Mimetic discretizations: new opportunities for pre- and post-processing

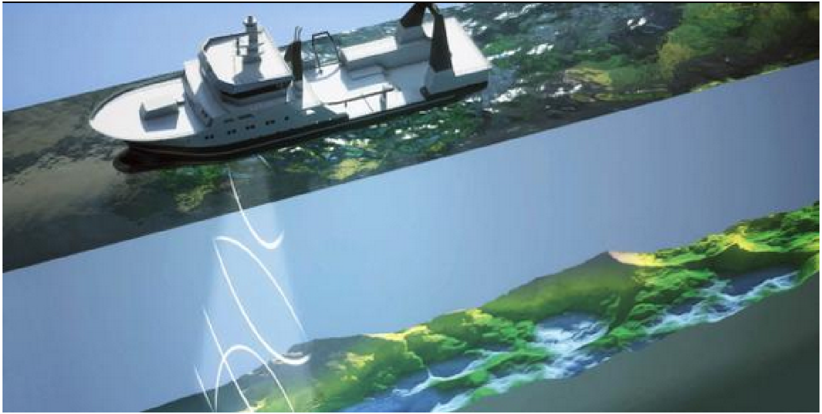
Alexander Pletzer (NeSI/NIWA), Wolfgang Hayek (NIWA/NeSI) and Jorge Bornemann (NIWA)
(alexander.pletzer@nesi.org.nz)
Mathematical Sciences Institute
Australian National University

11 Dec 2017

New Zealand eScience Infrastructure

National Institute for Water and Atmospheric Research Ltd (NIWA)

NIWA Vessels



Overview

- What are mimetic discretizations and why we care
- Why existing interpolation methods won't work
  - Solution: make pre/post processing consistent with discretization
- Addressing the problem using a differential form/exterior calculus approach
  - 4 types of fields, 4 types of discretization and 4 types of interpolation
  - the interpolation method is not a choice!
- Algorithms
  - Requires computing collision (intersection) of a grid with an object (point, line, area or volume)
- Results
- Summary and further work

What we mean by mimetic

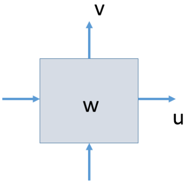
- Discretization that preserves the mathematical properties of a field
  - $\nabla \times \nabla = 0$ and $\nabla \cdot \nabla \times = 0$
  - $\int \nabla \times E \cdot dS = \oint E \cdot d\ell$ and $\int \nabla \cdot D dV = \oint D \cdot dS$
- Conserves quantities to near machine accuracy
  - line integral
  - surface flux
  - volume integral
- Free of spurious modes and numerical pollution
  - Mixed finite element methods based on H(curl) and H(div) elements
  - Finite Difference Finite Time (FDFT) domain
  - Discrete Exterior Calculus (DEC) methods
- Bossavit (1989), Hiptmaier (1997), Hirani (2003), Arnold (2002), Gilette & Bajaj (2010), Cotter & Shipton (2014), Samtaney & Mohamed (2015)

Motivation

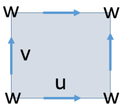
Want answers to following questions

How should we interpolate vector fields with staggered components?

- Arakawa C/D grids
- Components are on cell faces or edges
- Arises in computational fluid dynamics and electromagnetics



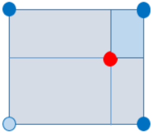
Arakawa C-grid



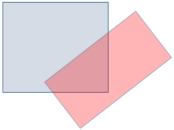
Arakawa D-grid

Can we unify existing interpolation methods?

- Linear, used since Babylonian times (2000-1700 BC)
- Conservative or area weighted, used in climate studies to enforce conservation
- In both cases the interpolation weights are ratios of areas (volumes)
- When is bilinear applicable and when should one use conservative?
- Should vector fields with staggered components use bilinear or conservative? Or something else?



Bilinear

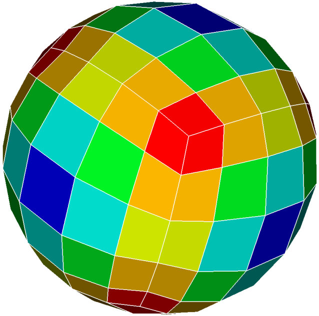


Conservative

How to handle curvilinear grids with non-orthogonal cells?

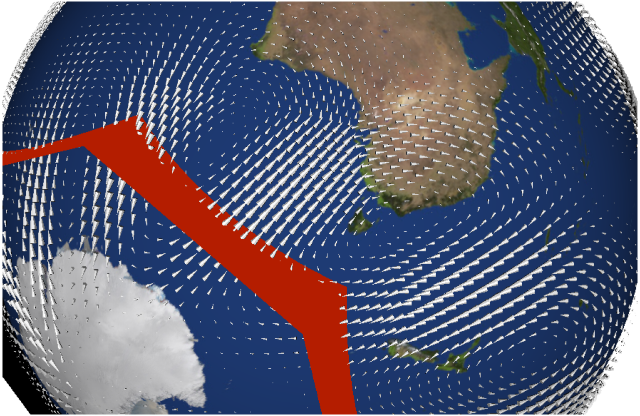
Example: cubed-sphere grid

- Project grids on the surface of a cube onto a sphere
- Six logically rectangular grids (cannot be represented as a single structured grid)
- No pole-like singularity but some distortion where three tiles meet



How to compute surface fluxes?

For example compute water fluxes across area



The type of field determines the discretization

Some notation

Coordinates

- ξ^i is curvilinear coordinate, $i = 1, 2, 3$
- x is position (in physical space)

Wedge product

$\wedge$ (antisymmetric, like cross-product $\times$)

Exterior derivative and contravariant bases

- d
- similar to ∇, $\nabla\times$, or $\nabla\cdot$
- $d\xi^i$ is basis function $\leftrightarrow \nabla\xi^i$

Why four types of fields?

One derivative - 4 fields

- 0-form: $\alpha(x)$ (just a function of space, one component)
  - appropriate for scalar fields that don't change under coordinate transformations
  - Example: temperature
- 1-form: $\beta = \sum_i \beta_i d\xi^i$, basis is $d\xi^1$, 3 components
  - appropriate for vector fields
  - Examples: electric field E and induction H
- 2-form: $\gamma = \sum_{i<j} \gamma_{ij} d\xi^i \wedge d\xi^j$, basis is $d\xi^i \wedge d\xi^j$, 3 components
  - appropriate for pseudo-vector fields, obtained by applying cross product of 1-forms
  - Examples: magnetic field B and displacement field D
- 3-form: $\omega = \omega_{123} d\xi^1 \wedge d\xi^2 \wedge d\xi^3$, basis is $d\xi^1 \wedge d\xi^2 \wedge d\xi^3$, one component
  - appropriate for pseudo-scalar, density-like fields

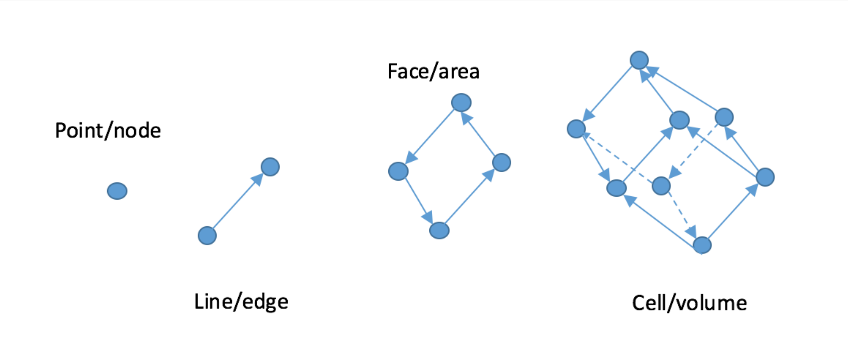
Discretized versions of the fields

Association of form with cell elements

- 0-form: on nodes
  - $\int \alpha = \alpha$ (integral is a no op)
- 1-form: on edges
  - $\int \beta$ is a line integral
- 2-form: on faces
  - $\int \gamma$ is a surface integral
- 3-form: cell centred
  - $\int \omega$ is a volume integral

Differential forms like to be integrated

Ordering of vertices determines orientation of line, surface, volume



One rule for $\nabla \times \nabla = 0$ and $\nabla \cdot \nabla\times = 0$

What it means to be mimetic (1): $d^2 = 0$!

- $d\alpha^0 \leftrightarrow \nabla\alpha$
- $d\beta^1 \leftrightarrow \nabla \times \beta$
- $d^2\alpha^0 \leftrightarrow \nabla \times \nabla\alpha = 0$
- $d\gamma^2 \leftrightarrow \nabla \cdot \gamma$
- $d^2\beta^1 \leftrightarrow \nabla \cdot \nabla \times \beta = 0$

What it means to be mimetic (2); divergence, Stokes' and gradient theorems

$\int_M d\alpha^k = \oint_{\partial M} \alpha^k$

It's easy to write integral of a differential form in any coordinate system

Pullback of a differential form introduces Jacobian

- $\int \beta(\xi) d\xi = \int \beta(t) \frac{d\xi}{dt} dt$ (t is line parametrization)
- $\int \gamma(\xi) d\xi^1 \wedge d\xi^2 = \int \gamma(\eta) \det\left(\frac{\partial \xi}{\partial \eta}\right) d\eta^1 \wedge d\eta^2$ (η^1 and η^2 are surface parametrization)
- $\int \omega(\xi) d\xi^1 \wedge d\xi^2 \wedge d\xi^3 = \int \omega(\eta) \det\left(\frac{\partial \xi}{\partial \eta}\right) d\eta^1 \wedge d\eta^2 \wedge d\eta^3$

We will see in next section that the above are exactly what we need for interpolation.

Interpolation

Generalizing “interpolation”

Making interpolation work for nodal, edge, face and cell fields

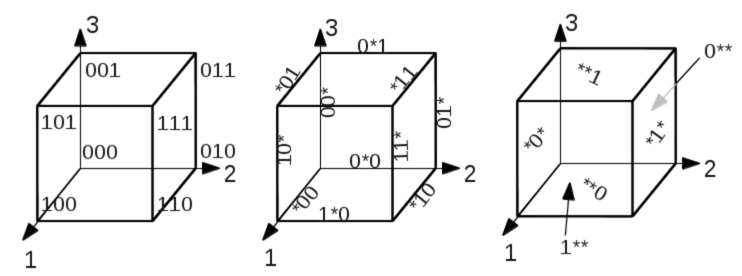
$\int f = \sum_i f_i \int_T \phi_i$

- ϕ_i is basis k-form, $k = 0$, 1, 2 or 3
- T is target (point, line, area or volume)
- f_i is field integral over cell element k (node, edge, face or cell)
- $\int_T \phi_i$ is the interpolation weight
- i index runs over all the degrees of freedom (points, edges, faces etc., as appropriate)

Identifing cell elements for low order basis forms

$i = (i_1, i_2, i_3)$ with $i_j \in \{0, 1, \star\}$

- 0 is low side
- 1 is the high side
- $\star$ element varies in this direction
- $(\star, \star, \star)$ is the cell



Want the basis functions to satisfy orthogonality condition

$\int_i \phi_j = \delta_{ij}$

- i is cell element (node, edge, face, cell)
- j is basis function index
- nodal bases are zero on all nodes expect one where it is one
- edge bases have zero integral on all edges except itself where it is one
- face bases have zero integral on all faces except itself where it is one

The $k = 0$ bases are the usual tent functions

- $\phi_{000} = (1 - \xi^1)(1 - \xi^2)(1 - \xi^3)$
- $\phi_{001} = (1 - \xi^1)(1 - \xi^2)\xi^3$
- $\phi_{010} = (1 - \xi^1)\xi^2(1 - \xi^3)$
- $\phi_{011} = (1 - \xi^1)\xi^2\xi^3$
- $\ldots$
- $\phi_{111} = \xi^1\xi^2\xi^3$

The $k = 1, 2, 3$ bases basis forms

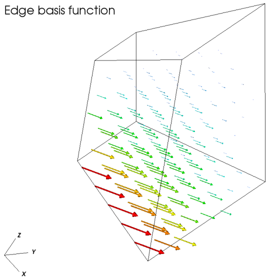
$k = 1$ bases are $H(curl)$ finite elements

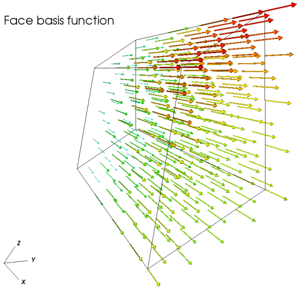
- $\phi_{00\star} = (1 - \xi^1)(1 - \xi^2) d\xi^3$
- $\phi_{01\star} = (1 - \xi^1)\xi^2 d\xi^3$
- $\dots$
- $\phi_{\star 11} = \xi^2 \xi^3 d\xi^1$

$k = 2$ bases are $H(div)$ finite elements

- $\phi_{0\star\star} = (1 - \xi^1) d\xi^2 \wedge d\xi^3$
- $\phi_{1\star\star} = \xi^1 d\xi^2 \wedge d\xi^3$
- $\dots$
- $\phi_{\star\star 1} = \xi^3 d\xi^1 \wedge d\xi^2$

Observe how the $k = 1, 2$ bases satisfy the orthogonality condition





Edge basis is perpendicular to neighbouring edges

Face basis is tangent to neighbouring faces

Computing the interpolation weights

Assume target is a simplex

$T : \lambda \to \xi = \xi_0 + \sum_{i=1}^{k} \lambda_i(\xi_i - \xi_0)$

Interpolation weight is the pullback, $\det(\partial\xi/\partial\lambda)$ is size of simplex in ξ space

$\int_T \phi = \int T^*\phi(\xi) = \int \phi(\lambda)\det(\partial\xi/\partial\lambda)$ independent of ξ coordinate system!

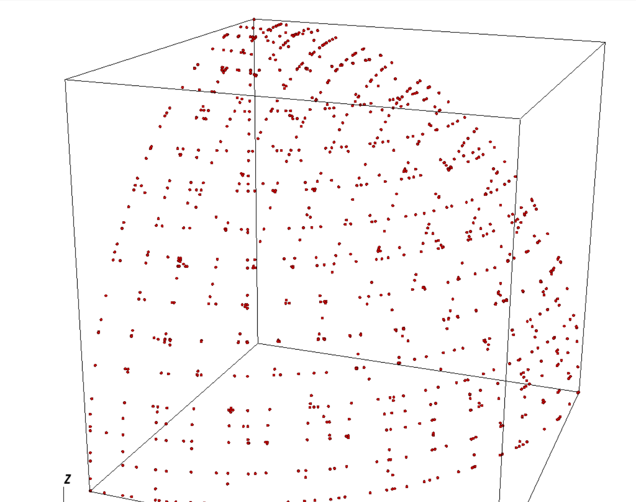
For 1-form $(1-\xi^1)\xi^2 d\xi^3$ and $\xi = \xi_0 + \lambda(\xi_1 - \xi_0)$:

$\int_T \phi = (x_1 - x_0)\int_0^1 d\lambda \phi(\xi_0 + \lambda(\xi_1 - \xi_0))$

Integrals can be computed analytically for polynomial bases and target simplices

Computing intersection of grid with target

grid cells with target points, grid faces with target edges, grid edge with target faces, etc

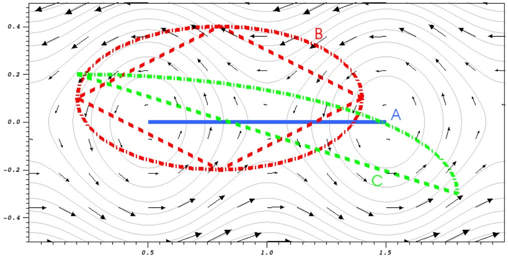


Results

Divergence-free field in Cartesian coordinates

Stream function

- $v = dz \wedge d\psi$ with stream function $\psi = \frac{\cos 2\pi x}{4\pi} + y^2$
- Closed surface fluxes B and C are exact
- Flux on A is exact because start/end points are nodes



Vector field with singularity

$v = \frac{xdx+ydy}{2\pi(x^2+y^2)}$ chosen such that $\oint v$ is 1 if contour contains singularity, 0 otherwise

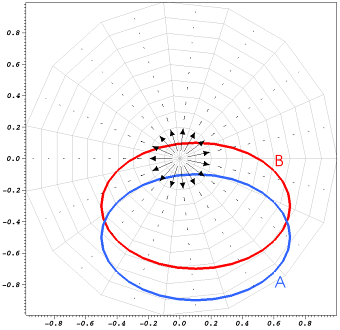


Figure: Radial field

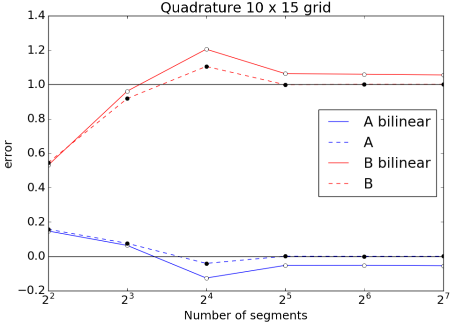
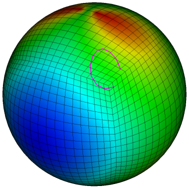


Figure: Numerical error

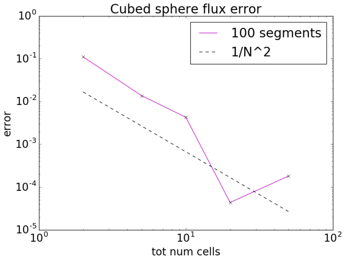
Flux computation on the cubed sphere $v = d\psi \wedge dr$

Error depends distance of start/end point from edge/face

- ψ is a function of longitude and latitude
- Some cells have 120 deg angle between edges



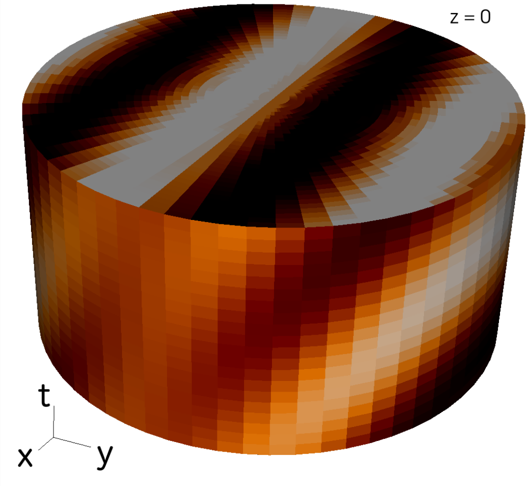
Integration path/surface



Error is $\sim 1/N^2$

Checking Maxwell's equations

$F = E \wedge dt + B$. Faraday's law is $dF = 0$ (no charge). Plane wave.



Summary

Different types of field $\leftrightarrow$ different staggerings

- nodal for scalar fields (e.g. temperature)
- edge for vector fields (e.g. electric field)
- face for pseudo-vector fields (e.g velocity)
- cell for pseudo-scalar fields (e.g. density)
- type of field $\rightarrow$ interpolation method
- field values set via cell, face and edge integrals (instead of vector field components)
- who needs vector fields?

Masking and partially valid cells?

Ok if taking account of partial cell, faces, edges when setting cell, face and edge integrals. Done!

Basis function extensions

What about tetrahedra?

Similar approach except that the basis functions are Whitney's

Higher order basis functions?

Initial work indicates that higher order basis functions can be used. These also satisfy the orthogonality condition $\int_i \phi_j = \delta_{ij}$ on sub-cell edges, faces and cells. Quadratic elements effectively split each cell into 8 sub-cells, each face into 4 sub-faces and each edge into 2 sub-edges.

Goal is to apply the rigour of dynamical cores to pre- and post-processing tools. Overtime we expect the distinction between dynamical core and pre-/post-processing to diminish.

Thank You

NeSI
New Zealand eScience Infrastructure