Recent progresses in the development of a new generation adaptive DG dynamical core for RegCM

Giovanni Tumolo

The Abdus Salam ICTP and OGS - Trieste, < gtumolo@ictp.it >

Trieste, May 24, 2016

- In collaboration with Luca Bonaventura (MOX-Politecnico di Milano)

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

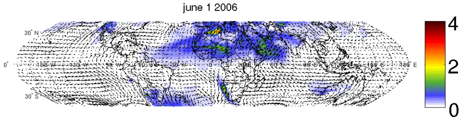
- and acknowledgements for funding from
  - The Abdus Salam International Centre for Theoretical Physics
  - Istituto Nazionale di Oceanografia e Geofisica Sperimentale (whithin the HPC-TRES framework)
  - The INdAM-GNCS
  - The Royal Meteorological Society

Outline

- Motivation and introduction to the p-SISLDG formulation.
- Review of a novel SISL time integration approach.
- 1st extension: mass conservative mixed Eulerian/semi-Lagrangian variant.
- 2nd extension: meshes of deformed quadrilaterals on the sphere and on a vertical plane.
- $p-$adaptivity.
- Numerical validation:
  - horizontal:
    - efficiency gain by TR-BDF2: unsteady flow with analytic solution
    - efficiency gain by $p-$adaptivity: Williamson's test 6
    - effects of the mesh deformation on the solution (mass conservative version): Williamson's test 2 and unsteady flow with analytic solution
  - $p-$adaptive tracers transport:
    - Solid body rotation
    - Deformational flow
    - Coupling with SWE solver: advection by Rossby-Haurwitz wave
  - vertical:
    - Interacting bubbles test
    - Linear hydrostatic lee waves
    - Nonlinear nonhydrostatic lee waves
- Where is the 3D dycore?? Current status and HPC requirements for using p-SISLDG as a three dimensional dynamical core.

Motivation

- Goal: use DG methods for the design of a new generation dynamical core for the *regional* climate modelling system RegCM of ICTP.



- This is challenging for:
  - stability restrictions with explicit time stepping:

    “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
  - computational cost : DG requires more d.o.f. per element than CG .
- How to increase computational efficiency of DG ?
  - coupling DG to semi-implicit semi-Lagrangian (SI-SL) technique (no CFL)
  - introduction of $p-$ adaptivity (flexible degrees of freedom)
- $\Longrightarrow$ p-SISLDG method.

p-SISLDG: main features

Main novel features of the proposed p-SISLDG formulation:

- is the first unconditionally stable DG formulation for the shallow water and for the Euler equations,
- is based on the first fully second order two-time-level SISL time integrator,
- is the first extensive application of $p-$adaptivity strategies in NWP,
- employs a unified discretization approach for the horizontal and vertical.

A novel approach for SISL time integration: TR-BDF2

Given a Cauchy problem for a system of ODEs:

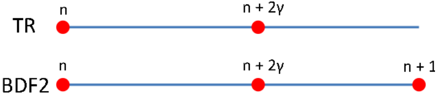
$$
\begin{aligned}
\mathbf{y}' &= \mathbf{f}(\mathbf{y}, t), \\
\mathbf{y}(0) &= \mathbf{y}_0,
\end{aligned}
$$

the TR-BDF2 method is defined by the two following implicit stages (Bank et al. IEEE trans. 1985):

$$
\begin{aligned}
\mathbf{u}^{n+2\gamma} - \gamma\Delta t\mathbf{f}(\mathbf{u}^{n+2\gamma}, t_n + 2\gamma\Delta t) &= \mathbf{u}^n + \gamma\Delta t\mathbf{f}(\mathbf{u}^n, t_n), \\
\mathbf{u}^{n+1} - \gamma_2\Delta t\mathbf{f}(\mathbf{u}^{n+1}, t_{n+1}) &= (1-\gamma_3)\mathbf{u}^n + \gamma_3\mathbf{u}^{n+2\gamma},
\end{aligned}
$$

with $\gamma \in (0, 1/2]$ fixed implicitness parameter and

$$
\gamma_2 = \frac{1-2\gamma}{2(1-\gamma)}, \quad \gamma_3 = \frac{1-\gamma_2}{2\gamma}.
$$

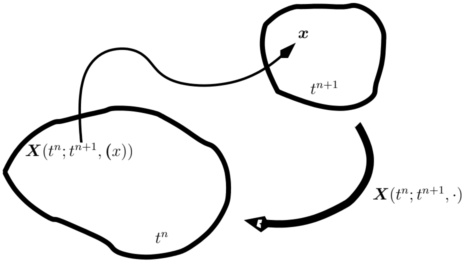


SL reinterpretation of TR-BDF2

If suitable semi-Lagrangian approximate evolution operators for scalar and vector valued functions are introduced: $[E(t^n, \Delta t)G](\mathbf{x}) = G(t^n, \mathbf{x}_D)$

where $\mathbf{x}_D = \mathbf{x} - \int_{t^n}^{t^{n+1}} \mathbf{u}^n\left(\mathbf{X}(t; t^{n+1}, \mathbf{x})\right) dt$ and $\mathbf{X}(t; t^{n+1}, \mathbf{x})$ is the solution of:

$$\begin{cases} \frac{d}{dt}\mathbf{X}(t; t^{n+1}, \mathbf{x}) = \mathbf{u}^n\Big(\mathbf{X}(t; t^{n+1}, \mathbf{x})\Big) \\ \mathbf{X}(t^{n+1}; t^{n+1}, \mathbf{x}) = \mathbf{x} \end{cases},$$



i.e. *two* steps are required to compute $[E(t^n, \Delta t)G]\,(\mathbf{x})$:

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

SL reinterpretation of TR-BDF2

... and if governing equations in advective form are to be solved:

(being $\frac{D}{Dt}$ the Lagrangian derivative operator)

(SWE) Shallow Water Eqs. (no Coriolis force):

$$\frac{Dh}{Dt} + h\nabla \cdot \mathbf{u} = 0,$$

$$\frac{D\mathbf{u}}{Dt} + g\nabla h = -g\nabla b,$$

with $h, \mathbf{u} = (u, v)^T$ and b being fluid depth, horizontal velocity and bathymetry elevation respectively,

(VSE) Euler eqs. (no Coriolis force) on a Vertical Slice ($\frac{\partial}{\partial y} = 0$):

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

with $\Theta = T\left(\frac{p}{p_0}\right)^{-R/c_p}, \Pi = \left(\frac{p}{p_0}\right)^{R/c_p}$, $p, T, \mathbf{u} = (u, w)^T$, pressure, temperature and vertical velocity, c_p, c_v, R specific heats and gas constant of dry air, and

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

... then SISL-TR steps for SWE and VSE are isomorphic

$$h^{n+2\gamma} + \gamma\Delta t\ h^n\ \boldsymbol{\nabla}\cdot\mathbf{u}^{n+2\gamma} = E\left(t^n, 2\gamma\Delta t\right)\left[h - \gamma\Delta t\ h\ \boldsymbol{\nabla}\cdot\mathbf{u}\right],$$

$$\mathbf{u}^{n+2\gamma} + \gamma\Delta t\ g\boldsymbol{\nabla} h^{n+2\gamma} = -\gamma\Delta t\ g\boldsymbol{\nabla} b + E\left(t^n, 2\gamma\Delta t\right)\left\{\mathbf{u} - \gamma\Delta t\left[g(\boldsymbol{\nabla} h + \boldsymbol{\nabla} b)\right]\right\}.$$

$$\pi^{n+2\gamma} + \gamma\Delta t\ \left(c_p/c_v - 1\right)\Pi^n\boldsymbol{\nabla}\cdot\mathbf{u}^{n+2\gamma} = -\pi^* + E\left(t^n, 2\gamma\Delta t\right)\left[\Pi - \gamma\Delta t\left(c_p/c_v - 1\right)\Pi\ \boldsymbol{\nabla}\cdot\mathbf{u}\right],$$

$$u^{n+2\gamma} + \gamma\Delta t\ c_p\Theta^n\frac{\partial\pi}{\partial x}^{n+2\gamma} = E(t^n, 2\gamma\Delta t)\left[u - \gamma\Delta t\ c_p\Theta\frac{\partial\pi}{\partial x}\right],$$

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

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

then SISL-BDF2 steps for SWE and VSE are isomorphic

$$h^{n+1} + \gamma_2\Delta t\ h^{n+2\gamma}\ \nabla \cdot \mathbf{u}^{n+1} =$$
$$(1-\gamma_3)E(t^n, \Delta t)h$$
$$+\gamma_3 E(t^n + 2\gamma\Delta t, (1-2\gamma)\Delta t)h,$$

$$\pi^{n+1} + \gamma_2\Delta t\,(c_p/c_v - 1)\,\Pi^{n+2\gamma}\,\nabla \cdot \mathbf{u}^{n+1} =$$
$$-\pi^* + (1-\gamma_3)[E\,(t^n, \Delta t)\ \Pi]$$
$$+\gamma_3[E\,(t^n + 2\gamma\Delta t, (1-2\gamma)\Delta t)\ \Pi],$$

$$u^{n+1} + \gamma_2\Delta t\ c_p\Theta^{n+2\gamma}\,\frac{\partial \pi}{\partial x}^{n+1} =$$
$$(1-\gamma_3)[E\,(t^n, \Delta t)\ u]$$
$$+\gamma_3[E\,(t^n + 2\gamma\Delta t, (1-2\gamma)\Delta t)\ u],$$

$$\mathbf{u}^{n+1} + \gamma_2\Delta t\ g\nabla h^{n+1} =$$
$$-\gamma_2\Delta t\ g\nabla b$$
$$+(1-\gamma_3)E(t^n, \Delta t)\mathbf{u}$$
$$+\gamma_3 E(t^n + 2\gamma\Delta t, (1-2\gamma)\Delta t)\mathbf{u}.$$

$$\left(1 + (\gamma_2\Delta t)^2\,\frac{g}{\theta*}\,\frac{d\theta^*}{dz}\right) w^{n+1} + \gamma_2\Delta t\ c_p\Theta^{n+2\gamma}\,\frac{\partial \pi}{\partial z}^{n+1} =$$
$$(1-\gamma_3)[E\,(t^n, \Delta t)\ w] + \gamma_3[E\,(t^n + 2\gamma\Delta t, (1-2\gamma)\Delta t)\ w] +$$
$$\gamma_2\Delta t\,\frac{g}{\theta*}\,\{(1-\gamma_3)[E\,(t^n, \Delta t)\ \theta] + \gamma_3[E\,(t^n + 2\gamma\Delta t, (1-2\gamma)\Delta t)\ \theta]\}$$

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

1st Extension: mass conservation, SISL-TR-BDF2 time discretization

Considering the continuity equation in Eulerian flux form, while the momentum one in advective vector form:

$$\frac{\partial \eta}{\partial t} = -\boldsymbol{\nabla} \cdot (h\mathbf{u}),$$

$$\frac{D\mathbf{u}}{Dt} = -g\boldsymbol{\nabla}\eta - f\mathbf{k} \times \mathbf{u},$$

then, the TR stage of the SISL time discretization of previous equations is:

$$\eta^{n+2\gamma} + \gamma\Delta t\, \boldsymbol{\nabla} \cdot \left(h^n \mathbf{u}^{n+2\gamma}\right) = \eta^n - \gamma\Delta t\, \boldsymbol{\nabla} \cdot (h^n \mathbf{u}^n),$$

$$\mathbf{u}^{n+2\gamma} + \gamma\Delta t \left(g\boldsymbol{\nabla}\eta^{n+2\gamma} + f\mathbf{k} \times \mathbf{u}^{n+2\gamma}\right)$$
$$= E\left(t^n, 2\gamma\Delta t\right)\left[\mathbf{u} - \gamma\Delta t\left(g\boldsymbol{\nabla}\eta + f\mathbf{k} \times \mathbf{u}\right)\right].$$

The TR stage is then followed by the BDF2 stage:

$$\eta^{n+1} + \gamma_2\Delta t\, \boldsymbol{\nabla} \cdot (h^{n+2\gamma}\mathbf{u}^{n+1}) = (1-\gamma_3)\eta^n + \gamma_3\, \eta^{n+2\gamma},$$

$$\mathbf{u}^{n+1} + \gamma_2\Delta t \left(g\boldsymbol{\nabla}\eta^{n+1} + f\mathbf{k} \times \mathbf{u}^{n+1}\right)$$
$$= (1-\gamma_3)E\left(t^n, \Delta t\right)\mathbf{u} + \gamma_3 E\left(t^n + 2\gamma\Delta t, (1-2\gamma)\Delta t\right)\mathbf{u}.$$

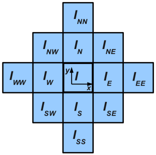
DG space discretization

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

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

- modal bases are used to span P_h, T_h, V_h,
- L^2 projection against test functions (chosen equal to the basis functions),
- introduction of (centered) numerical fluxes,
- substitution of velocity d.o.f. from momentum eqs. into the continuity eq.,
- give raise, at each SI step, to a discrete (vector) Helmholtz equation in the fluid depth / pressure unknown only,

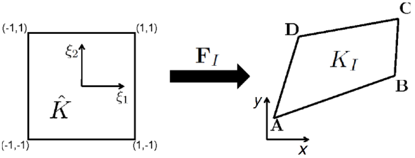
i.e. a sparse block structured nonsymmetric linear system is solved by GMRES with *block* diagonal (for the moment) preconditioning.



Potential of $p-$adaptivity for atmospheric modelling applications

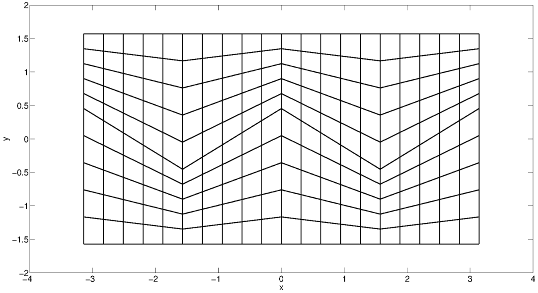
- No remeshing required of many physical quantities like orography profiles, data on land use and soil type, land-sea masks.
- Completely independent resolution for each single model variable.
- Easier coupling with SL technique, especially on unstructured meshes (no need to store two meshes).
- Possibility also of static p-adaptation: e.g. reduced p as counterpart of reduced grid, i.e. locally imposed p controlling the local Courant number ($\Longrightarrow$ significant $\#$gmres-iterations reduction).
- Main potential problem: dynamic load balancing is mandatory for massively parallel implementations.

2nd extension: mesh deformation on the sphere



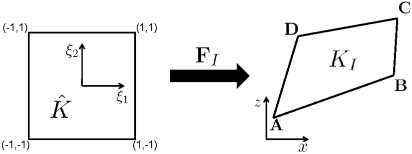
$$x = F_{I,1}(\xi_1,\xi_2) = x_I^A \frac{1-\xi_1}{2}\frac{1-\xi_2}{2} + x_I^B \frac{1+\xi_1}{2}\frac{1-\xi_2}{2} + x_I^C \frac{1+\xi_1}{2}\frac{1+\xi_2}{2} + x_I^D \frac{1-\xi_1}{2}\frac{1+\xi_2}{2},$$

$$y = F_{I,2}(\xi_1,\xi_2) = y_I^A \frac{1-\xi_1}{2}\frac{1-\xi_2}{2} + y_I^B \frac{1+\xi_1}{2}\frac{1-\xi_2}{2} + y_I^C \frac{1+\xi_1}{2}\frac{1+\xi_2}{2} + y_I^D \frac{1-\xi_1}{2}\frac{1+\xi_2}{2},$$



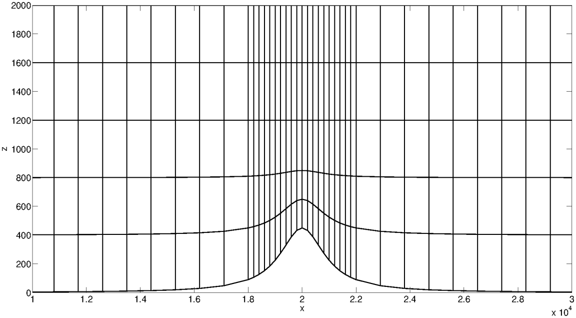
(adapted from Weller 2012)

2nd extension: mesh deformation on a vertical plane, topography in z coordinate



$$x = F_{I,1}(\xi_1,\xi_2) = x_I^A \frac{1-\xi_1}{2}\frac{1-\xi_2}{2} + x_I^B \frac{1+\xi_1}{2}\frac{1-\xi_2}{2} + x_I^C \frac{1+\xi_1}{2}\frac{1+\xi_2}{2} + x_I^D \frac{1-\xi_1}{2}\frac{1+\xi_2}{2},$$

$$z = F_{I,2}(\xi_1,\xi_2) = z_I^A \frac{1-\xi_1}{2}\frac{1-\xi_2}{2} + z_I^B \frac{1+\xi_1}{2}\frac{1-\xi_2}{2} + z_I^C \frac{1+\xi_1}{2}\frac{1+\xi_2}{2} + z_I^D \frac{1-\xi_1}{2}\frac{1+\xi_2}{2}.$$



Numerical Validation

Shallow Water Equations (SWE) on the sphere

Unsteady flow with analytic solution (Läuter 2005): TR-BDF2 vs off centerd Crank Nicolson

- Relative errors for TR-BDF2 at different resolutions, Δt in seconds:

$N_x \times N_y$	Δt	$l_1(h)$	$l_2(h)$	$l_\infty(h)$	q_2^{emp}
10×5	3600	5.46×10^{-3}	6.12×10^{-3}	9.54×10^{-3}	-
20×10	1800	1.25×10^{-3}	1.40×10^{-3}	2.14×10^{-3}	2.1
40×20	900	3.04×10^{-4}	3.41×10^{-4}	5.21×10^{-4}	2.0
80×40	450	7.55×10^{-5}	8.47×10^{-5}	1.29×10^{-4}	2.0

- Relative errors for off-centered Crank Nicolson ($\theta = 0.6$) at different resolutions:

$N_x \times N_y$	Δt	$l_1(h)$	$l_2(h)$	$l_\infty(h)$	q_2^{emp}
10×5	3600	1.44×10^{-2}	1.63×10^{-2}	2.40×10^{-2}	-
20×10	1800	8.74×10^{-3}	9.89×10^{-3}	1.44×10^{-2}	0.7
40×20	900	4.81×10^{-3}	5.45×10^{-3}	7.96×10^{-3}	0.9
80×40	450	2.53×10^{-3}	2.86×10^{-3}	4.18×10^{-3}	0.9

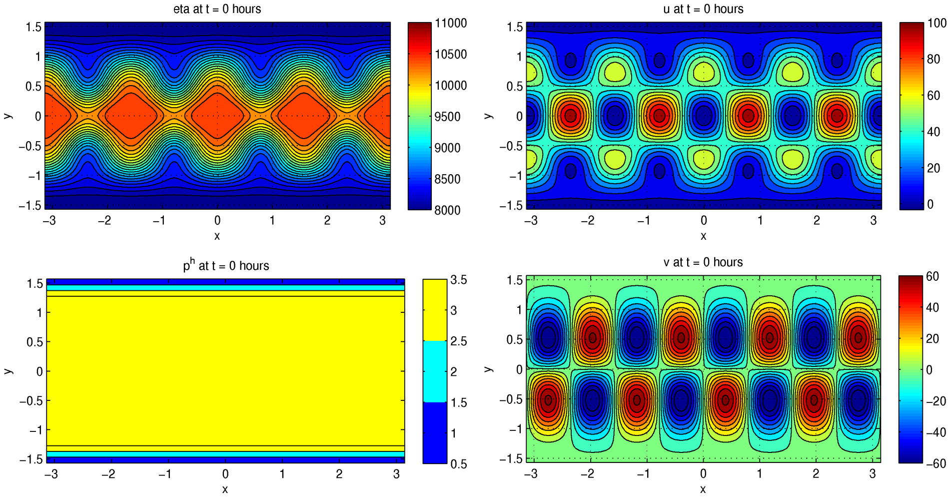
- At max. resolution in space and time (80×40 el., $\Delta t = 450$s) error norms for TR-BDF2 are around 34 times smaller than those of off-centered Crank Nicolson, while CPU time is equivalent (104.3s for a time step of TR-BDF2 vs 99.9s for a time step of off centerd CN).

- At fixed resolution in space (40×20 el.), off centered Crank Nicolson needs to be run with a 16 times smaller Δt in order to reach same level of accuracy of TR-BDF2 with $\Delta t = 900$s. $\Longrightarrow$ CPU time for TR-BDF2 is around 20% that of off-centered CN for same accuracy.

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

64×32 elements, $\max p^h = 4$, $\Delta t = 900s$ ($C_{cel} \approx 83$ without adaptivity).

$$\frac{\#\text{gmres-iterations}(p^h = \text{adapted})}{\#\text{gmres-iterations}(p^h = \text{uniform})} \approx 13\%, \qquad \Delta^n_{dof} = \frac{\sum_{l=1}^{N}(p_l^n + 1)^2}{N(p_{max}+1)^2} \approx 45\%.$$



Williamson's test 6: time convergence rate and p-adaptation efficiency

- Relative errors at $t_f = 15$ days for different number of elements, with respect to NCAR spectral model solution at resolution T511:

$N_x \times N_y$	Δt[min]	$l_1(h)$	$l_2(h)$	$l_\infty(h)$	q_2^{emp}
10×5	60	2.92×10^{-2}	3.82×10^{-2}	6.75×10^{-2}	-
20×10	30	5.50×10^{-3}	6.80×10^{-3}	1.11×10^{-2}	2.4
40×20	15	1.40×10^{-3}	1.80×10^{-3}	3.20×10^{-3}	2.0

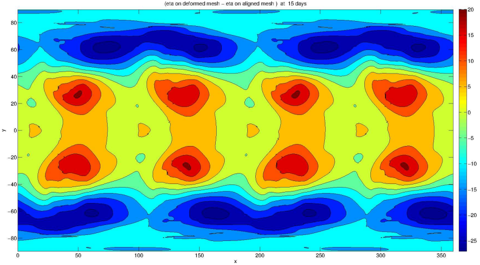
- Relative differences btw adaptive (tol. $\epsilon = 10^{-2}$) and nonadaptive solution at $t_f = 15$ days:

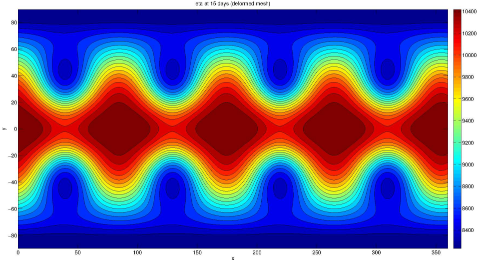
adaptivity	$l_1(h)$	$l_2(h)$	$l_\infty(h)$
static	2.182×10^{-4}	3.434×10^{-4}	2.856×10^{-4}
static + dynamic	3.407×10^{-4}	4.301×10^{-4}	7.484×10^{-4}

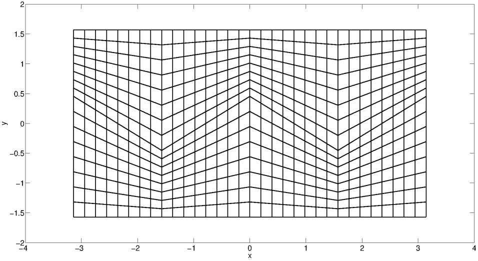
- CPU time: static and dynamic p-adaptive solution execution time is around 24% of that for nonadaptive solution.

Williamson's test 6: deformed vs. aligned mesh

$$p^{\eta} = 4, \quad p^{u} = 5, \quad N_x \times N_y = 32 \times 16, \qquad t_f = 15\text{days}$$







$$d_1 = 1.058 \times 10^{-3}$$

$$d_2 = 1.263 \times 10^{-3}$$

$$d_\infty = 2.568 \times 10^{-3}$$

Mass conservative formulation on deformed mesh: convergence rate, Williamson's test 2

$$p^\eta = p^u = 3$$

$N_x \times N_y$	Δt[s]	$l_1(\eta)$	$l_2(\eta)$	$l_\infty(\eta)$	q_2^{emp}
20×10	1800	5.59×10^{-5}	8.39×10^{-5}	1.20×10^{-3}	-
40×20	900	5.84×10^{-6}	8.66×10^{-6}	1.67×10^{-4}	3.3
80×40	450	7.50×10^{-7}	1.02×10^{-6}	6.88×10^{-6}	3.1

$N_x \times N_y$	Δt[s]	$l_1(u)$	$l_2(u)$	$l_\infty(u)$	q_2^{emp}
20×10	1800	7.72×10^{-4}	1.51×10^{-3}	1.35×10^{-2}	-
40×20	900	7.46×10^{-5}	2.78×10^{-4}	6.77×10^{-3}	2.5
80×40	450	4.69×10^{-6}	1.00×10^{-5}	2.03×10^{-4}	4.8

$N_x \times N_y$	Δt[s]	$l_1(v)$	$l_2(v)$	$l_\infty(v)$	q_2^{emp}
20×10	1800	8.23×10^{-4}	1.09×10^{-3}	9.98×10^{-3}	-
40×20	900	7.87×10^{-5}	1.58×10^{-4}	2.66×10^{-3}	2.8
80×40	450	8.12×10^{-6}	1.84×10^{-5}	4.70×10^{-4}	3.1

Mass conservative formulation: errors for deformed vs. aligned mesh, unsteady flow with analytic solution (Läuter 2005)

$p^{\eta} = 4, \quad p^{u} = 5, \quad N_x \times N_y = 20 \times 10, \qquad t_f = 5\text{days}$

mesh	$l_1(\eta)$	$l_2(\eta)$	$l_\infty(\eta)$
distorted	1.574439×10^{-3}	2.015191×10^{-3}	6.223918×10^{-3}
aligned	1.574433×10^{-3}	2.015189×10^{-3}	6.220938×10^{-3}

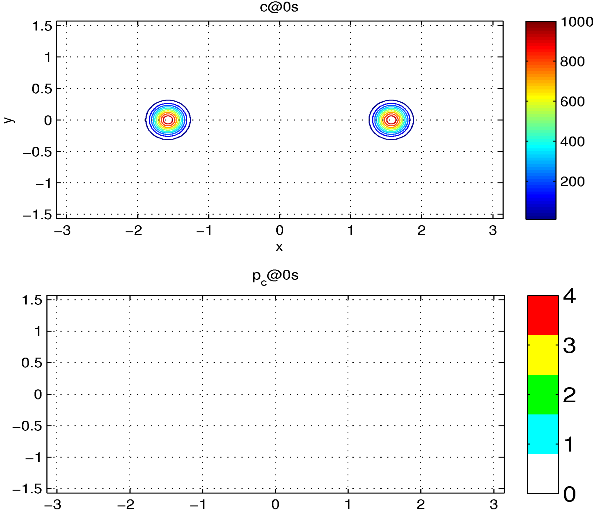
mesh	$l_1(u)$	$l_2(u)$	$l_\infty(u)$
distorted	3.062825×10^{-2}	3.816815×10^{-2}	7.295160×10^{-2}
aligned	3.062796×10^{-2}	3.816801×10^{-2}	7.293568×10^{-2}

mesh	$l_1(v)$	$l_2(v)$	$l_\infty(v)$
distorted	2.105254×10^{-2}	2.195037×10^{-2}	3.832416×10^{-2}
aligned	2.105328×10^{-2}	2.195039×10^{-2}	3.833373×10^{-2}

p-adaptive tracers advection

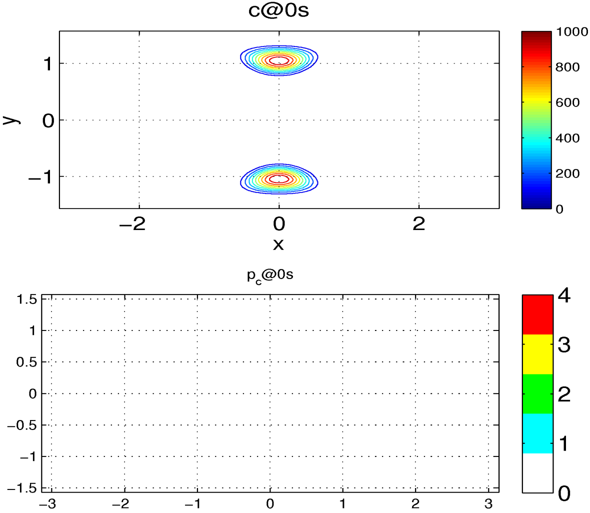
Solid body rotation on the sphere

120×60 elements, max $p^c = 4$, $\Delta t = 7200$s, $C_{vel,x} \approx 400$, $C_{vel,y} \approx 4$



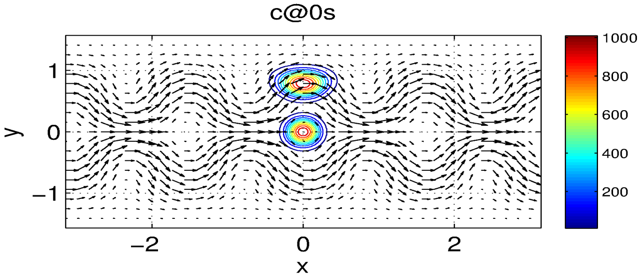
Deformational flow on the sphere (adapted from Nair, Lauritzen 2010)

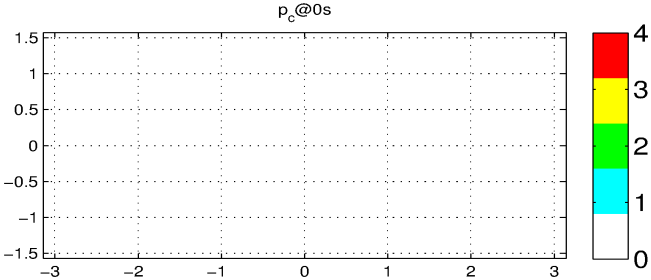
80×40 elements, max $p^c = 4$, $\Delta t = 1800$s



Rossby Haurwitz wave velocity field

120×60 elements, $\max p^c = 4$, $\Delta t = 900s$, $C_{vel,x} \approx 1$

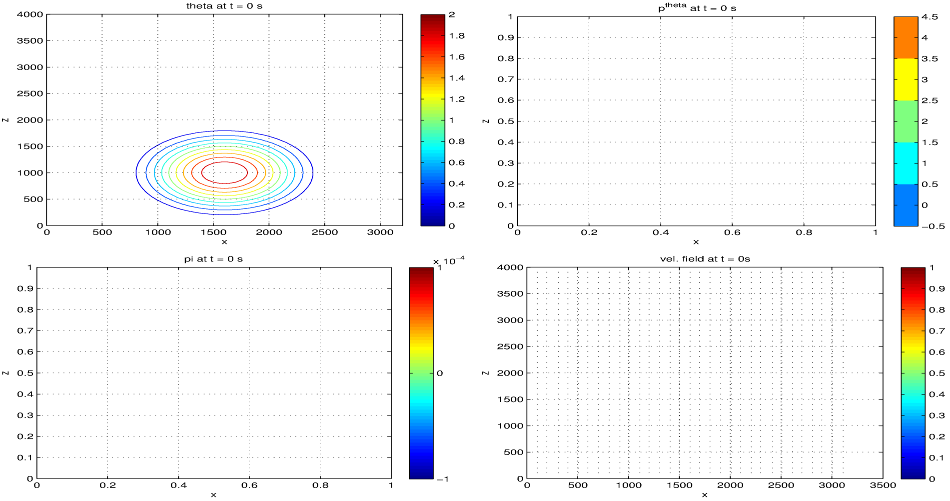




Euler equations on a Vertical Slice (VSE)

Warm bubble test (Carpenter et al., MWR 1990)

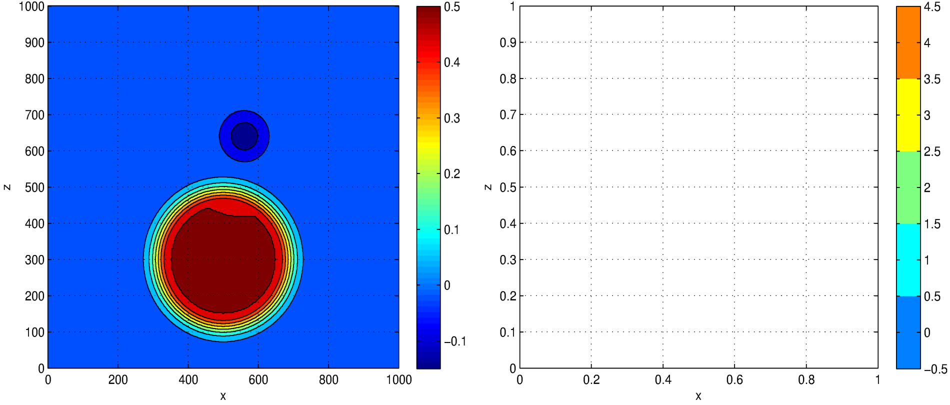
49×60 elements, $p^{\pi} = 4$, $p^{u} = 5$, $\Delta t = 1$ s, $C \approx 18$.



variable	l_1	l_2	l_∞
π	2.744×10^{-3}	4.92×10^{-3}	3.86×10^{-2}
θ	1.70×10^{-2}	4.38×10^{-2}	9.34×10^{-2}
u	3.64×10^{-4}	1.14×10^{-3}	3.60×10^{-2}

Interacting bubbles test (Robert, 1993)

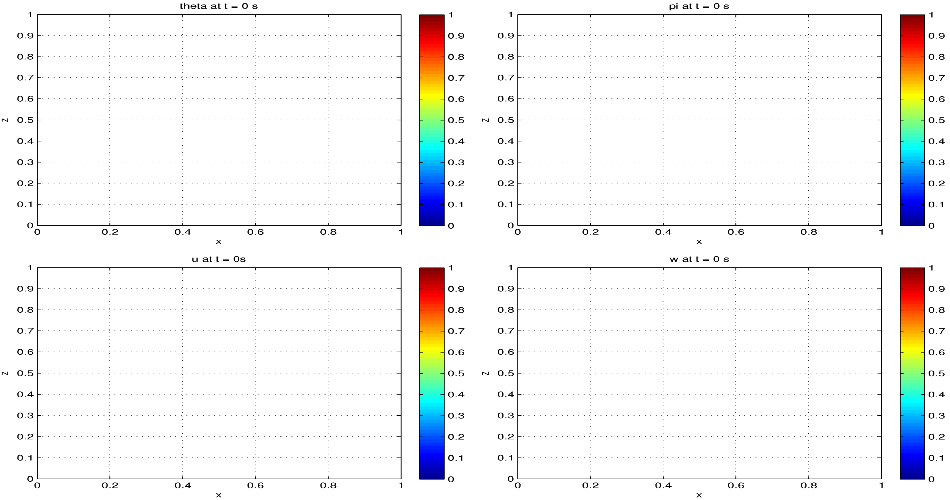
50×50 elements, $p^{\pi} = 4$, $p^{u} = 5$, $\Delta t = 1$ s, $C \approx 87$.



Linear hydrostatic lee waves

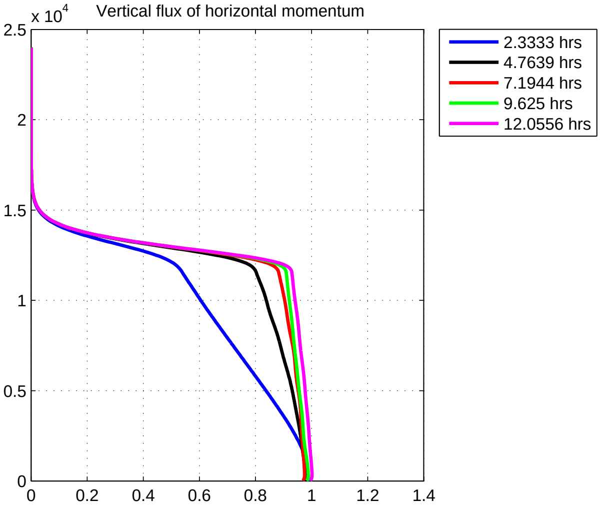
60×50 elements, $p^{\pi} = 4$, $p^{u} = 5$, $\Delta t = 7$ s, $C_V \approx 7, C_H \approx 9.$

(maximum space resolution 2 km)



Linear hydrostatic lee waves

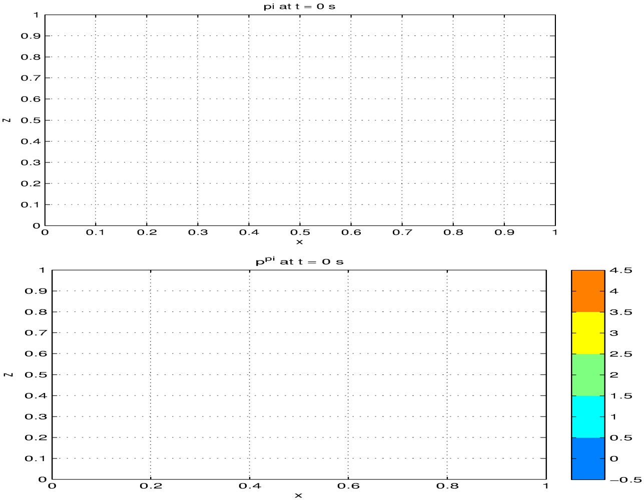
60×50 elements, $p^{\pi} = 4$, $p^{u} = 5$, $\Delta t = 7$ s, $C_V \approx 7, C_H \approx 9$.



Linear hydrostatic lee waves: adaptive run

60×50 elements, $p^{\pi} = p^{u} = 4$, $\Delta t = 7$ s, $C_V \approx 7, C_H \approx 9$.

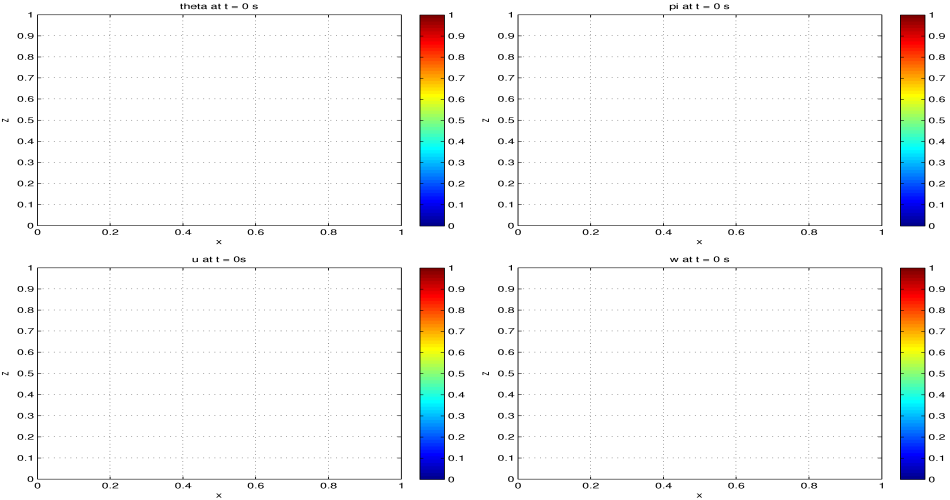
(maximum space resolution 2 km)



Nonlinear nonhydrostatic lee waves

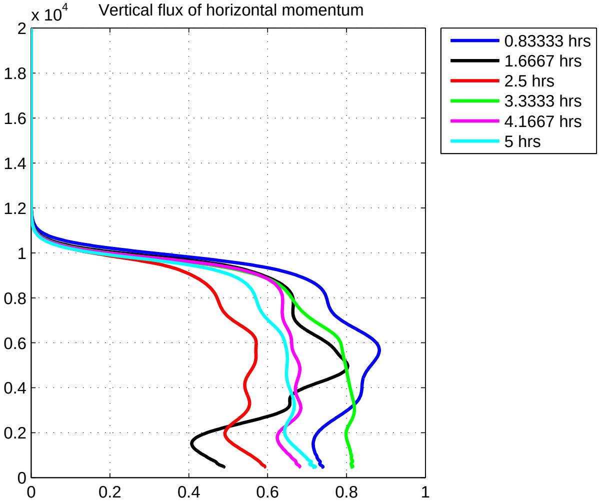
60×50 elements, $p^{\pi} = 4$, $p^{u} = 5$, $\Delta t = 2$ s, $C_V \approx 25$, $C_H \approx 13$.

(maximum space resolution 200m)



Nonlinear nonhydrostatic lee waves

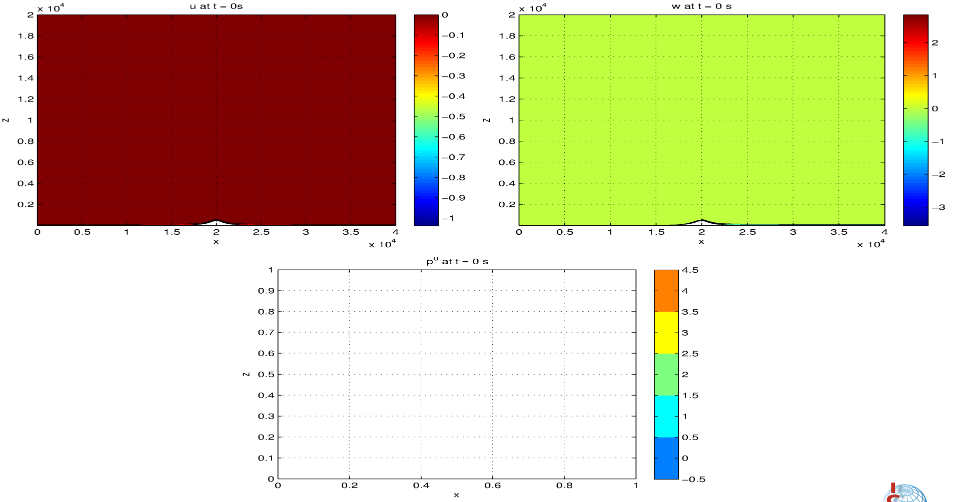
60 $\times$ 50 elements, $p^{\pi} = 4$, $p^{u} = 5$, $\Delta t = 2$ s, $C_V \approx 25$, $C_H \approx 13$.



Nonlinear nonhydrostatic lee waves: adaptive run

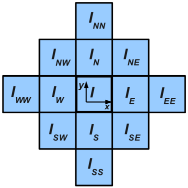
100×50 elements, $p^{\pi} = p^{u} = 4$, $\Delta t = 2$ s, $C_V \approx 25$, $C_H \approx 13$.

(maximum space resolution 200m)

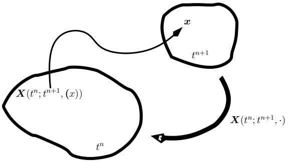


Main challenges towards p-SISLDG parallelization

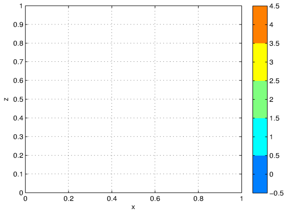
- semi-implicit stencil requires communication



- semi-Lagrangian advection stencil requires communication as well

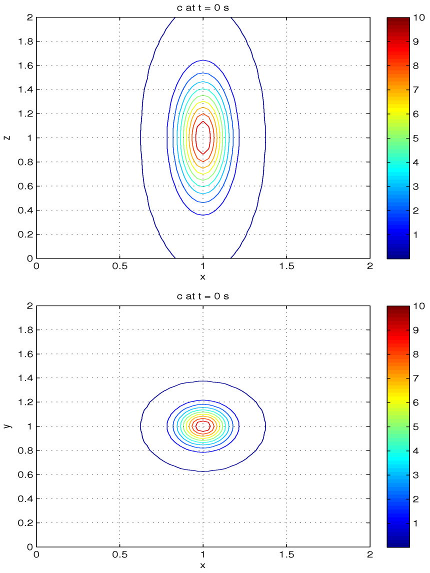


- dynamic $p-$adaptivity requires dynamic load balancing



Where is the 3D dynamical core ??

Preliminary results with the 3D dycore



Solid body advection of a tracer concentration:

Eulerian formulation,

explicit Runge-Kutta 4 time integrator.

Towards a three dimensional dynamical core

- Up to now, no general HPC infrastructures for efficient and natively p-adaptive implementations of DG methods on massively parallel were available;
- hence a new design of the three dimensional code was needed by using
  - indirect adressing for the elements;
  - direct adressing for element degrees of freedom;
  - advanced data types (object-oriented programming);
  - global arrays of pointers to local data structures to avoid the use of linked lists and to make easier the migration of elements between processes;
- this is the basis for the parallel programming design of p-SISLDG around these criteria:
  - SPMD style programming (like MPI + OpenMP);
  - dynamic load balancing based for example on the use of Space Filling Curves (SFCs) (under evaluation as other options like collection of adaptive octrees);
  - overlap between computation and communication;
  - adoption of standards / reduction use of external libraries (for example by use of Fortran coarrays).

Open issues and future perspectives

- Improvement of the linear solver for the SI step: hierachical Krylov solver;
- introduction of the SL discretization of diffusive terms; (see L. Bonaventura and R. Ferretti, SIAM J. Sci. Comp. 2014);
- development of a conservative fully semi-Lagrangian version (along the lines of M. Restelli, M. Bonaventura, R. Sacco, *J. Comput. Phys.*, 2006);
- from next October available a new resource (provided by Politecnico di Milano and OGS whithin the HPC-TRES framework) working on related topics.

Conclusions

-
  - a novel TR-BDF2-based SISL discretization has been presented within the DG framework for the rotating SWE as well as for the Euler equations on a vertical slice, that can be effectively applied to all geophysical scale flows.
  - The resulting scheme is
    - unconditionally stable,
    - full second order accurate in time,
    - arbitrary high order in space,
    - adapting the number of degrees of freedom in each element in order to balance accuracy and computational cost,
    - extended to deformed meshes with no grid imprinting and extendable to arbitrary non-structured even non-conforming meshes,
    - equipped with mass conservative version,
    - multiscale i.e. the same unified model (and therefore architecture) can be successfully run at a range of scales from global to regional (self generation of BC's).
  - Numerical experiments confirm the potential of the proposed formulation.
  - Parallel 3D version at advanced stage of development (nontrivial challenge from the HPC side).

Conclusions

- - a novel TR-BDF2-based SISL discretization has been presented within the DG framework for the rotating SWE as well as for the Euler equations on a vertical slice, that can be effectively applied to all geophysical scale flows.
  - The resulting scheme is
    - unconditionally stable,
    - full second order accurate in time,
    - arbitrary high order in space,
    - adapting the number of degrees of freedom in each element in order to balance accuracy and computational cost,
    - extended to deformed meshes with no grid imprinting and extendable to arbitrary non-structured even non-conforming meshes,
    - equipped with mass conservative version,
    - multiscale i.e. the same unified model (and therefore architecture) can be successfully run at a range of scales from global to regional (self generation of BC's).
  - Numerical experiments confirm the potential of the proposed formulation.
  - Parallel 3D version at advanced stage of development (nontrivial challenge from the HPC side).
- THANK YOU FOR YOUR ATTENTION!

References

- Tumolo G., Bonaventura L., and Restelli M. 2013. A semi-implicit, semi-Lagrangian, p-adaptive discontinuous Galerkin method for the shallow water equations. *Journal of Computational Physics.* Vol. 232, pp. 46-67

- Tumolo G., Bonaventura L. 2015. A semi-implicit, semi-Lagrangian discontinuous Galerkin framework for adaptive numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society.* DOI: 10.1002/qj.2544.

- Tumolo G. 2016. A mass conservative TR-BDF2 semi-implicit semi-Lagrangian DG discretization of the shallow water equations on general structured meshes of quadrilaterals. *Communications in Applied and Industrial Mathematics.* in press.

- Bonaventura L., Ferretti R. 2014. Semi-Lagrangian methods for parabolic problems in divergence form. *SIAM Journal on Scientific Computing*, DOI:10.1137/140969713.

- Restelli M., Bonaventura L., Sacco R. 2006. A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flows. *Journal of Computational Physics*, Vol. 216, pp. 195-215.

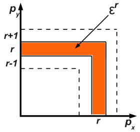
Dynamic p-adaptation: the strategy

- p-adaptivity easier by the use of modal bases: here tensor products of Legendre polynomials;
- hence, the representation for a model variable α becomes ($l = (l_x, l_y)$ multi-index):

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

- and its 2-norm is given by (in planar geometry):

$$\mathcal{E}_l^{tot} = \sum_{k,l=1}^{p_l^\alpha+1} \alpha_{l,k,l}^2 = \sum_{r=1}^{p_l^\alpha+1} \mathcal{E}_l^r, \quad \mathcal{E}_l^r := \sum_{max(k,l)=r} \alpha_{l,k,l}^2,$$



- while the quantity $w_l^r = \sqrt{\frac{\mathcal{E}_l^r}{\mathcal{E}_l^{tot}}}$ will measure the relative 'weight' of the $r-$ degree modes
- *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.*