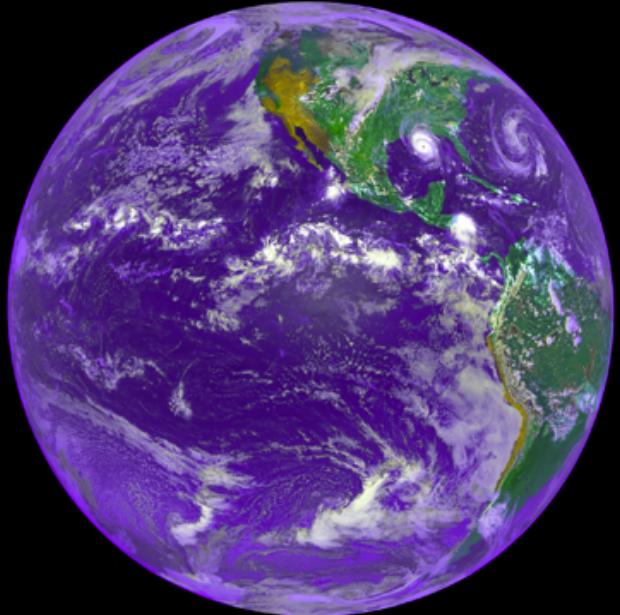


# A radial basis function partition of unity method for transport on the sphere\*



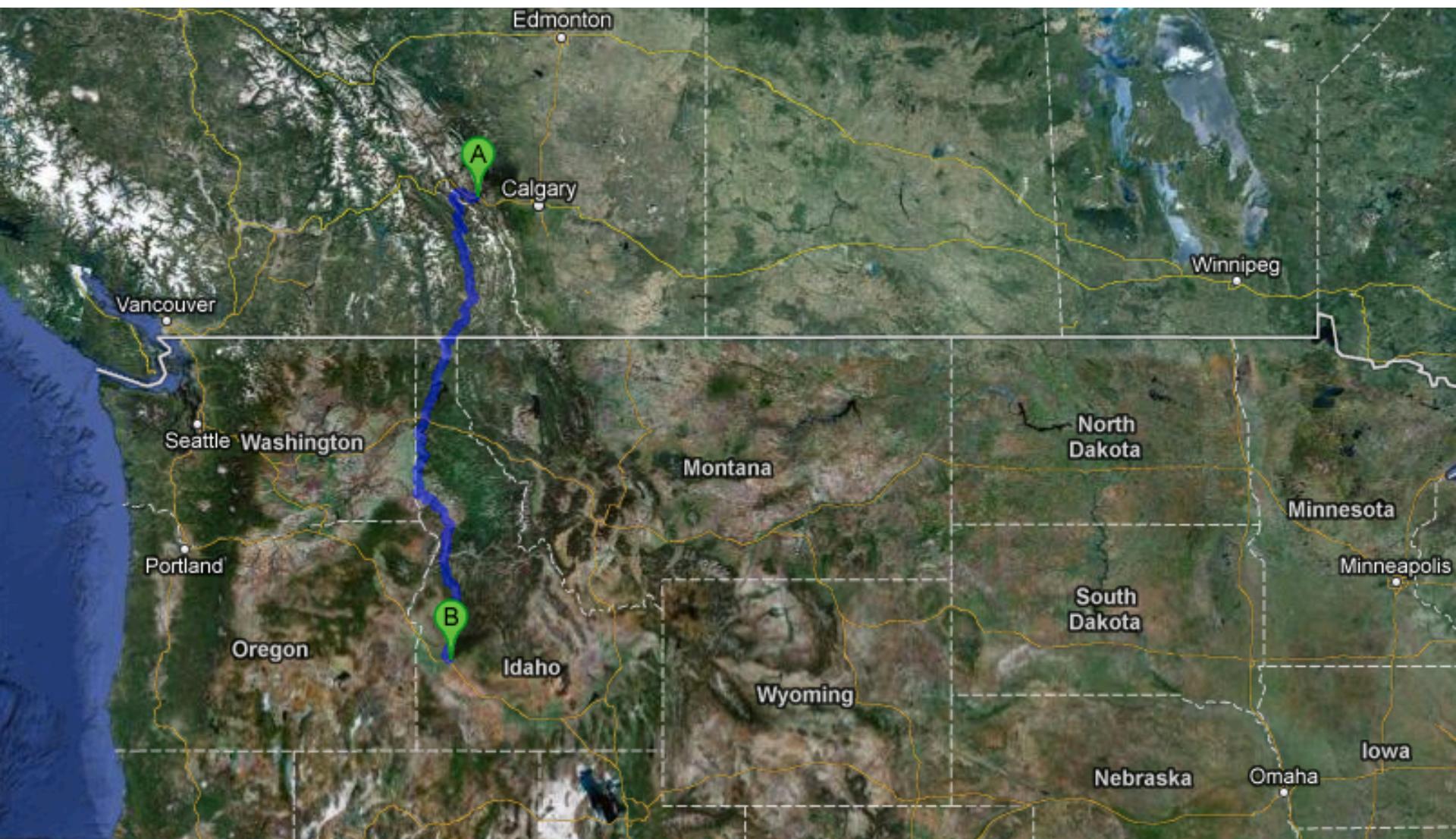
Grady B. Wright  
Boise State University

Collaborator: Kevin Aiton, Boise State University

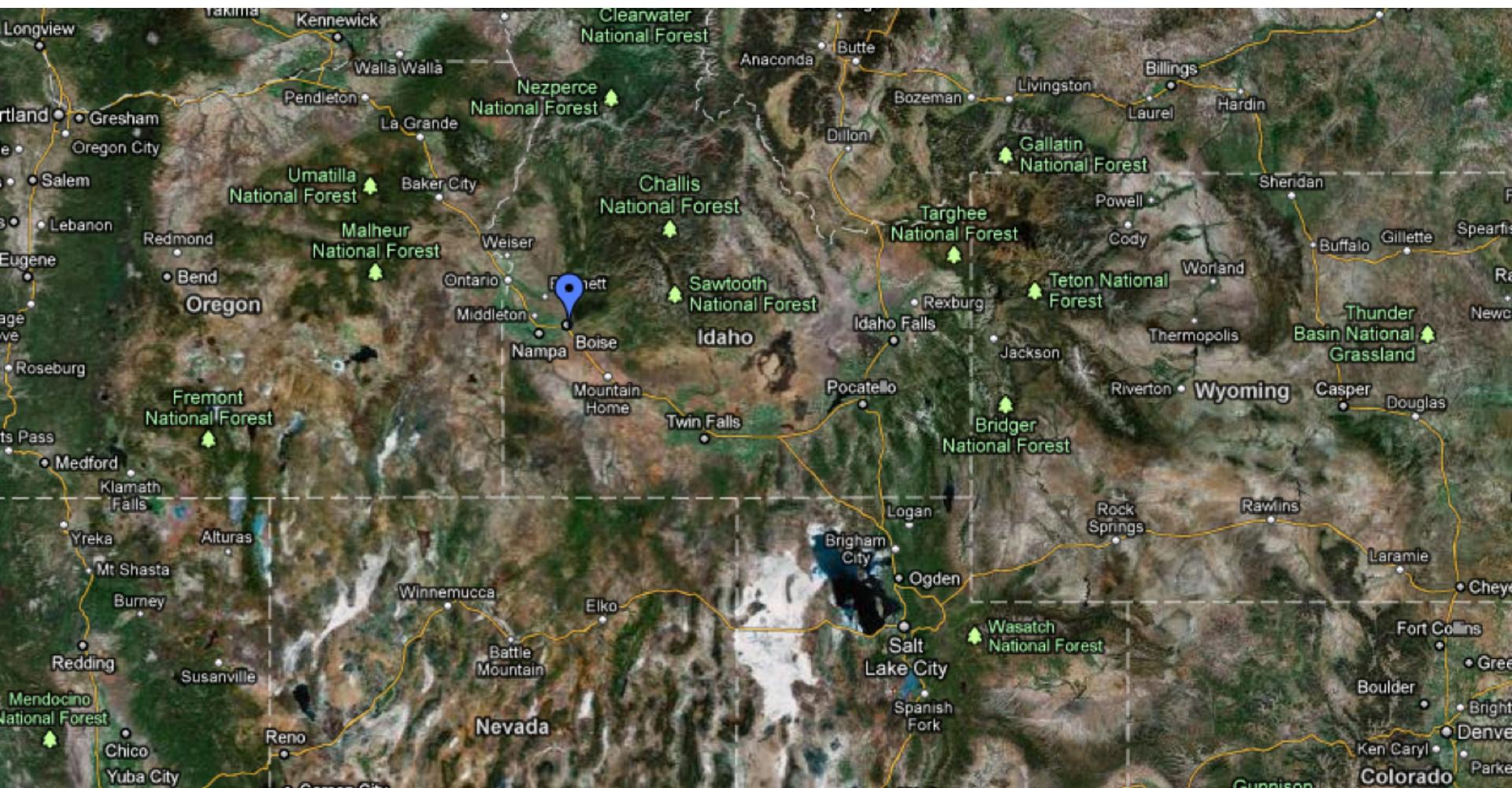
\*This work is supported by NSF CMG grant DMS 0934581

# First some geography

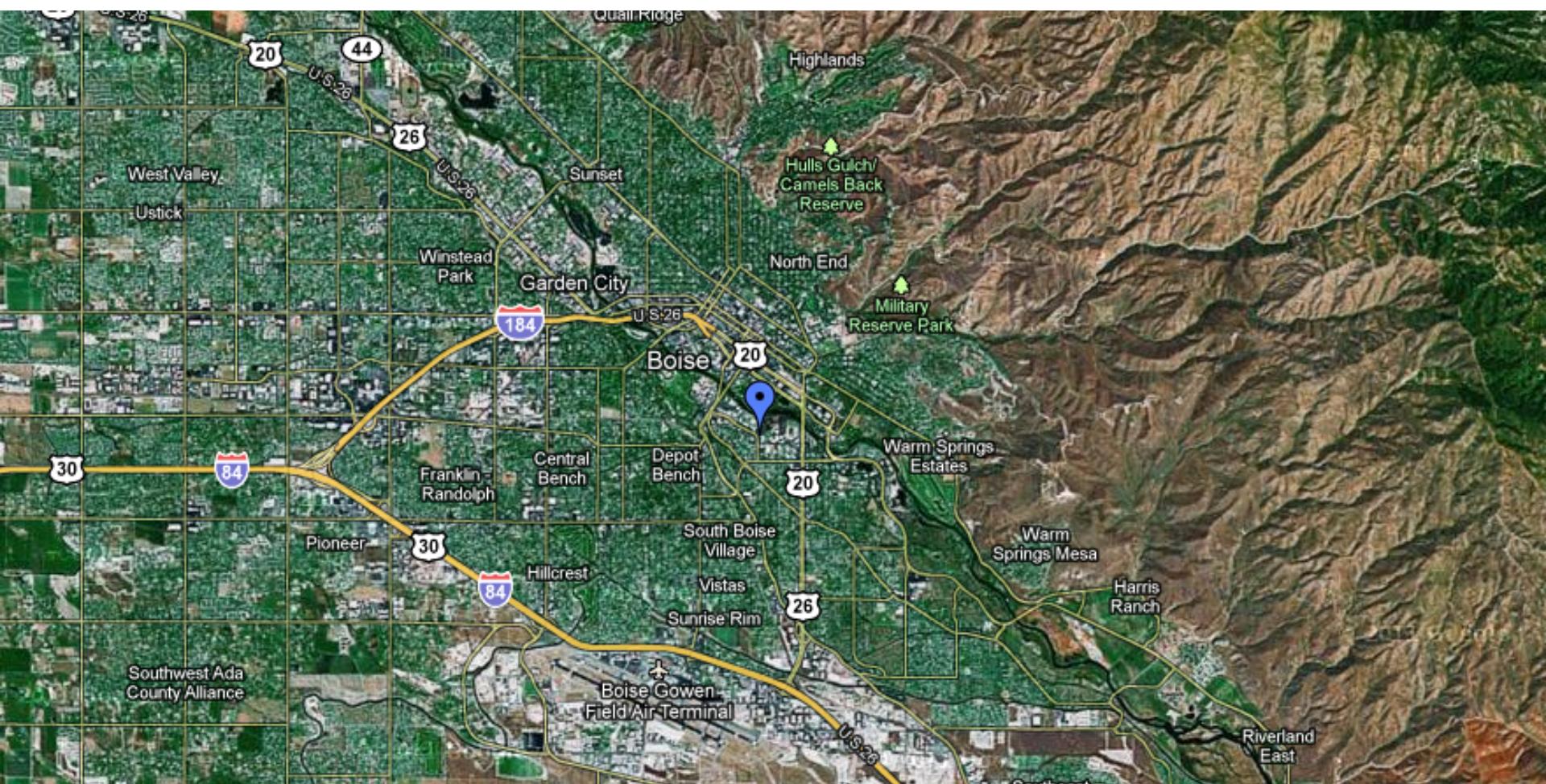
---



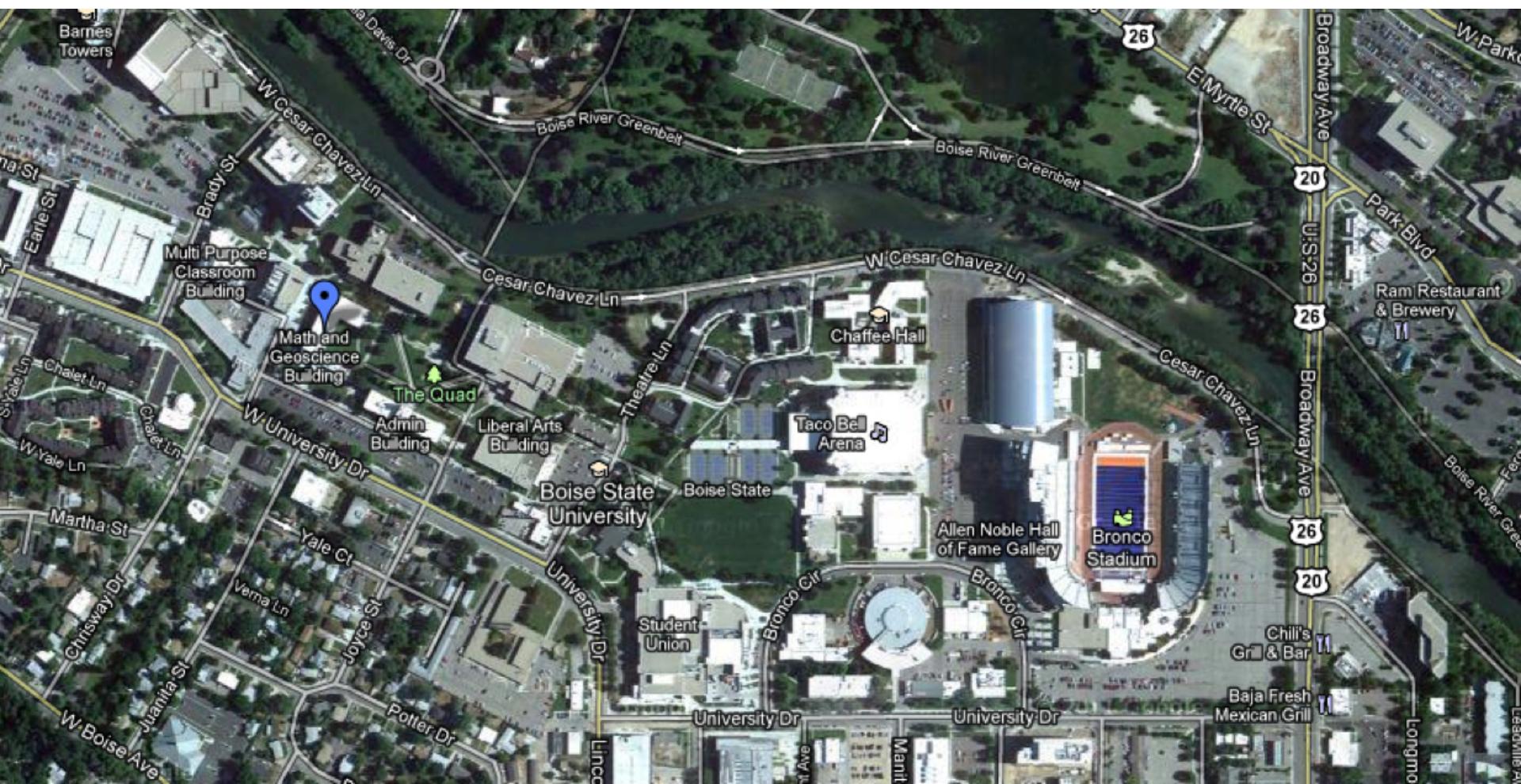
# First some geography



# First some geography



# First some geography



# First some geography

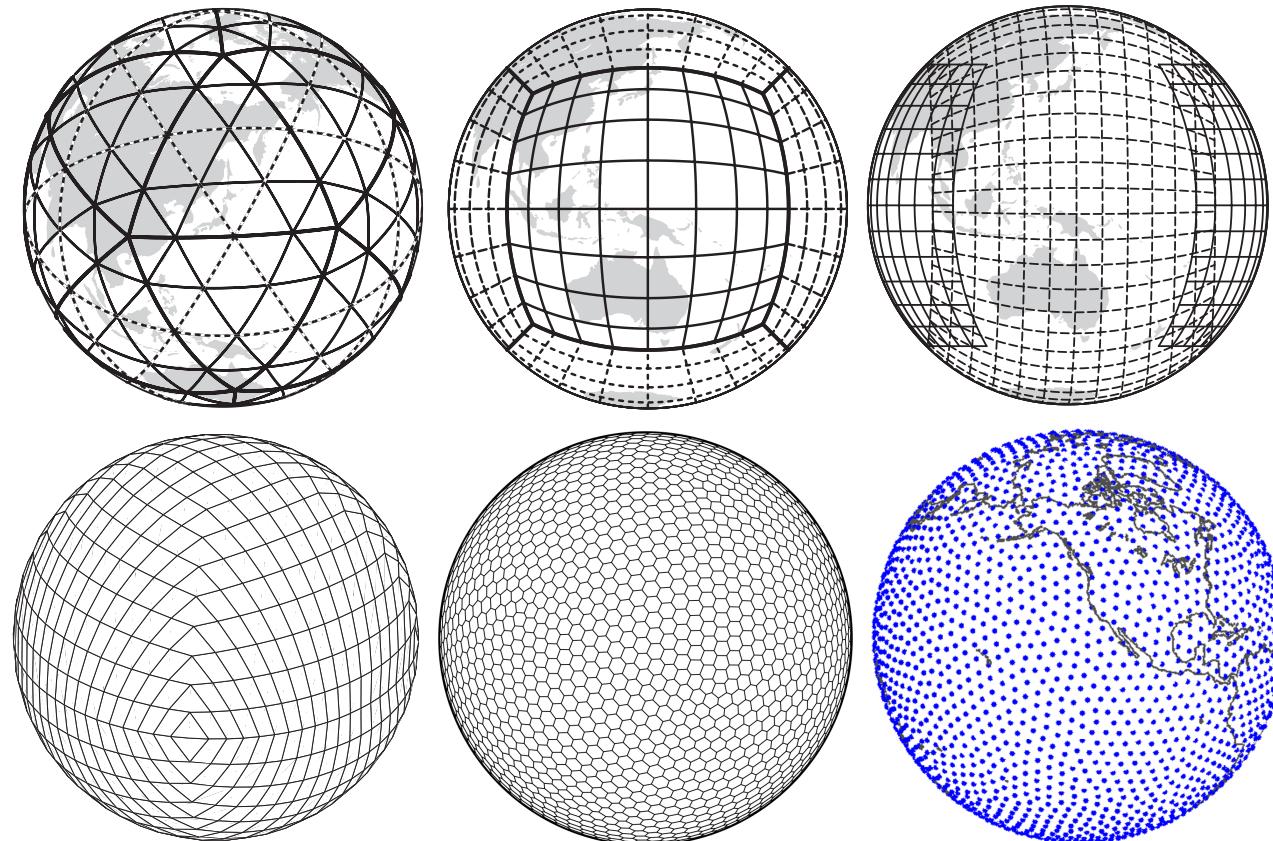
---

BIRS Workshop  
June 28, 2012



# Overview

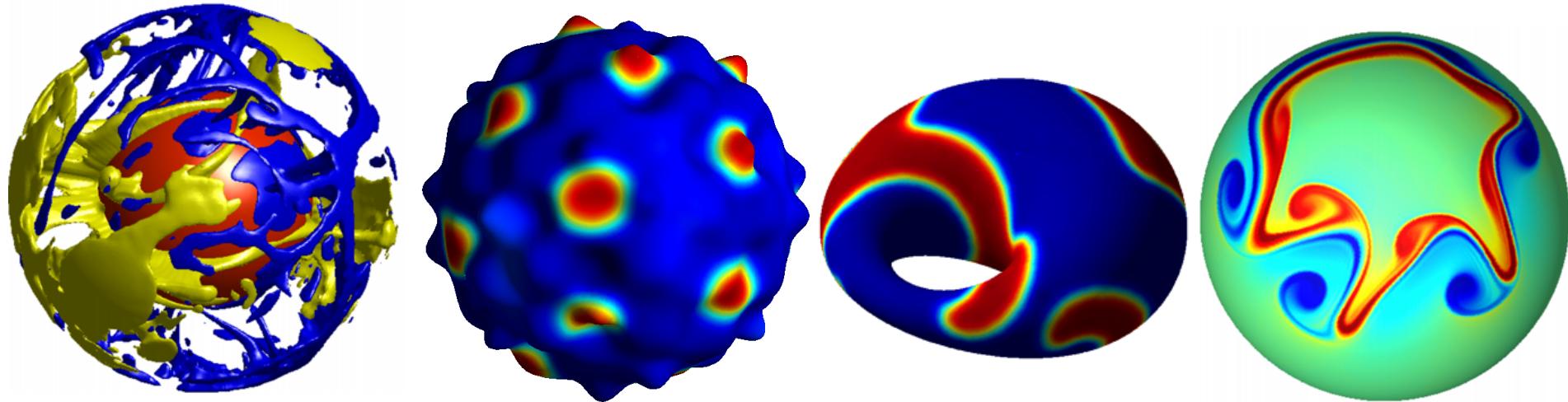
- Transport (or advective) processes dominate atmospheric fluid motions on all scales.
- Numerical methods for complex atmospheric models have to get transport correct.
- Existing techniques for transport (and more) in spherical geometry:



# Overview: Radial Basis Functions (RBFs)

BIRS Workshop  
June 28, 2012

- Meshfree methods for PDEs



Collaborators: N. Flyer, A.L. Fogelson, E. J. Fuselier, L.H Kellogg, R.M. Kirby, E. Lehto, D.A. Yuen

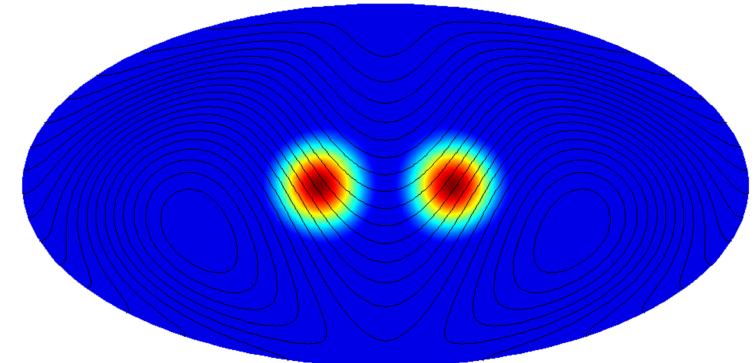
- Some references:
  1. E. J. Kansa, *Comput. Math. Appl.*, 19:147–161, 1990.
  2. G. E. Fasshauer. *Meshless Approximation Methods with MATLAB*. World Scientific Publishers, Singapore, 2007.
  3. H. Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge, 2005.
  4. N. Flyer and B. Fornberg, *Computers and Fluids*, 26 (2011), 23–32.

# Transport equation on the sphere

---

- We consider the **transport** of a scalar valued quantity  $h$  on the surface of the unit sphere in an incompressible velocity field  $\mathbf{u}$ .
- The governing PDE can be written in **Cartesian coordinates** as:

$$h_t + \mathbf{u} \cdot (\mathbf{P} \nabla h) = 0$$



$\mathbf{P}$  projects arbitrary three-dimensional vectors onto a plane tangent to the unit sphere at  $\mathbf{x}$ :

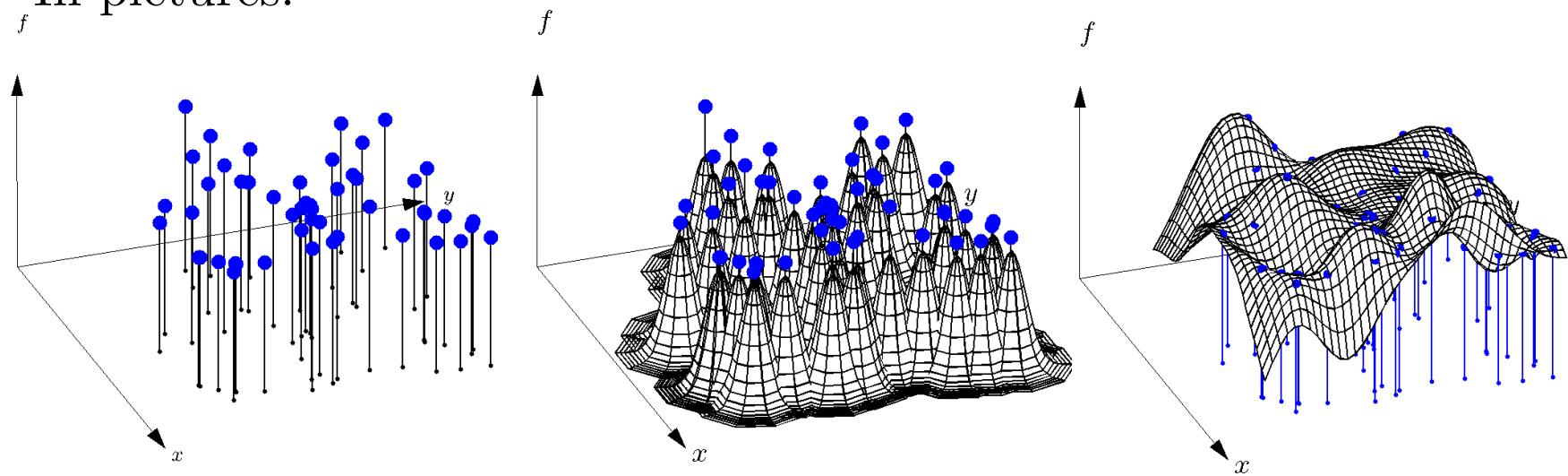
$$\mathbf{P} = \mathbf{I} - \mathbf{x}\mathbf{x}^T = \begin{bmatrix} (1-x^2) & -xy & -xz \\ -xy & (1-y^2) & -yz \\ -xz & -yz & (1-z^2) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x^T \\ \mathbf{p}_y^T \\ \mathbf{p}_z^T \end{bmatrix}$$

- $\mathbf{u}$  is tangent to the sphere and can be defined by a stream function:

$$\mathbf{u} = \mathbf{x} \times \nabla \psi$$

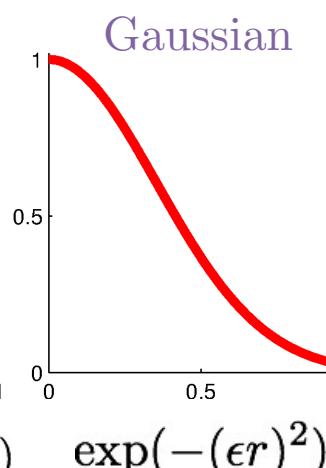
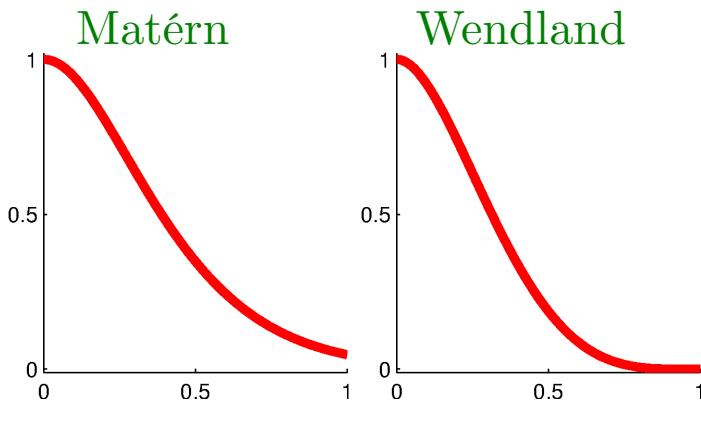
# Intro to RBFs via interpolation

- In pictures:

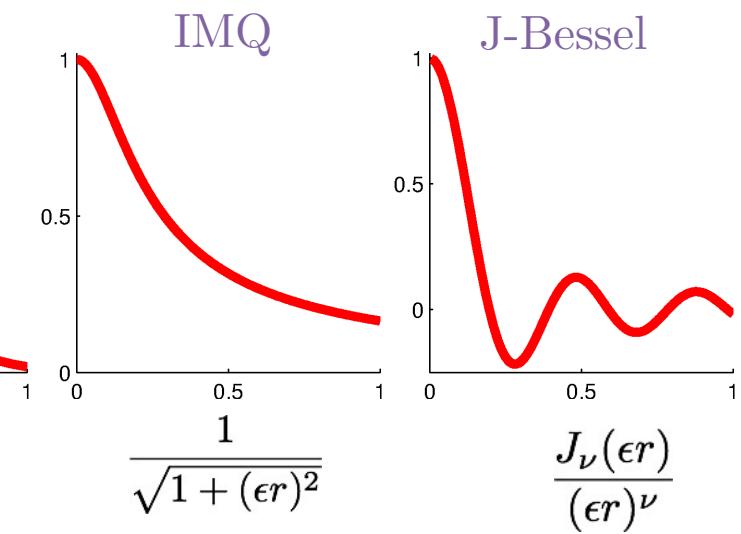
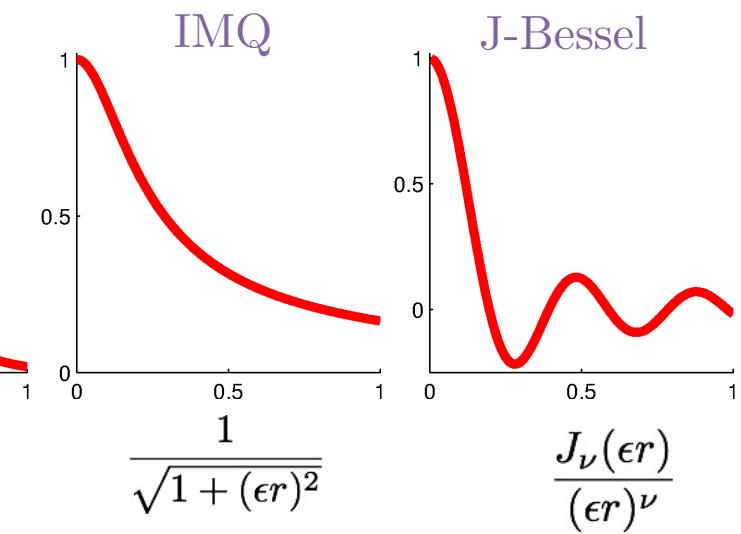
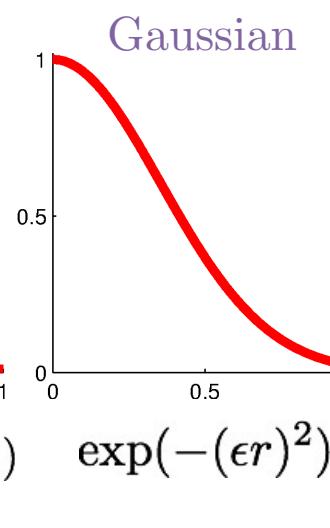


- Examples of radial kernels:  $\phi(\|\mathbf{x}\|_2) := \phi(r)$

Finite-smoothness



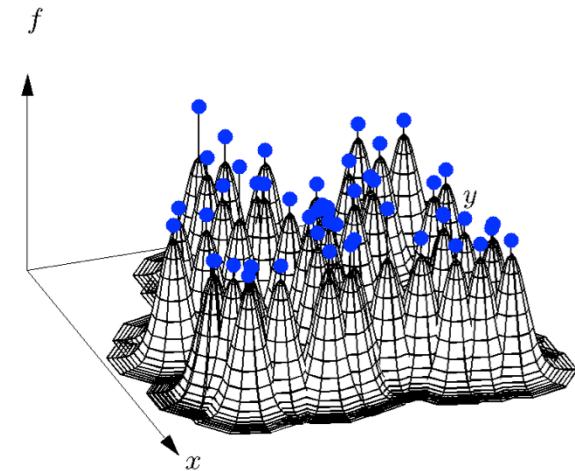
Infinite-smoothness



# Intro to RBFs via interpolation

- Let  $X = \{\mathbf{x}_j\}_{j=1}^N$  be a distinct set of scattered nodes on  $\mathbb{R}^d$  and  $f$  some target function sampled at  $X$ .
- RBF interpolant of  $f|_X$  is given by

$$s_X(\mathbf{x}) = \sum_{j=1}^N c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|),$$



where  $\|\cdot\|$  is the standard two-norm.

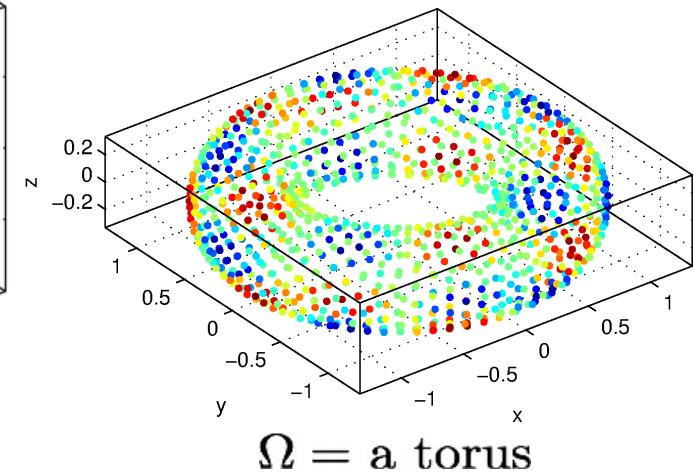
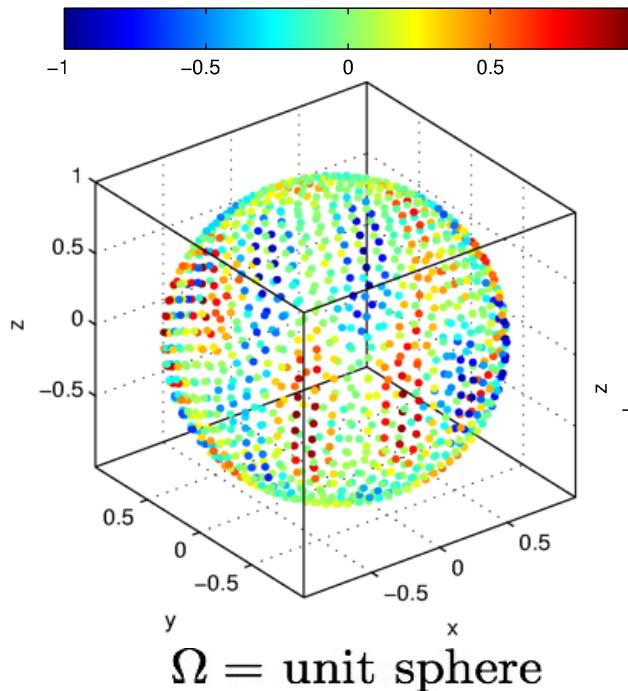
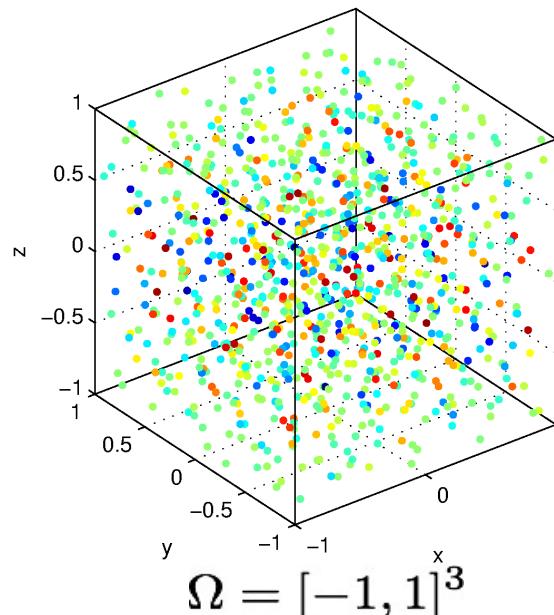
- The coefficients  $c_j$  are determined from collocation:

$$\underbrace{\begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix}}_{A_X} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}}_{\underline{c}} = \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}}_{\underline{f}}$$

# Geometric flexibility of RBFs

- Consider interpolation of  $f$  from a set of scattered nodes  $X$  on  $\Omega \subset \mathbb{R}^d$

Examples:



- Form of the RBF interpolant does not change:

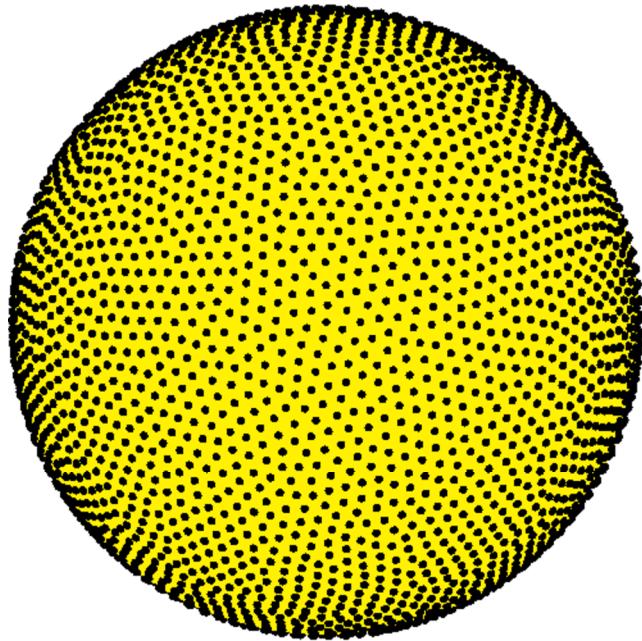
$$s_X(\mathbf{x}) = \sum_{j=1}^N c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|)$$

- Can use Cartesian (extrinsic) coordinates!

# RBFs and partition-of-unity on the sphere

---

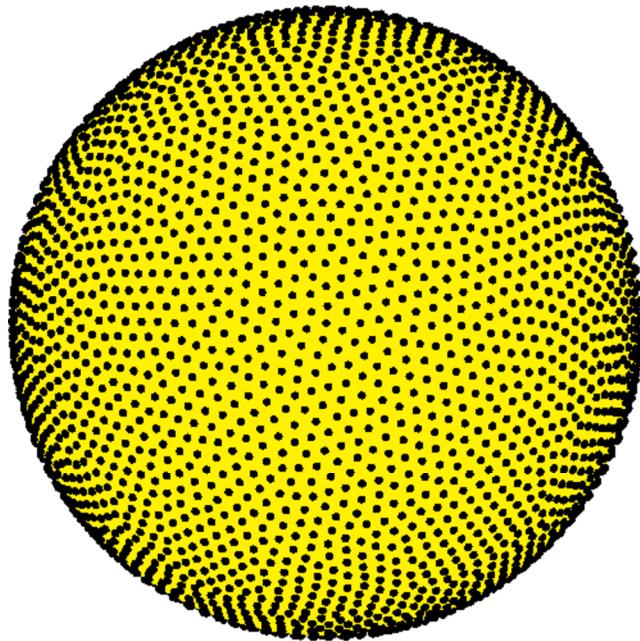
- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



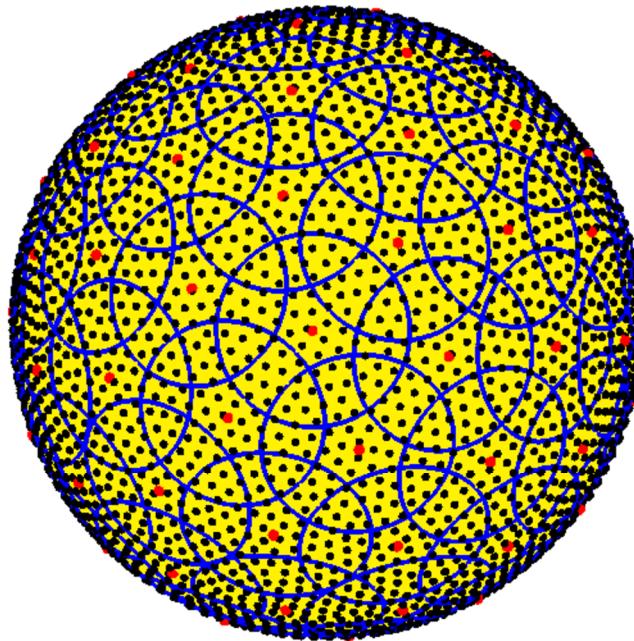
Key Steps:

1. Generate a set of overlapping patches (spherical caps)  $\Omega = \{\Omega_k\}_{k=1}^M$  that creates a uniform partition of the sphere with respect to the density of the nodes in  $X$ , and each patch contains roughly  $n$  nodes of  $X$ .

# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patch  $\Omega_k$

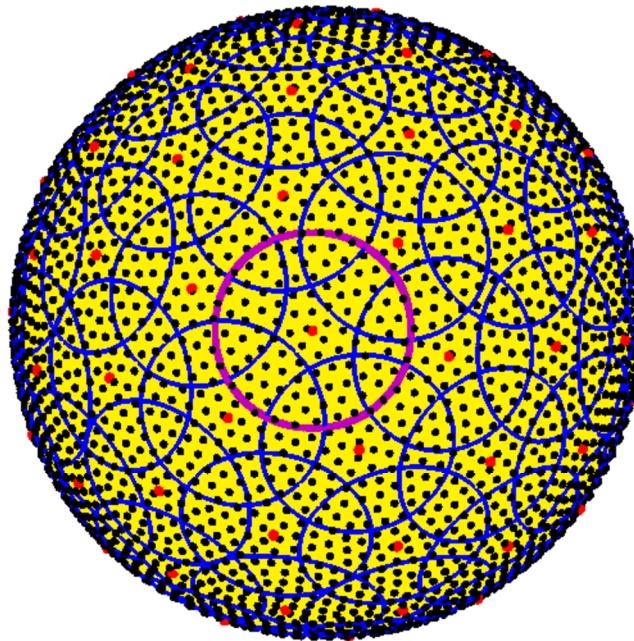
## Key Steps:

1. Generate a set of overlapping patches (spherical caps)  $\Omega = \{\Omega_k\}_{k=1}^M$  that creates a uniform partition of the sphere with respect to the density of the nodes in  $X$ , and each patch contains roughly  $n$  nodes of  $X$ .

# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patches  $\Omega_k$

Key Steps:

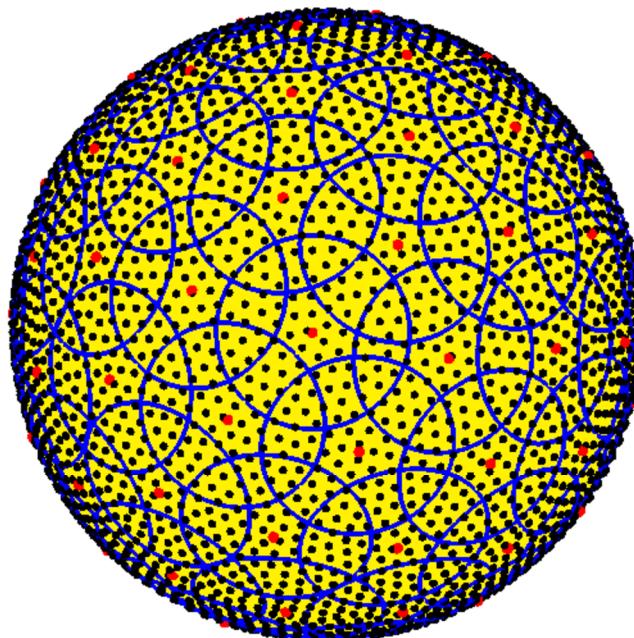
- Letting  $X_k$  denote the set of nodes in patch  $\Omega_k$ , construct RBF interpolants  $s_k$ , for  $k = 1, \dots, M$ :

$$s_k(\mathbf{x}) = \sum_{j=1}^n c_j^k \phi(\|\mathbf{x} - \mathbf{x}_j^k\|)$$

# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patches  $\Omega_k$

Key Steps:

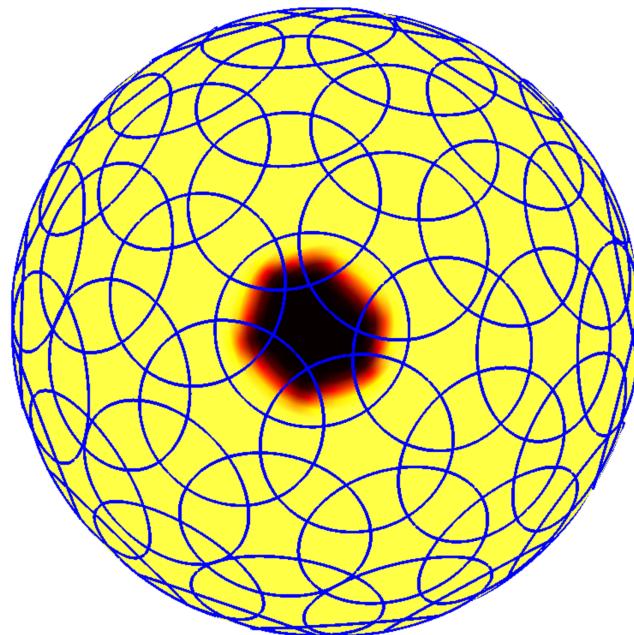
3. Define weight functions  $w_k : \mathbb{S}^2 \rightarrow \mathbb{R}$ ,  $k = 1 \dots, M$ , with the properties that each  $w_k$  is compactly supported over  $\Omega_k$  and the set of all  $w_k$  form a partition-of-unity over  $\Omega$ :

$$\sum_{k=1}^M w_k(\mathbf{x}) \equiv 1, \mathbf{x} \in \Omega$$

# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patches  $\Omega_k$

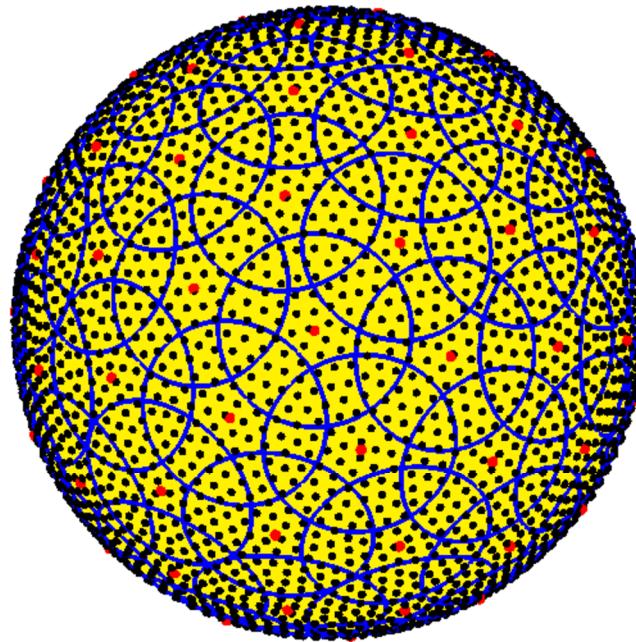
Key Steps:

3. Define weight functions  $w_k : \mathbb{S}^2 \rightarrow \mathbb{R}$ ,  $k = 1 \dots, M$ , with the properties that each  $w_k$  is compactly supported over  $\Omega_k$  and the set of all  $w_k$  form a partition-of-unity over  $\Omega$ :

$$\sum_{k=1}^M w_k(\mathbf{x}) \equiv 1, \mathbf{x} \in \Omega$$

# RBFs and partition-of-unity on the sphere

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patches  $\Omega_k$

## Key Steps:

Weight function details:

$$\psi_k(\mathbf{x}) = \psi\left(\frac{\|\mathbf{x} - \xi_k\|}{\rho_k}\right)$$

$\rho_k$  = radius of patch  $\Omega_k$

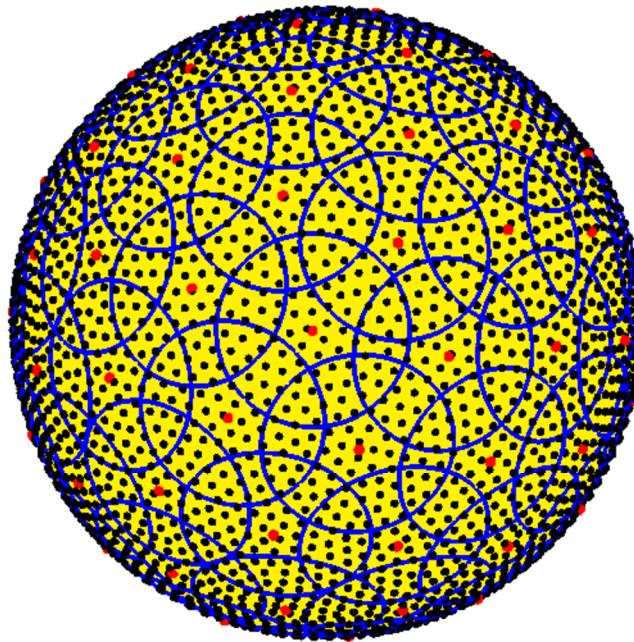
$\psi$  has compact support over  $[0, 1]$ .

$$w_k(\mathbf{x}) = \frac{\psi_k(\mathbf{x})}{\sum_{i=1}^M \psi_i(\mathbf{x})}$$

# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patches  $\Omega_k$

Key Steps:

4. Create a global interpolant for  $X$  as

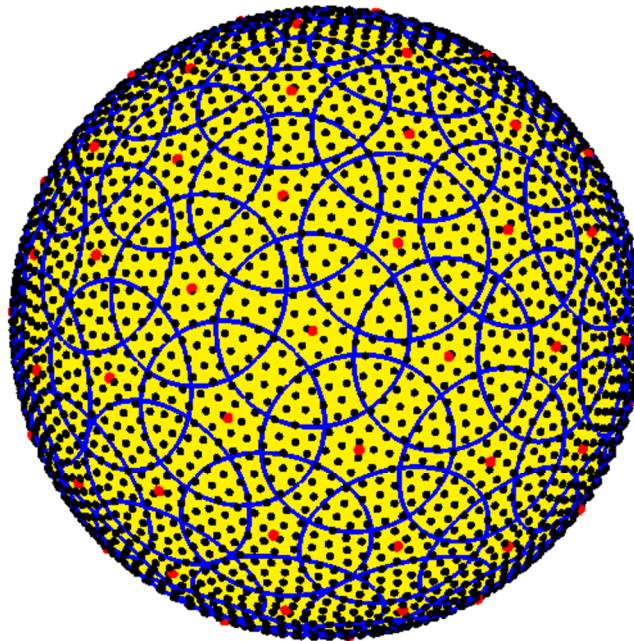
$$s(\mathbf{x}) = \sum_{k=1}^M w_k(\mathbf{x}) s_k(\mathbf{x})$$

R, Cavoretto and A. De Rossi. Spherical interpolation using the partition of unity method: An efficient and flexible algorithm. *Appl. Math. Lett.*, In press 2011.

# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patches  $\Omega_k$

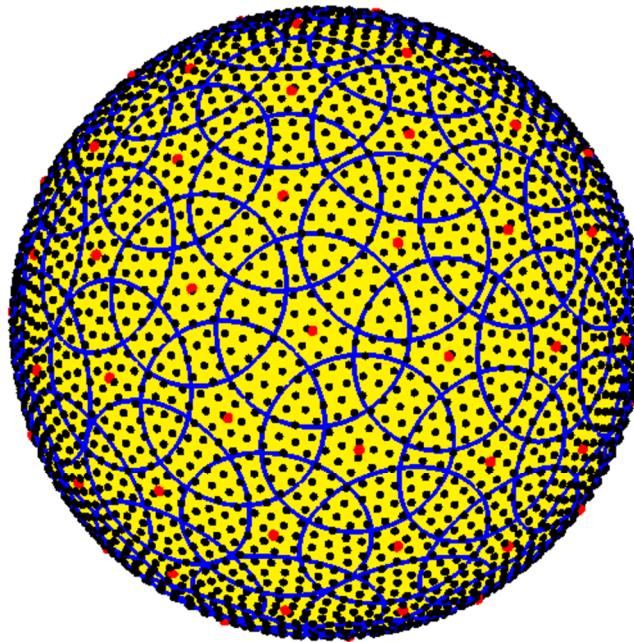
## Cost Comparison:

Collocation	Global RBF	RBF-PUM
Construction:	$O(N^3)$	$O(n^3M)$
Evaluation:	$O(N^2)$	$O(nN)$

# RBFs and partition-of-unity on the sphere

BIRS Workshop  
June 28, 2012

- Consider  $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ , where  $\mathbf{x}_j = (x_j, y_j, z_j)$ :



$M$  total patches  
 $n$  nodes per patch  
 $\xi_k$  = center of patches  $\Omega_k$

## Accuracy Comparison:

Theory for interpolation with RBF-PUM in  $\mathbb{R}^d$  says same convergence orders should be expected as global RBFs.

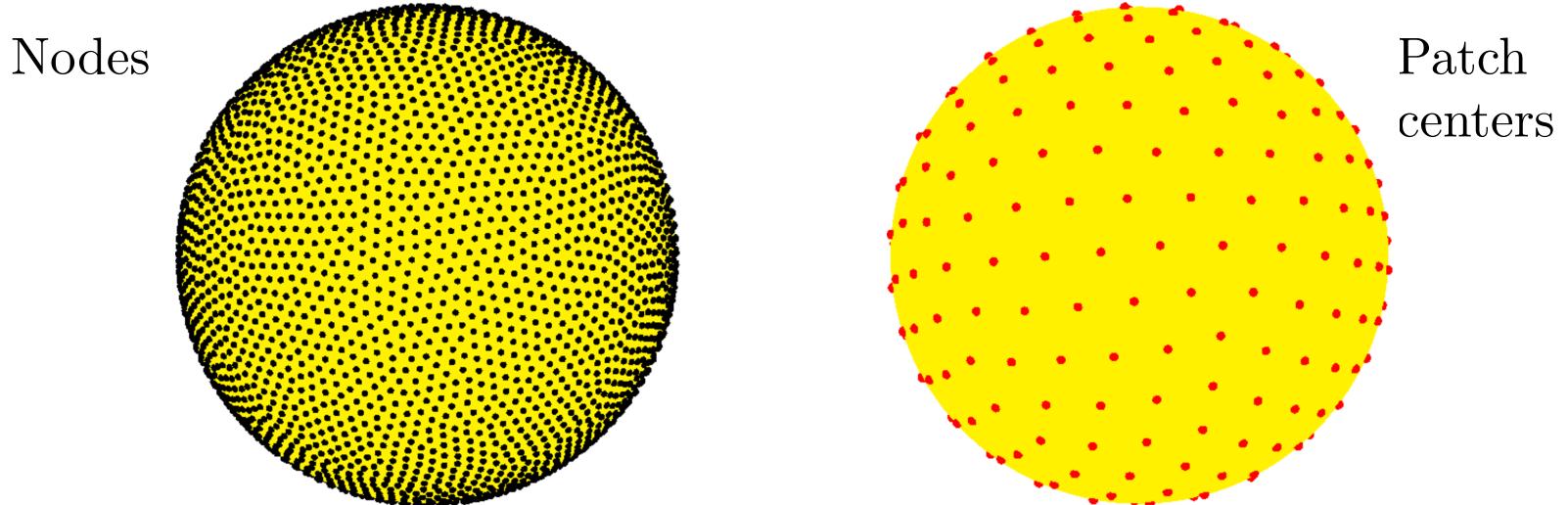
Spectral orders are therefore theoretically possible.

No theory for  $\mathbb{S}^2$ , but should expect similar results.

# Choosing the nodes and patches

Nodes: We use the *maximal determinant* (MD) node sets, which are quasi-uniformly distributed over the sphere. R.S. Womersley, I. Sloan (2001)

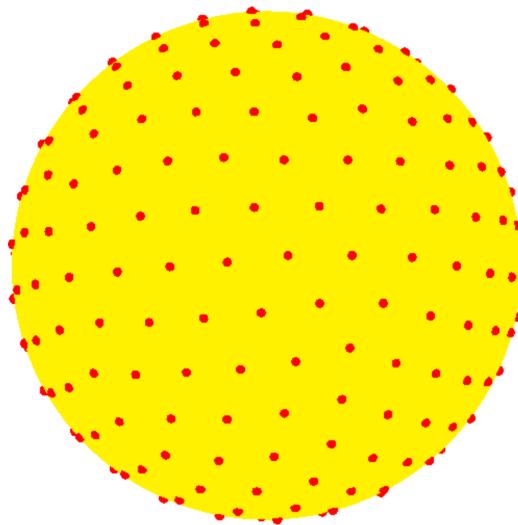
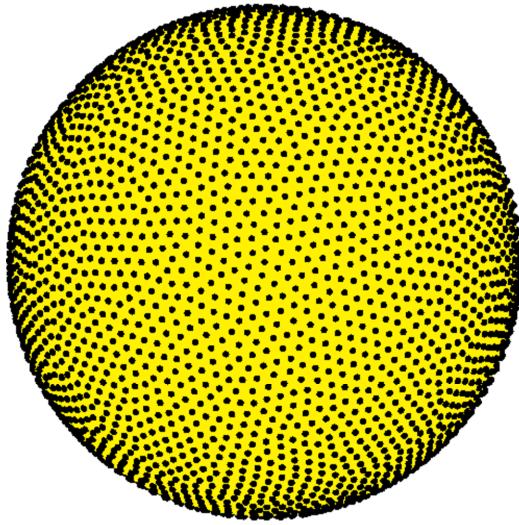
Patches: We use *minimum energy* (ME) points, which are also quasi-uniformly distributed over the sphere. D.P. Hardin, E.B. Saff (2004)



Parameters: Given  $N$  nodes, there are 2 parameters to choose for determining the total number of patches  $M$ :

- $n$  = approx. number of nodes in each patch;
- $q$  = measure of the amount the patches overlap.

# Choosing the nodes and patches



- Using the quasi-uniformity of the nodes and patches, we compute the **radii of the patches** using the approximation:

$$\rho \approx 2\sqrt{\frac{n}{N}}$$

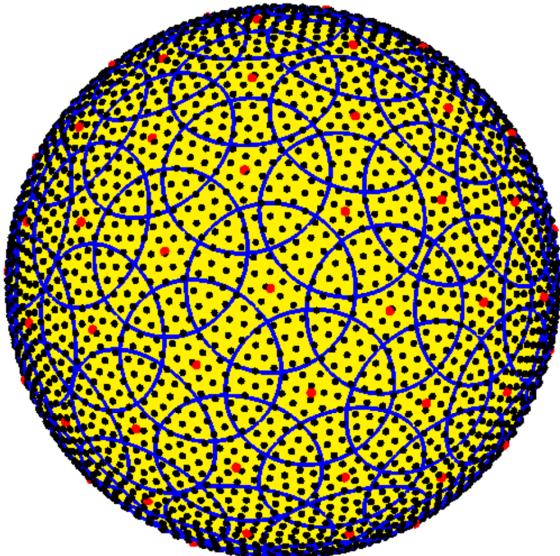
- The overlap parameter  $q$  determines the **average number of patches a node belongs to**, and satisfies the relationship:

$$\frac{4\pi}{M} \approx \frac{\pi\rho^2}{q} \quad \Rightarrow M = \left\lceil q \frac{N}{n} \right\rceil$$

# Choosing the nodes and patches

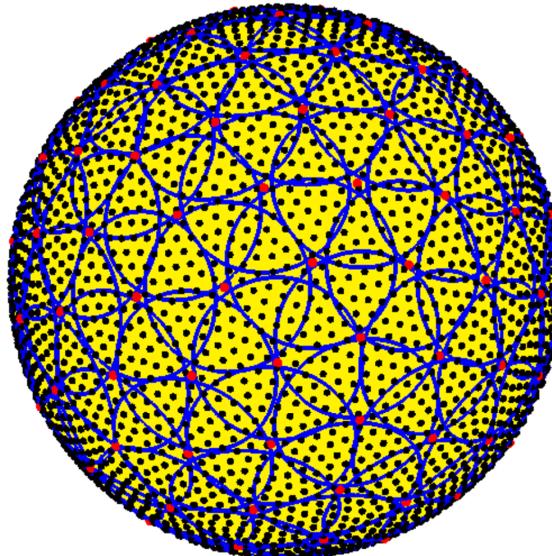
- Illustration of the patches for  $N=4096$ ,  $n=100$ , and different  $q$ :

$q=2$



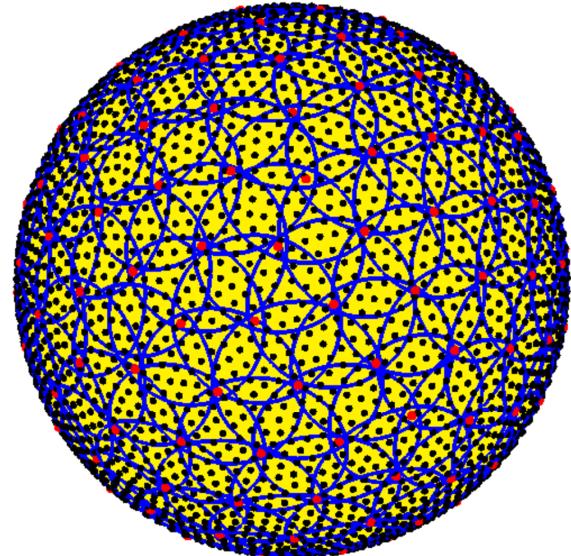
$M=82$

$q=3$



$M=123$

$q=4$



$M=184$

The computational cost for evaluating the interpolant grows at most linearly with  $q$  and  $n$ .

# Collocation of surface gradient operator

---

- Surface gradient operator:

$$\mathbf{P}\nabla = (\mathbf{I} - \mathbf{x}\mathbf{x}^T)\nabla = \begin{bmatrix} (1-x^2) & -xy & -xz \\ -xy & (1-y^2) & -yz \\ -xz & -yz & (1-z^2) \end{bmatrix} \begin{bmatrix} \partial_x \\ \partial_y \\ \partial_z \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x \cdot \nabla \\ \mathbf{p}_y \cdot \nabla \\ \mathbf{p}_z \cdot \nabla \end{bmatrix}$$

- Let  $s(\mathbf{x})$  be the RBF-PUM interpolant of  $f$  at the nodes, then we can approximate the  $x$ -component of the surface gradient of  $f$  by:

$$\begin{aligned} [\mathbf{p}_x \cdot \nabla f(\mathbf{x})] \Big|_{\mathbf{x}=\mathbf{x}_j} &\approx [\mathbf{p}_x \cdot \nabla s(\mathbf{x})] \Big|_{\mathbf{x}=\mathbf{x}_j}, \quad j = 1, \dots, N \\ &= \sum_{k=1}^M [\mathbf{p}_x \cdot \nabla (w_k(\mathbf{x})s_k(\mathbf{x}))] \Big|_{\mathbf{x}=\mathbf{x}_j}, \quad j = 1, \dots, N \\ &= D_N^x \underline{f} \end{aligned}$$

- We can similarly compute the differentiation matrices  $D_N^y$  and  $D_N^z$  for  $(\mathbf{p}_y \cdot \nabla)$  and  $(\mathbf{p}_z \cdot \nabla)$ .

# RBF-PUM for transport equation

- Continuous transport equation:

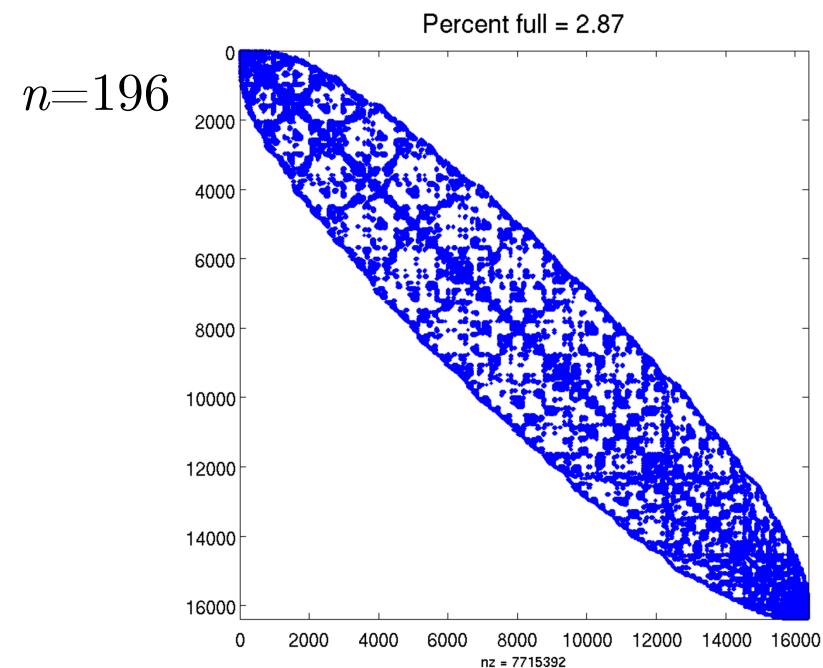
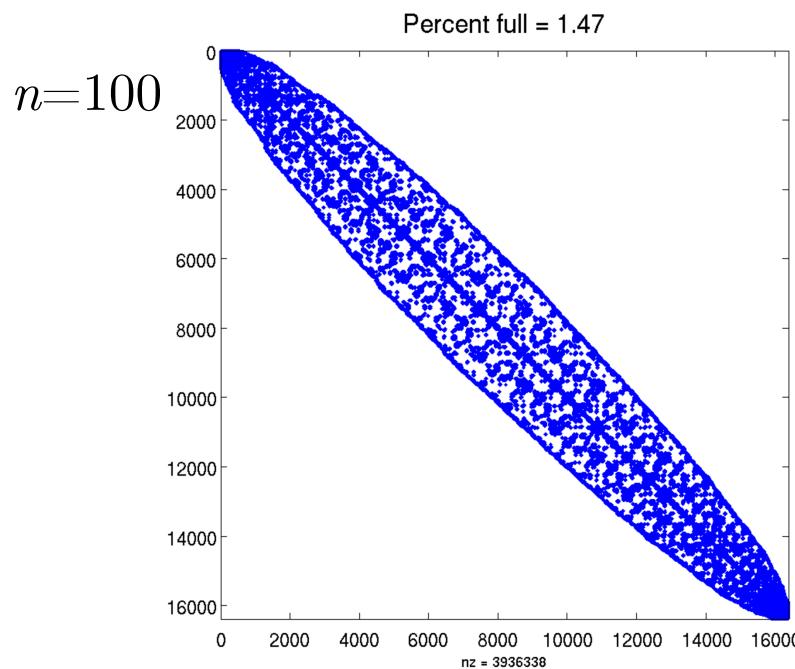
$$h_t + \mathbf{u} \cdot (\mathbf{P} \nabla h) = 0 \quad \mathbf{u} = (u, v, w) \in T_{\mathbf{x}} \mathbb{S}^2$$

- Semi-discrete formulation:

$$\underline{h}_t = -(\text{diag}(\underline{u}) D_N^x + \text{diag}(\underline{v}) D_N^y + \text{diag}(\underline{w}) D_N^z) \underline{h} = -D_N \underline{h}$$

- Unlike global RBF-type methods the matrix  $D_N$  above is *sparse!*

Illustrations of the sparsity of  $D_N$  for  $N=16384$  and  $q=4$

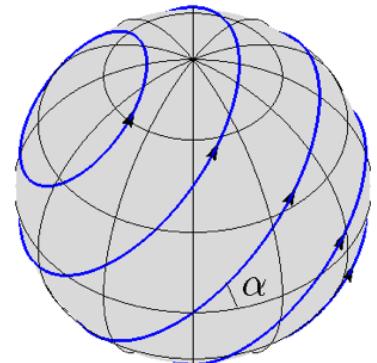


# What about stability?

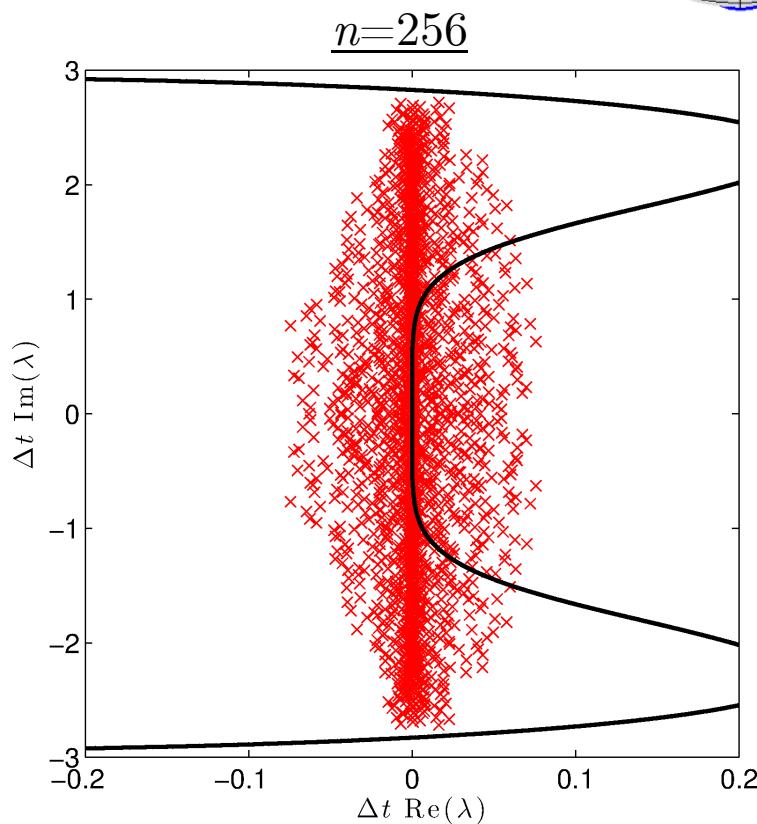
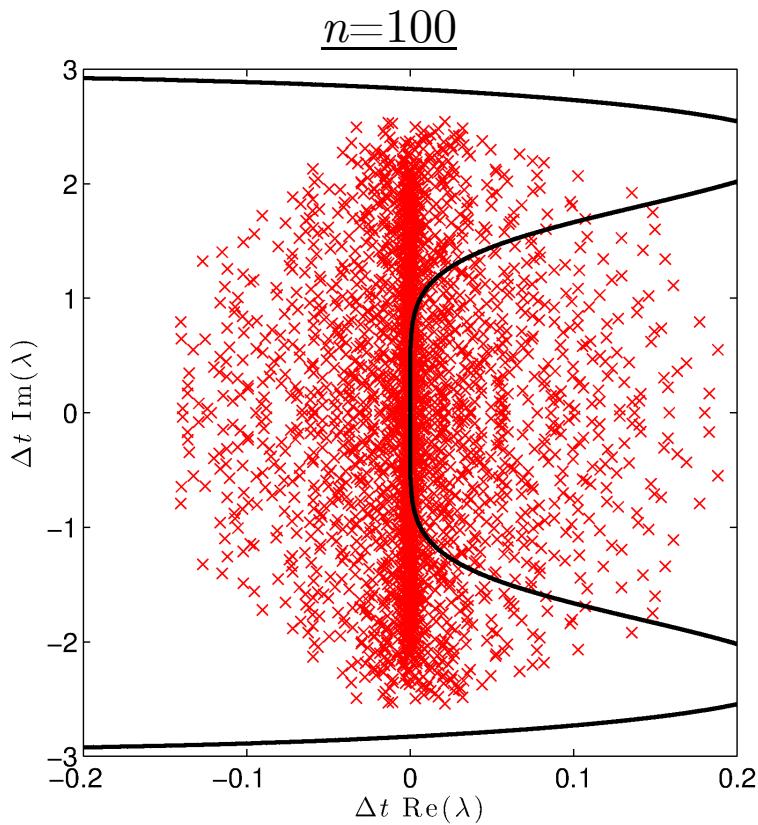
$$\underline{h}_t = -(\text{diag}(\underline{u})D_N^x + \text{diag}(\underline{v})D_N^y + \text{diag}(\underline{w})D_N^z)\underline{h} = -D_N\underline{h}$$

- Consider the case of where

$$\mathbf{u} = \mathbf{x} \times \nabla(\cos(\alpha)z + \sin(\alpha)y)$$



Illustrations of eigenvalues of  $-D_N$  for  $N=4096$  and  $q=4$

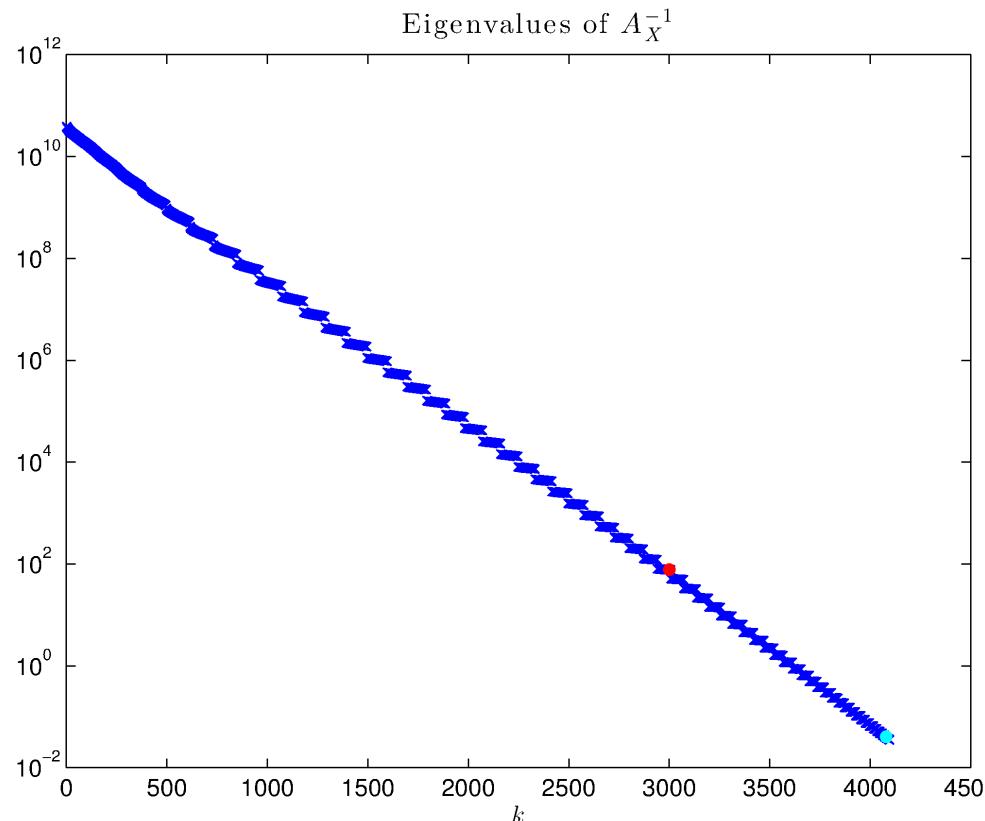


# Stabilization via “global” hyperviscosity

BIRS Workshop  
June 28, 2012

- We need a filter to damp all these spurious eigenmodes while leaving the physically relevant ones essentially intact.
- Idea for stabilizing global RBF methods (Fornberg & Lehto, JCP (2011))

$$\frac{d}{dt} \underline{h} = -\mu(A_X)^{-1} \underline{h}, \quad \mu > 0$$

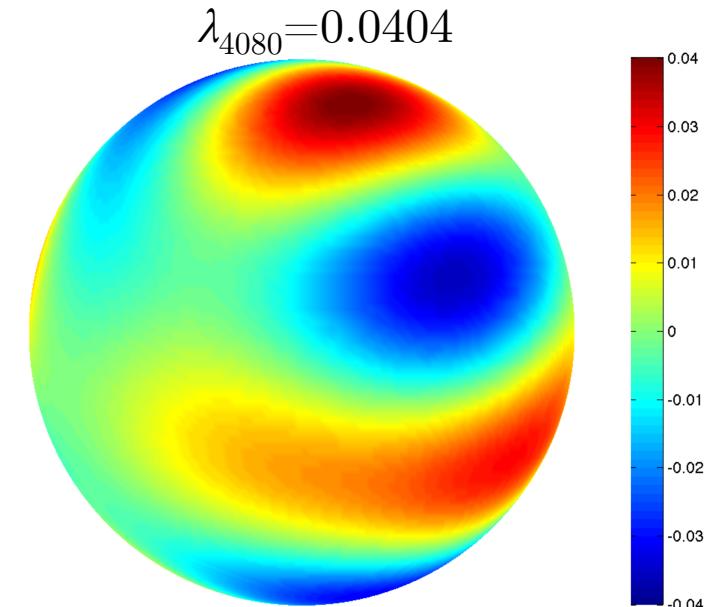
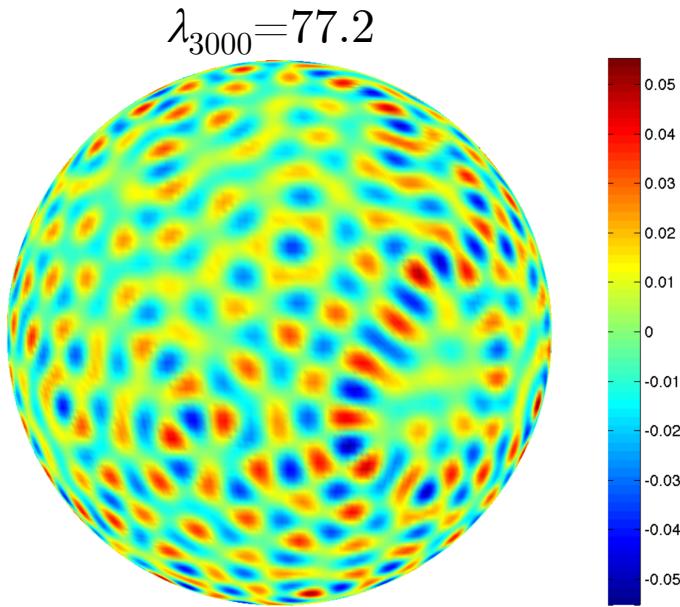


# Stabilization via “global” hyperviscosity

- We need a filter to damp all these spurious eigenmodes while leaving the physically relevant ones essentially intact.
- Idea for stabilizing global RBF methods (Fornberg & Lehto, JCP (2011))

$$\frac{d}{dt} \underline{h} = -\mu(A_X)^{-1} \underline{h}, \quad \mu > 0$$

Eigenvectors:



$(A_X^{-1})$  damps high-frequency modes extremely fast and low ones slowly.

# Stabilization via “global” hyperviscosity

---

- We need a filter to damp all these spurious eigenmodes while leaving the physically relevant ones essentially intact.
- Idea for stabilizing global RBF methods (Fornberg & Lehto, JCP (2011))

$$\frac{d}{dt} \underline{h} = -\mu(A_X)^{-1} \underline{h}, \quad \mu > 0$$

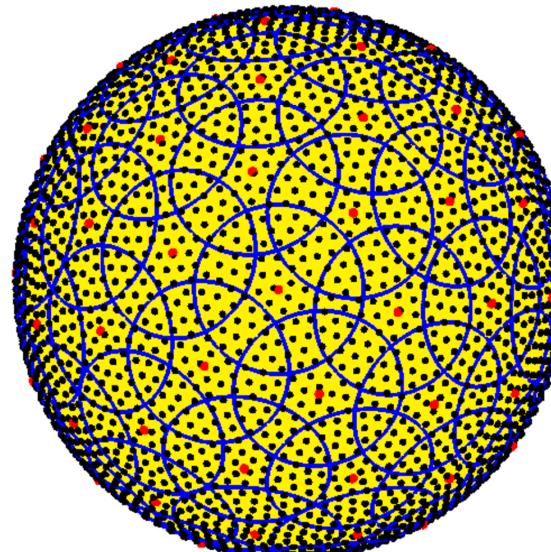
- Modified PDE to solve:

$$\frac{d}{dt} \underline{h} = - (D_X + \mu(A_X)^{-1}) \underline{h}, \quad \mu > 0$$

# Stabilization via “local” hyperviscosity

BIRS Workshop  
June 28, 2012

- We need a filter to damp all these spurious eigenmodes while leaving the physically relevant ones essentially intact.
- Idea for stabilizing RBF-PUM:



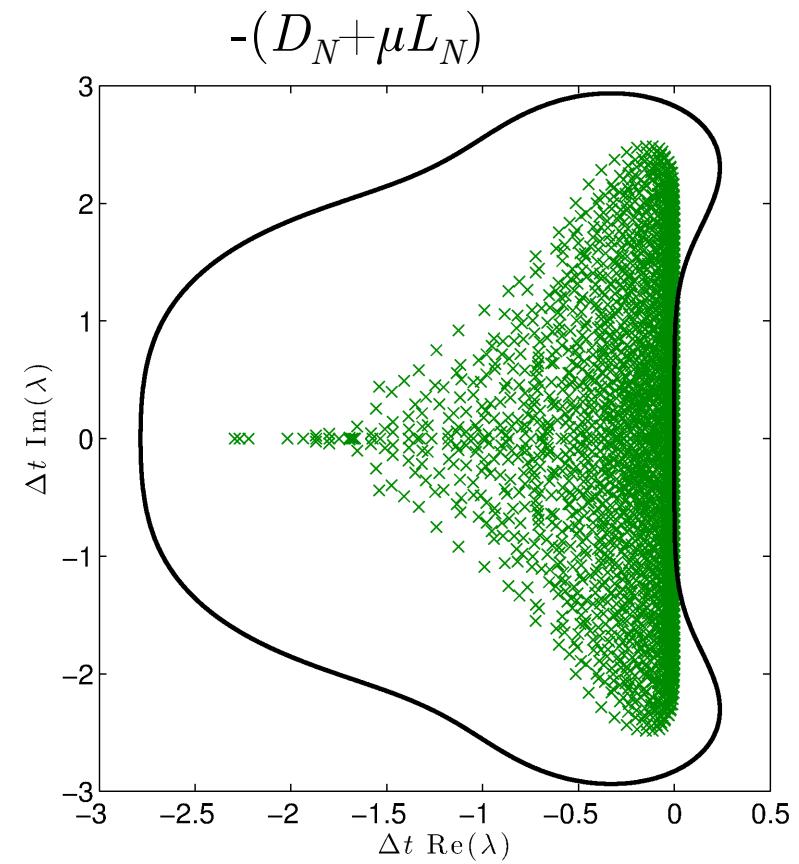
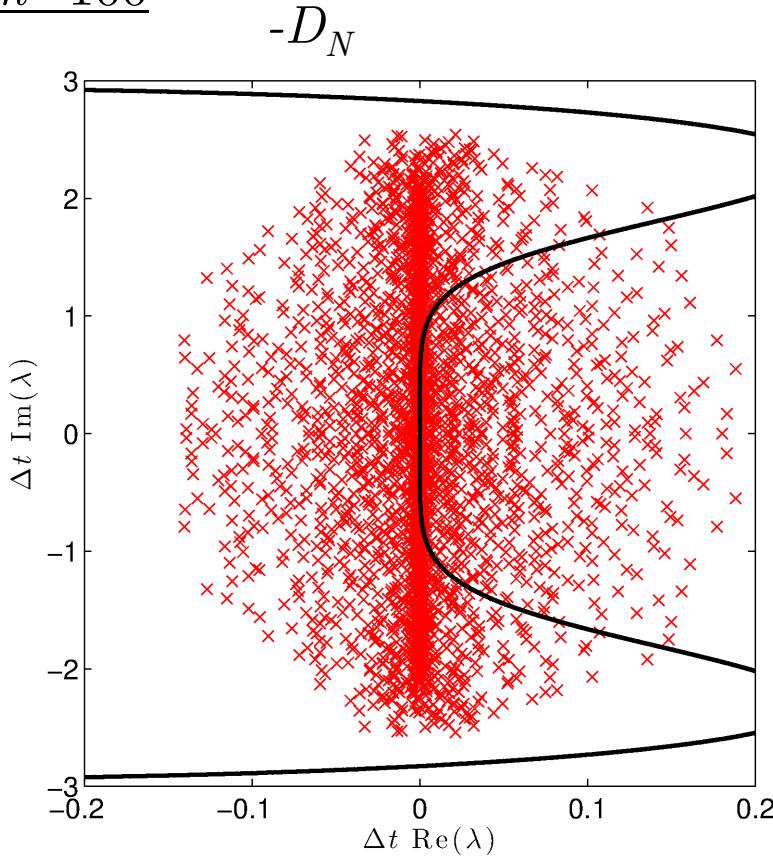
1. On each patch  $\Omega_k$ , apply  $A_k^{-1}$  to  $\underline{h}_k$  (solution at time  $t$  on  $\Omega_k$ ).
  2. Use the PUM weight functions  $w_k$  to sum up all the  $A_k^{-1}\underline{h}_k(t)$ .
- Modified PDE to solve:

$$\frac{d}{dt} \underline{h} = - (D_N + \mu L_N) \underline{h}, \quad \mu > 0$$

# Stabilization via “local” hyperviscosity

Illustrations of the eigenvalues for  $N=4096$  and  $q=4$ :

$n=100$



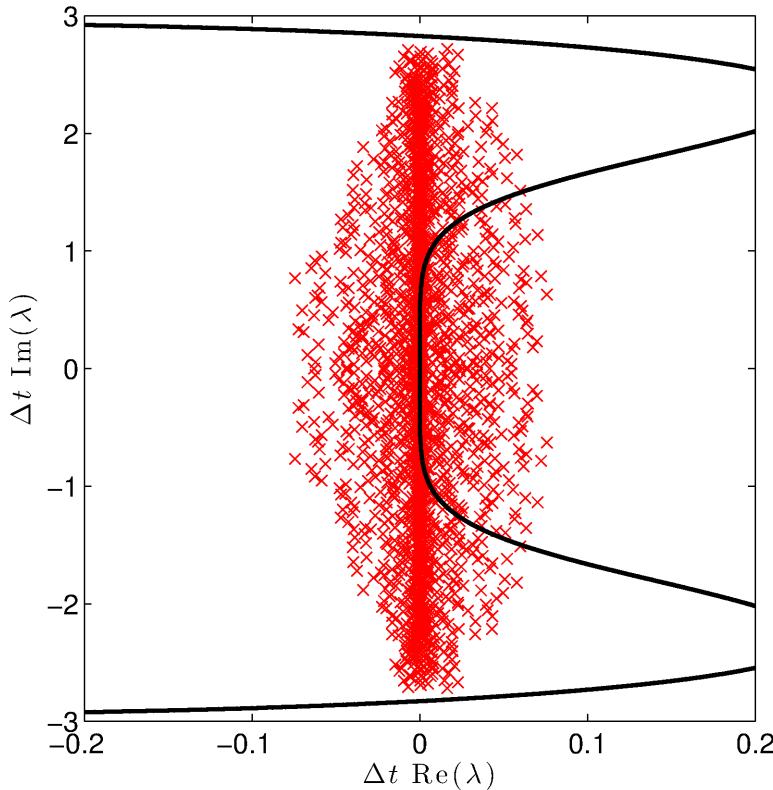
- Eigenvalues have been scaled by  $\Delta t = 2\pi/125$ ,  $\mu = 8 \times 10^{-9}$
- Black solid line is the stability domain of RK4

# Stabilization via “local” hyperviscosity

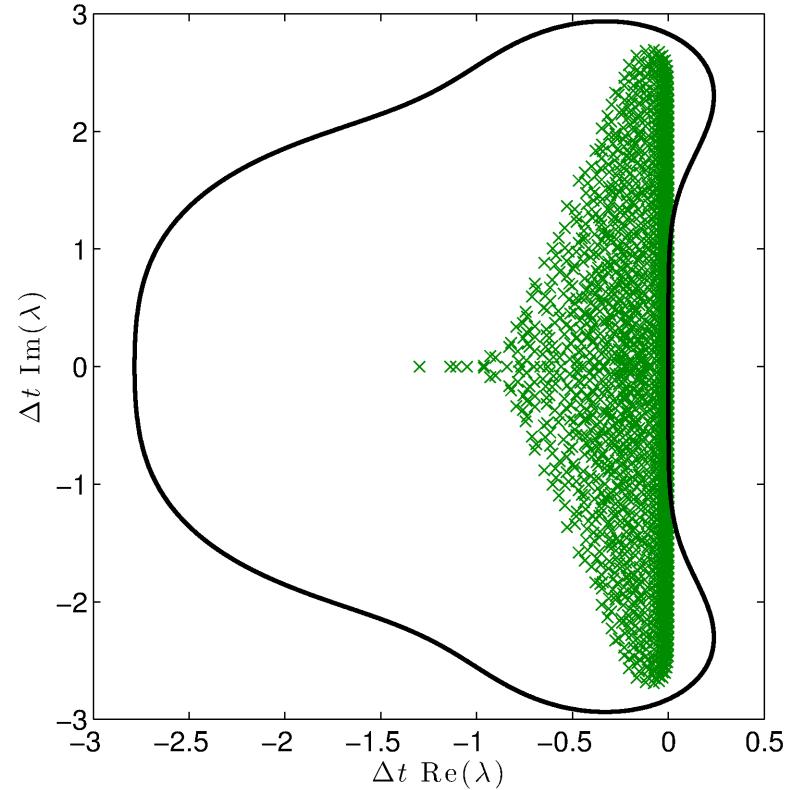
Illustrations of the eigenvalues for  $N=4096$  and  $q=4$ :

$n=256$

$-D_N$



$-(D_N + \mu L_N)$



- Eigenvalues have been scaled by  $\Delta t = 2\pi/125$ ,  $\mu = 8 \times 10^{-9}$
- Black solid line is the stability domain of RK4

# Numerical results: solid body rotation

BIRS Workshop  
June 28, 2012

Details for all numerical results:

- We use the Gaussian RBF
- Time-step  $\Delta t$  is not optimized
- Overlap is set to  $q=4$
- We set  $\epsilon = a_n \sqrt{N} + b_n$

- Solid body rotation of a non-smooth cosine bell  
(Williamson et. al. JCP (1992))

Stream Function for flow

$$\psi(\mathbf{x}) = \cos(\alpha)z + \sin(\alpha)y \quad \alpha = \pi/2 \text{ (flow over the poles)}$$

Initial condition

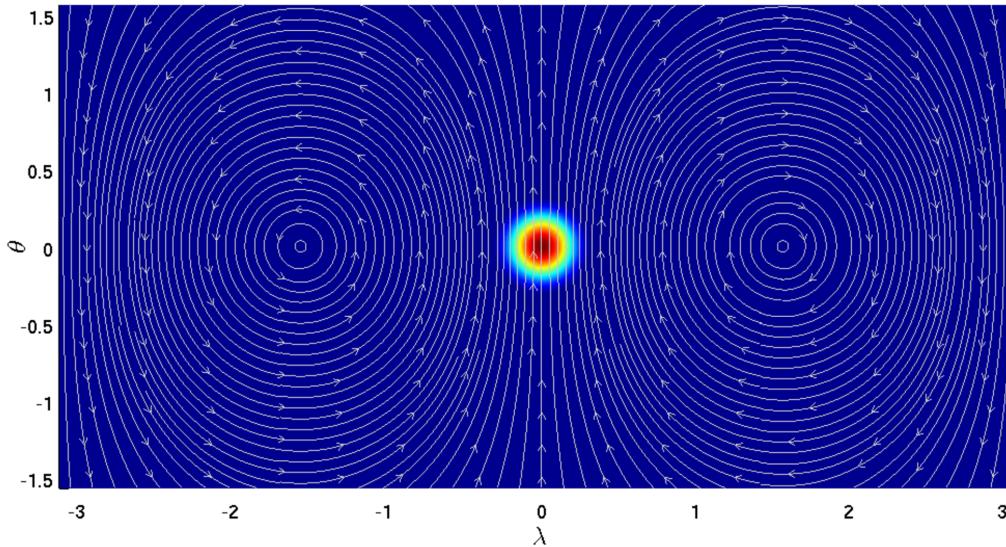
$$h(\mathbf{x}) = \begin{cases} \frac{1}{2} (1 + \cos(3\pi r(\mathbf{x}))) & r(\mathbf{x}) < 1/3 \\ 0 & r(\mathbf{x}) \geq 1/3 \end{cases}$$

$$r(\mathbf{x}) = \arccos(x)$$

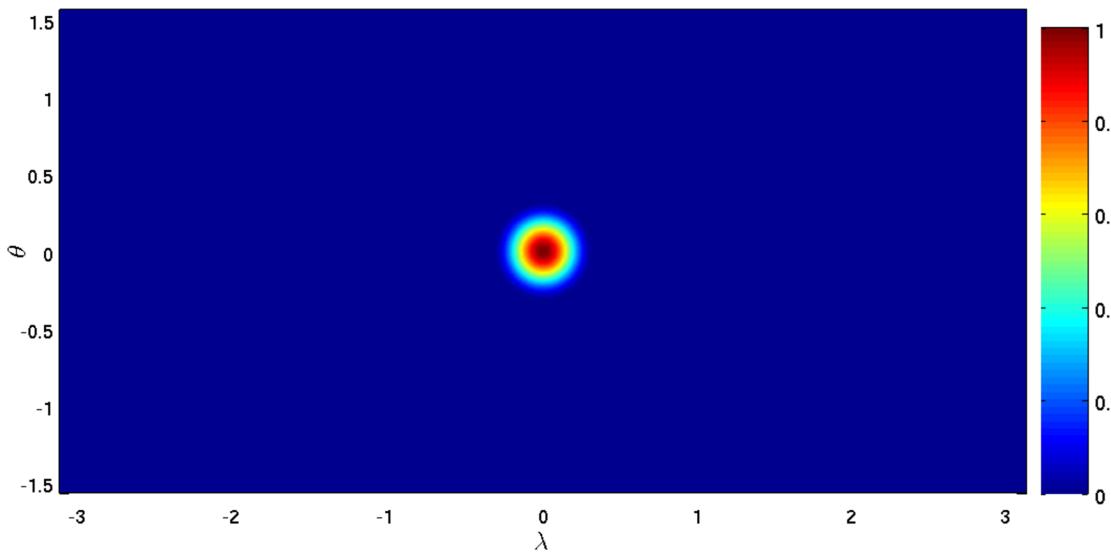
# Numerical results: solid body rotation

Plots of the solution for  $N=12544$ ,  $n=144$ ,  $\Delta t=2\pi/1600$ ,  $\mu=8\times 10^{-9}$ :

Initial condition,  $t=0$ , with streamlines



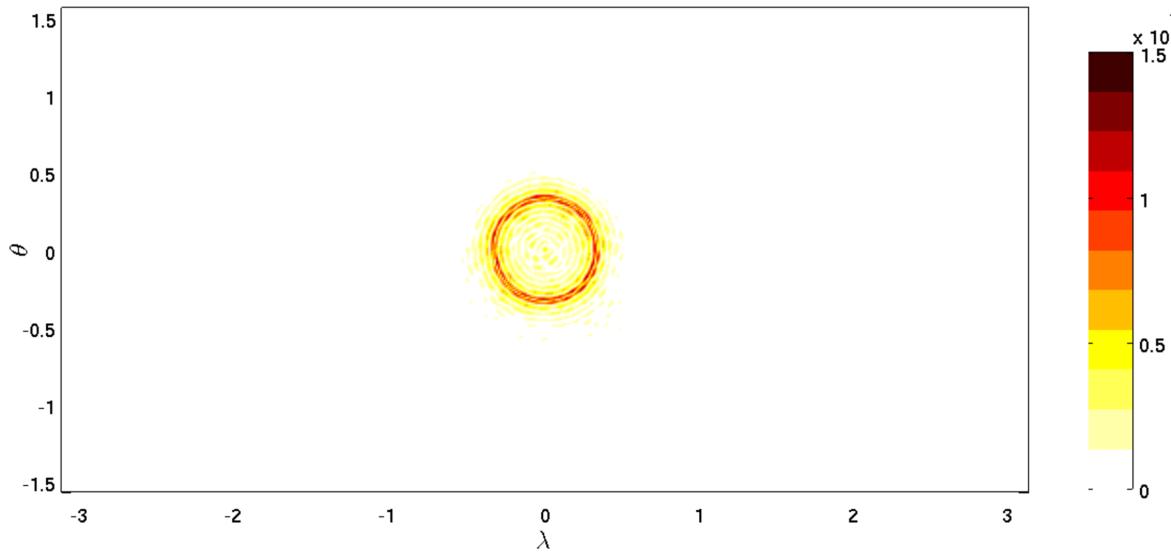
Solution after one revolution,  $t=2\pi$



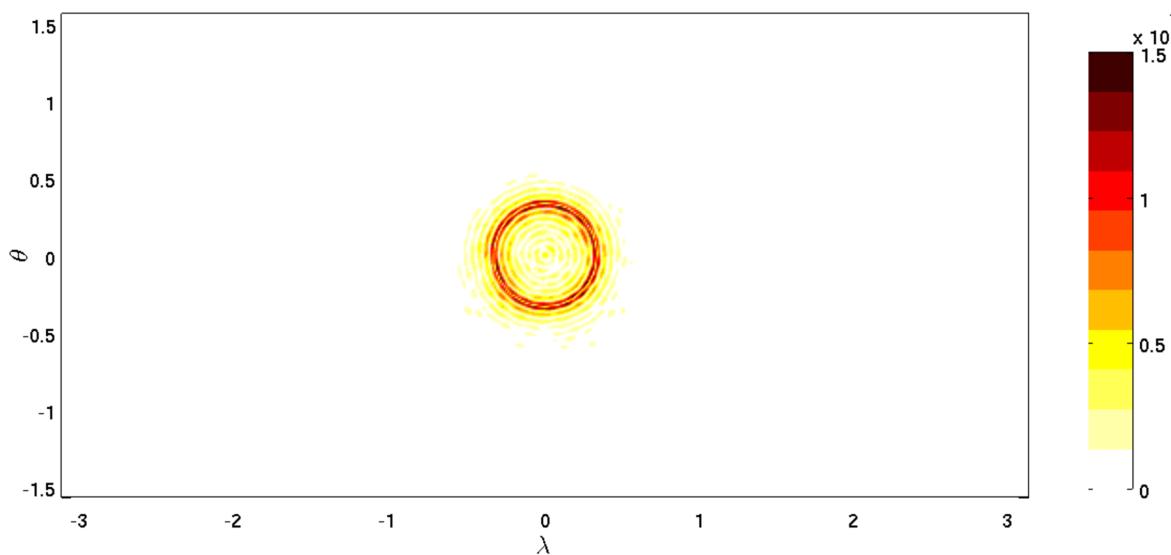
# Numerical results: solid body rotation

Plots of the error for  $N=12544$ ,  $n=144$ ,  $\Delta t=2\pi/1600$ ,  $\mu=8\times 10^{-9}$ :

Magnitude of the error after one revolution

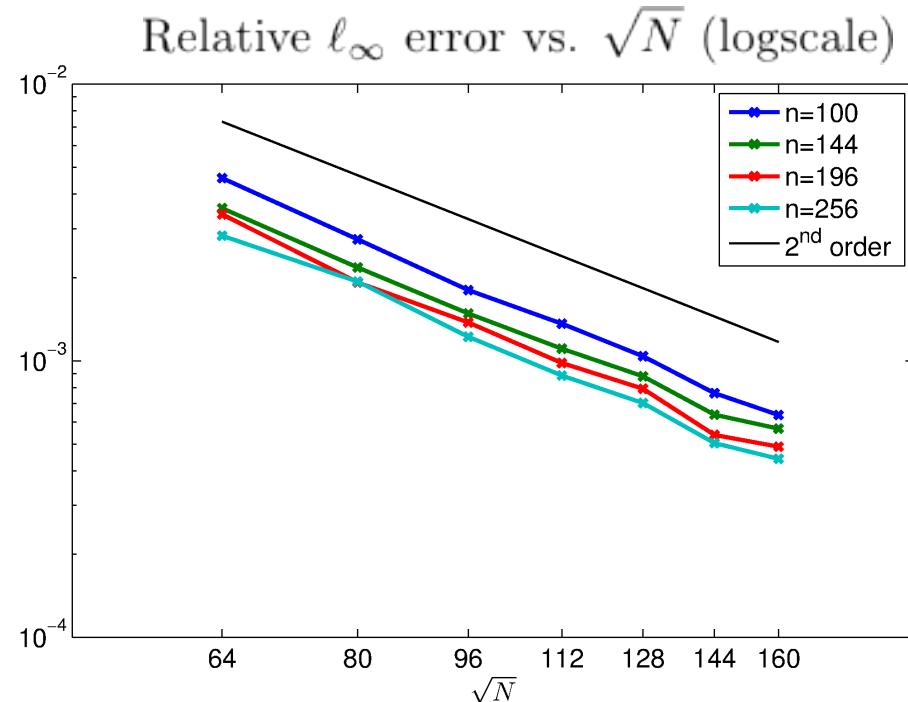
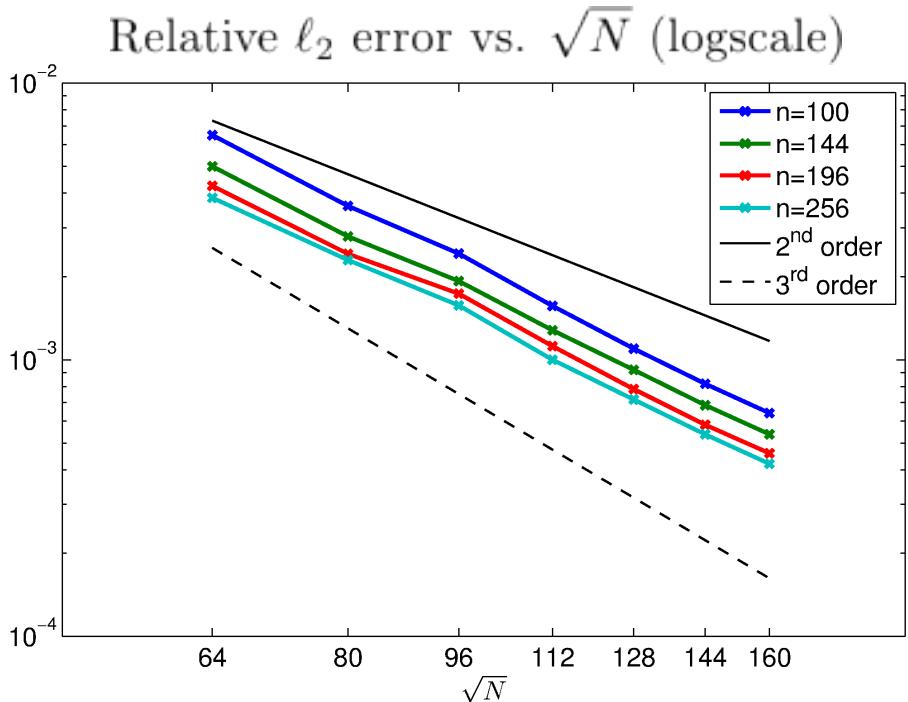


Magnitude of the error after ten revolutions



# Numerical results: solid body rotation

Convergence plots for increasing  $N$  and  $n$ ,  $\Delta t=2\pi/1600$ ,  $\mu=8\times 10^{-9}$ :

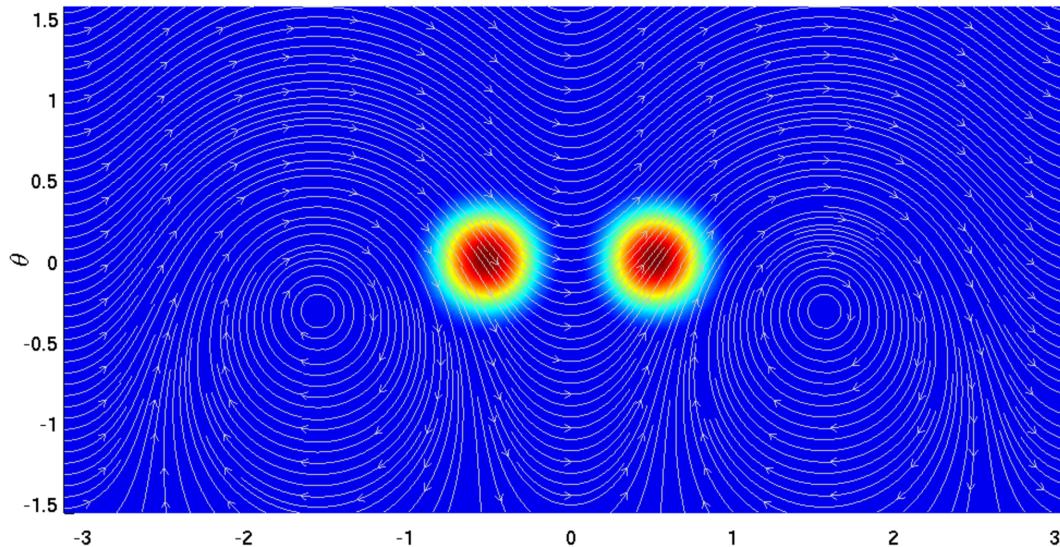


Convergence rates are as expected given smoothness of the initial condition.

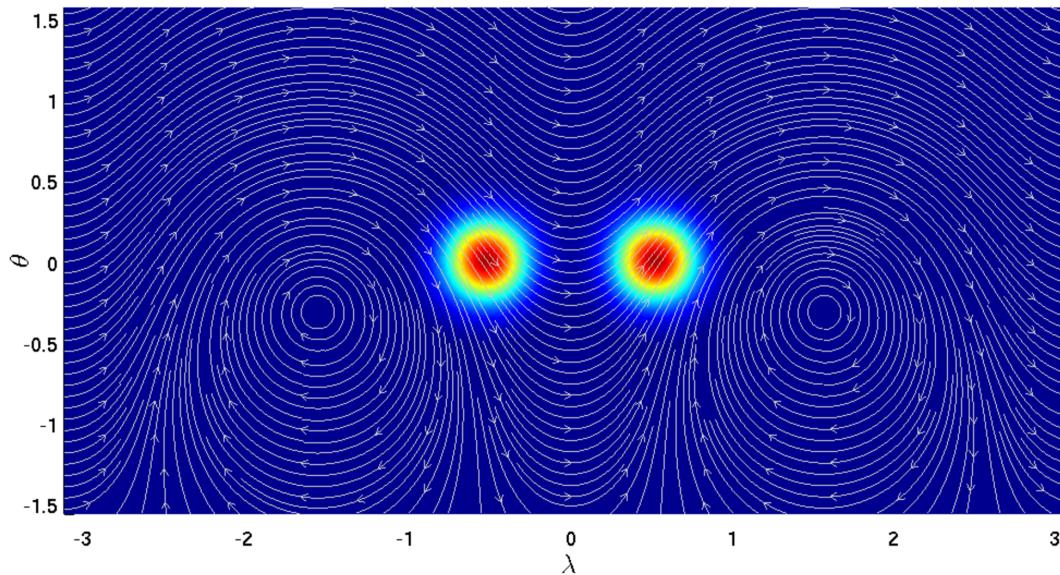
# Numerical results: deformational flow

- Deformational/rotational flow (R.D. Nair and P.H. Lauritzen, JCP (2010))

Non-smooth  
initial condition:



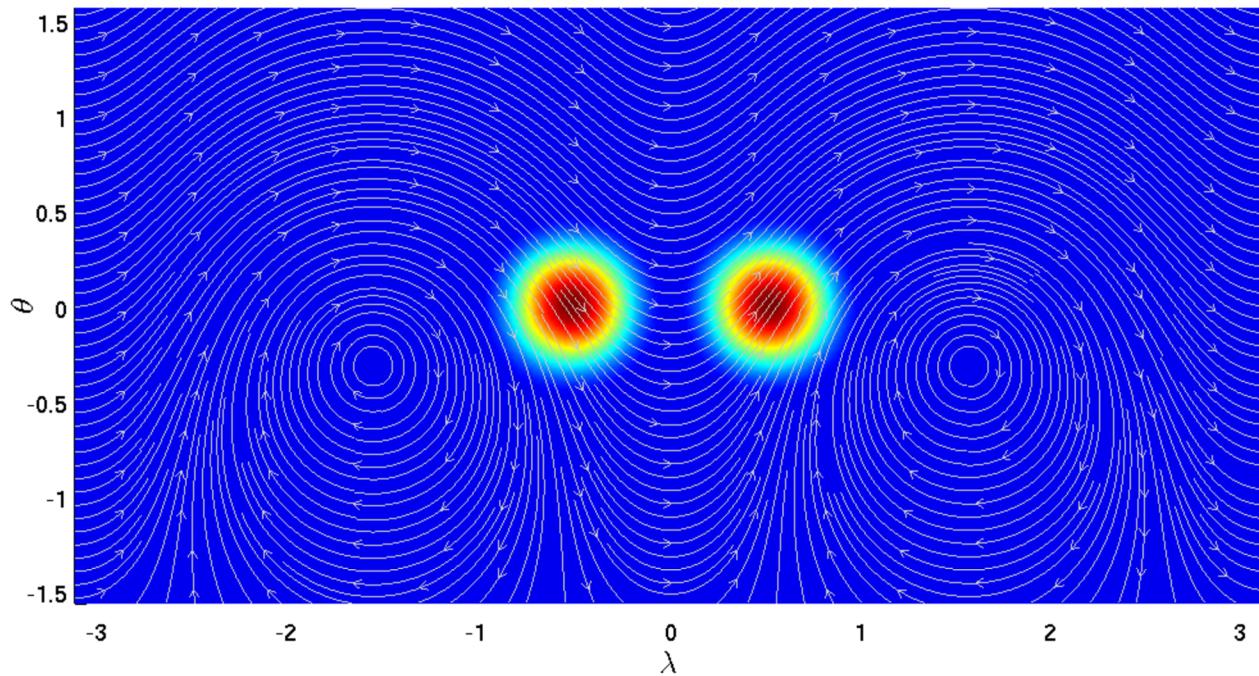
Smooth  
initial condition:



# Numerical results: deformational flow

BIRS Workshop  
June 28, 2012

Simulation for non-smooth IC,  $N=20736$ ,  $n=100$ ,  $\Delta t=5/2400$ ,  $\mu=10^{-8}$

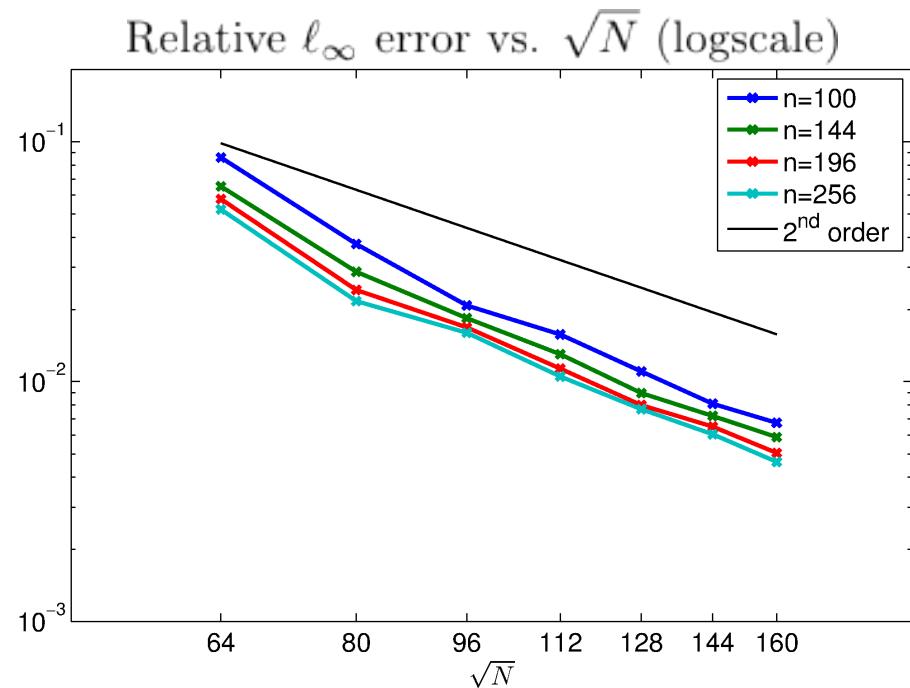
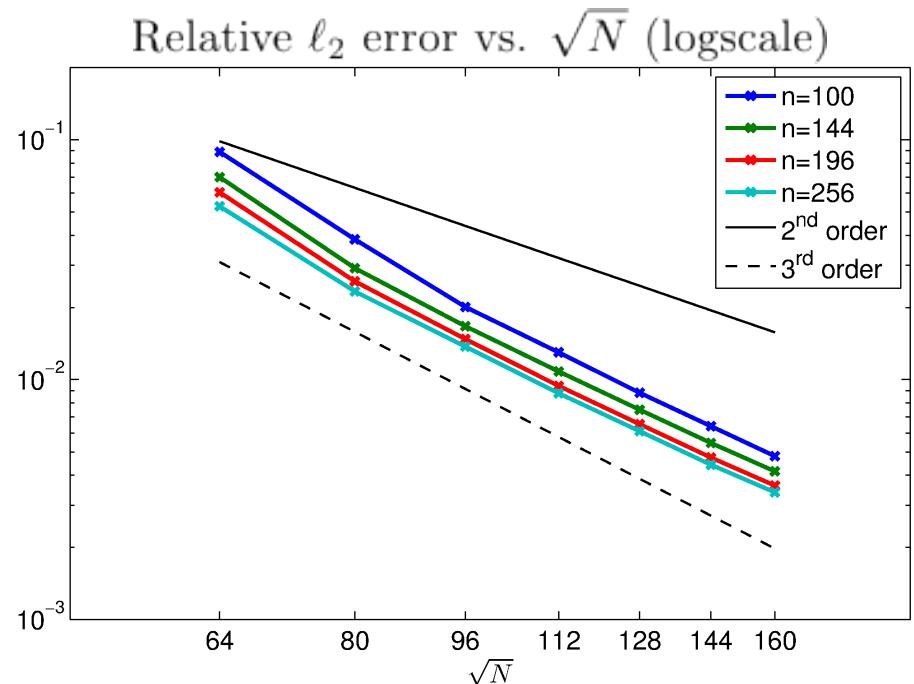


# Numerical results: deformational flow

BIRS Workshop  
June 28, 2012

Convergence plots for increasing  $N$  and  $n$ ,  $\Delta t=5/2400$ ,  $\mu=10^{-8}$

- Non-smooth initial condition:



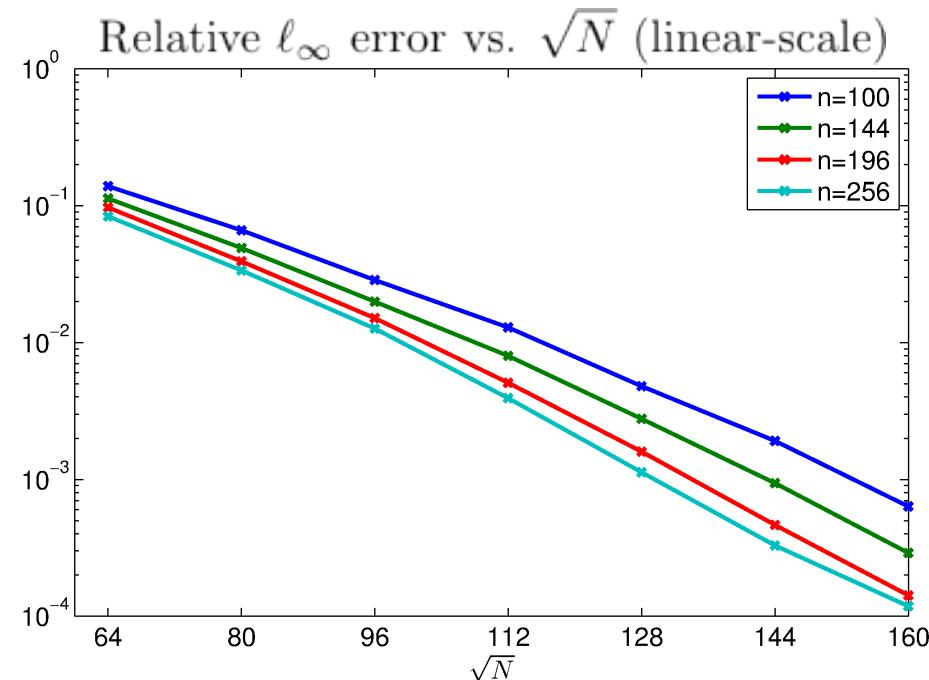
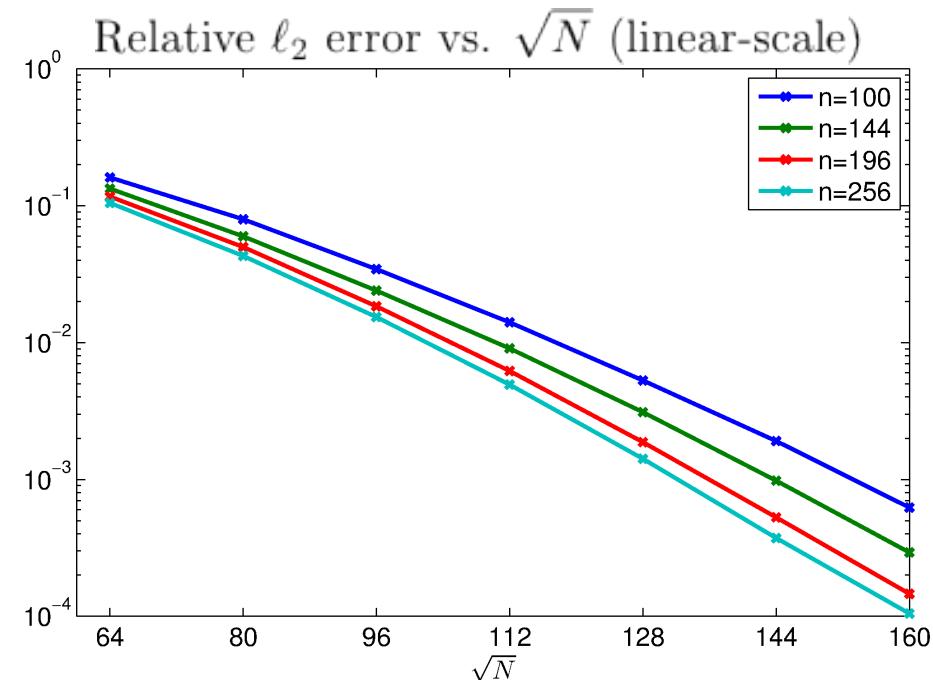
Convergence rates are as expected given smoothness of the initial condition.

# Numerical results: deformational flow

BIRS Workshop  
June 28, 2012

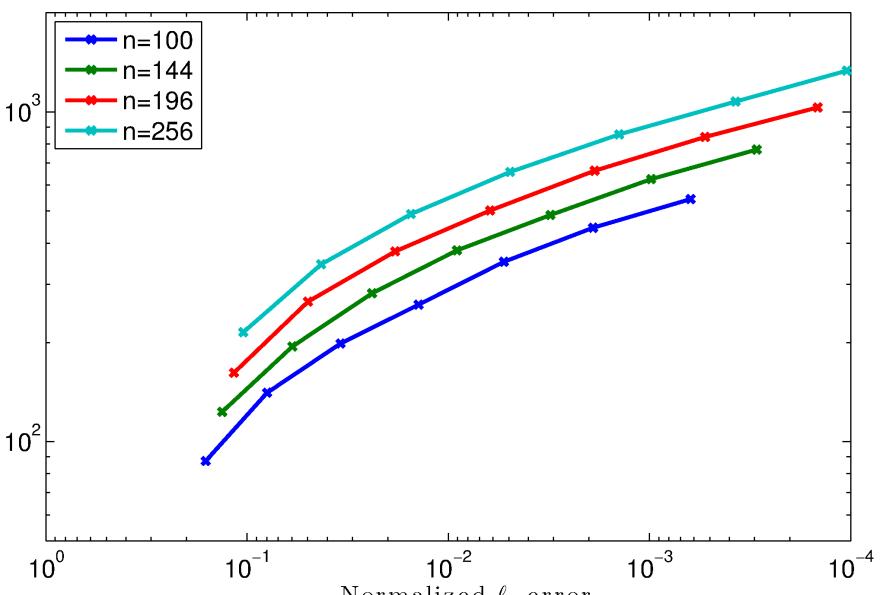
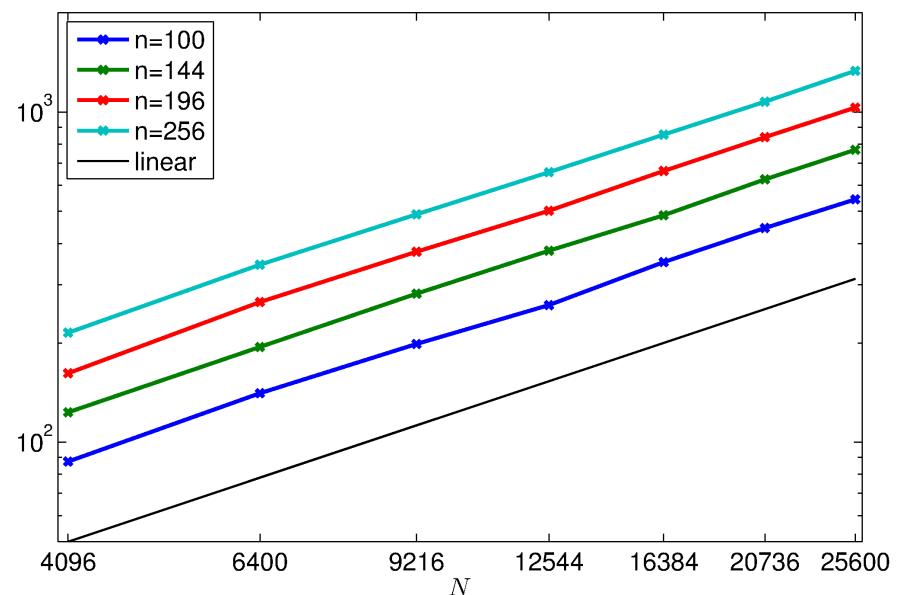
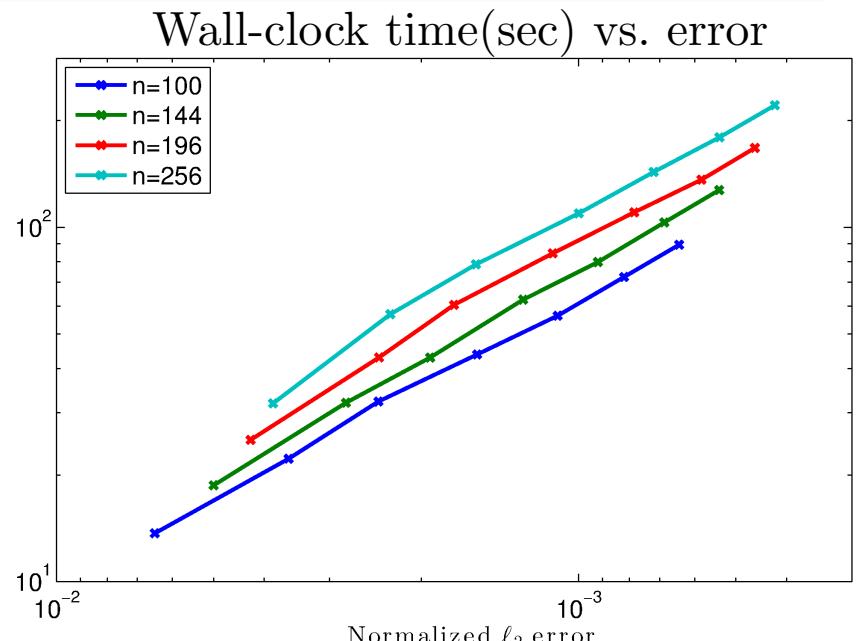
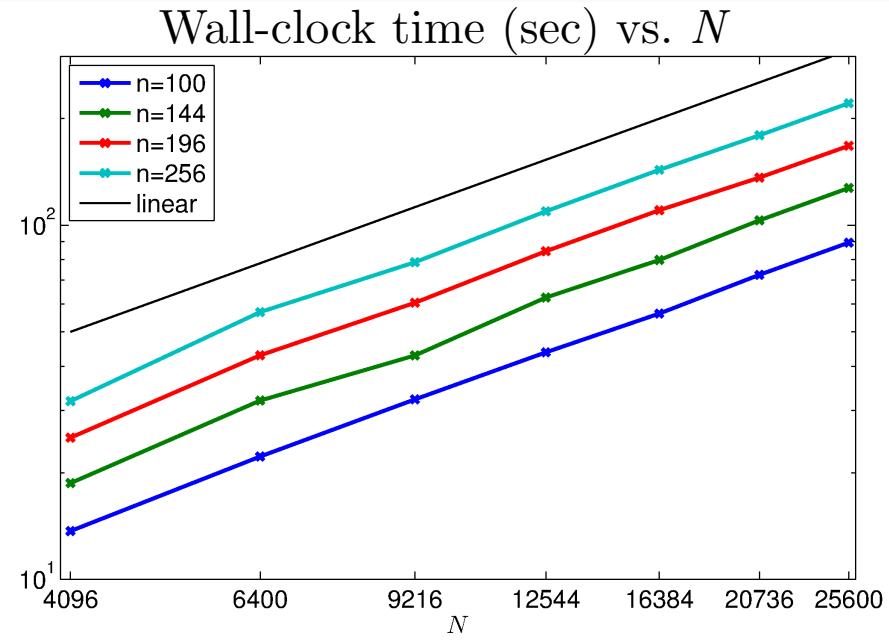
Convergence plots for increasing  $N$  and  $n$ ,  $\Delta t=5/2400$ ,  $\mu=10^{-8}$

- Smooth initial condition:



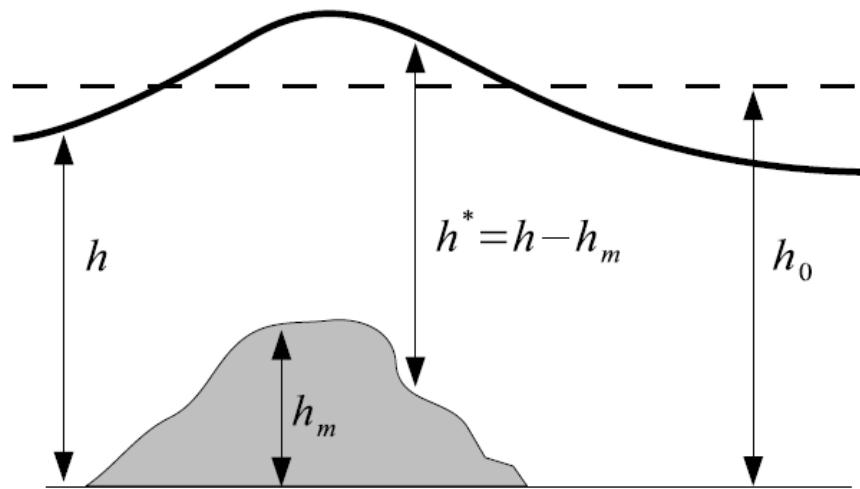
Observed convergence rate is spectral for the smooth initial condition.

# Computational performance: both tests



# Full shallow water wave equations

- Model for the nonlinear dynamics of a shallow, hydrostatic, homogeneous, and inviscid fluid layer.



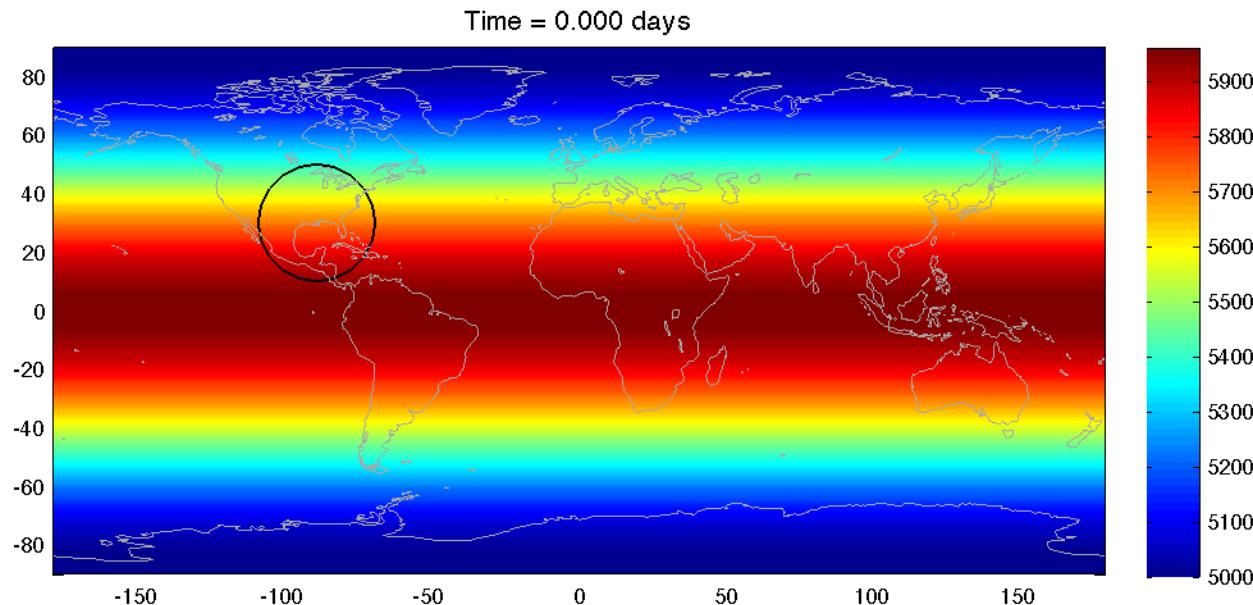
- Governing equations (in Cartesian form):

$$\frac{\partial \mathbf{u}_c}{\partial t} + P \begin{bmatrix} (\mathbf{u}_c \cdot P \nabla_c) u_c + f(\mathbf{x} \times \mathbf{u}_c) \cdot \hat{\mathbf{i}} + g(P \hat{\mathbf{i}} \cdot \nabla_c) h \\ (\mathbf{u}_c \cdot P \nabla_c) v_c + f(\mathbf{x} \times \mathbf{u}_c) \cdot \hat{\mathbf{j}} + g(P \hat{\mathbf{j}} \cdot \nabla_c) h \\ (\mathbf{u}_c \cdot P \nabla_c) w_c + f(\mathbf{x} \times \mathbf{u}_c) \cdot \hat{\mathbf{k}} + g(P \hat{\mathbf{k}} \cdot \nabla_c) h \end{bmatrix} = 0$$

$$\frac{\partial h^*}{\partial t} + (P \nabla_c) \cdot (h^* \mathbf{u}_c) = 0$$

# RBF-PUM for shallow water wave eqs.

- Numerical simulation: Flow over an isolated mountain  
(Test Case 5 from Williamson et. al., JCP (1992))



# Concluding remarks

---

- Computational cost of the RBF-PUM scales linearly with  $N$ .
- Spectral-type convergence appears to be possible for smooth solutions.
- Hyperviscosity procedure seems to be effective for stabilizing the method.

## Plans for the future:

- Explore ways to reduce overlap ( $q$ ) and still maintain stability.
- Parallelization of the method.
- Implementation of stable *flat* RBF algorithms for avoiding saturation errors.
- Extend the method for the non-linear shallow water wave equations.