Compressible atmospheric modelling on unstructured grids

Oswald Knoth

Leibniz Institute for Tropospheric Research, Leipzig

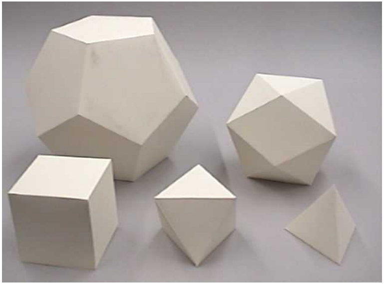
27.01.2016

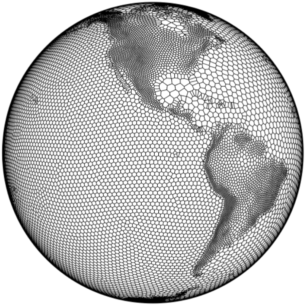
Contents

1. Introduction
2. Equations
3. Discretization of the conservative form
4. Operators
5. Operators
6. Application
7. Conclusion

Unstructured grids

- Computational domain is decomposed by polyhedrons





- Two polyhedrons have in common a face, an edge, or a vertex, or are disjunct
- Each vertex is adjacent to three edges
- Two dimensional grids, which are stacked over, have this property.

Global models

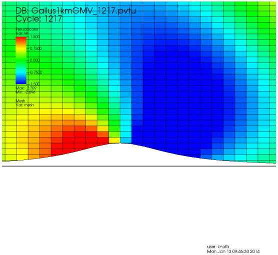
Overview over global atmospheric models with unstructured grid

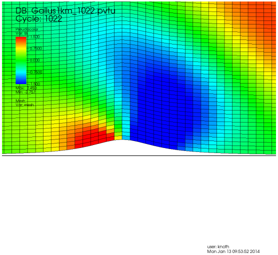
- Model for Prediction Across Scales (MPAS), developed by NCAR and LANL
- ICOsahedral Nonhydrostatic (ICON) of the DWD and MPI Hamburg
- Nonhydrostatic ICosahedral Atmospheric Model (NICAM), Japan Agency for Marine-Earth Science and Technology
- Ocean–Land–Atmosphere Model (OLAM)
- NCAR High-Order Method Modeling Environmen (HOMME)
- GungHo Project (UK MetOffice)
- PantaRhei project (ECMWF Reading)
- And many others!

Grids

Available grids in ASAMUnstructured

- Cartesian grids, cut cell and boundary following

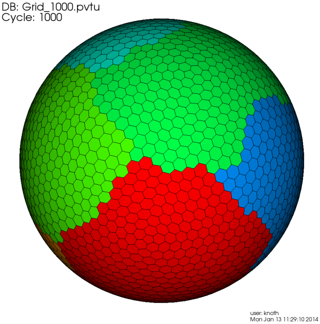


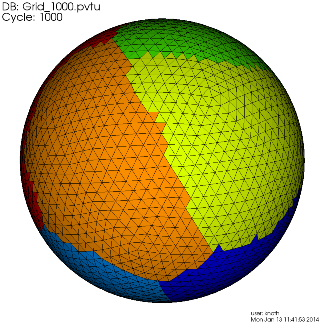


- Cut cell grids in three dimension are not simple

Available grids ASAMUnstructured

- Grids on the sphere
- Hexagon and triangular

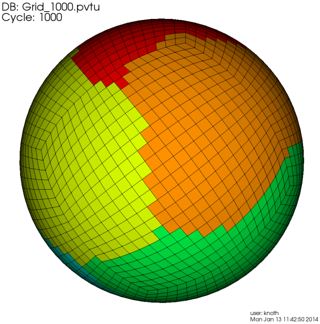


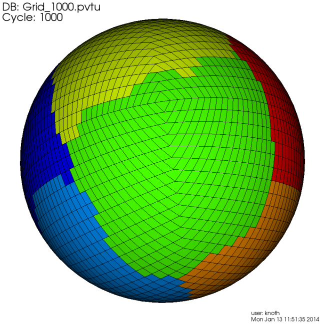


Grids

Available grids ASAMUnstructured

- Grids on the sphere
- Cubed sphere and two fold





1 Introduction

2 Equations

3 Discretization of the conservative form

4 Operators

5 Operators

6 Application

7 Conclusion

Preliminary considerations

Model variables:

- Define flux quantities (mass conservation!), Momentum, potential temperature, further species:
  $\mathbf{V} \equiv \rho\mathbf{v}$, $\Theta \equiv \rho\theta$, $c \equiv \rho\mu$.
- c stands for moisture, solids (cloud, rain, ice), and chemical substances

Conservative

The compressible Euler equations:

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \boldsymbol{\nabla}(\rho\mathbf{v}\mathbf{v}) - \boldsymbol{\nabla}\boldsymbol{\tau} = -\nabla p - \rho g\mathbf{k} - 2\boldsymbol{\Omega} \times (\rho\mathbf{v})$$

$$\frac{\partial\rho}{\partial t} + \nabla(\rho\mathbf{v}) = 0$$

$$\frac{\partial(\rho\theta)}{\partial t} + \nabla(\rho\theta\mathbf{v}) - \nabla(\rho\mathbf{K}_\theta\nabla\theta) = Q_\theta$$

$$\frac{\partial(\rho\mu)}{\partial t} + \nabla(\rho\mu\mathbf{v}) - \nabla(\rho\mathbf{K}_\mu\nabla\mu) = Q_\mu$$

$$p = \left(\frac{R_d\rho\theta}{p_0^\kappa}\right)^{1/(1-\kappa)}$$

- Pressure is determined by equation of state: $p = P(\rho, \theta, \mathbf{c})$
- Moist potential temperature θ
- In the dry case $\theta = T\left(\frac{p_0}{p}\right)^\kappa$ with $\kappa \equiv R_d/c_{pd} =$ const.

Conservative

The shallow water equations:

$$\frac{\partial (h\mathbf{v})}{\partial t} + \boldsymbol{\nabla}(h\mathbf{v}\mathbf{v}) = -gh\nabla(h + h_S) - 2\boldsymbol{\Omega} \times (h\mathbf{v})$$
$$\frac{\partial h}{\partial t} + \nabla(h\mathbf{v}) = 0$$

- With $\rho = h$ and a special pressure definition the compressible Euler equation has the form of the shallow water equation

Vector invariant

The vector invariant form of the compressible Euler equations:

$$\frac{\partial \mathbf{v}}{\partial t} + (2\boldsymbol{\Omega} + \boldsymbol{\omega}) \times \mathbf{v} - \frac{1}{\rho}\boldsymbol{\nabla}\boldsymbol{\tau} = -\frac{1}{\rho}\nabla p - g\mathbf{k} - \frac{1}{2}\nabla(\mathbf{v} \cdot \mathbf{v})$$

where

$$\boldsymbol{\omega} = \nabla \times \mathbf{v}$$

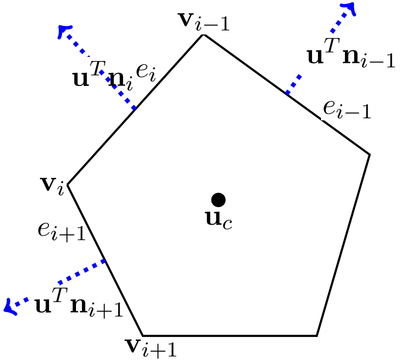
- Used now in new global models, ICON (DWD, MPI), ICON-IAP, MPAS(NCAR, Livermore)
- Through sophisticated spatial and temeporal discretisiation conservation of energy and potential vorticity

Variable position and operators

- Method of lines, first in space, than in time
- Encoding of variables
  - Scalar on cells or vertices
  - Full vector on cells or vertices
  - Projected vector on faces $(\mathbf{u}^T\mathbf{n})$ or edges $(\mathbf{u}^T\mathbf{t})$
- Operators for vectors
  - FaceToCell, CellToFace
  - EdgeToCell, CellToEdge
  - Divergence, Curl
- Operators for scalars or Cartesian components in cell center
  - AdvCell
  - Gradient

FaceToCell

- Reconstruct $\mathbf{u}_c$ from $\mathbf{u}^T\mathbf{n}_i, \quad i = 1, \ldots, n_e$
- Constant reconstruction
- Barycentric coordinates (reconstruct first in nodes)
- Polynomial reconstruction
- Reconstrution with radial basis function



Generalized barycentric coordinates

Let the vertices of the polyhedron P be denoted by $\mathbf{v}_1, \ldots, \mathbf{v}_N$. Then a set of functions $\lambda_1, ..., \lambda_{|V|}$ are generalized barycentric coordinates for P if the following holds:

B1 $\sum_{i=1}^{|V|} \lambda_i(\mathbf{x}) = 1$

B2 $\lambda_i(\mathbf{v}_j) = \delta_{ij}$

B3 $\lambda_i(\mathbf{x}) \geq 0$

B4 $\sum_{i=1}^{|V|} \lambda_i(\mathbf{x})\mathbf{v}_i = \mathbf{x}$

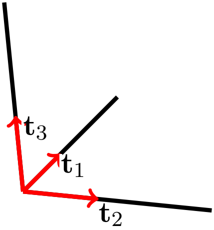
- Can be used to interpolate every information given at the vertices

Velocity interpolation

Let the outer unit normal of the faces of the polyhedron P be denoted by $\mathbf{n}_1, \ldots, \mathbf{n}_{|F|}$ and the midpoints by $\mathbf{x}_1^F, \ldots, \mathbf{x}_{|F|}^F$.
A given velocity field $\mathbf{u}(\mathbf{x})$ will be encoded by $u^F = \mathbf{n} \cdot \mathbf{u}(\mathbf{x}^F)$.
Task 1: Reonstruct $\mathbf{u}(\mathbf{x})$ for each polyhedron P from the given normal face components u^F.

$$\mathbf{u}^R(\mathbf{x}) = \sum_{i=1}^{|V|} \lambda_i(\mathbf{x}) \mathbf{u}_i^V$$

Corner velocity



$$\mathbf{n}_1 = \mathbf{t}_2 \times \mathbf{t}_3$$

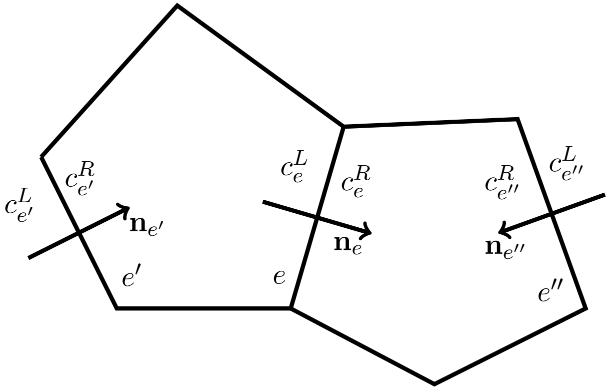
$$\mathbf{n}_2 = \mathbf{t}_3 \times \mathbf{t}_1$$

$$\mathbf{n}_3 = \mathbf{t}_1 \times \mathbf{t}_2$$

$$\mathbf{u} = (\mathbf{u} \cdot \mathbf{n}_1)\mathbf{t}_1 + (\mathbf{u} \cdot \mathbf{n}_2)\mathbf{t}_2 + (\mathbf{u} \cdot \mathbf{n}_3)\mathbf{t}_3$$

$$|\Omega_k|\frac{d}{dt}\bar{c}_k(t) = -\sum_{i=1}^{n^k}\int_{e_i^k}\mathbf{n}_i^k\cdot\mathbf{u}(\mathbf{x},t)c(\mathbf{x},t)dS = -\sum_{i=1}^{n^k} f_i^k. \tag{1}$$

$$\hat{f}_e^L = -\hat{f}_e^R = |e|\left[\max(\mathbf{u}\cdot\mathbf{n}_e,0)c_e^L + \min(\mathbf{u}\cdot\mathbf{n}_e,0)c_e^R\right]. \tag{2}$$



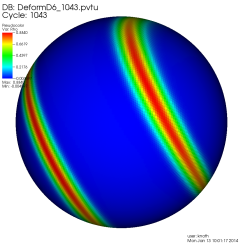
AdvCell

- Reconstruct face values from cell centered values
- Logarithmic reconstruction
  - Compute first first derivatives on the faces

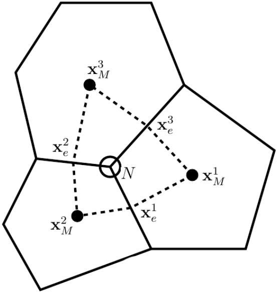
$$\varphi_k(\mathbf{x}) = \phi_0 + \sum_{i=1}^{n^k} b_i \cdot log(\mathbf{n}_i^\intercal \cdot (\mathbf{x}_i - \mathbf{x}) + \tilde{h}_i), \quad \tilde{h}_i > 0, \quad i = 1, \ldots, n^k.$$

  - Function φ interpolates derivatives at the boundaries and the cell value
- Linear reconstruction (compute gradient at cell center)
- Polynomial reconstruction with different stencil width

Advection


DB: DeformD6_1043.pvtu
Cycle: 1043
Pseudocolor
Var: Rho
0.8840
0.6619
0.4397
0.2176
user: knoth
Mon Jan 13 10:01:17 2014

Gradient computation of cell centered values



- Explanation in two dimension, a scalar c is given in ech cell
- Barycentric subdivision, laeding to the interaction volumes around corners
- Define linear functions $L_i(\mathbf{x})$ for each sub-cell wit additional unknowns at the edge values
- Two neighbouring linear ansatz functions have the same directional derivative

$$\mathbf{n}_1^\intercal \cdot \nabla L_1(\mathbf{x}_e^1) = \mathbf{n}_1^\intercal \cdot \nabla L_2(\mathbf{x}_e^1).$$

Coriolis term

- Multyply cell centered reconstructed velocity by

$$k = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- $$k \times \mathbf{u}_C$$

Application

- Gravity Waves
- Shallow water, flow over hill
- Baroclinic instability

Future work

- Geostrophic balance, accuracy of the Coriolis term
- Cut cell in three dimension, cell merging
- Encoding of curvature in geometry
- More efficient parellel pressure solver
- More physics