A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed sphere

Peter H. Lauritzen[1] Ramachandran D. Nair[2] Paul A. Ullrich[3]

[1]Climate and Global Dynamics Division
National Center for Atmospheric Research (NCAR)

[2]Institute for Mathematics Applied to Geosciences (IMAGe)
National Center for Atmospheric Research (NCAR)

[3]Department of Atmospheric, Oceanic and Space Sciences
University of Michigan

Solution of Partial Differential Equations on the Sphere, Santa Fe

Overview

- Motivation
- Derivation of scheme in Cartesian geometry:
  - Semi-Lagrangian finite-volume method
  - Conversion of area integrals into line-integrals: Divergence theorem
  - Results from idealized tests in Cartesian geometry
- Extension to the cubed-sphere:
  - Divergence theorem in gnomonic (central) coordinates
  - Line-integrals along great-circle arcs
  - Results from idealized tests in spherical geometry
- Final Remarks

Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:

Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:
  - Example, the chemistry version of NCAR's Community Atmosphere Model (CAM) model uses $\mathcal{O}(100)$ prognostic tracers.

Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:
  - Example, the chemistry version of NCAR's Community Atmosphere Model (CAM) model uses $\mathcal{O}(100)$ prognostic tracers.
  - Cost of running the dynamics+tracers is dominated by computational cost of transporting tracers.

Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:
  - Example, the chemistry version of NCAR's Community Atmosphere Model (CAM) model uses $\mathcal{O}(100)$ prognostic tracers.
  - Cost of running the dynamics+tracers is dominated by computational cost of transporting tracers.
  - Some solutions:

Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:
  - Example, the chemistry version of NCAR's Community Atmosphere Model (CAM) model uses $\mathcal{O}(100)$ prognostic tracers.
  - Cost of running the dynamics+tracers is dominated by computational cost of transporting tracers.
  - Some solutions:
    - Use computationally very cheap scheme

Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:
  - Example, the chemistry version of NCAR's Community Atmosphere Model (CAM) model uses $\mathcal{O}(100)$ prognostic tracers.
  - Cost of running the dynamics+tracers is dominated by computational cost of transporting tracers.
  - Some solutions:
    - Use computationally very cheap scheme
    - Incremental remapping (Dukowicz and Baumgardner 2000): 'High' startup cost (geometry calculations), however, each additional tracer adds only relatively small cost: **'Get extra accuracy for free!'**

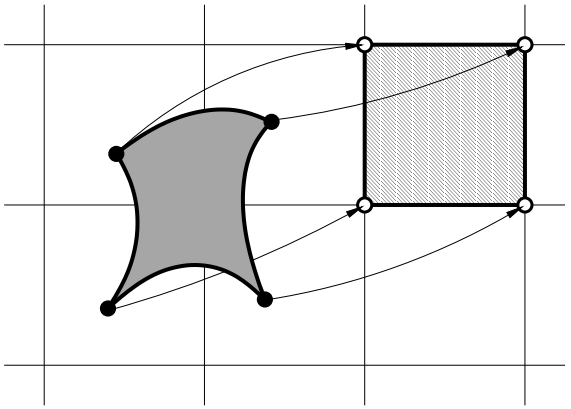
Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:
  - Example, the chemistry version of NCAR's Community Atmosphere Model (CAM) model uses $\mathcal{O}(100)$ prognostic tracers.
  - Cost of running the dynamics+tracers is dominated by computational cost of transporting tracers.
  - Some solutions:
    - Use computationally very cheap scheme
    - Incremental remapping (Dukowicz and Baumgardner 2000): 'High' startup cost (geometry calculations), however, each additional tracer adds only relatively small cost: **'Get extra accuracy for free!'**
    - Sub-cycling tracers (by accumulating fluxes)

Motivation

- Highly desirable that tracer transport scheme is efficient and adaptable for a large number of tracers:
  - Example, the chemistry version of NCAR's Community Atmosphere Model (CAM) model uses $\mathcal{O}(100)$ prognostic tracers.
  - Cost of running the dynamics+tracers is dominated by computational cost of transporting tracers.
  - Some solutions:
    - Use computationally very cheap scheme
    - Incremental remapping (Dukowicz and Baumgardner 2000): 'High' startup cost (geometry calculations), however, each additional tracer adds only relatively small cost: **'Get extra accuracy for free!'**
    - Sub-cycling tracers (by accumulating fluxes)
  - CSLAM follows the incremental remapping idea but higher-order and semi-Lagrangian formulation (although flux form possible)

Continuous equations



Consider 2D transport equation for a passive tracer:

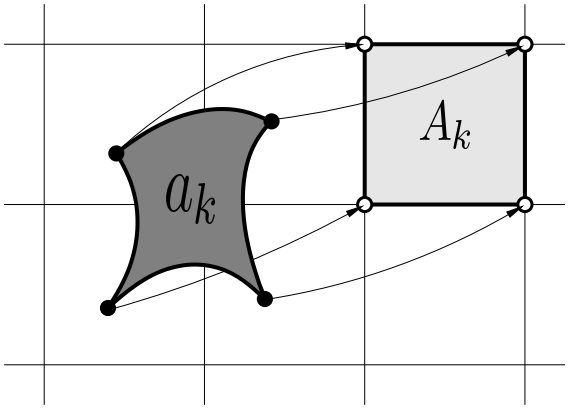
$$\frac{d}{dt}\int_{A(t)} \psi \, dA = 0, \tag{1}$$

where ψ density and $A(t)$ arbitrary Lagrangian area. A temporal discretization of (1) along the characteristics is

$$\int_{A(t+\Delta t)} \psi \, dA = \int_{A(t)} \psi \, dA, \tag{2}$$

where Δt is the time-step size.

Semi-Lagrangian finite-volume scheme



In an upstream semi-Lagrangian method:

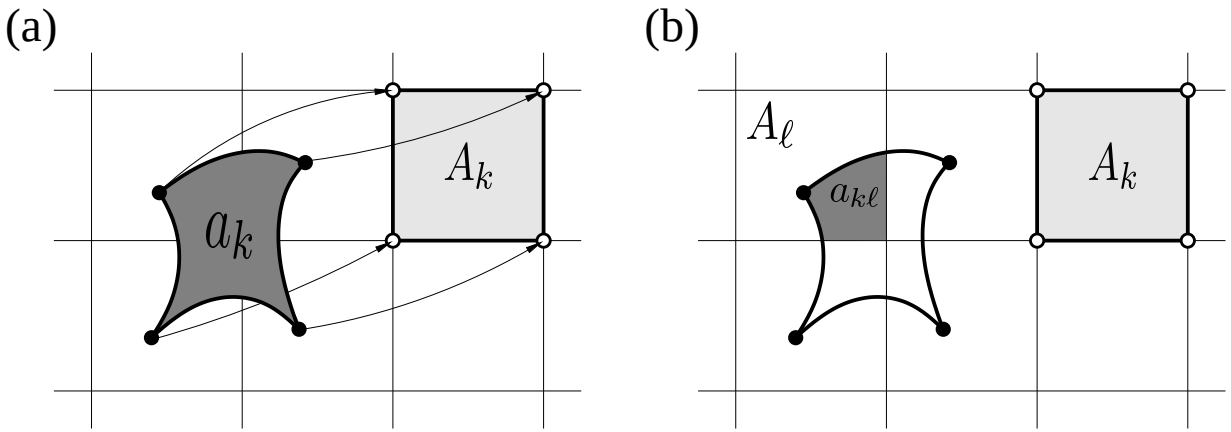
- $A(t + \Delta t)$ = static grid cell = A with area ΔA
- $A(t)$ = upstream deformed cell = a with area δa

Discretized transport equation for cell k can be written as

$$\overline{\psi}_k^{n+1} \, \Delta A_k = \overline{\psi_k^*}^n \, \delta a_k \tag{3}$$

where $\overline{\psi}_k^{n+1}$ and $\overline{\psi_k^*}^n$ is average tracer density in cell k at time-level $n + 1$ and n, respectively.

Semi-Lagrangian finite-volume scheme



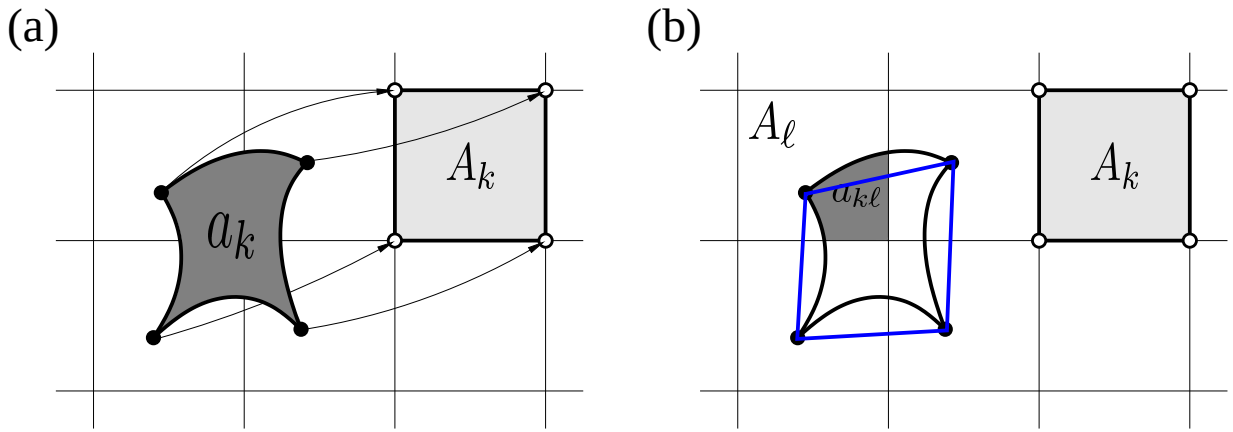
- The integral over the upstream cell can be broken up into the sum of integrals of $f_\ell(x,y)$ over non-empty overlap regions $a_{k\ell}$:

$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi}_k^{*n}\,\delta a_k = \sum_{\ell=1}^{L_k} \iint_{a_{k\ell}} f_\ell(x,y)\,dx\,dy, \tag{4}$$

where $f_\ell(x,y)$ sub-grid-scale reconstruction in A_ℓ and

$$a_{k\ell} = a_k \cap A_\ell, \quad a_{k\ell} \neq \emptyset; \quad \ell = 1,\ldots,L_k. \tag{5}$$

Approximation to upstream cell



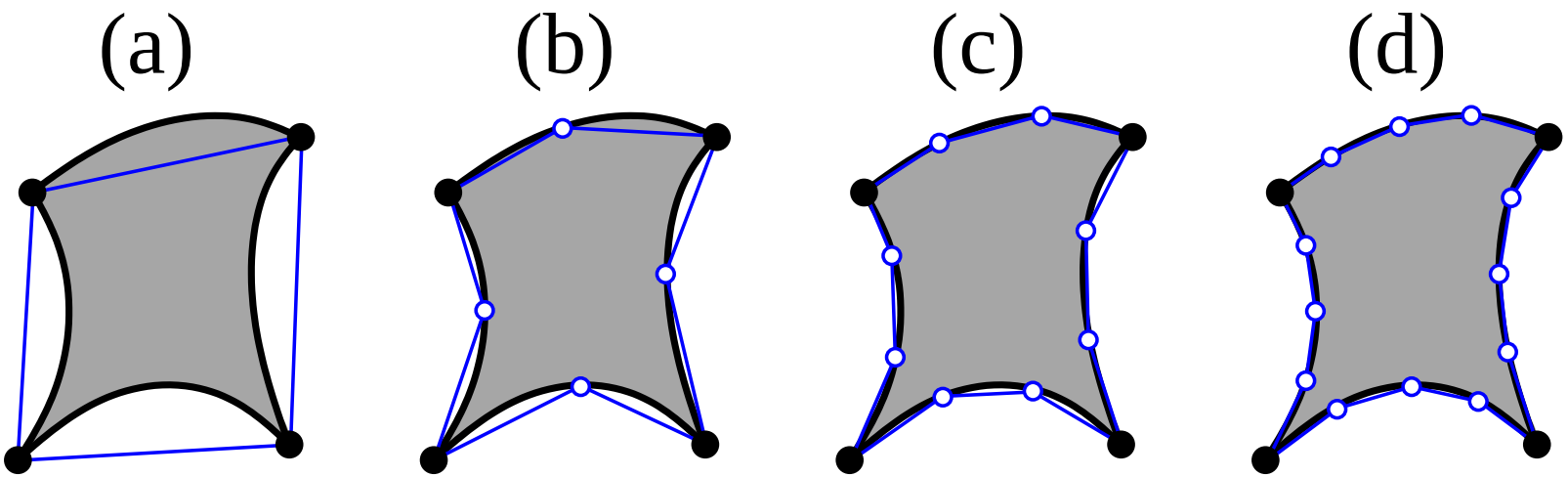
- The integral over the upstream cell can be broken up into the sum of integrals of $f_\ell(x,y)$ over non-empty overlap regions $a_{k\ell}$:

$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi}_k^{*n}\,\delta a_k = \sum_{\ell=1}^{L_k} \iint_{a_{k\ell}} f_\ell(x,y)\,dx\,dy, \tag{6}$$

where $f_\ell(x,y)$ sub-grid-scale reconstruction in A_ℓ and

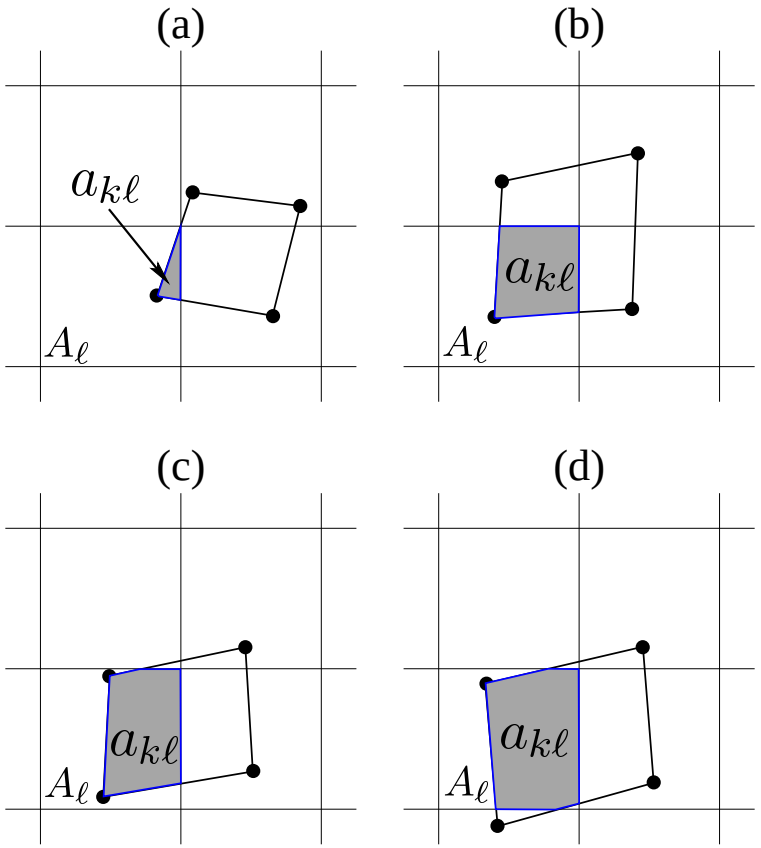
$$a_{k\ell} = a_k \cap A_\ell, \quad a_{k\ell} \neq \emptyset; \quad \ell = 1,\ldots,L_k. \tag{7}$$

Aside: More accurate approximation to upstream cell



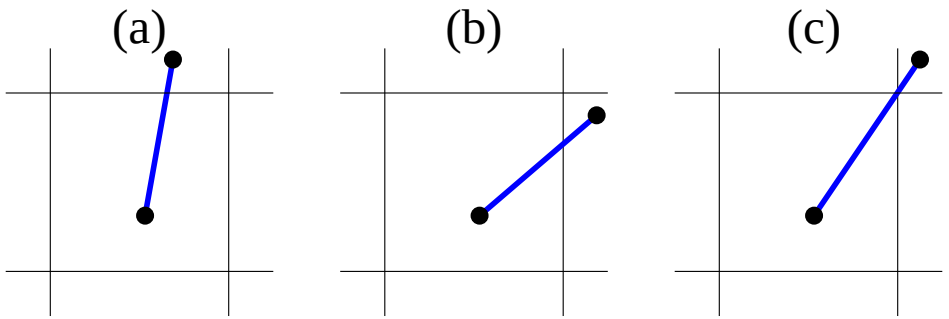
- To improve the approximation to the upstream cell one may introduce (b) one, (c) two or (d) three Lagrangian points along the cell sides (unfilled circles) and connect these by straight line segments to converge towards the exact upstream cell boundary.
- Would increase start-up cost of scheme, however, it would add no cost for each additional tracer transported!

Semi-Lagrangian finite-volume scheme



- The sub-domains $a_{k\ell}$ over which one must integrate can have many possible shapes. The practical difficulty in developing analytical integrals that cover all possible cases is, in general, somewhat complicated but not impossible (Rančić, 1992).

Semi-Lagrangian finite-volume scheme

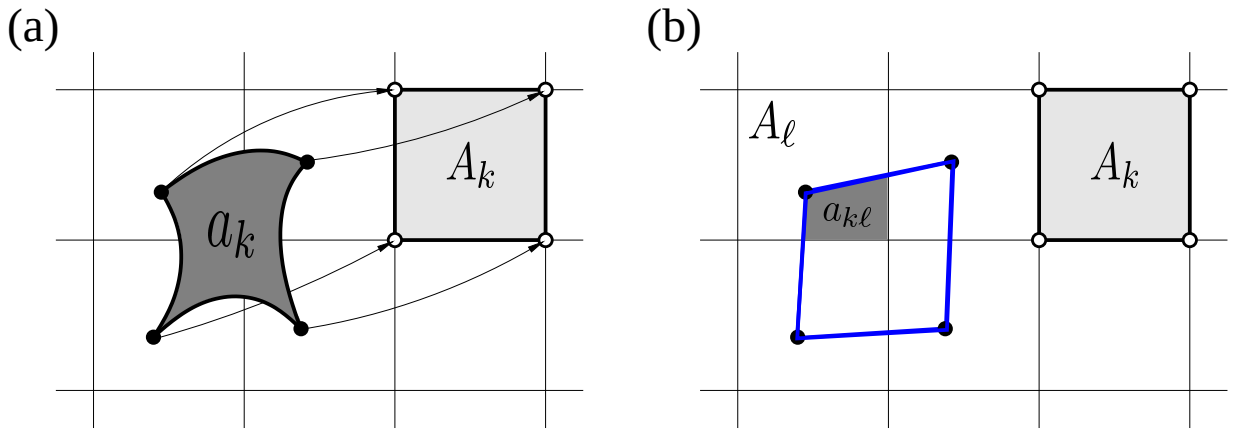


- Instead the problem can be greatly simplified by converting the area-integrals into line-integrals by appropriate use of the divergence theorem (Dukowicz 1984, Margolin and Shashkov 2003)

  Note: 2^{nd}-order version of Cartesian CSLAM = donor cell scheme of Margolin and Shashkov (2003)!

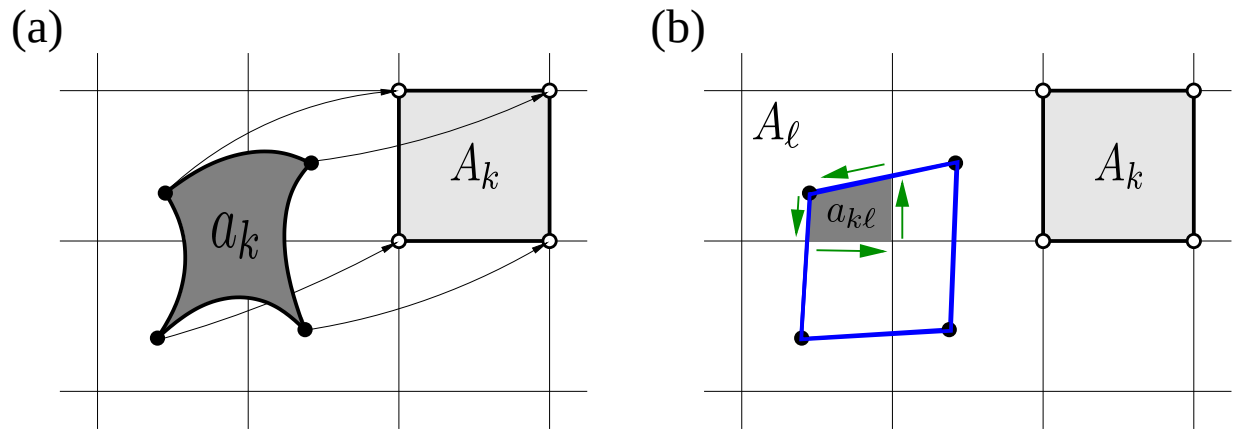
- Instead of dealing with numerous overlap area ‘types’ one only needs to deal with 3 types of crossings with coordinate lines.

Upstream integral



$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi_k^*}^{n}\,\delta a_k \quad = \quad \sum_{\ell=1}^{L_k} \iint_{a_{k\ell}} f_\ell(x,y)\,dx\,dy\,,$$

Upstream integral

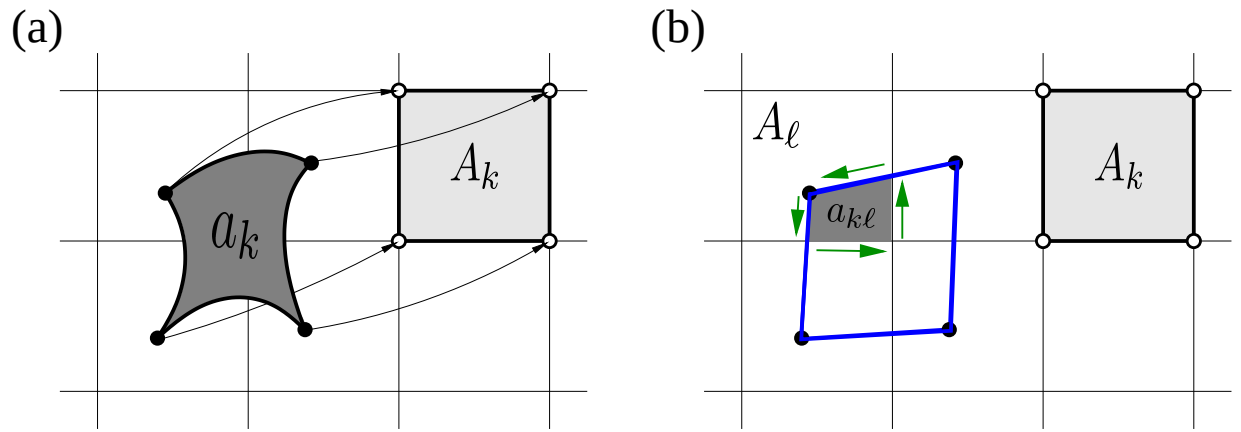


$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi_k^*}^{n}\,\delta a_k \quad = \quad \sum_{\ell=1}^{L_k} \oint_{\partial a_{k\ell}} \left[P\,dx + Q\,dy\right],$$

where $\partial a_{k\ell}$ is the boundary of $a_{k\ell}$ and

$$-\frac{\partial P}{\partial y} + \frac{\partial Q}{\partial x} = f_\ell(x, y)$$

Upstream integral



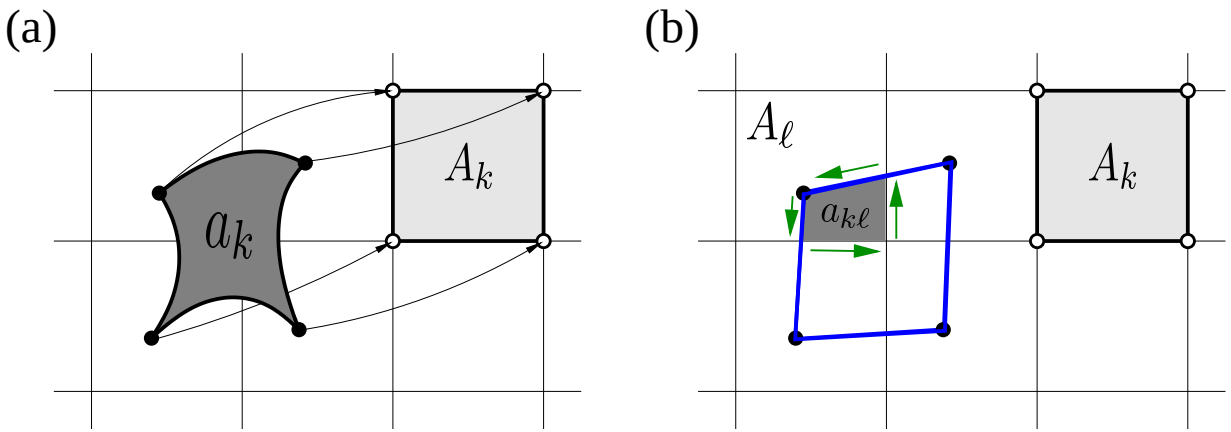
$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi_k^*}^{n}\,\delta a_k \quad = \quad \sum_{\ell=1}^{L_k} \oint_{\partial a_{k\ell}} \left[P\,dx + Q\,dy\right],$$

where $\partial a_{k\ell}$ is the boundary of $a_{k\ell}$ and

$$-\frac{\partial P}{\partial y} + \frac{\partial Q}{\partial x} = f_\ell(x,y) = \sum_{i+j\le 2} c_\ell^{(i,j)} x^i y^j, \quad i,j \in \{0,1,2\}.$$

Here we use PPM (Colella and Woodward 1984) in each coordinate direction and cross term approximation as in Jablonowski (2004).

Upstream integral - weights



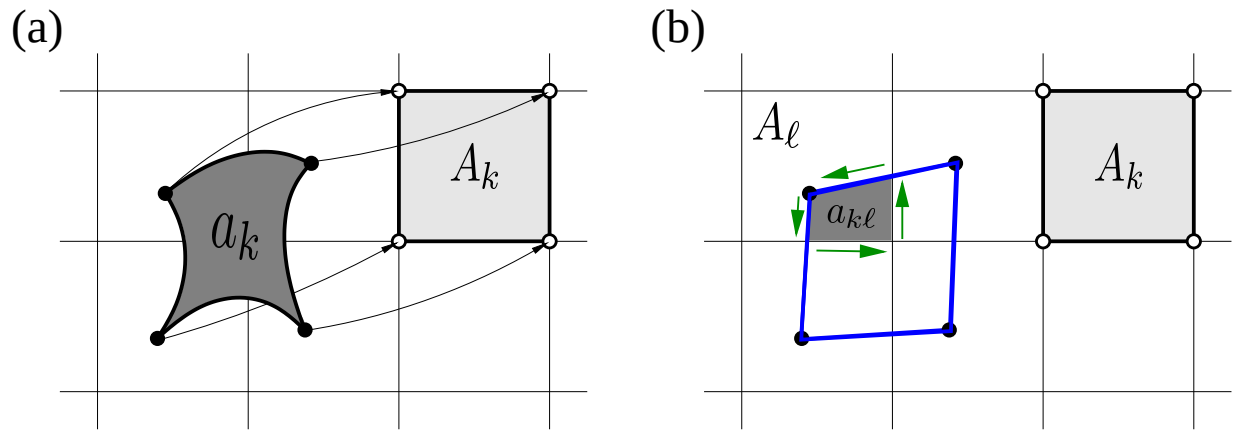
$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi}_k^{*n}\,\delta a_k \quad = \quad \sum_{\ell=1}^{L_k}\left[\sum_{i+j\le 2} c_\ell^{(i,j)} w_{k\ell}^{(i,j)}\right],$$

where weights $w_{k\ell}^{(i,j)}$ are functions of the coordinates of the vertices of $a_{k\ell}$ denoted $(x_{k\ell,h}, y_{k\ell,h})$, $h = 1, \ldots, N_h$:

$$w_{k\ell}^{(0,0)} \quad = \quad \frac{1}{2}\sum_{h=1}^{N_h}\left(x_{k\ell,h} + x_{k\ell,h-1}\right)\left(y_{k\ell,h} - y_{k\ell,h-1}\right),$$

$$\ldots$$

Upstream integral - weights



$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi}_k^{*n}\,\delta a_k \quad = \quad \sum_{\ell=1}^{L_k}\left[\sum_{i+j\le 2} c_\ell^{(i,j)} w_{k\ell}^{(i,j)}\right].$$

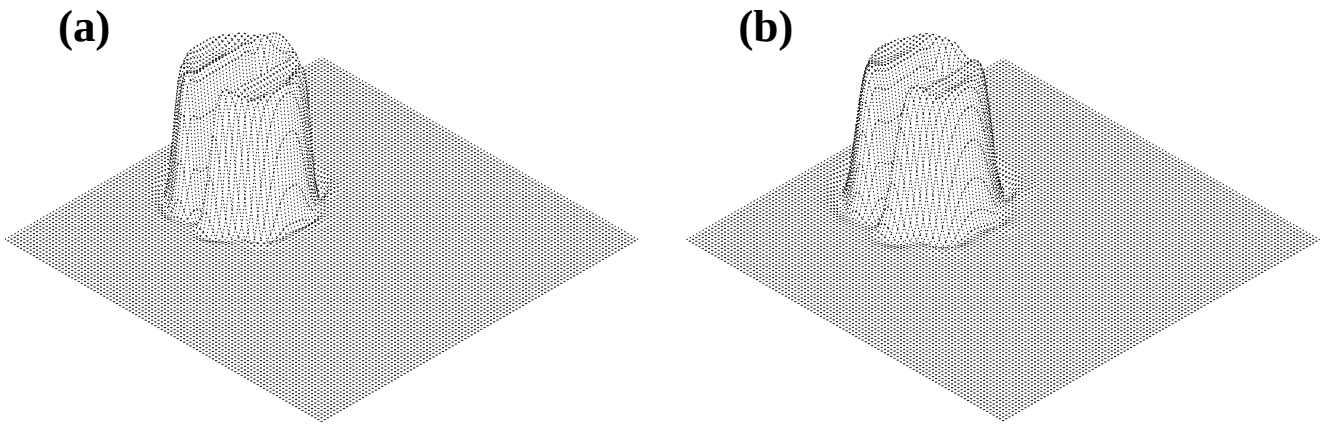
- Weights $w_{k\ell}^{(i,j)}$ can be reused for each additional tracer
  - multi tracer efficiency (Dukowicz and Baumgardner 2000)
- $c_\ell^{(i,j)}$ must be recomputed for every tracer.
  Very efficient monotone second-order reconstruction options exist (Lipscomb and Ringler 2005)

Results for CSLAM - Cartesian

- Bad news: Rigorous 2D scheme (CSLAM) is not worth the extra computational cost for applications in Cartesian geometry (at least not in idealized tests)!

Results: Solid-body advection in Cartesian geometry

6 revolutions of slotted cylinder



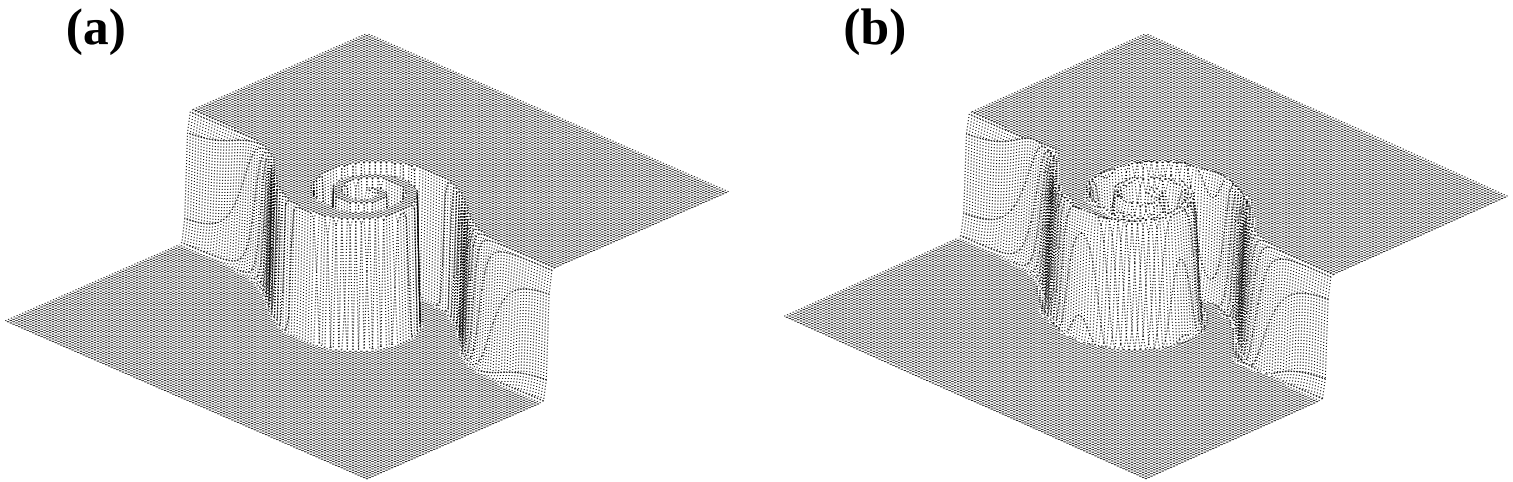
Scheme	RMS	l_1	l_2	l_∞	Reference
CSLAM	**0.0650**	**0.2723**	**0.2734**	**0.6591**	Lauritzen et al. (2009)
CSLAM*	**0.0681**	**0.2943**	**0.2863**	**0.6736**	Lauritzen et al. (2009)
SLICE	0.0673	-	-	-	Zerroukat et al. (2007)
CISL	0.0692	0.2988	0.2908	0.6754	Nair and Machenhauer (2002)

$\star$ = no cross term in the reconstruction function

- No major gain in accuracy by using rigorous 2D approach!
- The cross term in the reconstruction function is only having a marginal effect on accuracy.

Results: Solid-body advection in Cartesian geometry

Idealized cyclogenesis (deformational flow)

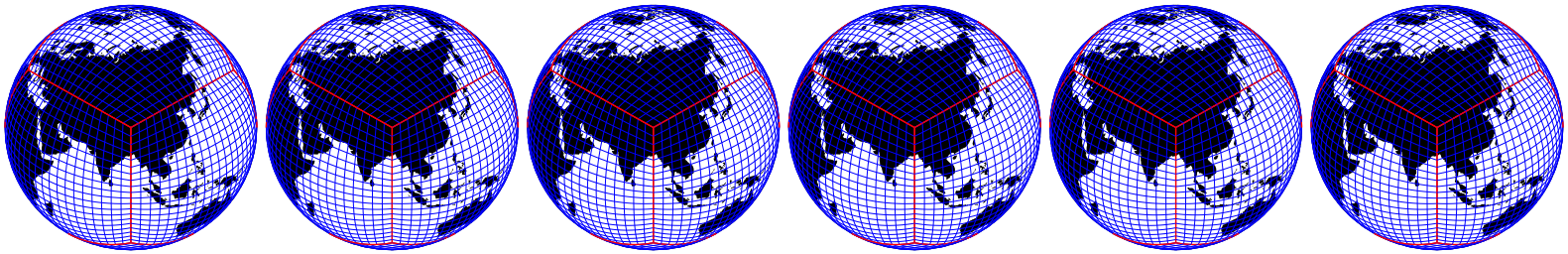


Scheme	RMS	l_1	l_2	l_∞	Reference
CSLAM	**0.0642**	**0.0113**	**0.0646**	**0.8802**	Lauritzen et al. (2009)
CSLAM*	**0.0653**	**0.0116**	**0.0656**	**0.8777**	Lauritzen et al. (2009)
SLICE	0.0693	-	-	-	Zerroukat et al. (2007)
CISL	0.0666	0.0119	0.0670	0.8737	Nair and Machenhauer (2002)

$\star$ = no cross term in the reconstruction function

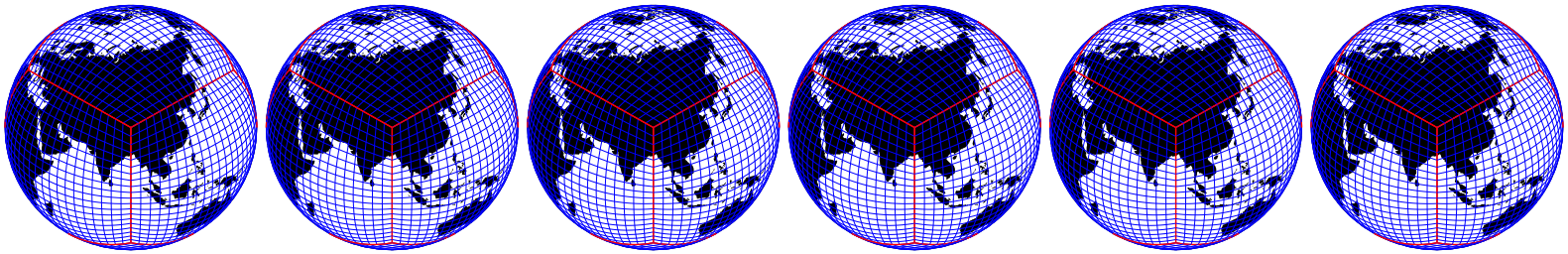
- No major gain in accuracy by using rigorous 2D approach!
- The cross term in the reconstruction function is only having a marginal effect on accuracy.

Extension to the cubed-sphere



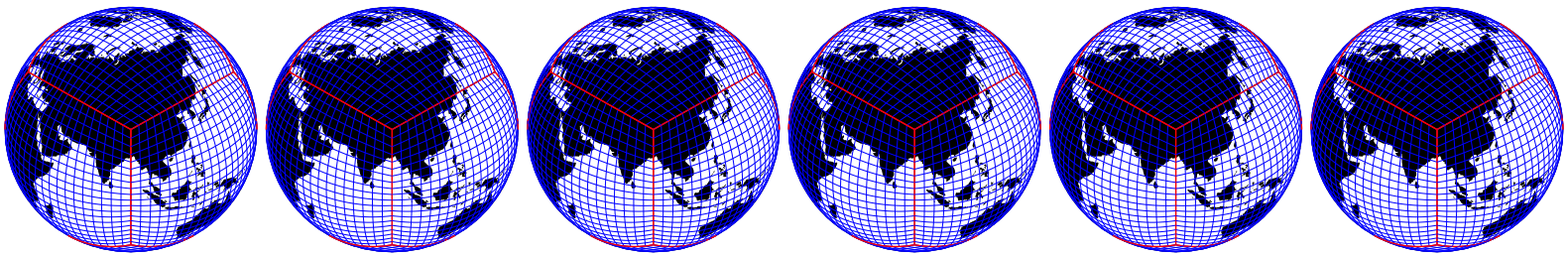
- Is there a ‘payoff’ for using rigorous fully two-dimensional ‘high-order’ CSLAM on cubed-sphere grid?

Extension to the cubed-sphere



- Is there a ‘payoff’ for using rigorous fully two-dimensional ‘high-order’ CSLAM on cubed-sphere grid?
  - Are fully 2D schemes necessary to ‘consistently’ treat the corners of the sphere or can dimensionally split approaches do the job?

Extension to the cubed-sphere

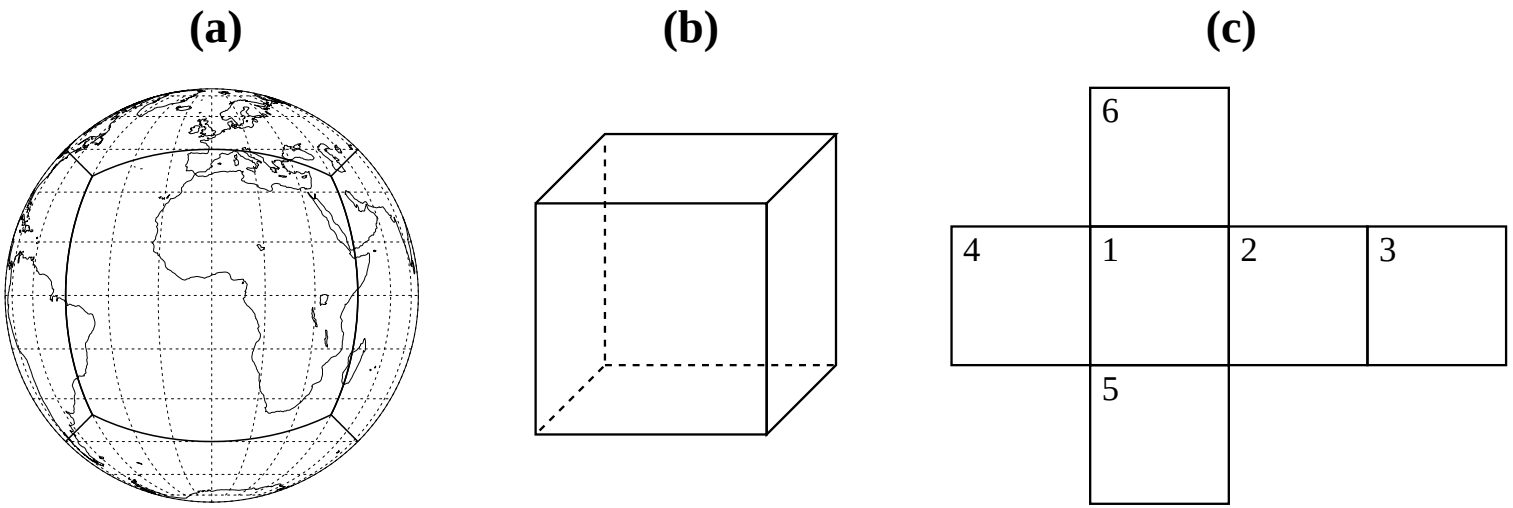


- Is there a ‘payoff’ for using rigorous fully two-dimensional ‘high-order’ CSLAM on cubed-sphere grid?
  - Are fully 2D schemes necessary to ‘consistently’ treat the corners of the sphere or can dimensionally split approaches do the job?
- Will try and answer this question by comparing cubed-sphere CSLAM with cubed-sphere version of the widely used Lin and Rood (1996) scheme:

$$\overline{\psi}_k^{n+1} = \overline{\psi}_k^n + F^x\left[\frac{1}{2}\left(\overline{\psi}^n + f^y\right)\right] + F^y\left[\frac{1}{2}\left(\overline{\psi}^n + f^x\right)\right], \tag{8}$$

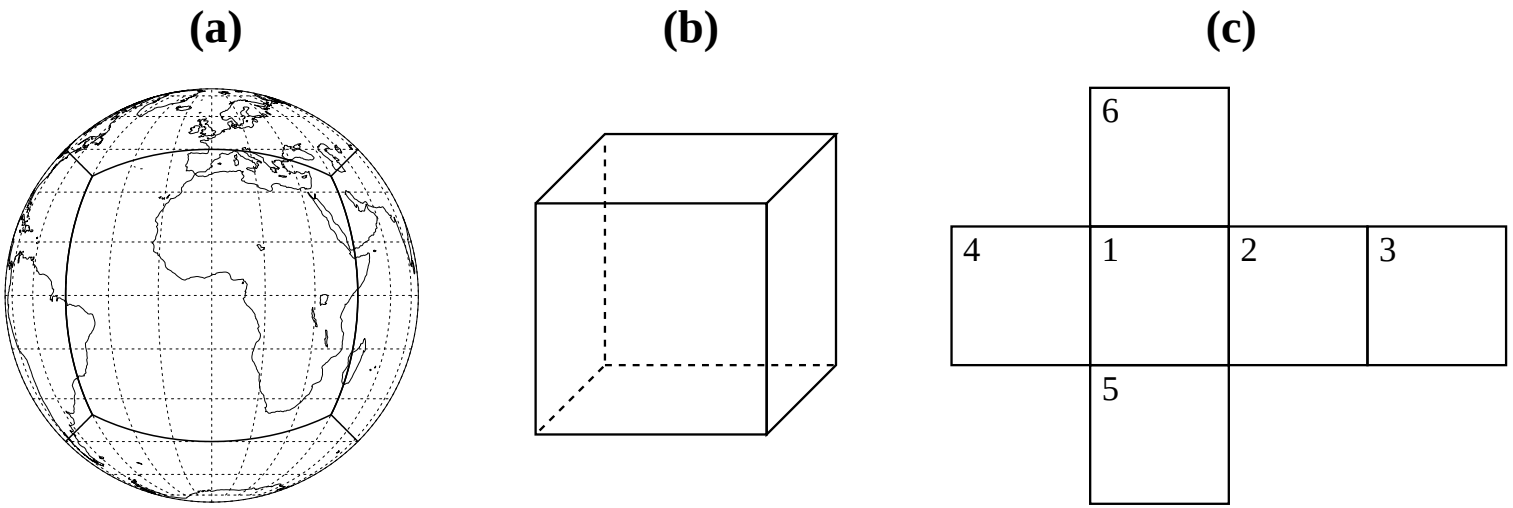
documented in Putman and Lin (2007).

Extension to the cubed-sphere



- Consider equi-angular cubed-sphere grid

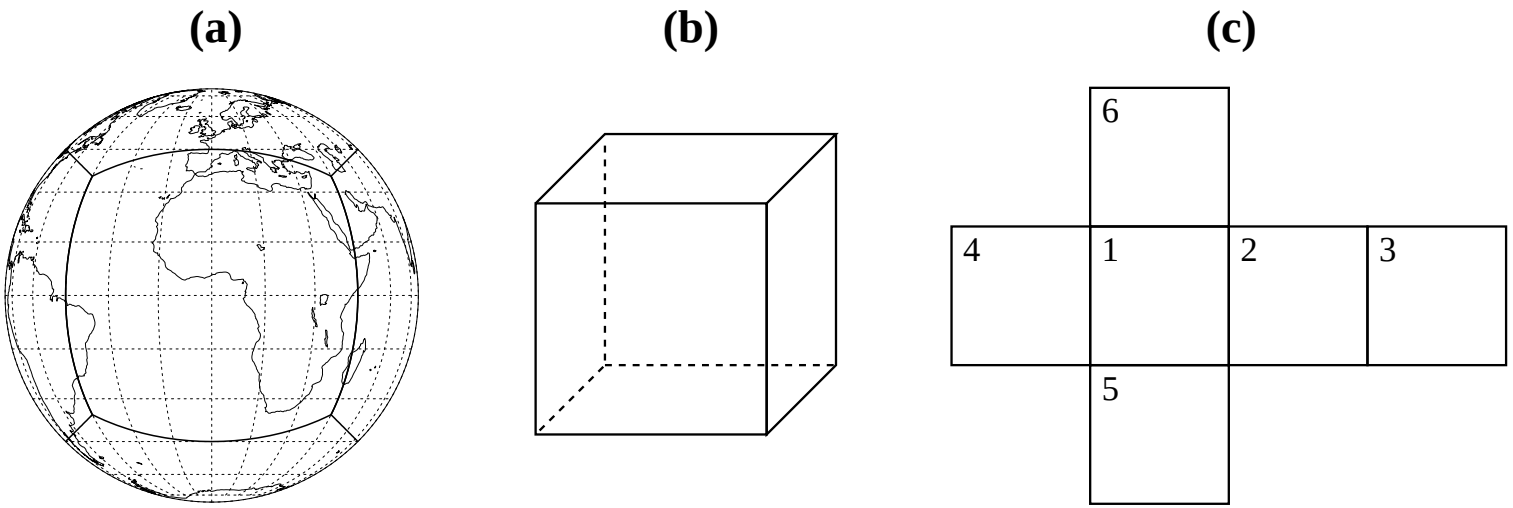
Extension to the cubed-sphere



- Consider equi-angular cubed-sphere grid
- All computations performed on gnomonic (central) projection:

$$x = \tan\alpha \quad \text{and} \quad y = \tan\beta; \quad \alpha, \beta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right], \tag{9}$$

Extension to the cubed-sphere

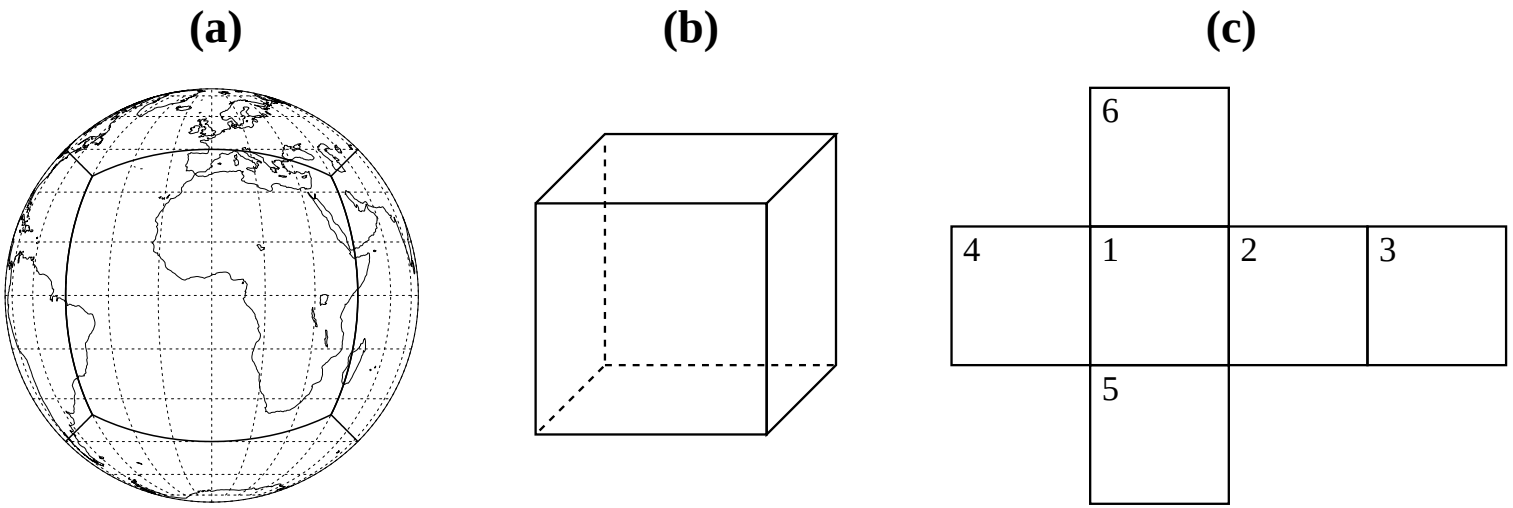


- Consider equi-angular cubed-sphere grid
- All computations performed on gnomonic (central) projection:

$$x = \tan\alpha \quad \text{and} \quad y = \tan\beta; \quad \alpha, \beta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right], \tag{9}$$

- Note that any straight line on the gnomonic projection corresponds to a great-circle arc on the sphere!

Extension to the cubed-sphere

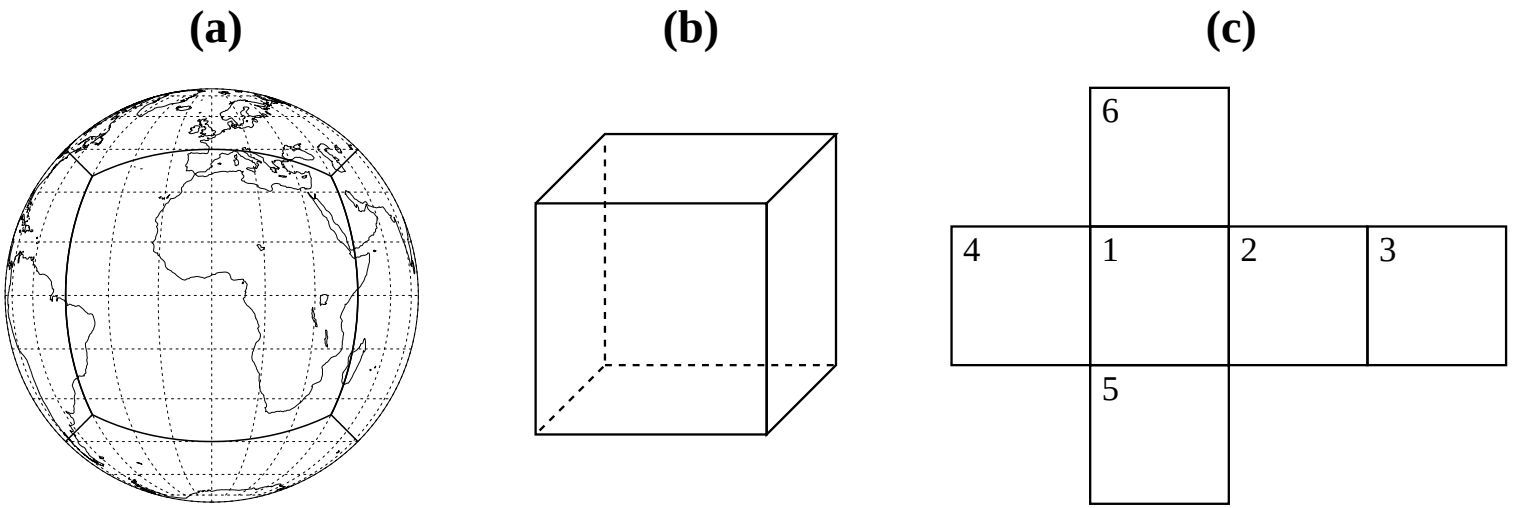


- Consider equi-angular cubed-sphere grid
- All computations performed on gnomonic (central) projection:

$$x = \tan\alpha \quad \text{and} \quad y = \tan\beta; \quad \alpha, \beta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right], \tag{9}$$

- Note that any straight line on the gnomonic projection corresponds to a great-circle arc on the sphere!
- $\Rightarrow$ reuse Cartesian algorithm except

Extension to the cubed-sphere

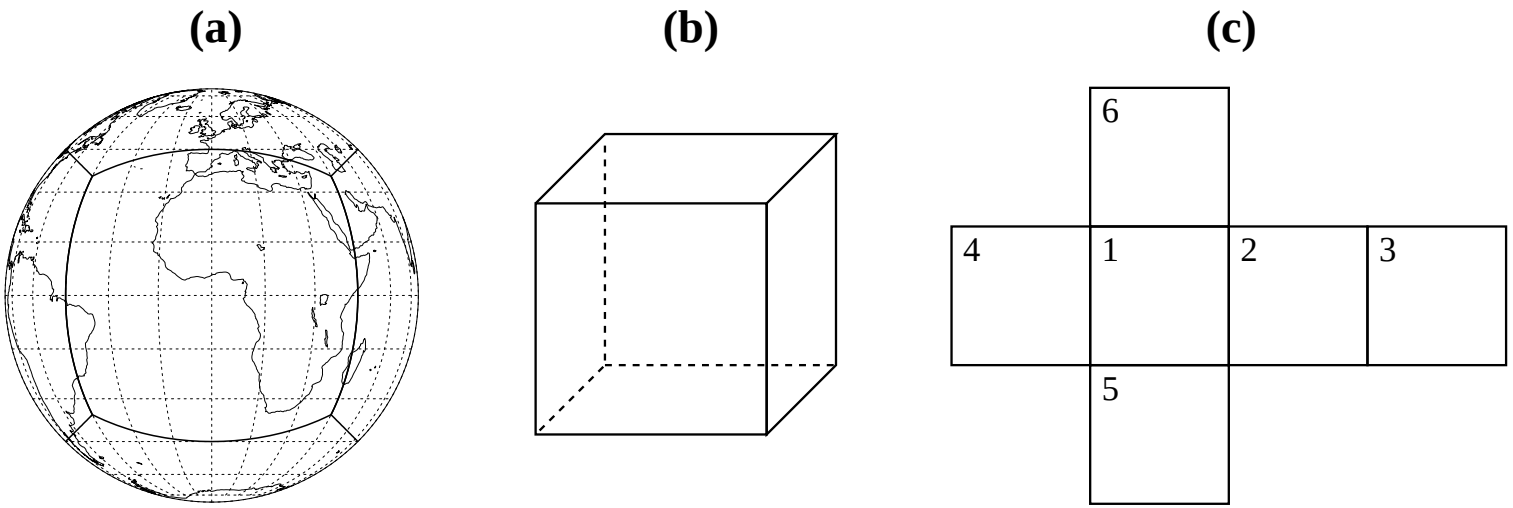


- Consider equi-angular cubed-sphere grid
- All computations performed on gnomonic (central) projection:

$$x = \tan\alpha \quad \text{and} \quad y = \tan\beta; \quad \alpha, \beta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right], \tag{9}$$

- Note that any straight line on the gnomonic projection corresponds to a great-circle arc on the sphere!
- $\Rightarrow$ reuse Cartesian algorithm except
  - Divergence theorem must be converted to gnomonic coordinates (and associated potentials must be found).

Extension to the cubed-sphere



- Consider equi-angular cubed-sphere grid
- All computations performed on gnomonic (central) projection:

$$x = \tan\alpha \quad \text{and} \quad y = \tan\beta; \quad \alpha, \beta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right], \tag{9}$$

- Note that any straight line on the gnomonic projection corresponds to a great-circle arc on the sphere!
- $\Rightarrow$ reuse Cartesian algorithm except
  - Divergence theorem must be converted to gnomonic coordinates (and associated potentials must be found).
  - Consistently couple the panel discretizations for the global domain.

Divergence theorem in gnomonic coordinates

Let Ψ be a vector field with contravariant components Ψ_x and Ψ_y in the direction of the unit basis vectors (e_x, e_y), i.e. $\Psi = \Psi_x e_x + \Psi_y e_y$. Divergence theorem:

$$\int_{A_{k\ell}} \nabla \cdot \Psi dV = - \oint_{\partial a_{k\ell}} \left[\widetilde{\Psi}_x \, dy + \widetilde{\Psi}_y \, dx \right], \tag{10}$$

where

$$\widetilde{\Psi}_x = \frac{\Psi_x}{\rho \sqrt{1+y^2}} \text{ and } \widetilde{\Psi}_y = \frac{\Psi_y}{\rho \sqrt{1+x^2}},$$

with $\rho = \sqrt{1+x^2+y^2}$. The divergence operator is given by

$$\nabla \cdot \Psi = \rho^3 \left[\frac{\partial \widetilde{\Psi}_x}{\partial x} + \frac{\partial \widetilde{\Psi}_y}{\partial y} \right]. \tag{11}$$

Divergence theorem in gnomonic coordinates

A choice of potentials for fully 2D (third-order, '2D PPM') reconstructions:

$$\widetilde{\Psi}_x^{(i,j)}(x,y) = 0, \quad i,j \in \{0,1,2\} \tag{12}$$

$$\widetilde{\Psi}_y^{(0,0)}(x,y) = \frac{1}{1+x^2}\frac{y}{\rho}, \tag{13}$$

$$\widetilde{\Psi}_y^{(1,0)}(x,y) = \frac{1}{1+x^2}\frac{xy}{\rho}, \tag{14}$$

$$\widetilde{\Psi}_y^{(0,1)}(x,y) = -\frac{1}{\rho}, \tag{15}$$

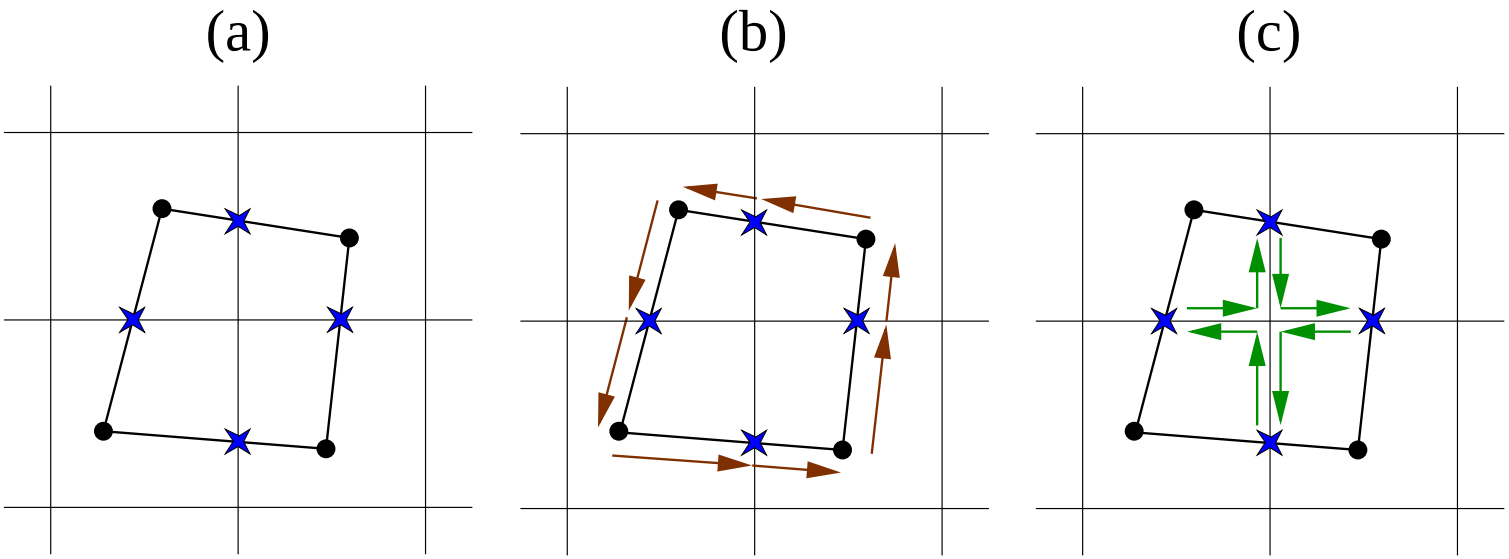
$$\widetilde{\Psi}_y^{(2,0)}(x,y) = \frac{1}{1+x^2}\frac{x^2 y}{\rho}, \tag{16}$$

$$\widetilde{\Psi}_y^{(0,2)}(x,y) = -\frac{y}{\rho} + \operatorname{arcsinh}\left(\frac{y}{\sqrt{1+x^2}}\right), \tag{17}$$

$$\widetilde{\Psi}_y^{(1,1)}(x,y) = -\frac{x}{\rho}, \tag{18}$$

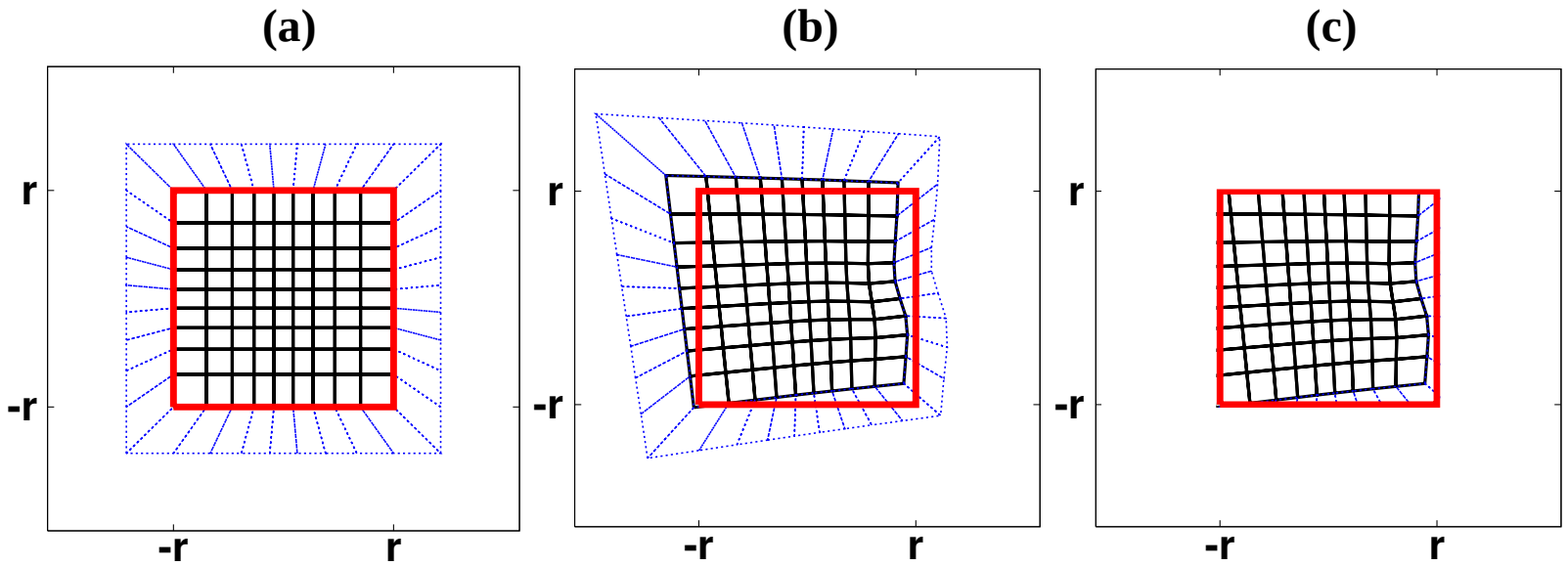
- Monotone filter: Barth and Jespersen (1989) that simply scales the sub-grid scale reconstruction so that its min-max values do not exceed the cell-averages of the neighboring cells.

Divergence theorem in gnomonic coordinates



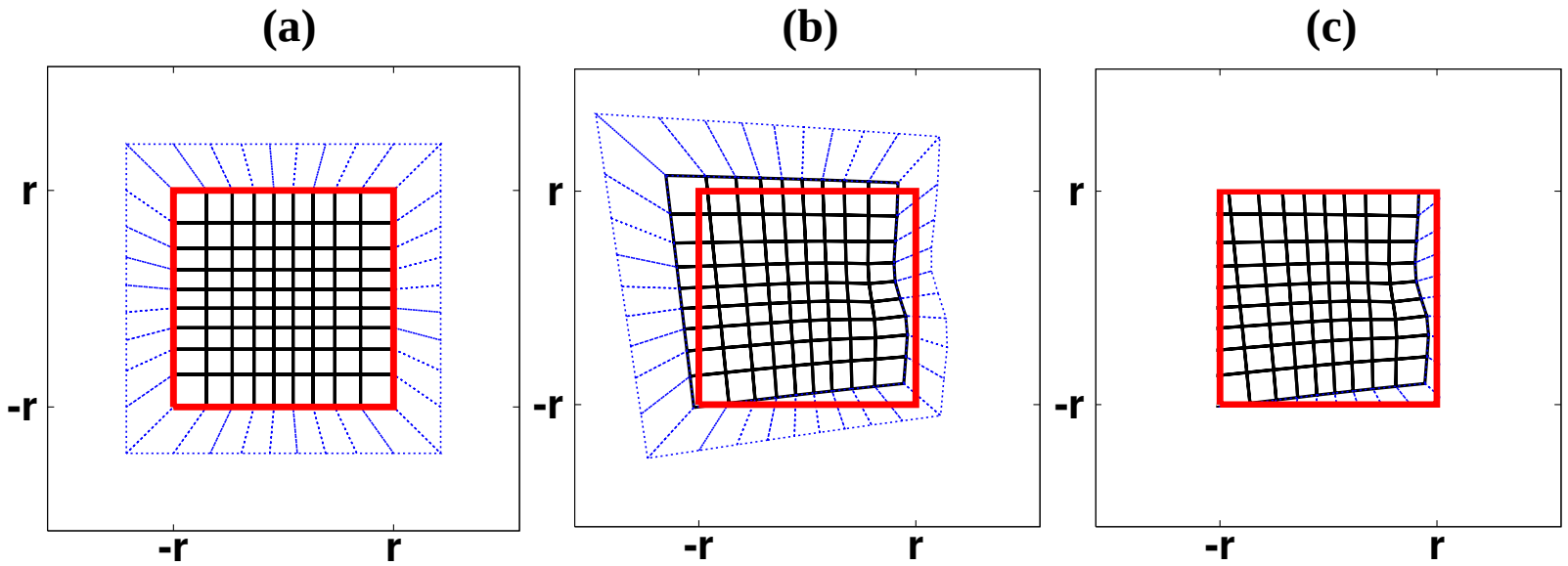
- Integrating the potentials along sides of $a_{k\ell}$:
  - (Fig. c) Along coordinates lines it is possible to compute the line integrals exactly (see Ullrich talk or Ullrich et al. (2009)).
  - (Fig. b) Along lines of arbitrary orientation we use Gaussian quadrature.

Coupling panels



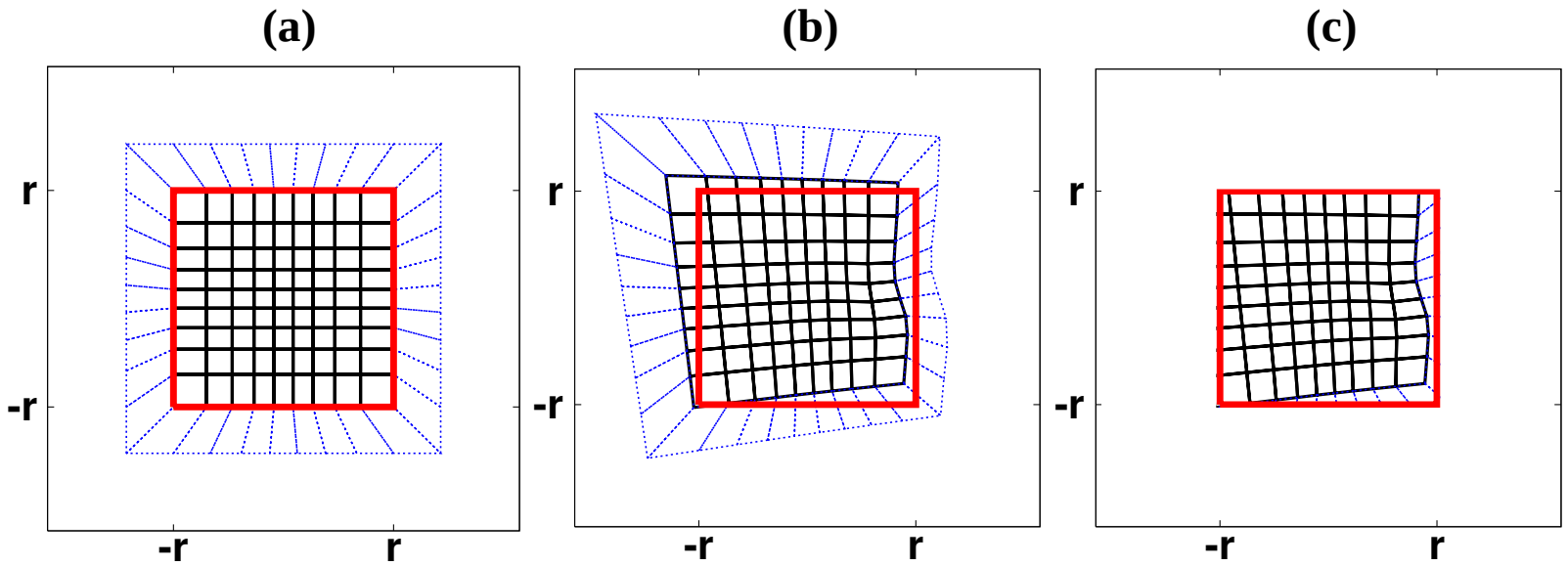
- (a) Halo for panel *p* (on panel *p*'s projection). Note that the cells on neighboring panels are deformed on panel *p*'s projection.

Coupling panels



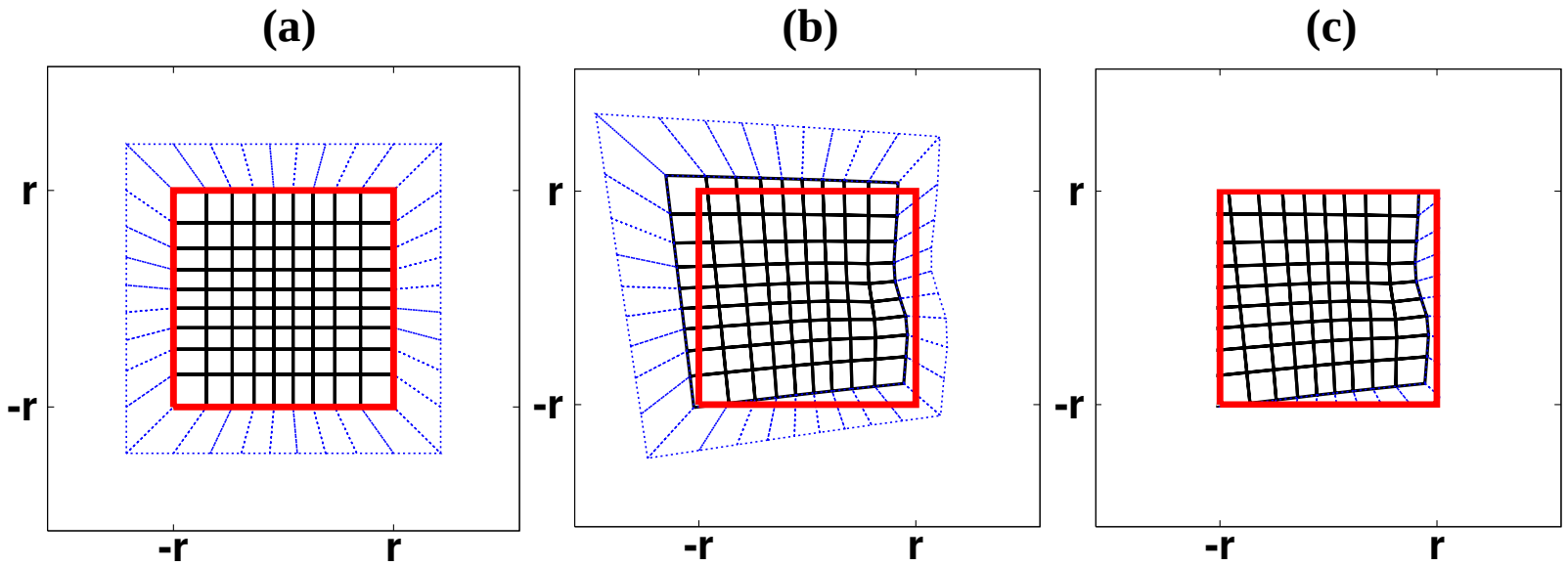
- (a) Halo for panel *p* (on panel *p*'s projection). Note that the cells on neighboring panels are deformed on panel *p*'s projection.
- (b) Compute deformed upstream grid (Figure shows departure grid for moving vortex test case). Note that cells entering from neighboring panels are 'naturally' skewed

Coupling panels



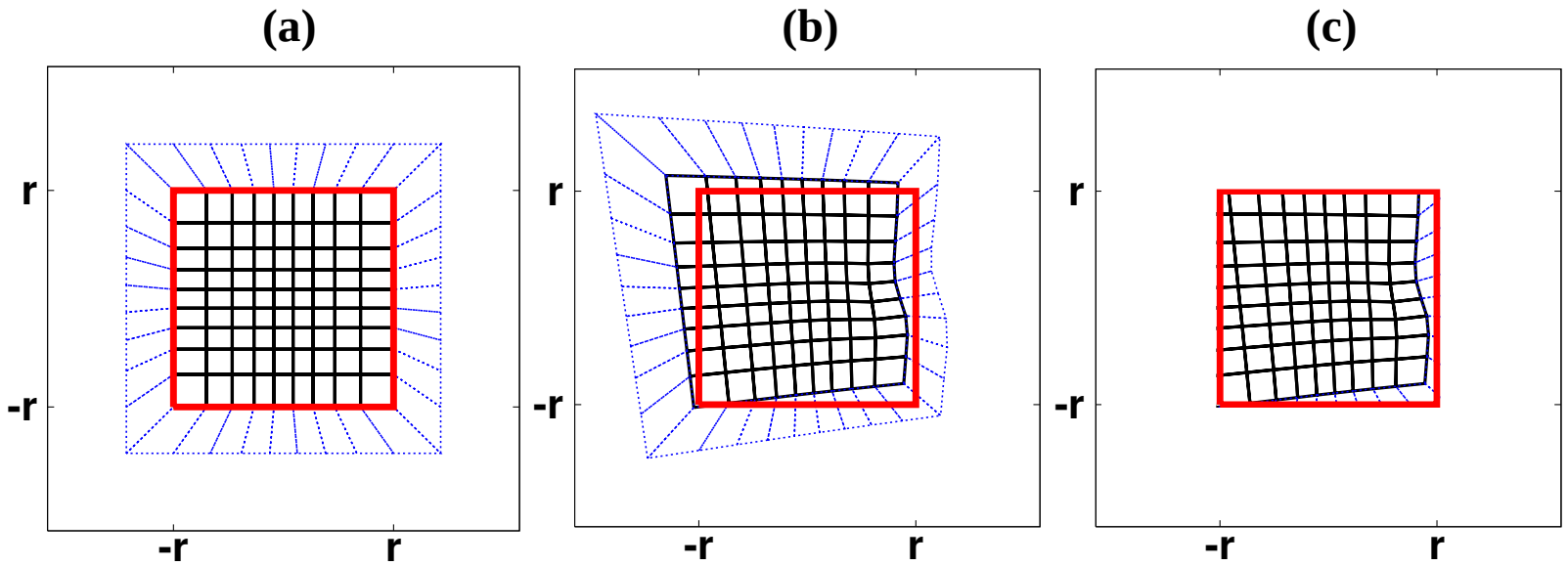
- (a) Halo for panel *p* (on panel *p*'s projection). Note that the cells on neighboring panels are deformed on panel *p*'s projection.
- (b) Compute deformed upstream grid (Figure shows departure grid for moving vortex test case). Note that cells entering from neighboring panels are 'naturally' skewed
- (c) 'Chop off' non-local overlap areas

Coupling panels



- (a) Halo for panel *p* (on panel *p*'s projection). Note that the cells on neighboring panels are deformed on panel *p*'s projection.
- (b) Compute deformed upstream grid (Figure shows departure grid for moving vortex test case). Note that cells entering from neighboring panels are 'naturally' skewed
- (c) 'Chop off' non-local overlap areas
- Do remapping local to panel (for each panel)

Coupling panels



- (a) Halo for panel *p* (on panel *p*'s projection). Note that the cells on neighboring panels are deformed on panel *p*'s projection.
- (b) Compute deformed upstream grid (Figure shows departure grid for moving vortex test case). Note that cells entering from neighboring panels are 'naturally' skewed
- (c) 'Chop off' non-local overlap areas
- Do remapping local to panel (for each panel)
- Collect contributions from neighboring panels.

Results: Solid-body advection on the sphere

Solid-body rotation of cosine bell on the sphere

Dimensions: $N_c = 32$, $\varphi = \pi/4$, $\Delta t = 4050$s, 256 Δt's for one revolution.

Scheme	number of Gauss quadrature points	l_1	l_2	l_∞
CSLAM*-N	**2**	**0.0949**	**0.0536**	**0.0288**
CSLAM-N	**2**	**0.0764**	**0.0414**	**0.0254**
CSLAM-N	3	0.0765	0.0414	0.0255
CSLAM-N	4	0.0765	0.0414	0.0255
CSLAM-N	5	0.0765	0.0414	0.0255

-N = non-monotone (unlimited) version of CSLAM

- Just two quadrature points is sufficient!

Results: Solid-body advection on the sphere

Solid-body rotation of cosine bell on the sphere

Dimensions: $N_c = 32$, $\varphi = \pi/4$, $\Delta t = 4050$s, 256 Δt's for one revolution.

Scheme	number of Gauss quadrature points	l_1	l_2	l_∞
CSLAM*-N	**2**	**0.0949**	**0.0536**	**0.0288**
CSLAM-N	**2**	**0.0764**	**0.0414**	**0.0254**
CSLAM-N	3	0.0765	0.0414	0.0255
CSLAM-N	4	0.0765	0.0414	0.0255
CSLAM-N	5	0.0765	0.0414	0.0255

-N = non-monotone (unlimited) version of CSLAM

- Just two quadrature points is sufficient!
- **Cross term in reconstruction function is very important**

Results: Solid-body advection on the sphere

Solid-body rotation of cosine bell on the sphere

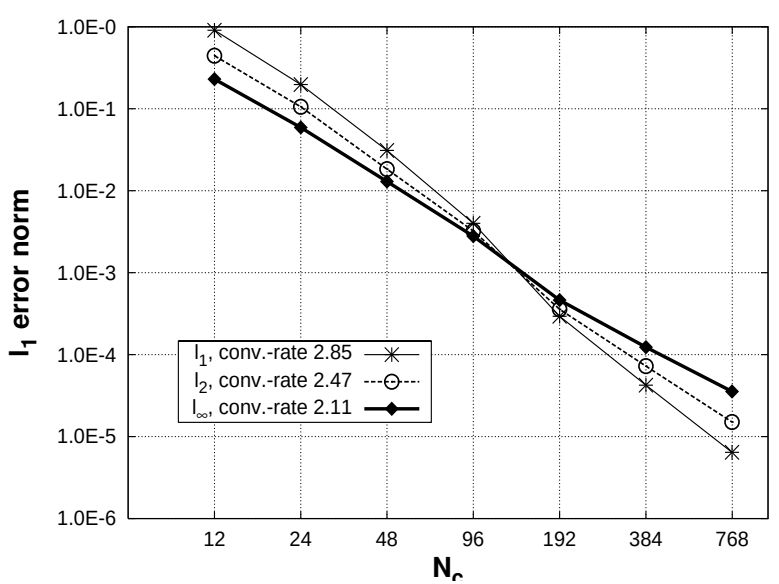
Dimensions: $N_c = 32$, $\varphi = \pi/4$, $\Delta t = 4050$s, 256 Δt's for one revolution.

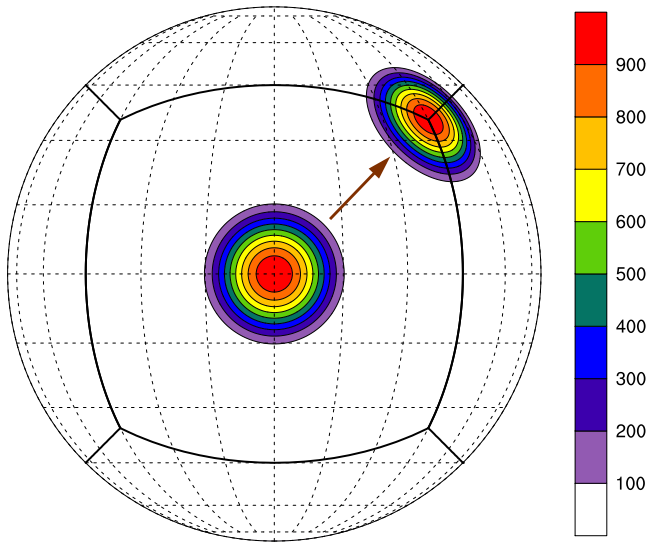
Scheme	number of Gauss quadrature points	l_1	l_2	l_∞
CSLAM*-N	**2**	**0.0949**	**0.0536**	**0.0288**
CSLAM-N	**2**	**0.0764**	**0.0414**	**0.0254**
CSLAM-N	3	0.0765	0.0414	0.0255
CSLAM-N	4	0.0765	0.0414	0.0255
CSLAM-N	5	0.0765	0.0414	0.0255

-N = non-monotone (unlimited) version of CSLAM

- Just two quadrature points is sufficient!
- **Cross term in reconstruction function is very important**
- $\Rightarrow$ By default we include cross term in reconstruction function and use two quadrature points for line integrals except along grid lines (where line-integrals are evaluated exactly).

Solid body advection of cosine bell

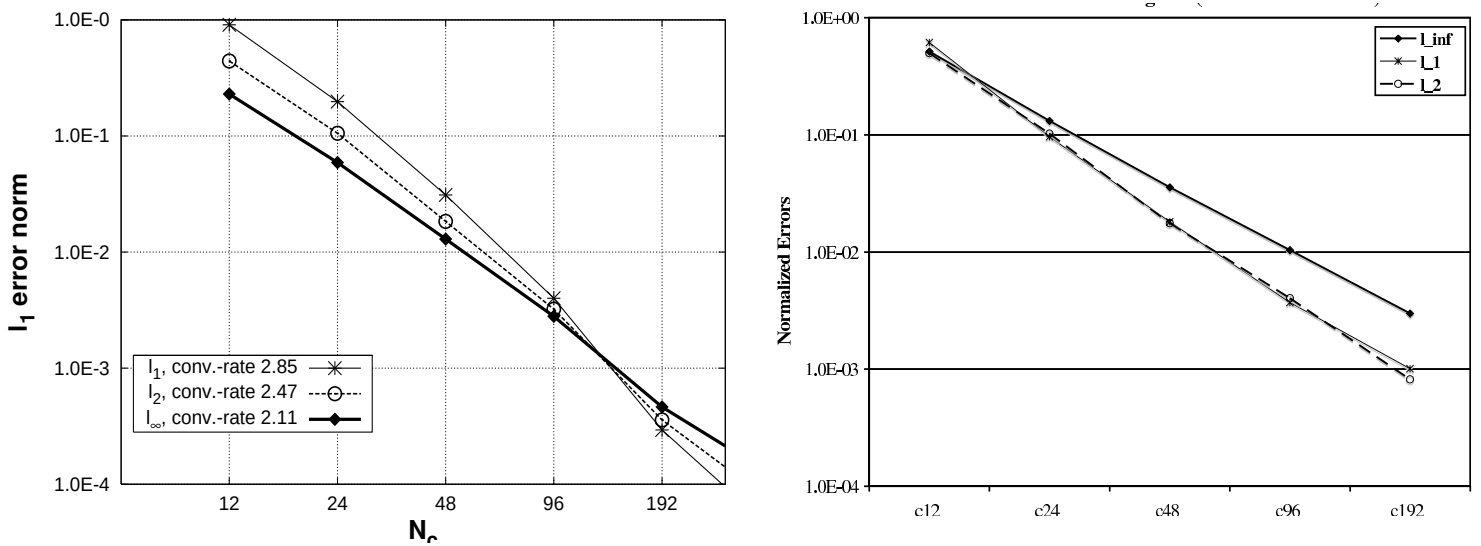




- CSLAM: Better than second-order convergence rates.

Solid body advection of cosine bell

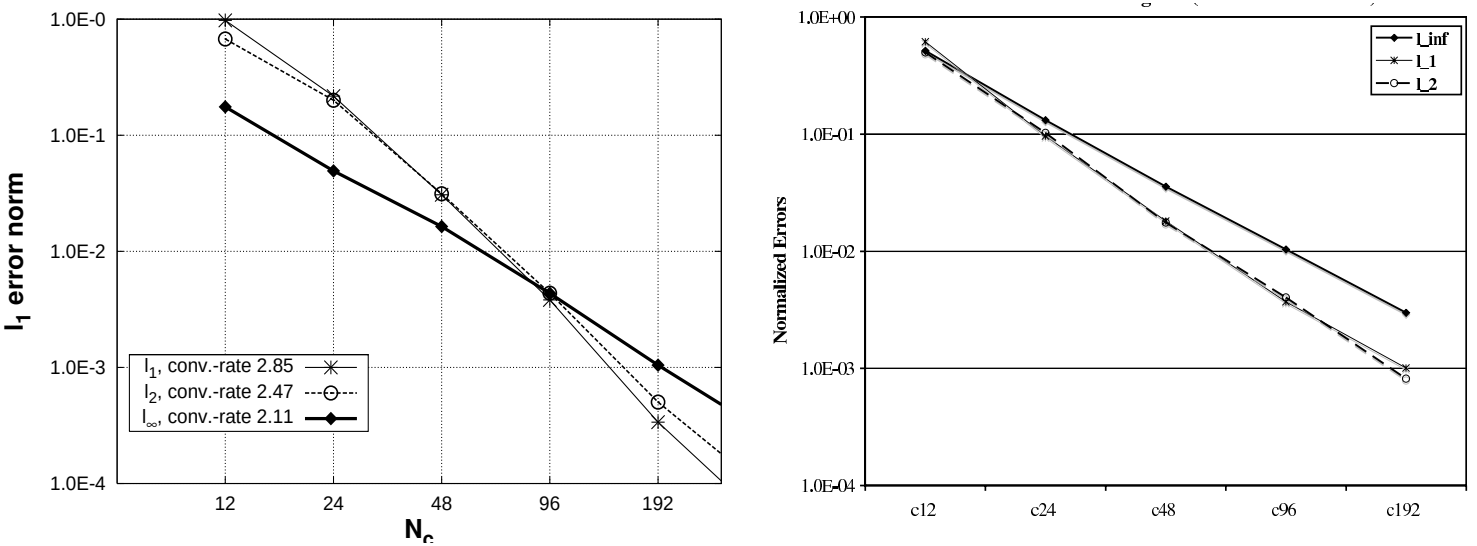
(left) CSLAM unlimited, (right) Putman and Lin (2007)



- CSLAM: Better than second-order convergence rates.
- Putman and Lin (2007): Convergence rates are 2.68, 2.54 and 1.97 for l_1, l_2 and l_∞.

Solid body advection of cosine bell

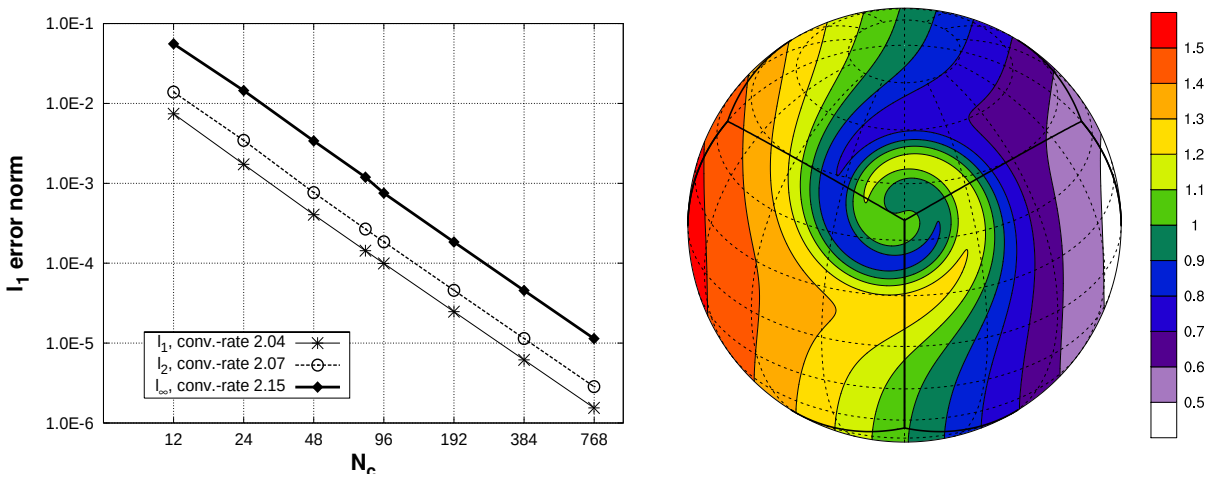
(left) CSLAM monotone (right) Putman and Lin (2007)



- CSLAM: Better than second-order convergence rates.
- Putman and Lin (2007): Convergence rates are 2.68, 2.54 and 1.97 for l_1, l_2 and l_∞.
- Monotone filter degrades accuracy

Static vortices (deformational flow)

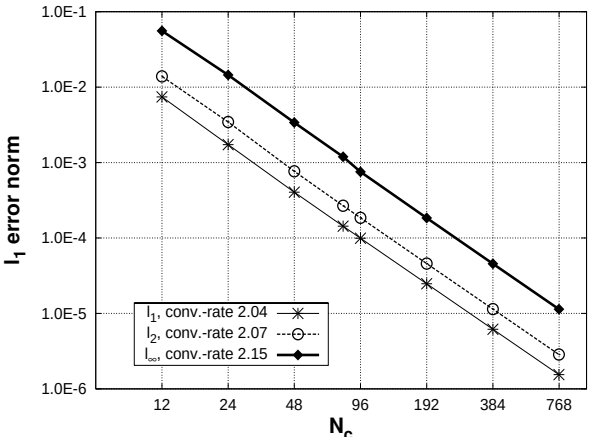
(left) monotone CSLAM, (right) day 6 solution

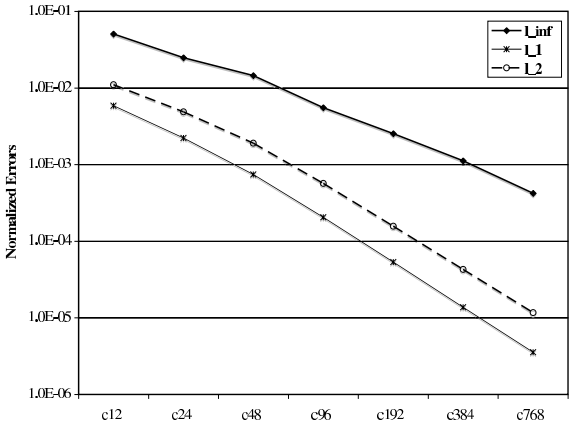


- CSLAM: Better than second-order convergence.

Static vortices (deformational flow)

(left) monotone CSLAM, (right) Putman and Lin (2007)

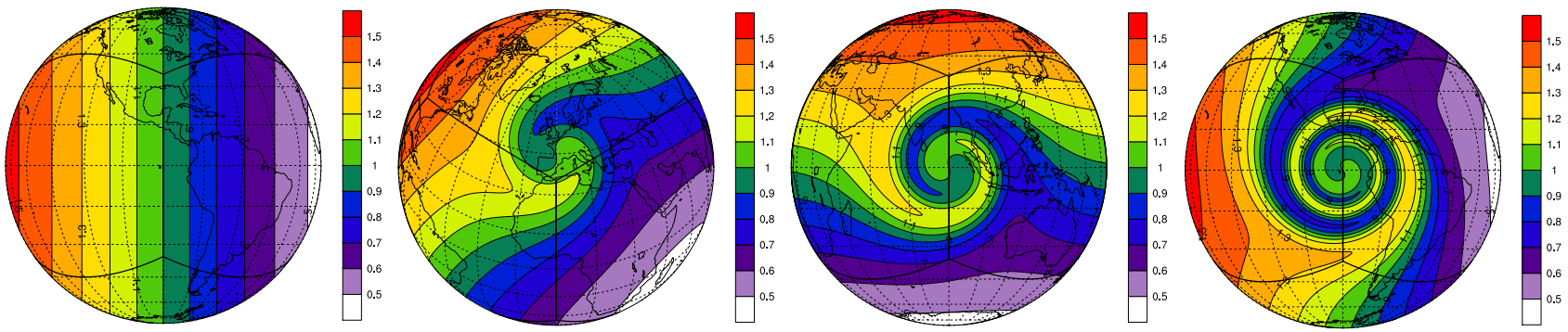




- CSLAM: Better than second-order convergence

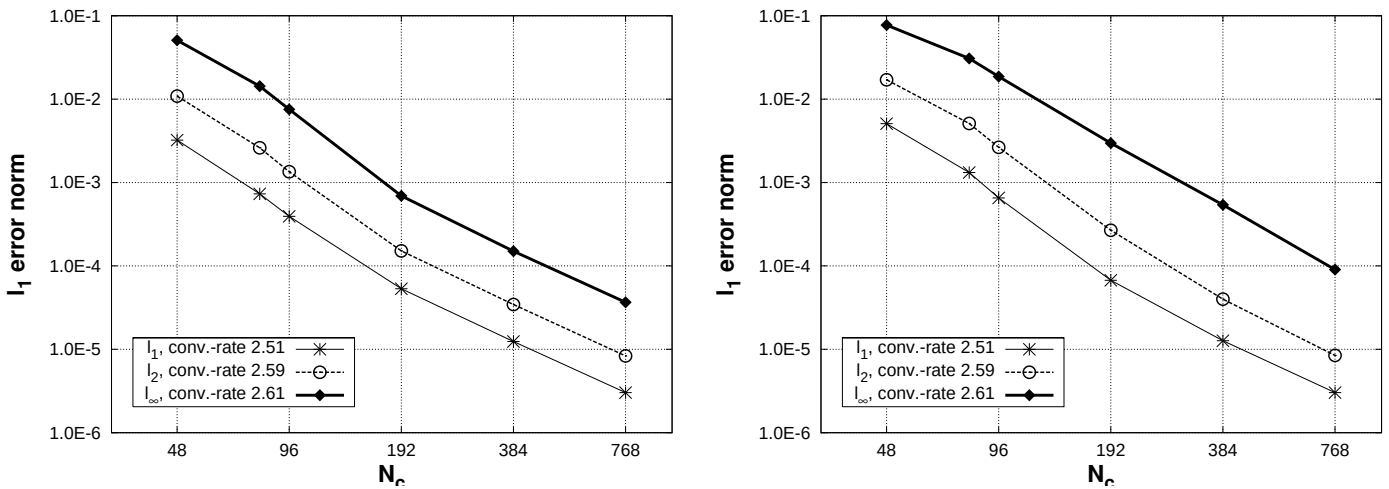
Moving vortices (solid body rotation + deformation)

Exact solution: Initial condition, 1/4, 1/2 and 1 revolution



Moving vortices (solid body rotation + deformation)

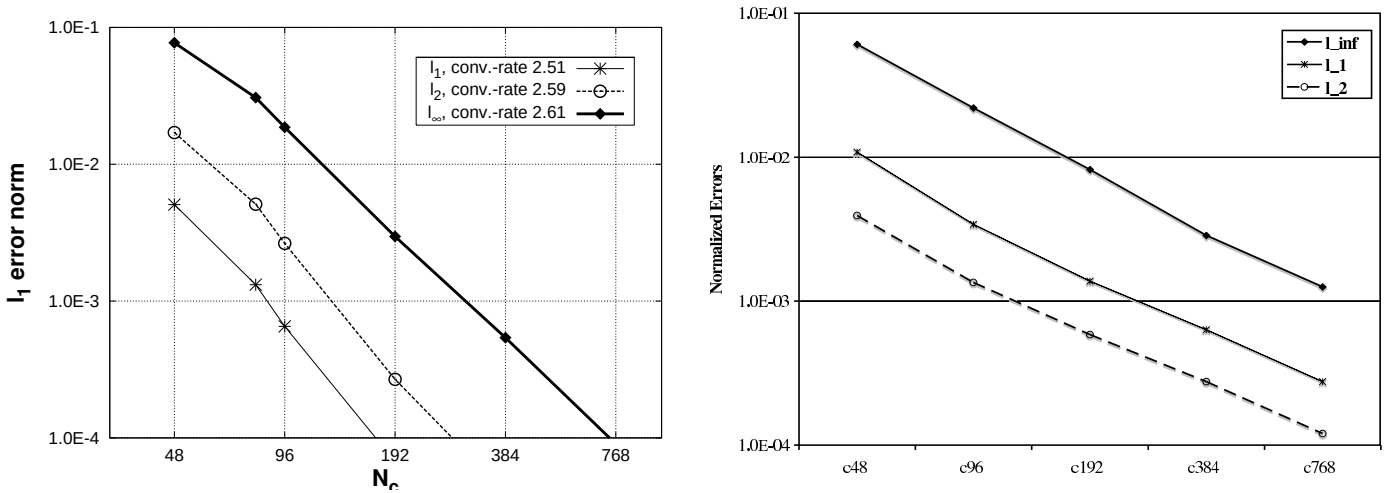
(left) CSLAM unlimited, (right) CSLAM monotone



- CSLAM: Better than second-order convergence.

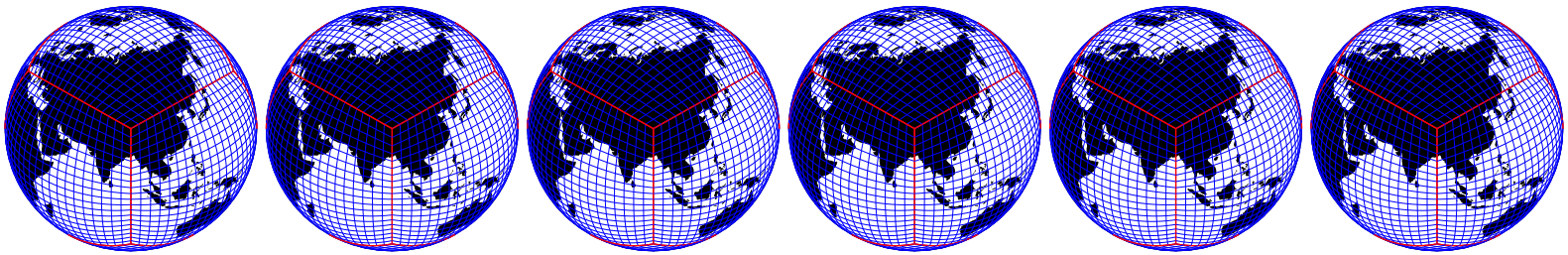
Moving vortices (solid body rotation + deformation)

(left) CSLAM monotone, (right) Putman and Lin (2007)



- Convergence rate for Putman and Lin (2007): 1.52, 1.67 and 1.54

Extension to the cubed-sphere



- Cubed sphere: Are fully 2D schemes necessary to ‘consistently’ treat the corners of the sphere or can dimensionally split approaches do the job?

Putman and Lin (2007) uses same order of reconstruction function (PPM) as CSLAM. So the main difference between the schemes is that CSLAM is fully 2D whereas Putman and Lin (2007) is dimensionally split.

Summary and Conclusions

- Rigorous 2D scheme developed:

Summary and Conclusions

- Rigorous 2D scheme developed:
  - Reuses geometric information for each additional tracer similar to Dukowicz and Baumgardner (2000) and Lipscomb and Ringler (2005).

Summary and Conclusions

- Rigorous 2D scheme developed:
  - Reuses geometric information for each additional tracer similar to Dukowicz and Baumgardner (2000) and Lipscomb and Ringler (2005).
  - Integrals are exact along coordinate lines (on the sphere); quadrature elsewhere.

Summary and Conclusions

- Rigorous 2D scheme developed:
  - Reuses geometric information for each additional tracer similar to Dukowicz and Baumgardner (2000) and Lipscomb and Ringler (2005).
  - Integrals are exact along coordinate lines (on the sphere); quadrature elsewhere.
  - Monotone option (fully 2D filter) - room for improvement!

Summary and Conclusions

- Rigorous 2D scheme developed:
  - Reuses geometric information for each additional tracer similar to Dukowicz and Baumgardner (2000) and Lipscomb and Ringler (2005).
  - Integrals are exact along coordinate lines (on the sphere); quadrature elsewhere.
  - Monotone option (fully 2D filter) - room for improvement!
- Future research:

Summary and Conclusions

- Rigorous 2D scheme developed:
  - Reuses geometric information for each additional tracer similar to Dukowicz and Baumgardner (2000) and Lipscomb and Ringler (2005).
  - Integrals are exact along coordinate lines (on the sphere); quadrature elsewhere.
  - Monotone option (fully 2D filter) - room for improvement!
- Future research:
  - Flux-form version of CSLAM (unlimited version will generate exactly the same results as semi-Lagrangian CSLAM but it might provide better/cheaper options for limiting)

Summary and Conclusions

- Rigorous 2D scheme developed:
  - Reuses geometric information for each additional tracer similar to Dukowicz and Baumgardner (2000) and Lipscomb and Ringler (2005).
  - Integrals are exact along coordinate lines (on the sphere); quadrature elsewhere.
  - Monotone option (fully 2D filter) - room for improvement!
- Future research:
  - Flux-form version of CSLAM (unlimited version will generate exactly the same results as semi-Lagrangian CSLAM but it might provide better/cheaper options for limiting)
  - Flux-form does not require $a_{k\ell}$ to exactly span the sphere!

Summary and Conclusions

- Rigorous 2D scheme developed:
  - Reuses geometric information for each additional tracer similar to Dukowicz and Baumgardner (2000) and Lipscomb and Ringler (2005).
  - Integrals are exact along coordinate lines (on the sphere); quadrature elsewhere.
  - Monotone option (fully 2D filter) - room for improvement!
- Future research:
  - Flux-form version of CSLAM (unlimited version will generate exactly the same results as semi-Lagrangian CSLAM but it might provide better/cheaper options for limiting)
  - Flux-form does not require $a_{k\ell}$ to exactly span the sphere!
  - 'a priori' reconstruction limiting or 'posteriori' with flux-limiters. Which is optimal for CSLAM?

References I

Barth, T., and D. Jespersen, 1989: The design and application of upwind schemes on unstructured meshes. *Proc. AIAA 27th Aerospace Sciences Meeting, Reno*.

Colella, P., and P. R. Woodward, 1984: The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.*, **54**, 174–201.

Dukowicz, J., 1984: Conservative rezoning (remapping) for general quadrilateral meshes. *J. Comput. Phys.*, **54**, 411–424.

Dukowicz, J. K., and J. R. Baumgardner, 2000: Incremental remapping as a transport/advection algorithm. *J. Comput. Phys.*, **160**, 318–335.

Jablonowski, C., 2004: Adaptive grids in weather and climate modeling, Ph.D. thesis, University of Michigan.

Lauritzen, P. H., R. D. Nair, and P. A. Ullrich, 2009: A conservative semi-Lagrangian multi-tracer transport scheme (cslam) on the cubed-sphere grid. *J. Comput. Phys.*. Submitted.

Lin, S., and R. Rood, 1996: Multidimensional flux-form semi-Lagrangian transport schemes. *Mon. Wea. Rev.*, **124**, 2046–2070.

Lipscomb, W. H., and T. D. Ringler, 2005: An incremental remapping transport scheme on a spherical geodesic grid. *Mon. Wea. Rev.*, **133**, 2335–2350.

Margolin, L. G., and M. Shashkov, 2003: Second-order sign-preserving conservative interpolation (remapping) on general grids. *J. Comput. Phys.*, **184**, 266–298.

Nair, R. D., and B. Machenhauer, 2002: The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere. *Mon. Wea. Rev.*, **130**(3), 649–667.

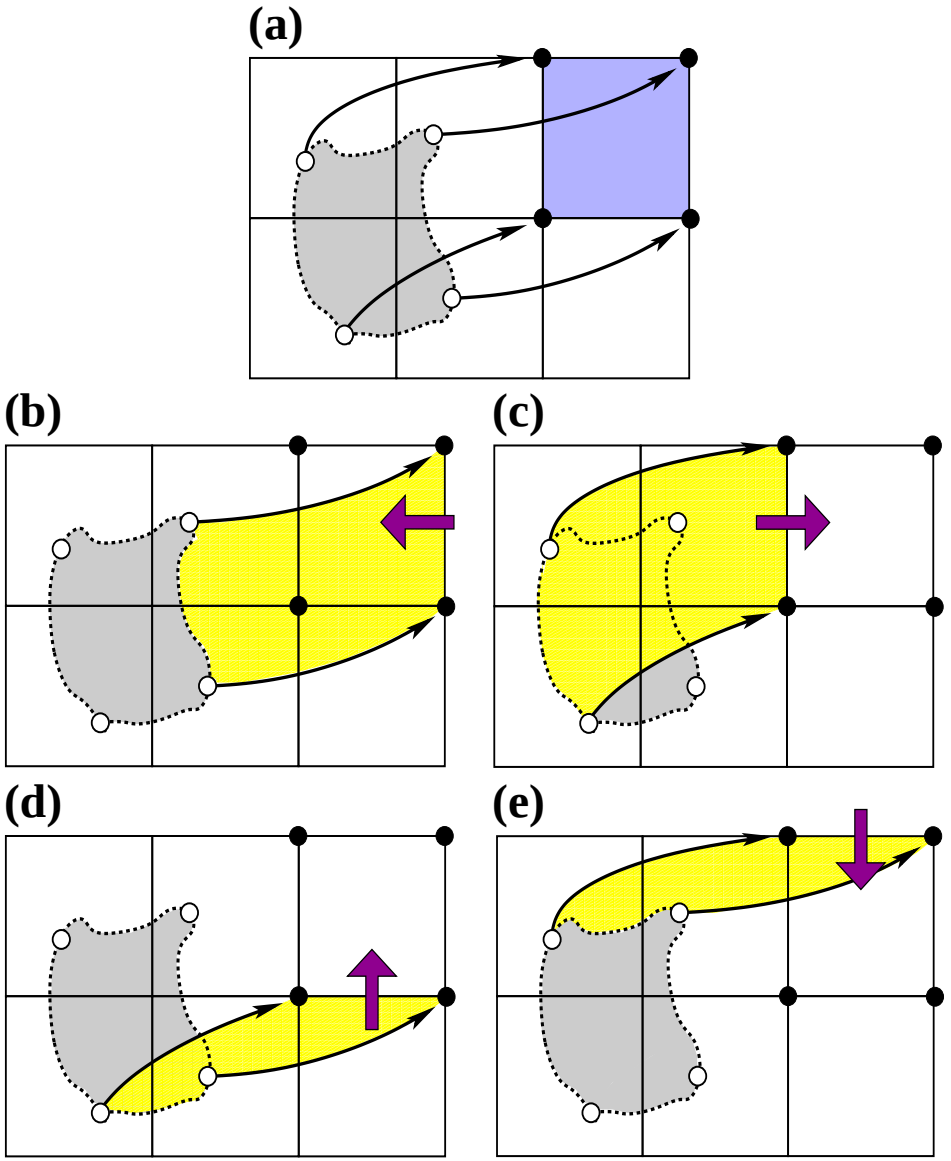
Putman, W. M., and S.-J. Lin, 2007: Finite-volume transport on various cubed-sphere grids. *J. Comput. Phys.*, **227**(1), 55–78.

Rančić, M., 1992: Semi-Lagrangian piecewise biparabolic scheme for two-dimensional horizontal advection of a passive scalar. *Mon. Wea. Rev.*, **120**, 1394–1405.

Ullrich, P. A., P. H. Lauritzen, and C. Jablonowski, 2009: Geometrically exact conservative remapping (GECoRe): Regular latitude-longitude and cubed-sphere grids. *Mon. Wea. Rev.*. To appear.

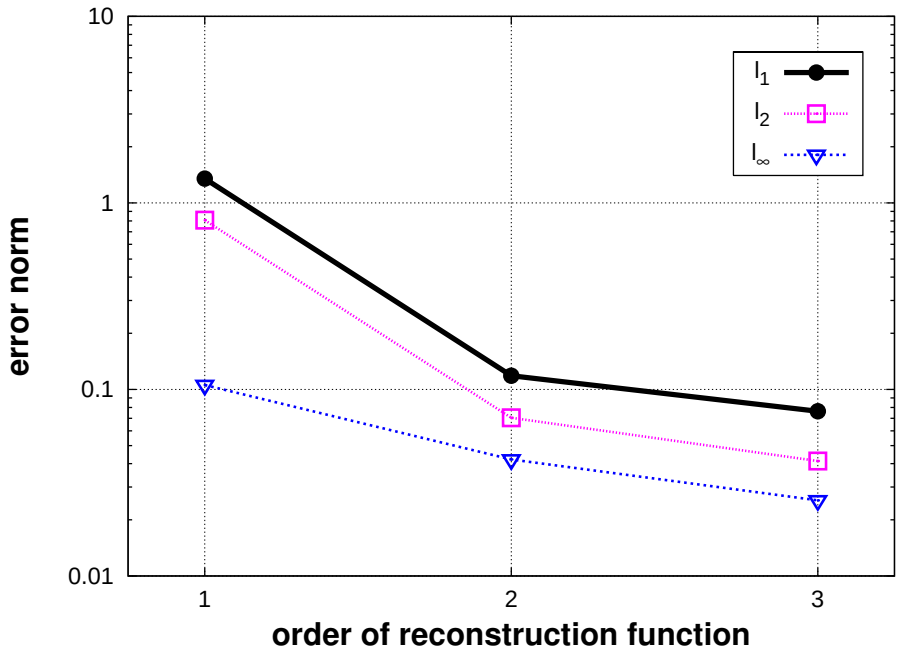
Zerroukat, M., N. Wood, and A. Staniforth, 2007: Application of the parabolic spline method (PSM) to a multi-dimensional conservative semi-lagrangian transport scheme (SLICE). *J. Comp. Phys.*, **225**, 935–948.

Flux-form versus upstream semi-Lagrangian



Convergence with order of reconstruction function

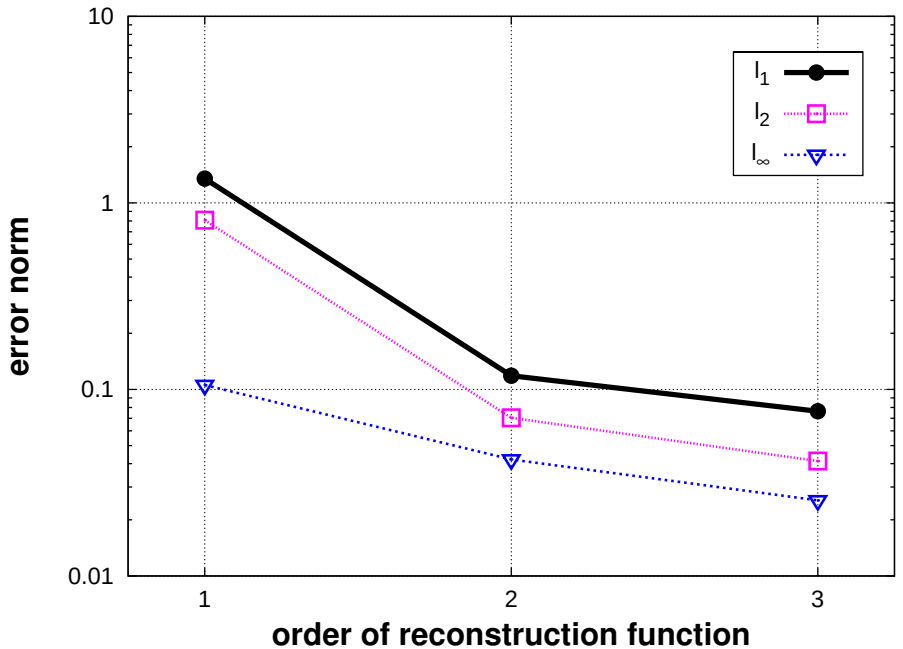
Same settings as previous slide



- As expected: Large improvement from 1st to 2nd (l_1 reduced by a factor 11), less (but still significant) improvement from 2nd to 3rd (l_1 reduced by a factor 1.6).

Convergence with order of reconstruction function

Same settings as previous slide



- As expected: Large improvement from 1st to 2nd (l_1 reduced by a factor 11), less (but still significant) improvement from 2nd to 3rd (l_1 reduced by a factor 1.6).
- all results shown beyond this point use third-order reconstruction functions (‘2D PPM’).