hat{\boldsymbol{i}}(\boldsymbol{x})\cdot\boldsymbol{G}^n(\boldsymbol{x}_D) = \mathcal{G}_x^n(\boldsymbol{x}_D)\,\hat{\boldsymbol{i}}(\boldsymbol{x})\cdot\hat{\boldsymbol{i}}(\boldsymbol{x}_D) + \mathcal{G}_y^n(\boldsymbol{x}_D)\,\hat{\boldsymbol{i}}(\boldsymbol{x})\cdot\hat{\boldsymbol{j}}(\boldsymbol{x}_D) + \mathcal{G}_z^n(\boldsymbol{x}_D)\,\hat{\boldsymbol{i}}(\boldsymbol{x})\cdot\hat{\boldsymbol{k}}(\boldsymbol{x}_D),$$

$$\hat{\boldsymbol{j}}(\boldsymbol{x})\cdot\boldsymbol{G}^n(\boldsymbol{x}_D) = \mathcal{G}_x^n(\boldsymbol{x}_D)\,\hat{\boldsymbol{j}}(\boldsymbol{x})\cdot\hat{\boldsymbol{i}}(\boldsymbol{x}_D) + \mathcal{G}_y^n(\boldsymbol{x}_D)\,\hat{\boldsymbol{j}}(\boldsymbol{x})\cdot\hat{\boldsymbol{j}}(\boldsymbol{x}_D) + \mathcal{G}_z^n(\boldsymbol{x}_D)\,\hat{\boldsymbol{j}}(\boldsymbol{x})\cdot\hat{\boldsymbol{k}}(\boldsymbol{x}_D),$$

$$\hat{\boldsymbol{k}}(\boldsymbol{x})\cdot\boldsymbol{G}^n(\boldsymbol{x}_D) = \mathcal{G}_x^n(\boldsymbol{x}_D)\hat{\boldsymbol{k}}(\boldsymbol{x})\cdot\hat{\boldsymbol{i}}(\boldsymbol{x}_D) + \mathcal{G}_y^n(\boldsymbol{x}_D)\hat{\boldsymbol{k}}(\boldsymbol{x})\cdot\hat{\boldsymbol{j}}(\boldsymbol{x}_D) + \mathcal{G}_z^n(\boldsymbol{x}_D)\hat{\boldsymbol{k}}(\boldsymbol{x})\cdot\hat{\boldsymbol{k}}(\boldsymbol{x}_D),$$

i.e.

$$\begin{pmatrix} \hat{\boldsymbol{i}}(\boldsymbol{x})\cdot[E(t^n,\Delta t)\boldsymbol{G}](\boldsymbol{x}) \\ \hat{\boldsymbol{j}}(\boldsymbol{x})\cdot[E(t^n,\Delta t)\boldsymbol{G}](\boldsymbol{x}) \\ \hat{\boldsymbol{k}}(\boldsymbol{x})\cdot[E(t^n,\Delta t)\boldsymbol{G}](\boldsymbol{x}) \end{pmatrix} = \underbrace{\begin{bmatrix} \hat{\boldsymbol{i}}(\boldsymbol{x})\cdot\hat{\boldsymbol{i}}(\boldsymbol{x}_D) & \hat{\boldsymbol{i}}(\boldsymbol{x})\cdot\hat{\boldsymbol{j}}(\boldsymbol{x}_D) & \hat{\boldsymbol{i}}(\boldsymbol{x})\cdot\hat{\boldsymbol{k}}(\boldsymbol{x}_D) \\ \hat{\boldsymbol{j}}(\boldsymbol{x})\cdot\hat{\boldsymbol{i}}(\boldsymbol{x}_D) & \hat{\boldsymbol{j}}(\boldsymbol{x})\cdot\hat{\boldsymbol{j}}(\boldsymbol{x}_D) & \hat{\boldsymbol{j}}(\boldsymbol{x})\cdot\hat{\boldsymbol{k}}(\boldsymbol{x}_D) \\ \hat{\boldsymbol{k}}(\boldsymbol{x})\cdot\hat{\boldsymbol{i}}(\boldsymbol{x}_D) & \hat{\boldsymbol{k}}(\boldsymbol{x})\cdot\hat{\boldsymbol{j}}(\boldsymbol{x}_D) & \hat{\boldsymbol{k}}(\boldsymbol{x})\cdot\hat{\boldsymbol{k}}(\boldsymbol{x}_D) \end{bmatrix}}_{R} \begin{pmatrix} \mathcal{G}_x^n \\ \mathcal{G}_y^n \\ \mathcal{G}_z^n \end{pmatrix}$$

SL evolution operator on *vector* valued functions

Four steps are then required to compute $[E(t^n, \Delta t)\boldsymbol{G}](\boldsymbol{x})$ components w.r.t. $\hat{\boldsymbol{i}}(\boldsymbol{x}), \hat{\boldsymbol{j}}(\boldsymbol{x}), \hat{\boldsymbol{k}}(\boldsymbol{x})$:

1. departure-point $\boldsymbol{x}_D$ computation;
2. interpolation at departure point $\boldsymbol{x}_D$ of $\boldsymbol{G}^n$ components in the unit vector triad at the *same* point $\boldsymbol{x}_D$ i.e. interpolation of $\mathcal{G}_x^n, \mathcal{G}_y^n, \mathcal{G}_z^n$;
3. computation of rotation matrix R, which transforms vector components in the departure-point unit vector triad $\hat{\boldsymbol{i}}(\boldsymbol{x}_D), \hat{\boldsymbol{j}}(\boldsymbol{x}_D), \hat{\boldsymbol{k}}(\boldsymbol{x}_D)$ into vector components in the arrival-point unit vector triad $\hat{\boldsymbol{i}}(\boldsymbol{x}), \hat{\boldsymbol{j}}(\boldsymbol{x}), \hat{\boldsymbol{k}}(\boldsymbol{x})$;
4. rotation of the interpolated components $\mathcal{G}_x^n(\boldsymbol{x}_D), \mathcal{G}_y^n(\boldsymbol{x}_D), \mathcal{G}_z^n(\boldsymbol{x}_D)$ by the matrix R.

- No explicit metric terms;
- in the limit $\Delta t \to 0$, off diagonal elements of R generate metric terms;
- no singularity at poles;
- under shallow-atmosphere approximation R reduces to $\Lambda = \begin{bmatrix} p & q \\ -q & p \end{bmatrix}$, where $p = (R_{11} + R_{22})/(1 + R_{33}),\ q = (R_{12} - R_{21})/(1 + R_{33})$.
  (A. Staniforth, A.A. White, N.Wood, Q.J.R.Meterol.Soc. 2010)

SISL time semi-discretized equations in component form

$$h^{n+1} + \alpha\Delta t\ h^n \nabla\cdot\boldsymbol{u}^{n+1} = E(t^n,\Delta t)\Big(h - (1-\alpha)\Delta t\ h\nabla\cdot\boldsymbol{u}\Big)$$

$$\begin{aligned}
u^{n+1} + \alpha\Delta t\left(\frac{g}{m_x}\frac{\partial h}{\partial x}^{n+1} - fu^{n+1}\right) &= -\alpha\Delta t\frac{g}{m_x}\frac{\partial b}{\partial x} + \\
\Lambda_{11}E(t^n,\Delta t)&\left[u - (1-\alpha)\Delta t\left(\frac{g}{m_x}\frac{\partial h}{\partial x} + \frac{g}{m_x}\frac{\partial b}{\partial x} - fv\right)\right] + \\
\Lambda_{12}E(t^n,\Delta t)&\left[v - (1-\alpha)\Delta t\left(\frac{g}{m_y}\frac{\partial h}{\partial y} + \frac{g}{m_y}\frac{\partial b}{\partial y} + fu\right)\right]
\end{aligned}$$

$$\begin{aligned}
v^{n+1} + \alpha\Delta t\left(\frac{g}{m_y}\frac{\partial h}{\partial y}^{n+1} + fu^{n+1}\right) &= -\alpha\Delta t\frac{g}{m_y}\frac{\partial b}{\partial y} + \\
\Lambda_{21}E(t^n,\Delta t)&\left[u - (1-\alpha)\Delta t\left(\frac{g}{m_x}\frac{\partial h}{\partial x} + \frac{g}{m_x}\frac{\partial b}{\partial x} - fv\right)\right] + \\
\Lambda_{22}E(t^n,\Delta t)&\left[v - (1-\alpha)\Delta t\left(\frac{g}{m_y}\frac{\partial h}{\partial y} + \frac{g}{m_y}\frac{\partial b}{\partial y} + fu\right)\right]
\end{aligned}$$

DG space discretization

Defined a tassellation $\mathcal{T}_h = \{K_l\}_{l=1}^N$ of domain Ω and chosen $\forall K_l \in \mathcal{T}_h$ two integers $p_l^h \geq 0$, $p_l^u \geq 0$, at each time level t^n, we are looking for approximate solution s.t.

$$\begin{aligned} h^n &\in H_h := \left\{ f \in L^2(\Omega) \ : \ f|_{K_l} \in \mathbb{Q}_{p_l^h}(K_l) \right\} \\ \boldsymbol{u}^n &\in V_h := \left\{ g \in L^2(\Omega) \ : \ g|_{K_l} \in \mathbb{Q}_{p_l^u}(K_l) \right\}^2, \end{aligned}$$

i.e. within each element K_l, the solution at time t^n will be represented as:

$$h^n(\boldsymbol{x})\big|_{K_l} = \sum_{r=1}^{(p_l^h+1)^2} \varphi_{l,r}(\boldsymbol{x}) h_{l,r}^n, \quad u^n(\boldsymbol{x})\big|_{K_l} = \sum_{r=1}^{(p_l^u+1)^2} \psi_{l,r}(\boldsymbol{x}) u_{l,r}^n, \quad v^n(\boldsymbol{x})\big|_{K_l} = \sum_{r=1}^{(p_l^u+1)^2} \psi_{l,r}(\boldsymbol{x}) v_{l,r}^n$$

The absence of a global continuity constraint, typical of discontinuous FEM,

- requires the definition of the solution at inter-element boundaries:
  pb1: how to choose numerical fluxes ?
  Centered fluxes are used (F. Bassi and S. Rebay, J. Comput. Phys. 1997);
- makes easy the introduction of adaptivity in space by locally varying p_l:
  pb2: at each t^n, how to choose p_l *locally*, i.e. for each element K_l ?
  A proper p-adaptation strategy is needed.

p-adaptivity (I): choice of basis functions

- p-adaptivity is further made easy by the use of modal bases.
- Since structured meshes of quadrilaterals are employed (on the sphere we use lon-lat coordinates), tensor products of Legendre polynomials are a good choice as :
  - hierarchical: good for adaptive computation of the p_l;
  - orthogonal in Cartesian domain: fully diagonal mass matrix in that case.
- Hence, within a given element K_l, the representation for a model variable α becomes

$$\alpha(\boldsymbol{x})\big|_{K_l} = \sum_{k=1}^{p_l^\alpha+1} \sum_{l=1}^{p_l^\alpha+1} \alpha_{l,k,l}\psi_{l_x,k}(x)\psi_{l_y,l}(y).$$

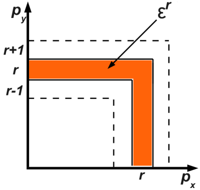
  with $l = (l_x, l_y)$ suitable multi-index.
- and its 2-norm is given by (in planar geometry):

$$\mathcal{E}_l^{tot} = \sum_{k,l=1}^{p_l^\alpha+1} \alpha_{l,k,l}^2$$

p-adaptivity (II): relative 'weights' of modal components

- Then, for a given element $K_l \in \mathcal{T}_h$, the '*energy*' contained in the $r-$ th modal components of $\alpha\big|_{K_l}$ is given by (again for planar geometry):

$$\mathcal{E}_l^r := \sum_{max(k,l)=r} \alpha_{l,k,l}^2$$



- while, for any integer $r = 1, \ldots, p_l^\alpha + 1$, the quantity

$$w_l^r = \sqrt{\frac{\mathcal{E}_l^r}{\mathcal{E}_l^{tot}}}$$

will measure the relative 'weight' of the $r-$th modal components of α with respect to the best approximation available for the L^2 norm of α.

p-adaptivity (III): adaptation algorithm

If α is a generic model variable, the following adaptation criterion is applied:

- Compute all model variables with p_{max} at initial time.
- *Given an error tolerance $\epsilon_l > 0$ for all $l = 1, \ldots, N$, at each time step repeat following steps:*

1) *compute w_{p_i}*

2.1) *if $w_{p_i} \geq \epsilon_i$, then*

2.1.1) *set $p_i(\alpha) := p_i(\alpha) + 1$*
2.1.2) *set $\alpha_{i,p_i} = 0$, exit the loop and go the next element*

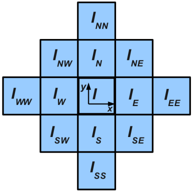
2.2) *if instead $w_{p_i} < \epsilon_i$, then*

2.2.1) *compute $w_{p_i - 1}$*
2.2.2) *if $w_{p_i - 1} \geq \epsilon_i$, exit the loop and go the next element*
2.2.3) *else if $w_{p_i - 1} < \epsilon_i$, set $p_i(\alpha) := p_i(\alpha) - 1$ and go back to 2.2.1.*

Fully discrete problem

- standard L^2 projection against test functions (chosen equal to the basis functions as in Direct Characteristic Galerkin scheme, Morton et al., M2AN 1988), followed by integration by parts (where necessary),
- introduction of (centered) numerical fluxes,
- expression of velocity d.o.f. in terms of depth d.o.f. from momentum equations and and their substitution into the continuity equation,

give raise, at each SI step, to a discrete (vector) Helmholtz equation in the fluid depth unknown only, with computational stencil surrounding the element K_I given by



sparse block structured nonsymmetric linear system solved by GMRES with *block* diagonal (for the moment) preconditioning.

Numerical Validation

Läuter unsteady flow: time convergence rate estimation ($\alpha = 0.50$).

$$p^h = 4, \quad p^u = 5, \quad \max(C_{cel}) \approx 1.7$$

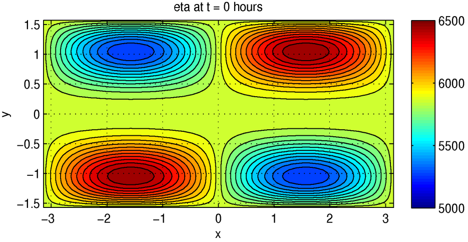
$N_x \times N_y$	Δt (min)	$E_1(h)$	$E_2(h)$	$E_\infty(h)$
10×5	60	3.287×10^{-3}	3.631×10^{-3}	6.044×10^{-3}
20×10	30	7.201×10^{-4}	7.871×10^{-4}	1.261×10^{-3}
40×20	15	1.680×10^{-4}	1.844×10^{-4}	2.966×10^{-4}

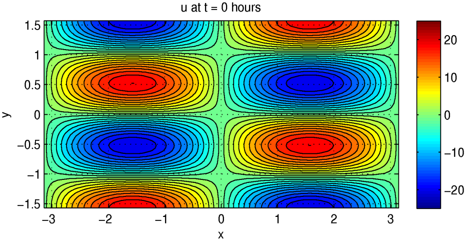
$N_x \times N_y$	Δt (min)	$E_1(u)$	$E_2(u)$	$E_\infty(u)$
10×5	60	3.748×10^{-2}	4.821×10^{-2}	1.679×10^{-1}
20×10	30	1.012×10^{-2}	1.288×10^{-2}	3.107×10^{-2}
40×20	15	2.574×10^{-3}	3.214×10^{-3}	7.557×10^{-3}

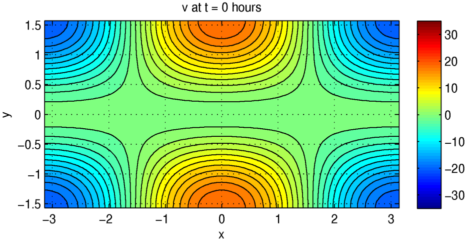
$N_x \times N_y$	Δt (min)	$E_1(v)$	$E_2(v)$	$E_\infty(v)$
10×5	60	6.549×10^{-2}	6.930×10^{-2}	2.744×10^{-1}
20×10	30	1.586×10^{-2}	1.676×10^{-2}	4.779×10^{-2}
40×20	15	3.956×10^{-3}	4.180×10^{-3}	1.491×10^{-2}

McDonald's and Bates cross-polar flow

30×15 elements, $p^h = 4$, $\Delta t = 900s$ ($C_{cel} \approx 21$, $C_{vel} \approx 1$ close to poles).

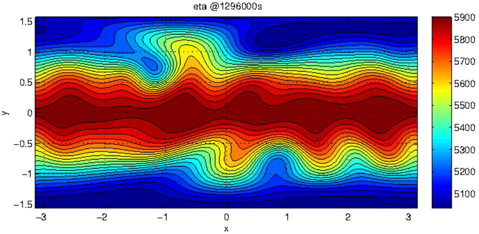


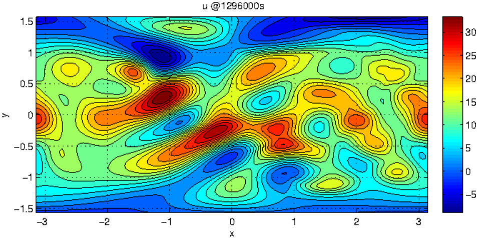


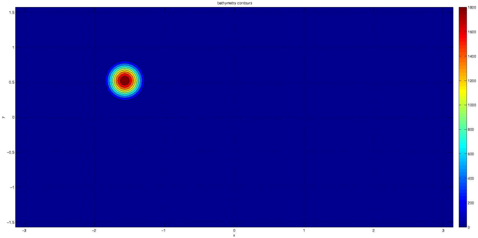


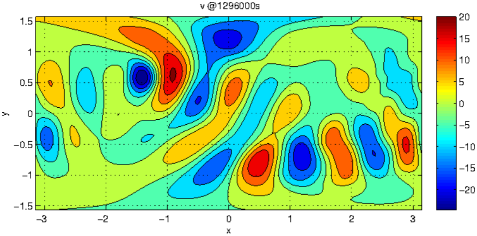
Williamson's test 5

Solution at day 15, 30×15 elements, max $p^h = 4$, $\Delta t = 900s$ ($C_{cel} \approx 11$ close to poles).



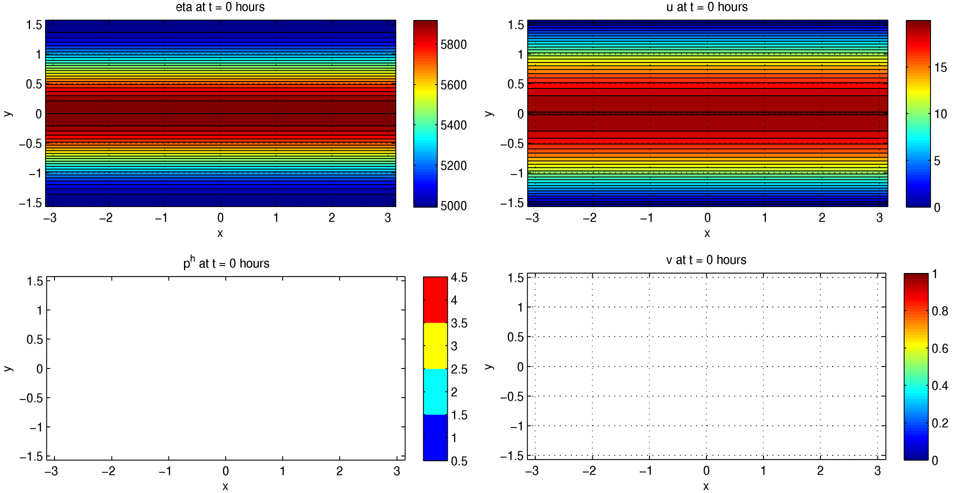






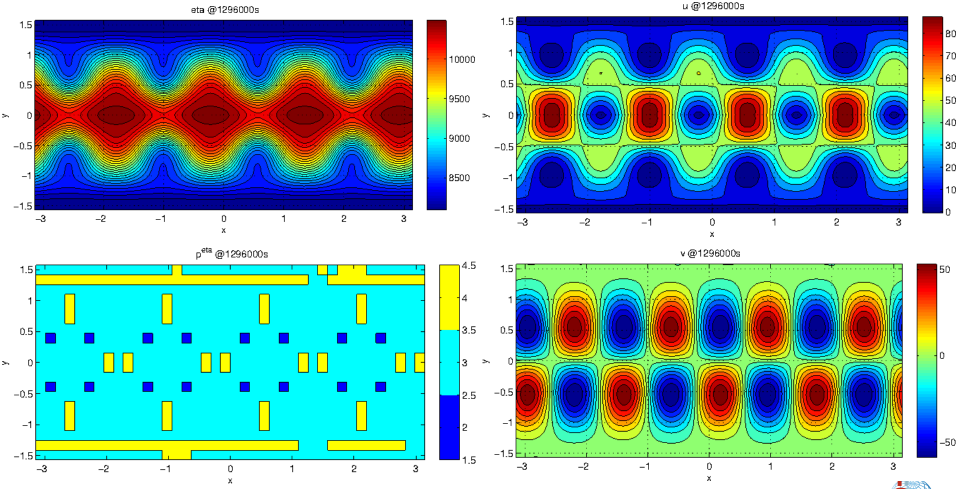
Williamson's test 5: dynamic p-adaptation.

30×15 elements, max $p^h = 4$, $\Delta t = 900s$ ($C_{cel} \approx 11$ close to poles).



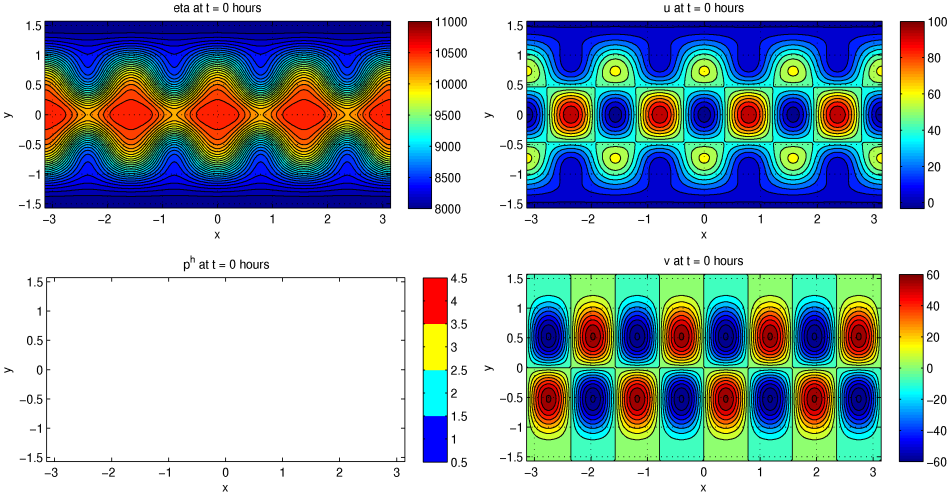
Williamson's test 6

Sol. at day 15, 40×20 elem., max $p^h = 5$, $\Delta t = 900s$ ($C_{cel} \approx 21$ close to poles).



Williamson’s test 6: dynamic p-adaptation

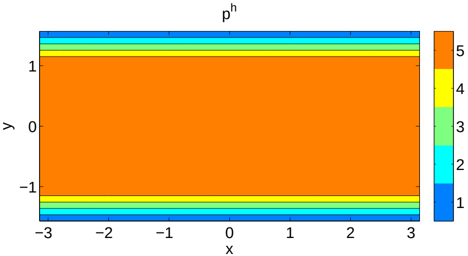
40×20 elements, $\max p^h = 5$, $\Delta t = 900s$ ($C_{cel} \approx 21$ close to poles).

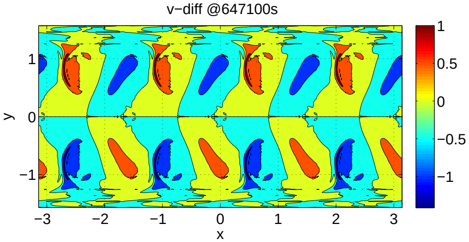


Williamson's test 6: static p-adaptation as control on Courant number

60×30 elements, $\max p^h = 5$, $\Delta t = 900s$ ($C_{cel} \approx 48$ close to poles).

max p^h can be imposed locally in order to control the local Courant number:





$\Longrightarrow$ this leads to significant efficiency improvement:

$$\frac{\#\text{gmres-iterations}(p^h = \text{adapted})}{\#\text{gmres-iterations}(p^h = \text{uniform})} \approx 13\%$$

(GMRES stopping criterion: $\frac{||residual||}{||rhs||} = 10^{-10}$)

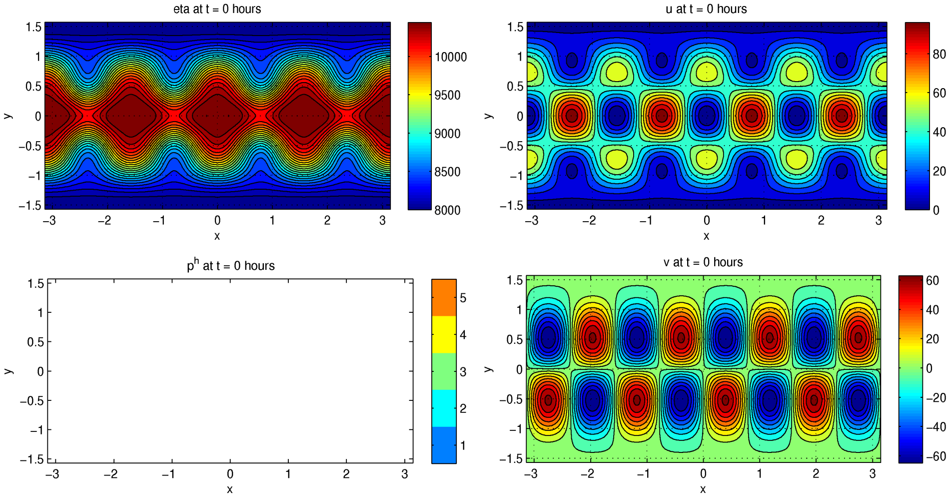
$$\Delta^n_{dof} = \frac{\sum_{l=1}^{N} (p^n_l + 1)^2}{N(p_{max} + 1)^2} \approx 67\%$$

($N = \#$ of elements).

Williamson's test 6: static + dynamic p-adaptation

50×25 elements, $\max p^h = 5$, $\Delta t = 900s$ ($C_{cel} \approx 33$ without adaptivity)

$$\frac{\#\text{gmres-iterations}(p^h = \text{adapted})}{\#\text{gmres-iterations}(p^h = \text{uniform})} \approx 18\%, \qquad \Delta^n_{dof} = \frac{\sum_{l=1}^{N}(p^n_l + 1)^2}{N(p_{max}+1)^2} \approx 40\%$$



Conclusions, open issues and future perspectives

- In summary:
  - a SISL DG discretization for rotating SWE has been presented, extending succesfully the SISL approach to DG framework;
  - the proposed algorithm is presented on structured meshes, but, in principle, it can be extended to arbitrary non-structured ones;
  - a simple $p-$adaptivity approach allows to reduce the computational cost;
  - numerical experiments prove the effectiveness of the proposed scheme.

- Now on the way:
  - improvement of the linear solver for the SI step: preconditioning strategy, from block diagonal to ILU;
  - improvement of the time-integration scheme: from θ-method to TR-BDF2;
  - completion of implementation of the SISLDG scheme for nonhydrostatic vertical slice equations.

- Future perspectives:
  - comparison with other stiff time integration techniques (e.g. Rosenbrock and exponential integrators);
  - parallelization strategy;
  - integration of SWE and vertical slice SISLDG discretizations to develop the nonhydrostatic dynamical core for RegCM;
  - development of a conservative version.