Continued development of the Vector Vorticity Model on the Icosahedral Geodesic Grid

Ross Heikes, C.S. Konor, Joon-Hee Jung and D. Randall

Dept. of Atmospheric Science
Colorado State University

CMMAP team meeting
University of California, Berkeley

12 January 2011

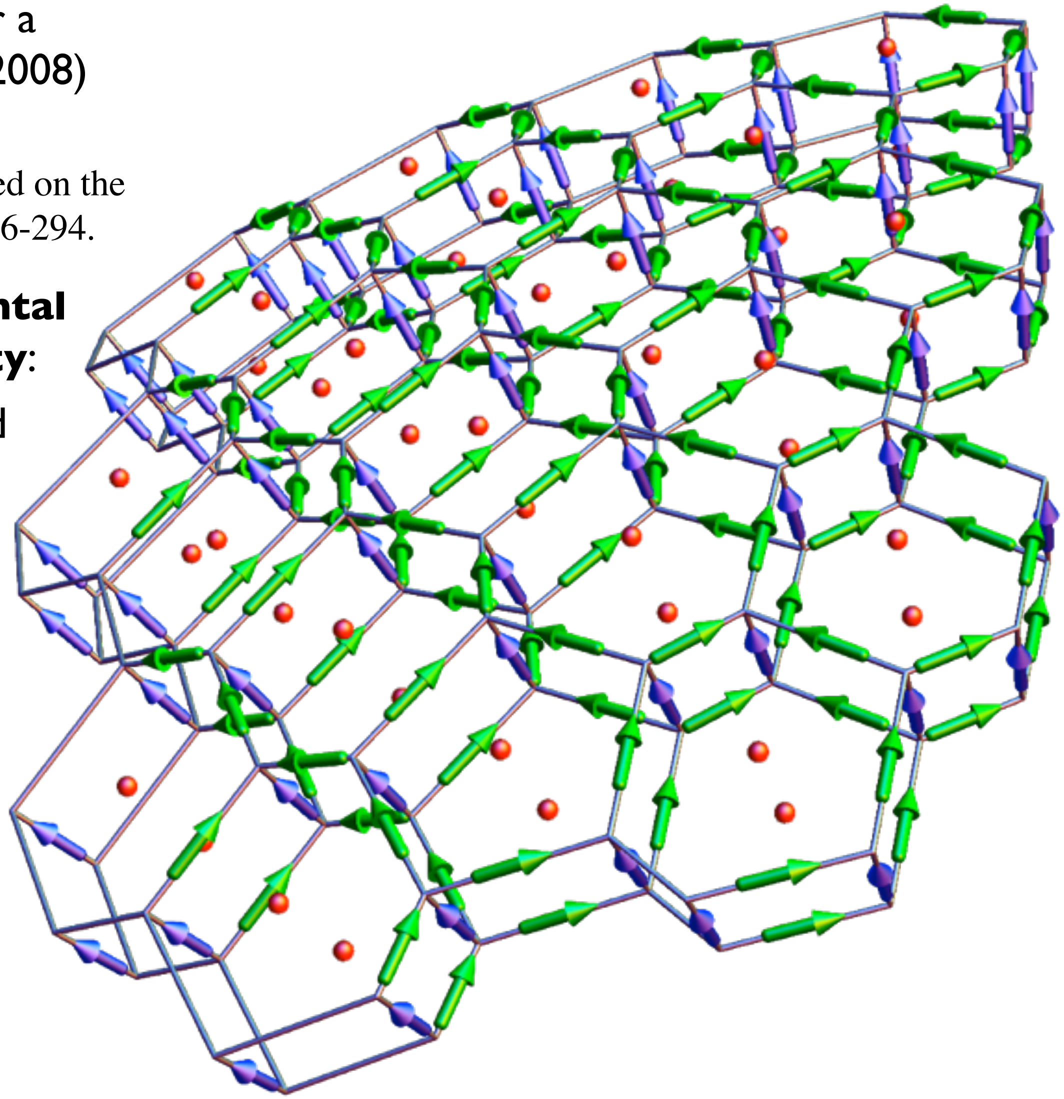
The Vector-Vorticity Dynamical Core (VVDC) on the Icosahedral Grid

- The VVDC was originally designed for a Cartesian grid by Jung and Arakawa (2008)

  Jung, J.-H., and A. Arakawa, 2008:
  A Three-Dimensional Anelastic Model Based on the Vorticity Equation. *Mon. Wea. Rev.*, **136**, 276-294.

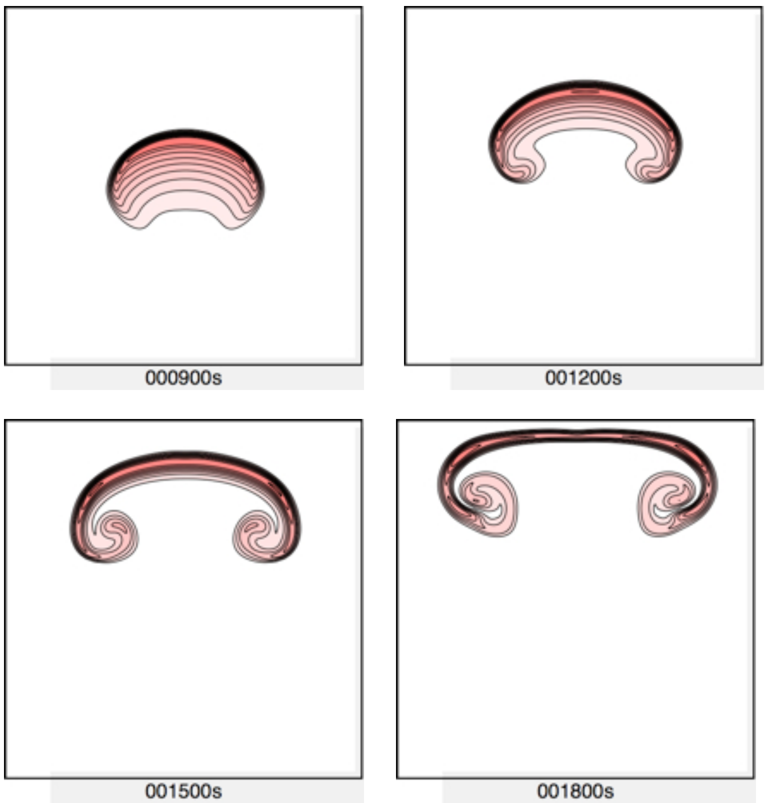
- The model predicts both the **horizontal** and **vertical** components of **vorticity**:
  - Horizontal (green arrows) is defined at cell edges and layer interfaces
  - Vertical (purple arrows) is defined cell corners and within a layer

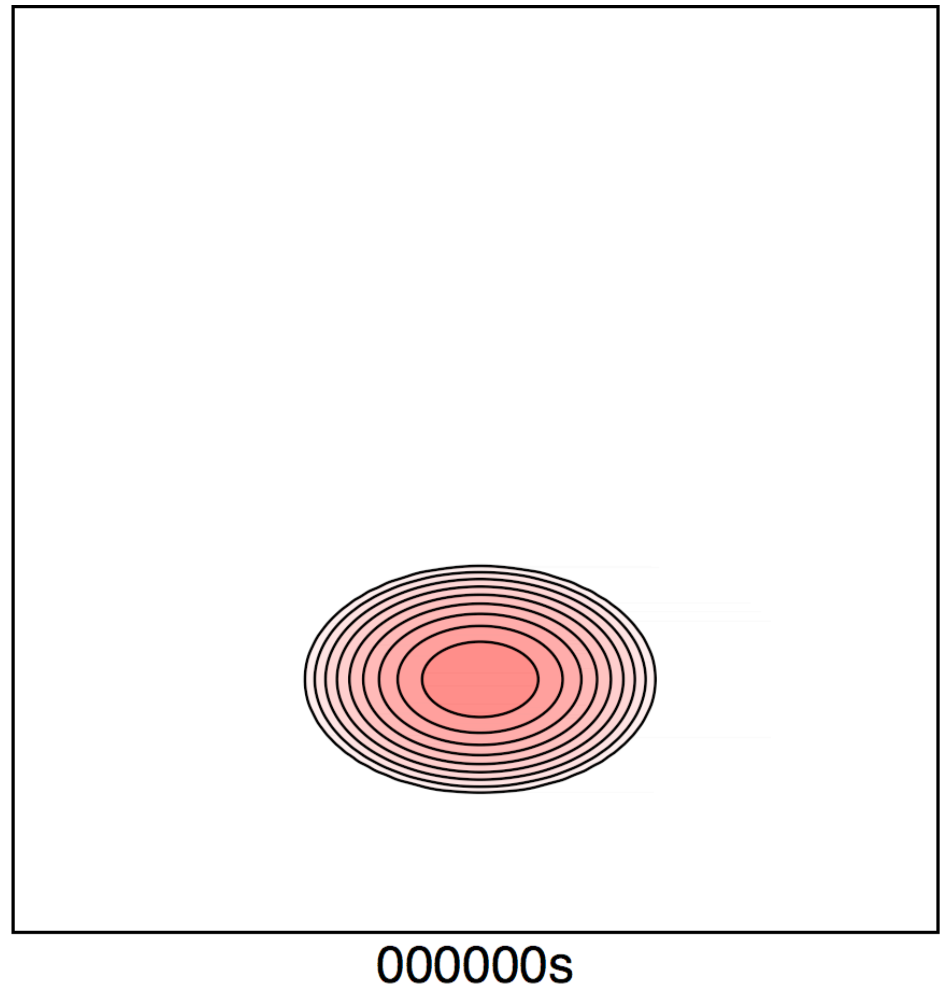
- The model predicts **Potential Temperature** (red dots) defined at layer interfaces



Rising bubble test case with the VVDC sphere

- This animation shows cross section of temperature through a rising warm bubble
- Contour interval 0.1 K



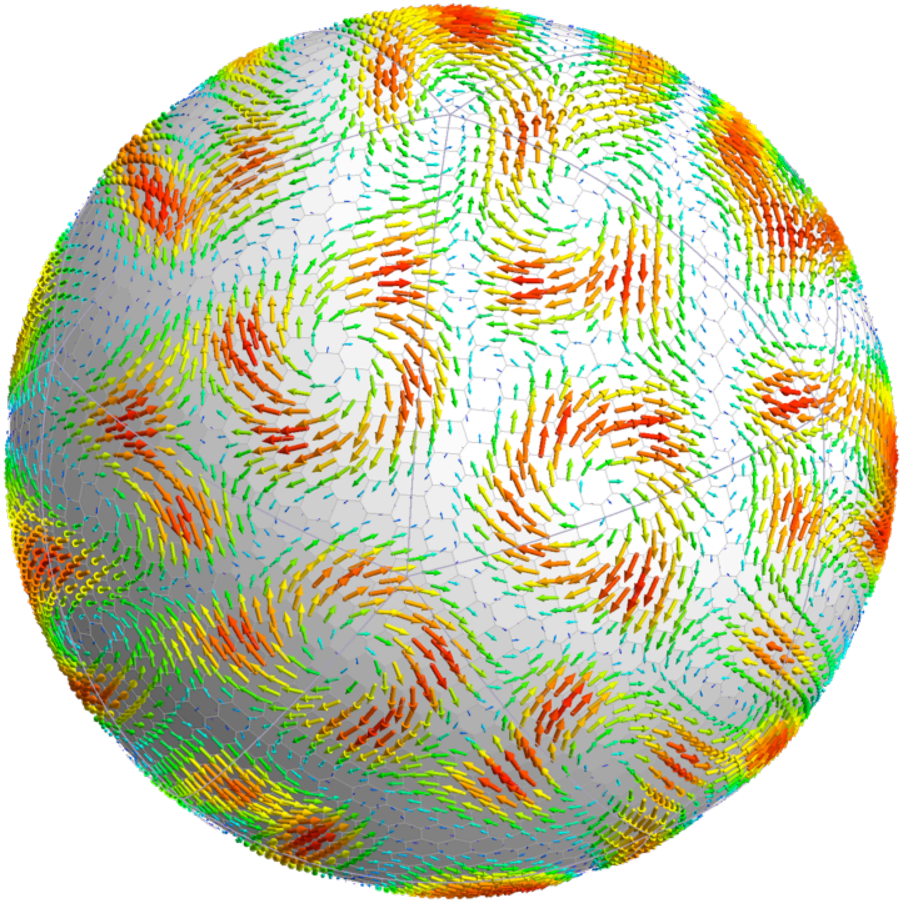


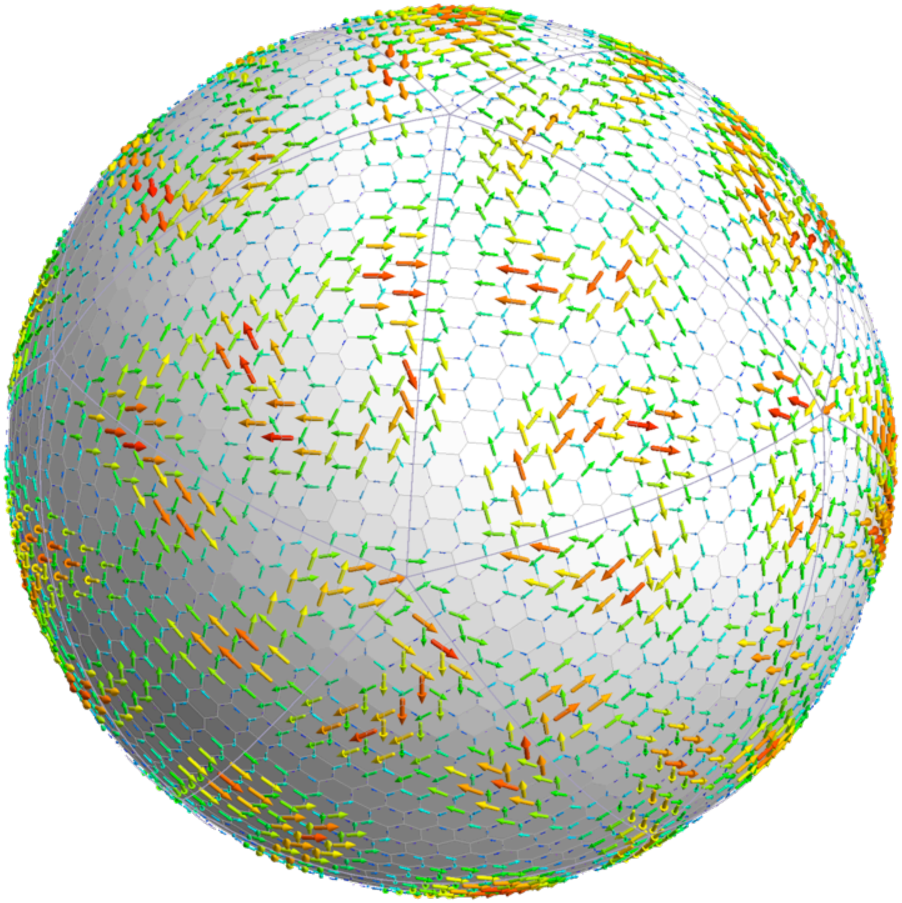
Radial basis functions -- motivation

✦ Current improvements to the model focus on aspects that are unique to the icosahedral grid

Consider a fully 3D field (tangent to surface of the sphere) defined at cell edges.

Projection into the direction tangent to each cell wall. **Can we reconstruct the 3D field from this information?**



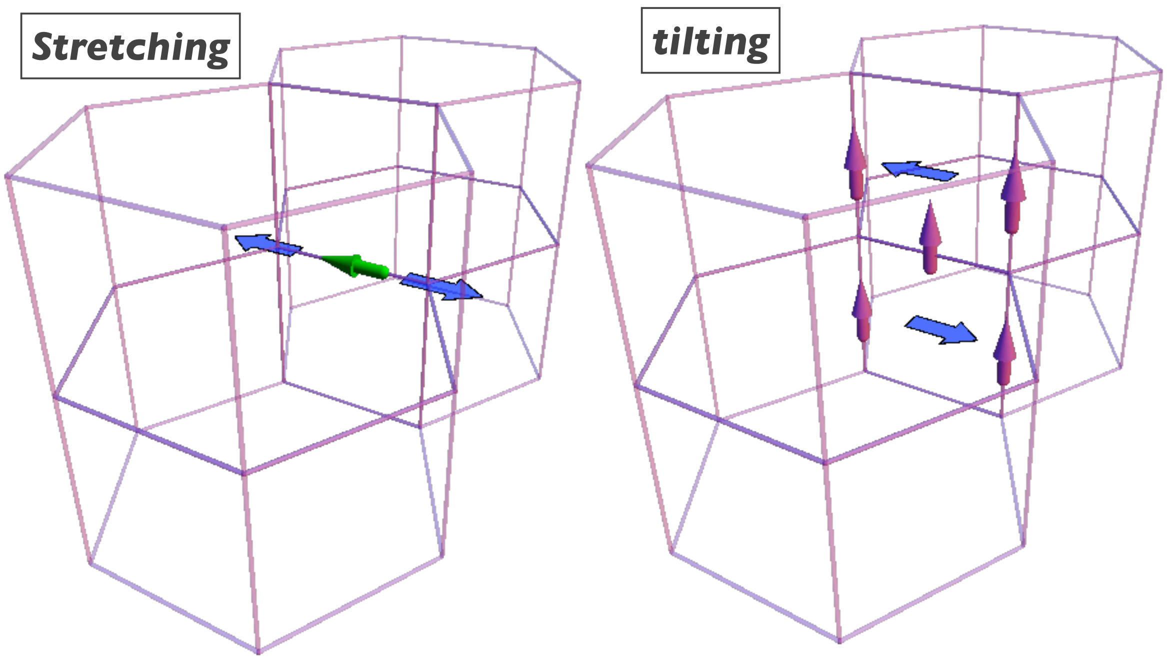


Radial basis functions -- motivation

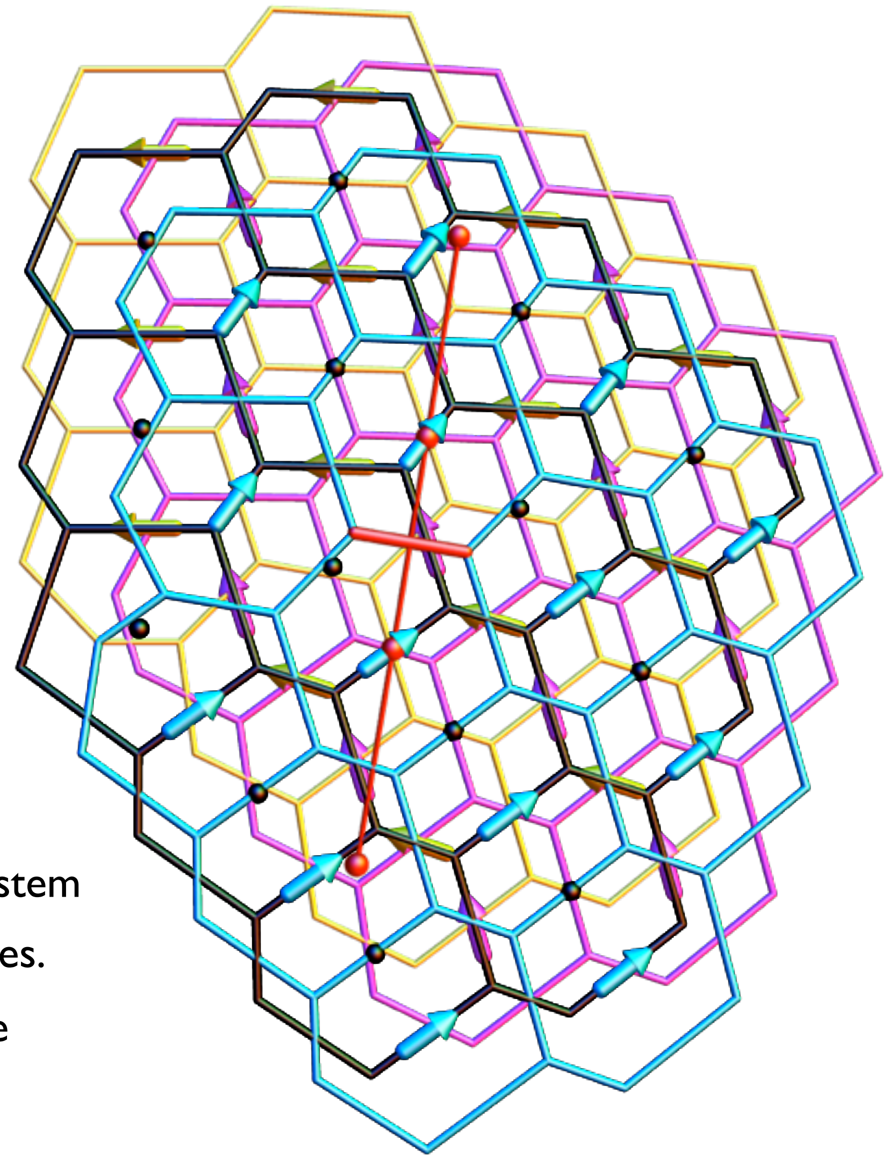
- **How would we use a 3D reconstruction?**
- Recall that the horizontal vorticity ($\boldsymbol{\eta}$) is defined tangent to cell walls and the horizontal wind ($\mathbf{v}$) is defined normal to cell walls.
- Consider the $\boldsymbol{\eta}$ equation

$$\frac{\partial \boldsymbol{\eta}}{\partial t} = -\overbrace{\nabla_H \bullet (\boldsymbol{\eta}_a \mathbf{v})}^{\substack{horizontal\\advection}} - \overbrace{\frac{\partial}{\partial z}(\boldsymbol{\eta}_a w)}^{\substack{vertical\\advection}} + \overbrace{(\boldsymbol{\eta}_a \bullet \nabla_H)\mathbf{v}}^{stretching} + \overbrace{\zeta_a \frac{\partial \mathbf{v}}{\partial z}}^{tilting} + \mathbf{v}\nabla_H \bullet (2\boldsymbol{\Omega}_H) - \overbrace{\mathbf{k} \times \nabla_H B}^{buoyancy} + \overbrace{\mathbf{k} \times \left(\frac{\partial \mathbf{F}_v}{\partial z} - \nabla_H F_w\right)}^{turbulent-drag\ force}$$

- So we need to approximate $\boldsymbol{\eta}$ and $\mathbf{v}$ in arbitrary directions.



Radial basis functions -- Advection of η



- Consider the mass grid (black points)
- **Three** cell edges are associated with each mass point. (cyan, magenta, yellow)
- We can construct Voronoi control volumes for each type (color) of edge.
- We would like to use the upstream biased advection method developed for the mass grid.
- Consider the flux of η across a particular cyan cell wall.
- Algorithm:
  - (1) Construct 3D η in a global coordinate system
  - (2) Construct an upstream flux. Sum the fluxes.
  - (3) Project 3D η back to the local coordinate system

Radial basis functions -- math

- This is based on

  Bonaventura et al. (2010) Kernal-based vector field reconstruction in computational fluid dynamic models. *Int. J Numer. Meth. Fluids*

- Suppose we wish to approximate a planar vector field

$$u : \mathbb{R}^2 \to \mathbb{R}^2 \text{ where } u = \left[u_1(\mathbf{x}), u_2(\mathbf{x})\right]$$

- Given a set of N **points** and N **unit vectors** associated with each point

$$\left\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\right\} \subset \mathbb{R}^2 \quad \text{and} \quad \mathbf{n}_i^T = \left[n_i^1, n_i^2\right] \in \mathbb{R}^2 \text{ for } 1 \le i \le N$$

- Given a set of N **scalar samples** of the field u

$$u_i = \mathbf{n}_i^T \bullet u(\mathbf{x}_i) \in \mathbb{R} \text{ for } 1 \le i \le N$$

Radial basis functions -- math

- The reconstruction function a **continuous** function

$$s:\mathbb{R}^2 \to \mathbb{R}^2 \quad \text{where} \quad s = \left[s_1(\mathbf{x}), s_2(\mathbf{x})\right]$$

with the property that

$$\mathbf{n}_i^T \bullet u(\mathbf{x}_i) = \mathbf{n}_i^T \bullet s(\mathbf{x}_i) \quad \text{for } 1 \le i \le N$$

- The reconstruction function s has the form

$$s(\mathbf{x}) = \sum_{i=1}^{N} c_i \begin{bmatrix} \phi(\mathbf{x}-\mathbf{x}_i) n_i^1 \\ \phi(\mathbf{x}-\mathbf{x}_i) n_i^2 \end{bmatrix} \quad \text{where} \quad \phi(\mathbf{x}) = e^{-\alpha\|\mathbf{x}\|}$$

- The unknown coefficients are computed from the linear system

$$Ac = (u_1, u_2, \ldots, u_N)^T$$

where the entries of the matrix

$$a_{ij} = n_j^1 n_i^1 \phi(\mathbf{x}_j - \mathbf{x}_i) + n_j^2 n_i^2 \phi(\mathbf{x}_j - \mathbf{x}_i) \quad \text{for } 1 \le i, j \le N$$

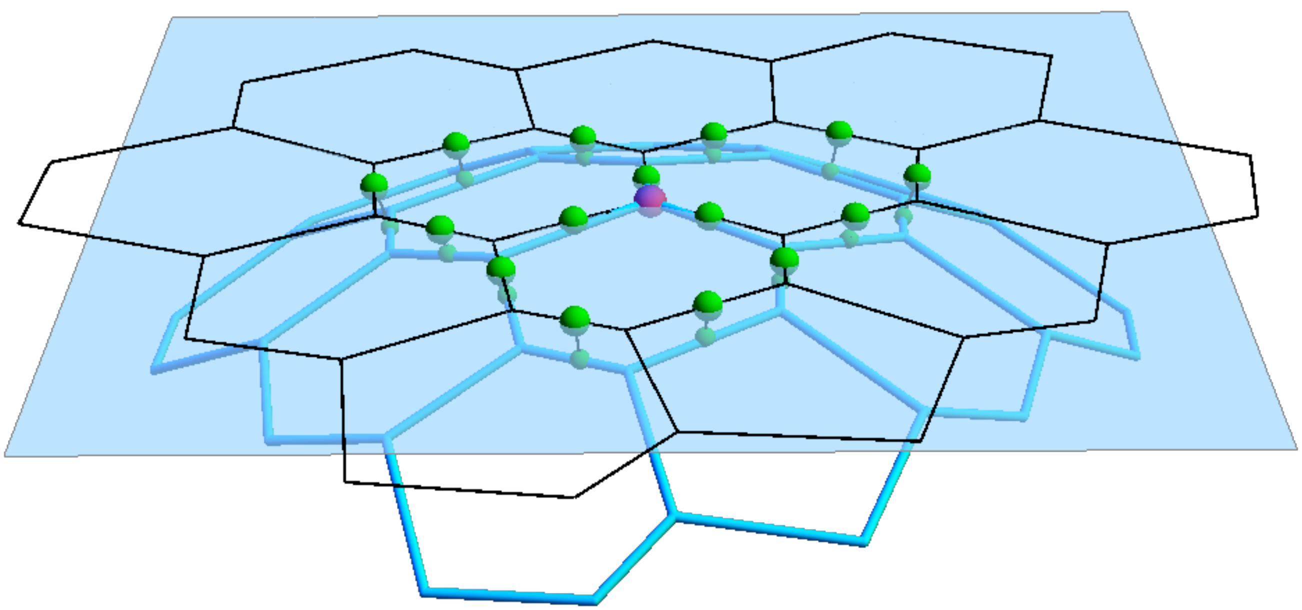
Radial Basis functions -- projection to tangent plane

- In the case of the sphere, we could construct

$$u : \mathbb{R}^3 \to \mathbb{R}^3$$

  This resulted in a very poorly conditioned matrix

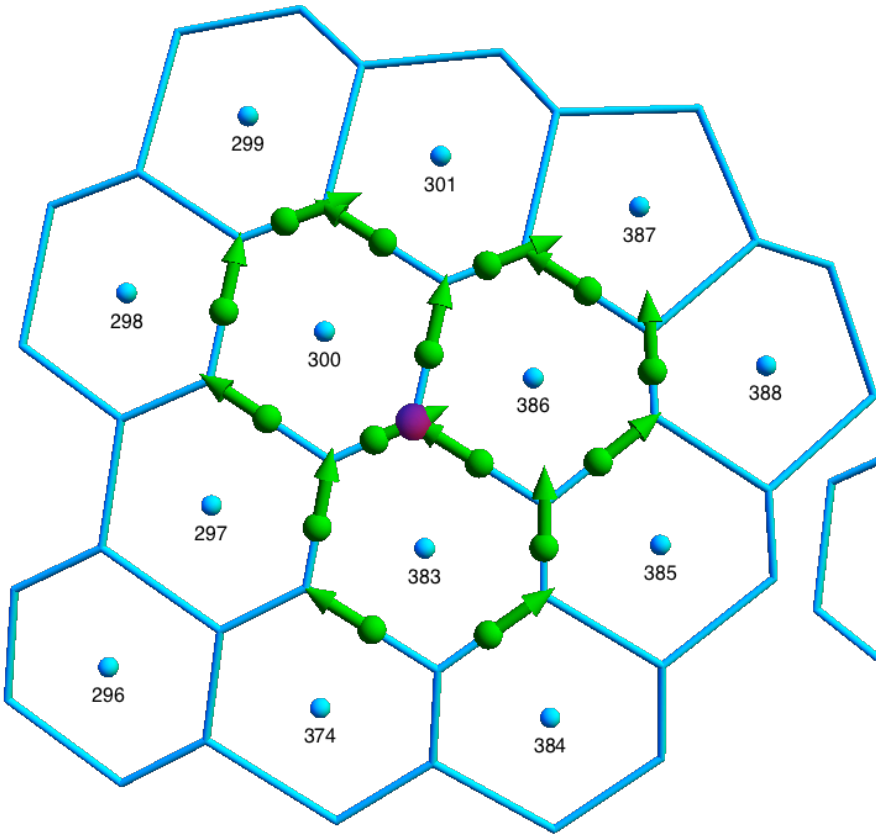
- Instead, the 3D icosahedral grid points are projected into a 2D plane tangent to the sphere
- And solve for the coefficients in the 2D plane

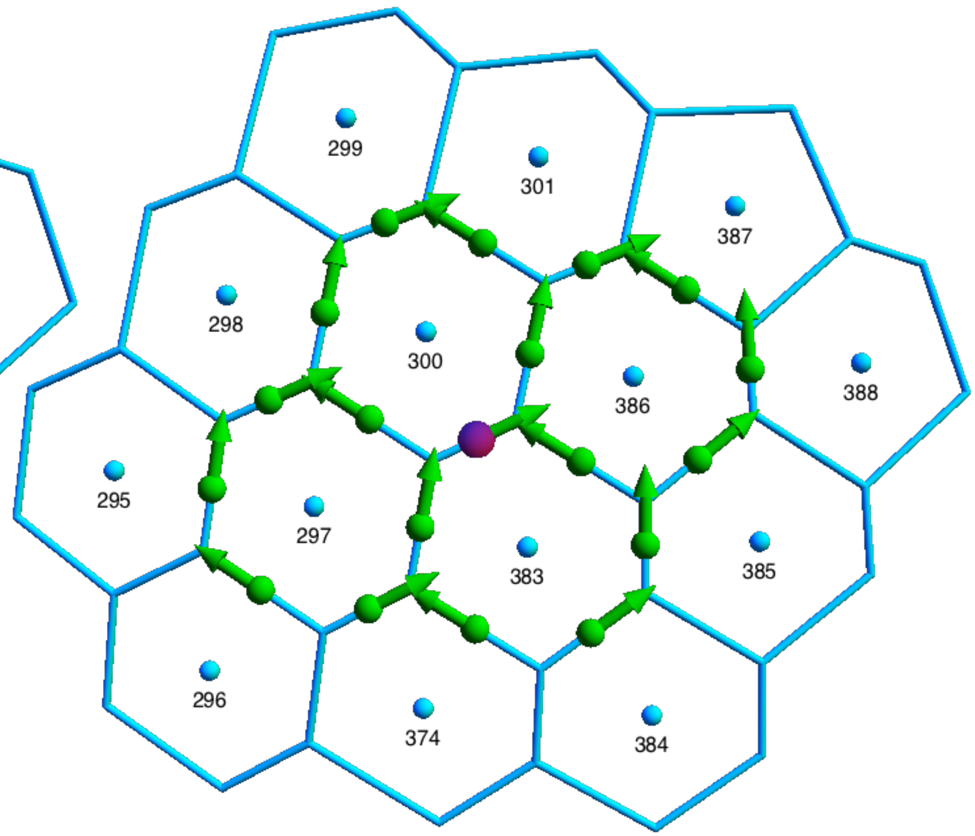


Radial basis functions -- stencil

✦ Stencils are defined at cell corners and cell edges (purple points):

- corners use 15 edge points
- edges use 19 (or 18 at pentagons) edge points





Radial basis functions -- practical stuff

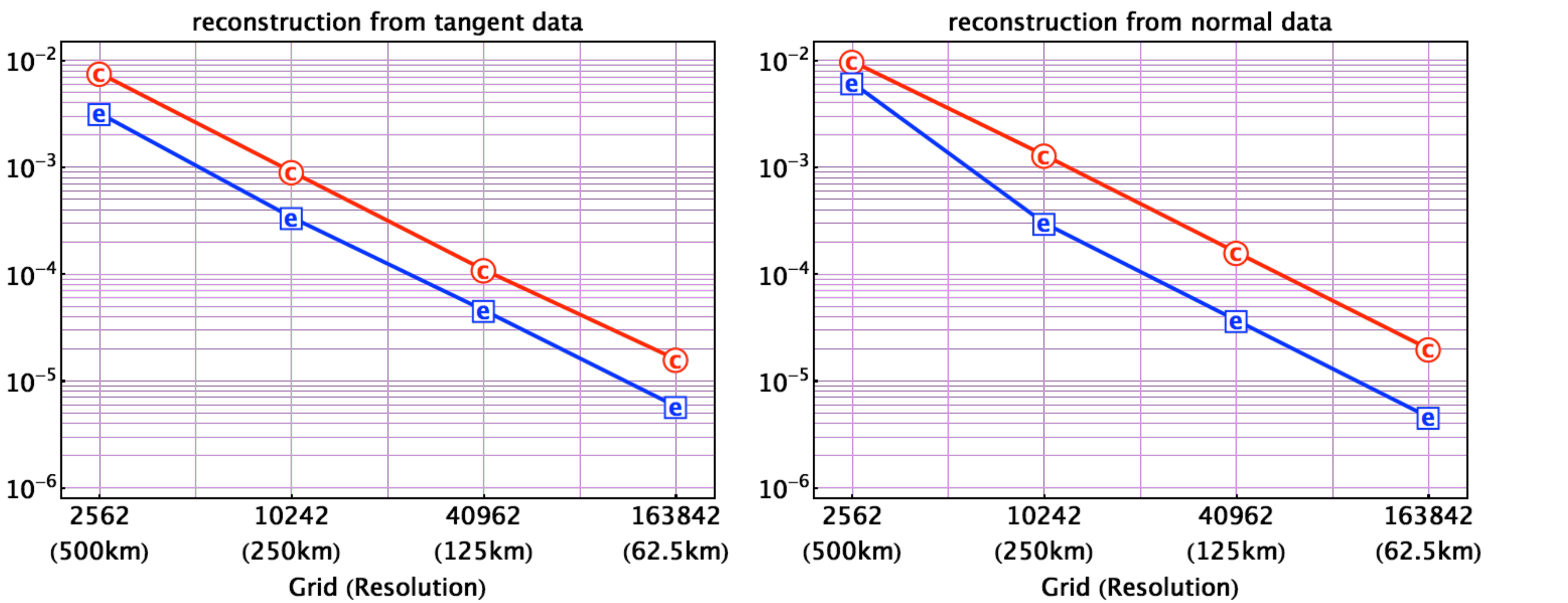
- A *Mathematica* code computes the coefficients only once and stores to a netCDF file
- To compute a 3D vector involves 3 dot products

Radial basis functions -- error

- Given an analytic test case plot the L_∞-norm error

$$error = \max\left\{ \left\| \mathbf{x}_{appx} - \mathbf{x}_{true} \right\|_2 \right\}$$

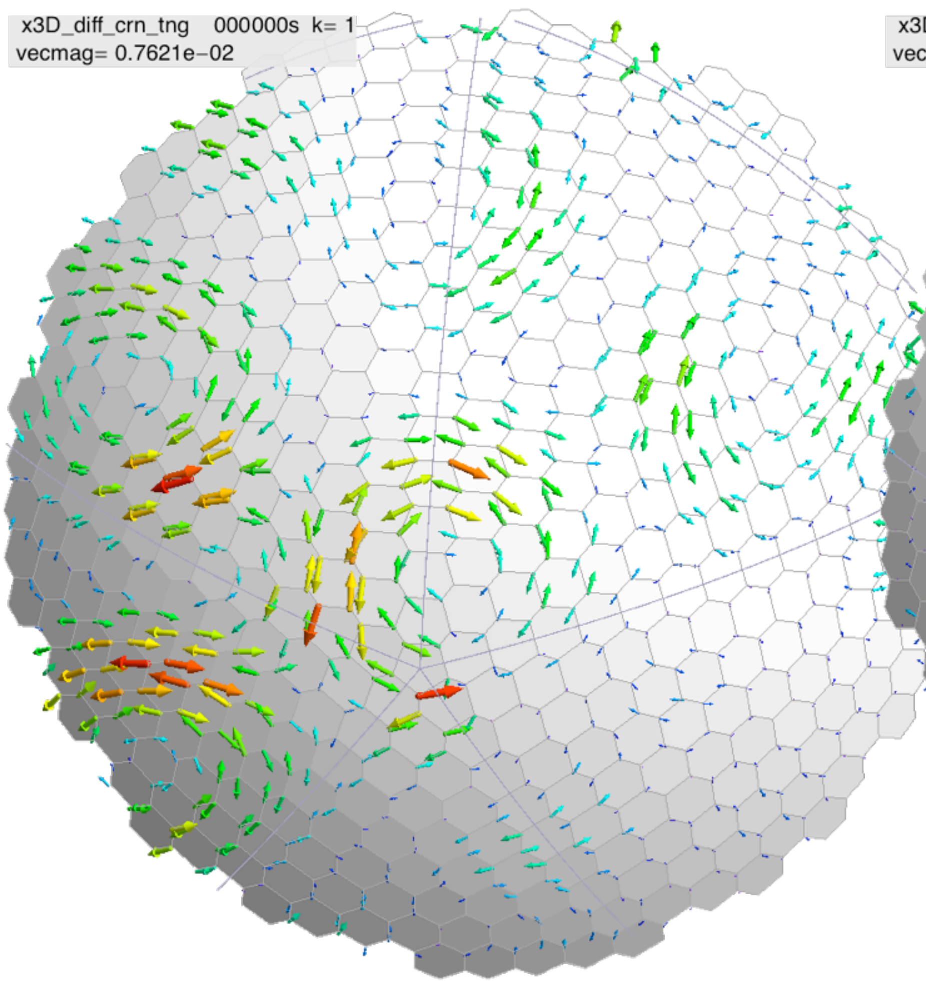
- With input data defined tangent and normal to cell walls
- The plots show the 3D reconstruction at:
  - cell corners (red lines)
  - cell edges (blue lines)



Radial basis functions -- structure of the error

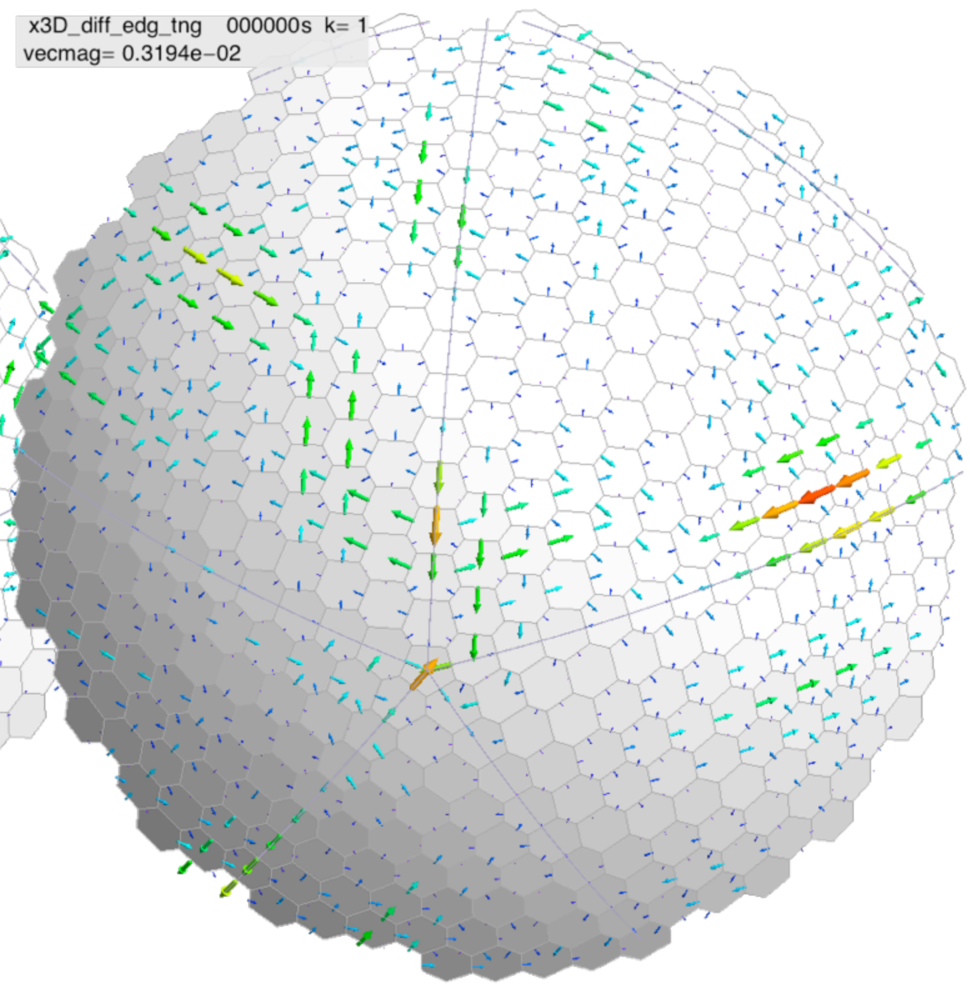
Reconstruction at **corners**:

- little structure from the grid
- opposing directions



Reconstruction at **edges**:

- more structure from the grid
- error only in tangent direction



Conclusions

- ✦ The use of RBFs offers a dramatic improvement over the previous approximation of 3D fields.
- ✦ RBFs provide a tool that could be used to improve other parts of the model, e.g.
  - finite-difference operators based on Gauss' theorem could more accurately approximate the line integral along walls of a given control volume
  - filtering to remove computational mode