Radial Basis Function Generated Finite Differences (RBF-FD): New Computational Opportunities for Solving PDEs

Bengt Fornberg University of Colorado, Boulder,
Department of Applied Mathematics

Natasha Flyer NCAR, IMAGe
Institute for Mathematics Applied to the Geosciences

in collaboration with

Brad Martin, Greg Barnett, Victor Bayona and Jonah Reeger

One main evolution path in numerical methods for PDEs:

Finite Differences (FD) (1910)
First general numerical approach for solving PDEs
FD weights obtained by using local polynomial approximations

➔

Pseudospectral (PS) (1970)
Can be seen either as the limit of increasing order FD methods, or as approximations by basis functions, such as Fourier or Chebyshev; often very accurate, but low geometric flexibility

➔

Radial Basis Functions (RBF) (1972)
Choose instead as basis functions translates of radially Symmetric functions:
PS becomes a special case, but now possible to scatter nodes in any number of dimensions, with no danger of singularities

➔

RBF-FD (2000)
Radial Basis Function-generated FD formulas. All approximations again local, but nodes can now be placed freely

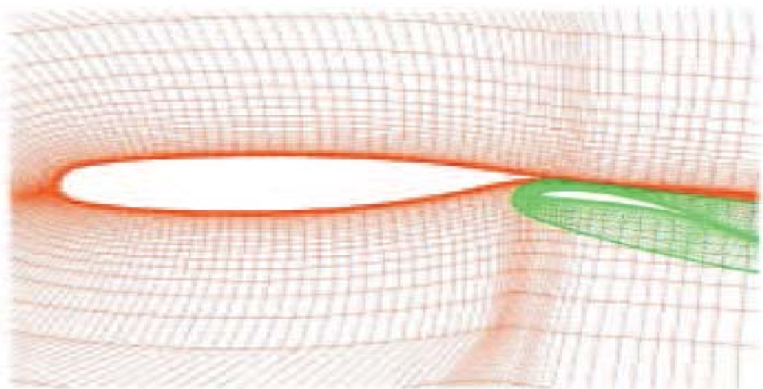
- Easy to achieve high orders of accuracy (4^{th} to 8^{th} order)
- Excellent for distributed memory computers / GPUs
- Local node refinement trivial in any number of dimensions (for ex. in 5+ dimensional mathematical finance applications).

Meshes vs. Mesh-free discretizations

Structured meshes:

Finite Differences (FD),
Discontinuous Galerkin (DG)
Finite Volumes (FV)
Spectral Elements (SE)

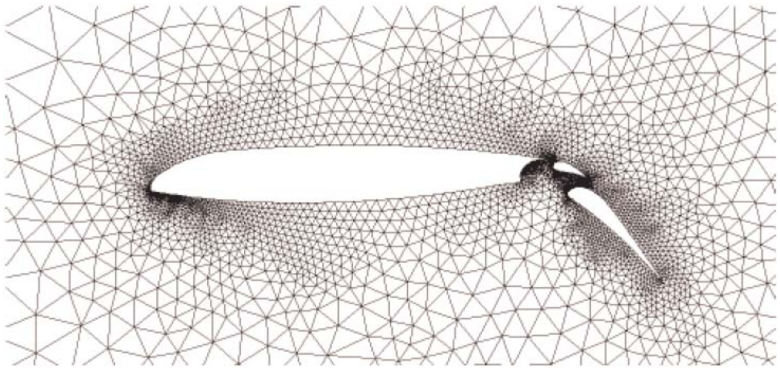
Require domain decomposition / curvilinear mappings



Unstructured meshes:

Finite Elements (FE)

Improved geometric flexibility; requires triangles, tetrahedrons, etc.

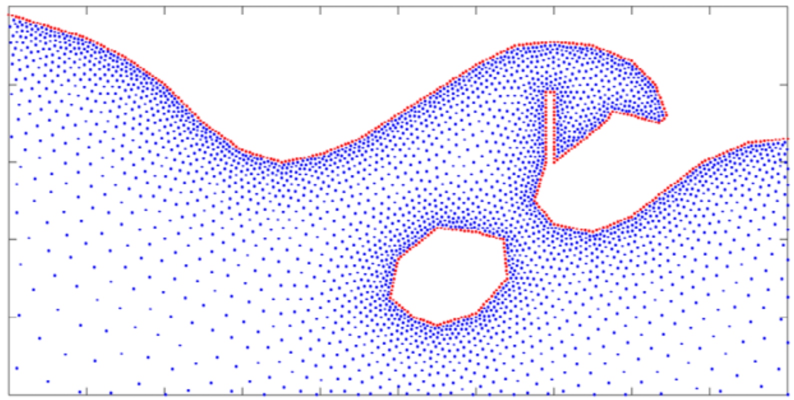


Mesh-free:

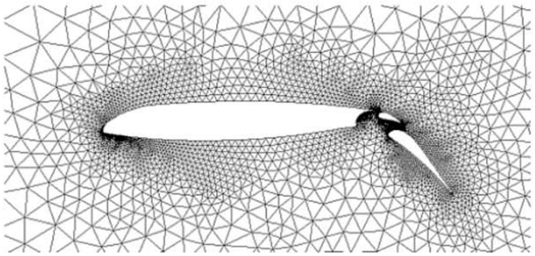
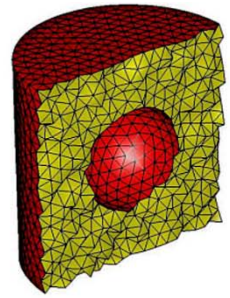
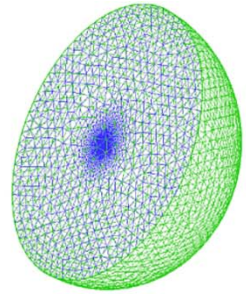
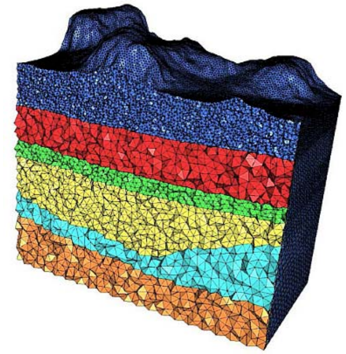
Radial Basis Function generated FD (RBF-FD)

Use RBF methods to generate weights in scattered node FD formulas

Total geometric flexibility;
needs just scattered nodes, but no connectivites, e.g. no triangles or mappings



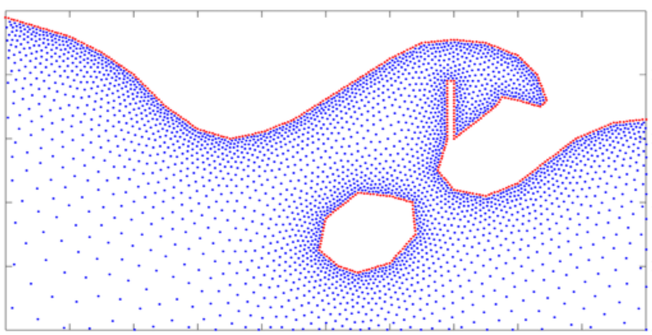
Unstructured meshes:

In 2-D: Quick to go from quasi-uniform nodes to well-balanced Delaunay triangularization
(no circumscribed circle will ever contain another node – guarantee against 'sliver' triangles).

In 3-D: Finding good tetrahedral sets can even become a dominant cost (especially in changing geometries)

Mesh-free:

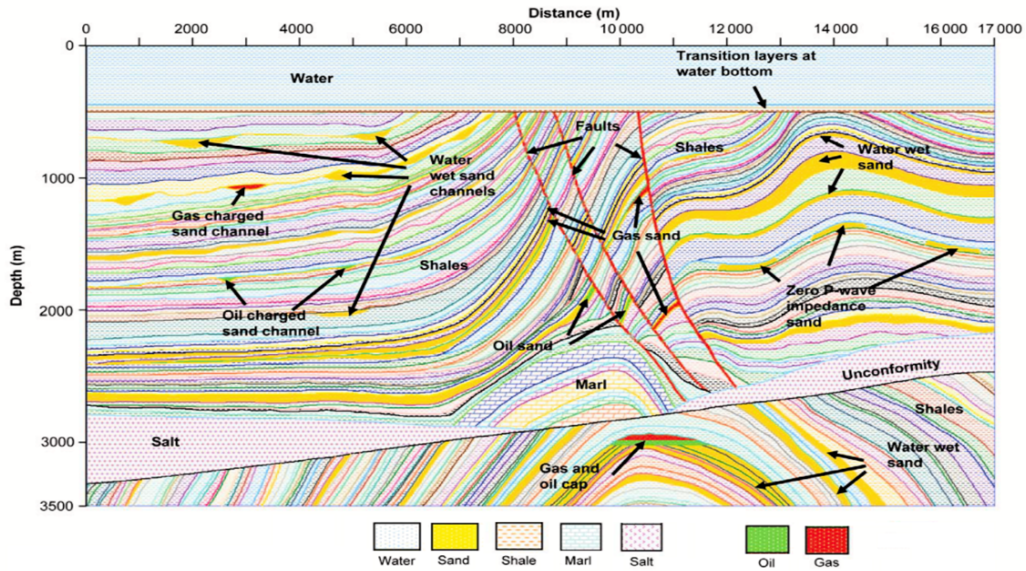


In both **2-D** and **3-D**, it is very fast to 'scatter' nodes quasi-uniformly, with prescribed density variations and aligning with boundaries.

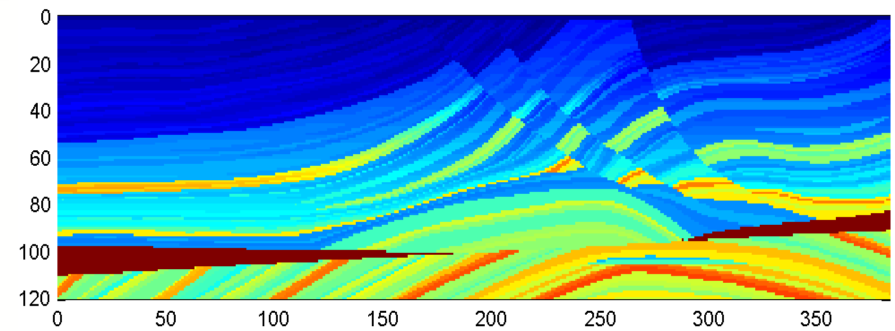
In **any-D**, all that RBF-FD needs for each node only a list of its nearest neighbors – total cost O(N log N) using *kd-tree*.

Example of an RBF-FD application: Seismic exploration

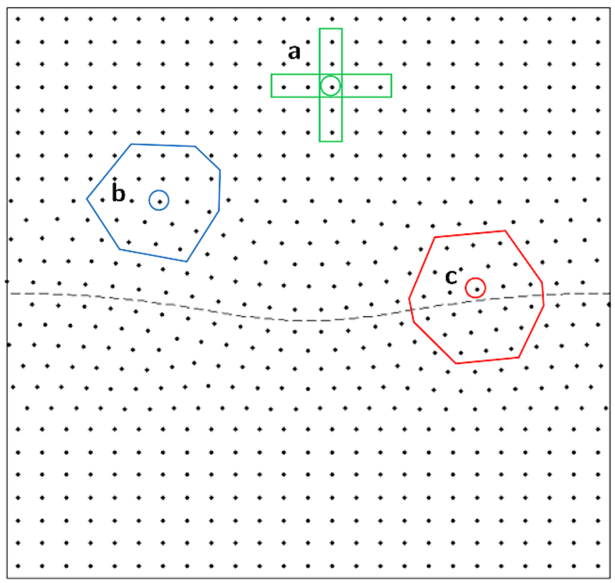
2-D slice off coast of Madagascar



Classical 2-D simplified test problem



RBF-FD idea



Regular Finite Differences (FD) are fine if:

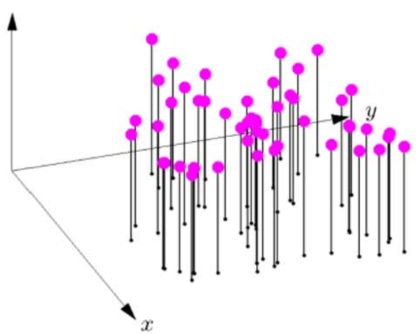
- of high order of accuracy,
- the material interfaces are aligned with the grid.

Mapping grids to realistic geometries is hopeless. So instead:

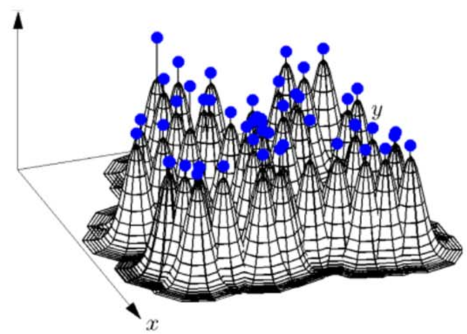
- align nodes locally to each interface
- can still use grid / regular FD away from interfaces (a)
- need to get high order accurate stencils for node sets such as (b) and (c).

Turns out: Cannot use polynomials to get scattered-node 'FD' weights beyond 1-D;
Works excellently if we replace polynomials with **Radial Basis Functions (RBFs)**
We will return to this example later

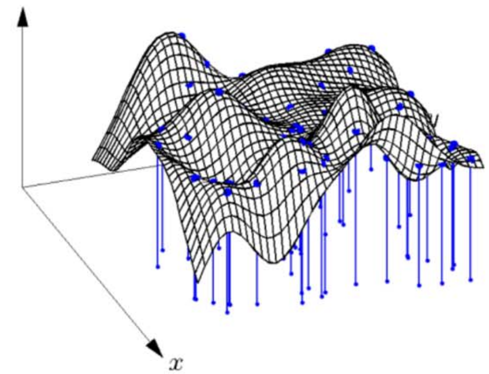
RBF idea, In pictures (for 2-D scattered data):



Scattered data within a 2-D region



Radial basis functions – here 'rotated' Gaussians

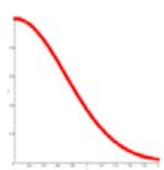


Linear combination of the basis functions that fits all the data

Many types of RBFs are available; Three examples:

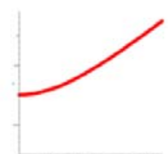
Gaussian (GA)

$e^{-(\varepsilon r)^2}$



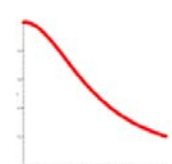
Multiquadric (MQ)

$\sqrt{1+(\varepsilon r)^2}$



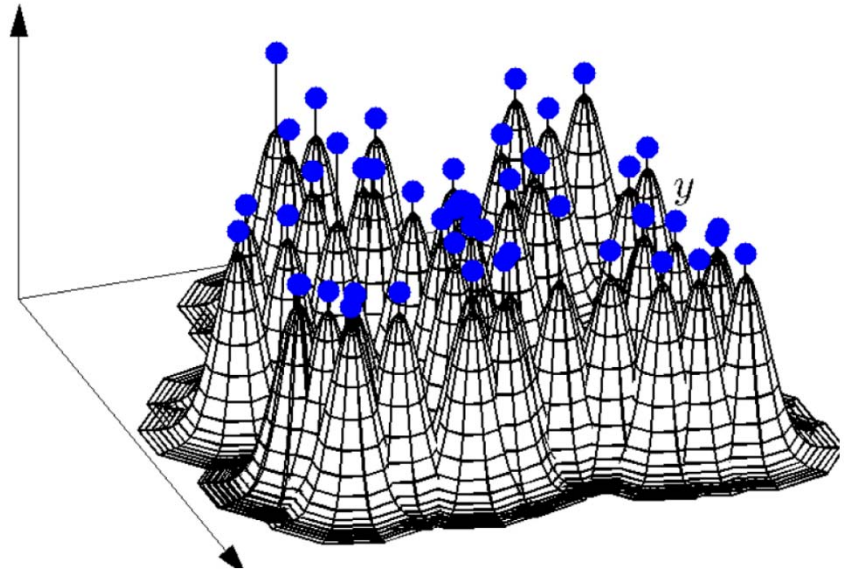
Inverse Quadratic (IQ)

$\dfrac{1}{1+(\varepsilon r)^2}$



Infinitely smooth RBFs (such as these ones) give spectral accuracy for interpolation and derivative approximations.

RBF idea, In formulas:



Given scattered data ($\underline{x}_k$, f_k), k = 1,2, … , n in d-D, the RBF interpolant is

$$s(\underline{x}) = \sum_{k=1}^{n} \lambda_k \phi(\| \underline{x} - \underline{x}_k \|)$$

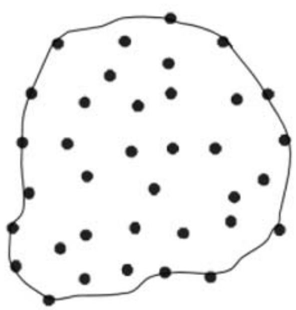
The coefficients λ_k can be found by collocation: s($\underline{x}_k$) = f_k, k = 1,2, … , n :

$$\begin{bmatrix} \phi(\| \underline{x}_1 - \underline{x}_1 \|) & \phi(\| \underline{x}_1 - \underline{x}_2 \|) & \cdots & \phi(\| \underline{x}_1 - \underline{x}_n \|) \\ \phi(\| \underline{x}_2 - \underline{x}_1 \|) & \phi(\| \underline{x}_2 - \underline{x}_2 \|) & \cdots & \phi(\| \underline{x}_2 - \underline{x}_n \|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\| \underline{x}_n - \underline{x}_1 \|) & \phi(\| \underline{x}_n - \underline{x}_2 \|) & \cdots & \phi(\| \underline{x}_n - \underline{x}_n \|) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

What is so special about expanding in RBFs?

No set of pre-specified basis functions (say, based on multivariate polynomials, spherical harmonics, etc.) can guarantee a non-singular system in case of scattered nodes.

Scattered nodes:

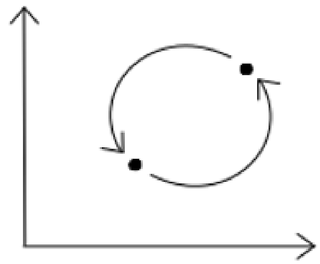


Interpolant: $s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \Psi_k(\underline{x})$

System that determines the expansion coefficients λ_k

$$\begin{bmatrix} \Psi_1(\underline{x}_1) & \Psi_2(\underline{x}_1) & \cdots & \Psi_N(\underline{x}_1) \\ \Psi_1(\underline{x}_2) & \Psi_2(\underline{x}_2) & \cdots & \Psi_N(\underline{x}_2) \\ \vdots & \vdots & & \vdots \\ \Psi_1(\underline{x}_N) & \Psi_2(\underline{x}_N) & \cdots & \Psi_N(\underline{x}_N) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}$$

Move any two nodes so they exchange locations:
Two rows of the matrix become interchanged; the determinant changes sign,
implies determinant was zero somewhere along the way.



What is different when using RBFs?
Two rows – but also two columns – become Interchanged;
the determinant kept its sign.

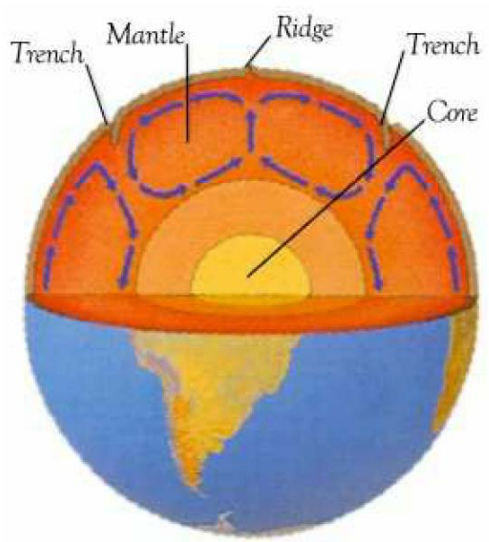
$$\begin{bmatrix} \phi(\|\underline{x}_1 - \underline{x}_1\|) & \phi(\|\underline{x}_1 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_1 - \underline{x}_N\|) \\ \phi(\|\underline{x}_2 - \underline{x}_1\|) & \phi(\|\underline{x}_2 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_2 - \underline{x}_N\|) \\ \vdots & \vdots & & \vdots \\ \phi(\|\underline{x}_N - \underline{x}_1\|) & \phi(\|\underline{x}_N - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_N - \underline{x}_N\|) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}$$

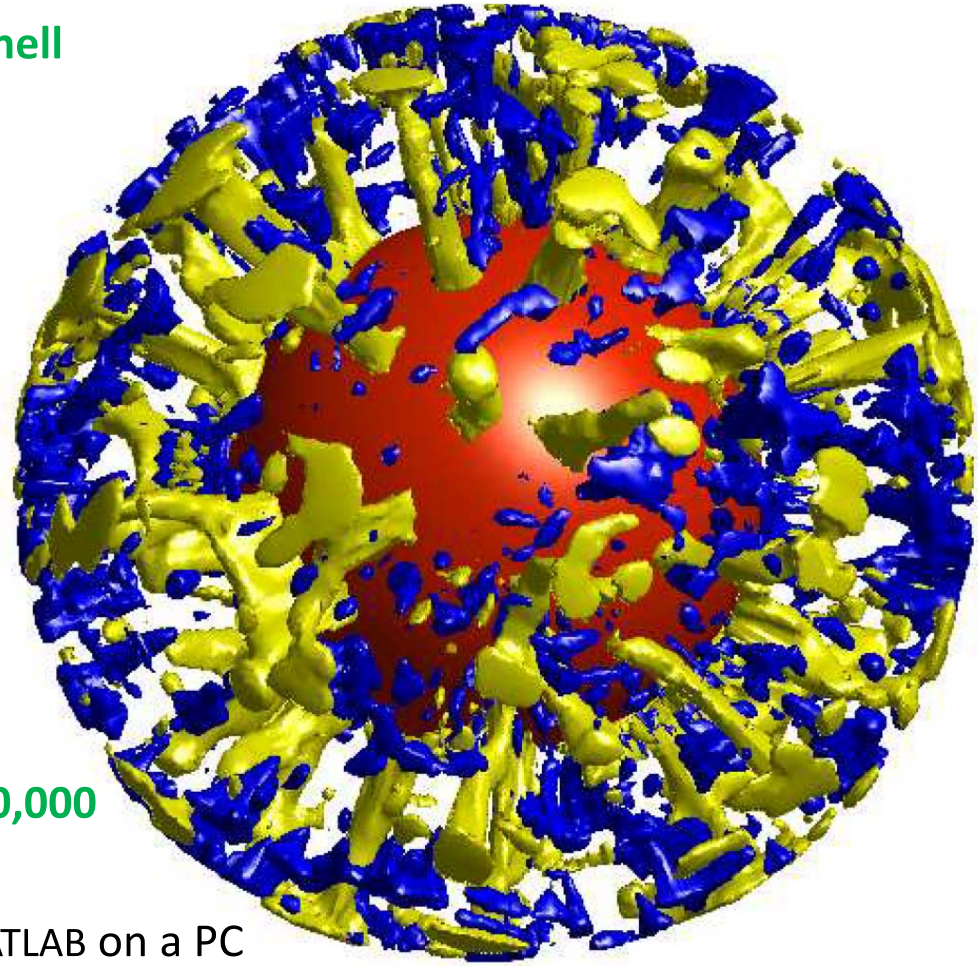
Key theorem: For most standard RBF choices, the RBF system can never be singular, no matter how any number of nodes are scattered in any number of dimensions.

Examples of two PDE applications using global RBFs

1. **Thermal Convection in a 3-D Spherical Shell**

   (Wright, Flyer and Yuen, 2009)





Example of computed solution for Ra = 500,000

Isosurfaces of perturbed temperature:
Single frame from a movie generated in MATLAB on a PC

At somewhat lower Ra number, a similar RBF calculation revealed a physical instability in an unexpected parameter regime, afterwards confirmed on the Japanese *Earth Simulator*.

Another global RBF example: Reaction-diffusion equations over curved biological surfaces

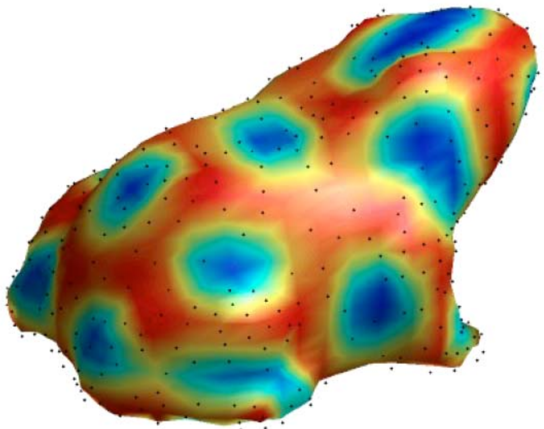
(Piret, 2012)

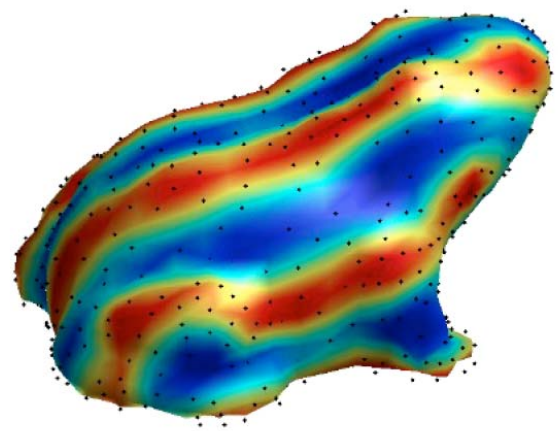
The *Brusselator equation*, modeling pattern formation, is solved here by global RBFs over the surface of a frog

- The 560 scattered nodes serve both as collocation points and to define the body shape
- Spectral accuracy: Only 2 points are needed per wave length to be resolved

Top row:

Snapshots from a computed time evolution for two different parameter values





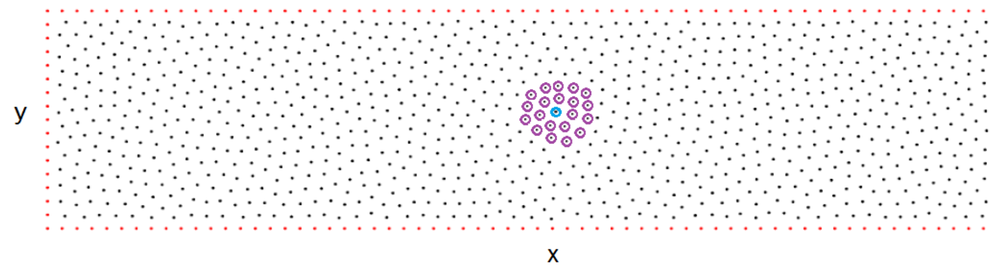
Bottom row:

Left: Tabasara rain frog

Right: Poison dart frog

Calculation of weights in RBF-FD stencil for a differential operator *L*

Choose *weights* so the result becomes
Exact for all RBFs interpolants of the form
$s(\underline{x}) = \sum_{k=1}^{n} \lambda_k \phi(\| \underline{x} - \underline{x}_k \|) + \{ p_m(\underline{x}) \}$
with constraints $\sum \lambda_k p_m(x_k) = 0$



System to solve for weights in case of 2-D, when also using up to linear polynomials with corresponding constraints $\Rightarrow$

$$\left[\begin{array}{ccc|ccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_n & y_n \\ \hline 1 & \cdots & 1 & & & \\ x_1 & \cdots & x_n & & 0 & \\ y_1 & \cdots & y_n & & & \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hline w_{n+1} \\ w_{n+2} \\ w_{n+3} \end{array}\right] = \left[\begin{array}{c} L\phi(\| \underline{x} - \underline{x}_1 \|) \,|_{\underline{x}=\underline{x}_c} \\ \vdots \\ L\phi(\| \underline{x} - \underline{x}_n \|) \,|_{\underline{x}=\underline{x}_c} \\ \hline L1 \,|_{\underline{x}=\underline{x}_c} \\ Lx \,|_{\underline{x}=\underline{x}_c} \\ Ly \,|_{\underline{x}=\underline{x}_c} \end{array}\right]$$

Same *A*-matrix as above; the entries w_{n+1}, … should be ignored.

Common RBF types: Infinitely smooth, e.g. GA: $\phi(r) = e^{-(\varepsilon r)^2}$, MQ: $\phi(r) = \sqrt{1+(\varepsilon r)^2}$

or finitely smooth, e.g. PHS: $\phi(r) = r^{2m} \log r, \quad \phi(r) = r^{2m+1}.$

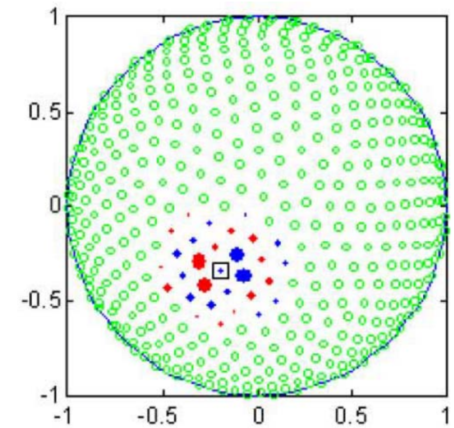
Some observations when using PHS with supporting polynomials:

- Non-singularity of linear system again assured,
- When refining, the polynomial part gradually 'takes over' from RBF part,
- With PHS, can use one-sided approximations at boundaries – like for splines an absence of Runge phenomena.

Convective flow around a sphere with the RBF-FD method

(Fornberg and Lehto, 2011)

RBF-FD stencil illustration: N = 800 ME nodes, n = 30. $\Rightarrow$

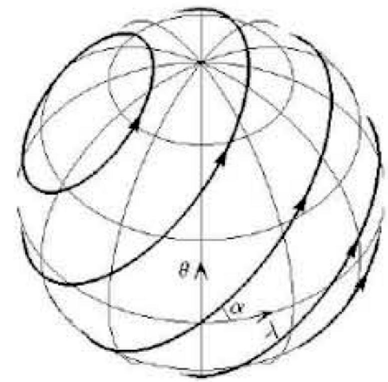


No surface bound coordinate system used

$\Rightarrow$ no counterpart to pole singularities

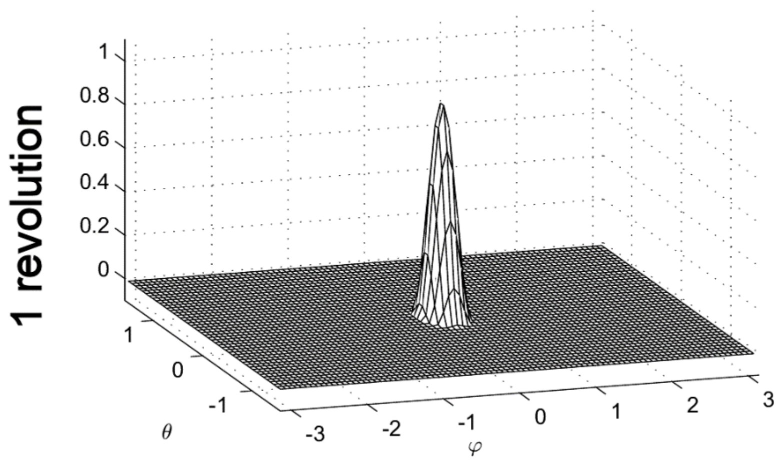
Test problem: Solid body rotation around a sphere $\Rightarrow$

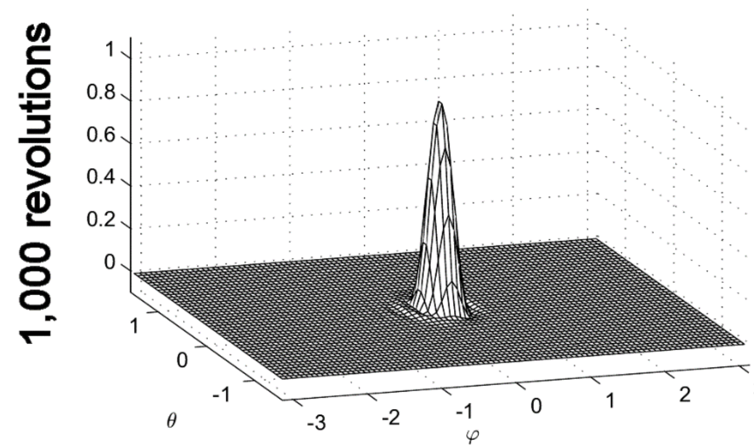
Initial condition: Cosine bell: N = 25,600, n = 74, RK4 in time



Numerical solution:

- No visible loss in peak height; minimal trailing wave trains
- For given accuracy, the most cost effective method available





Return to Seismic Application: Forward vs. Inverse Modeling

Recall the 2-D vertical slice near Madagascar:

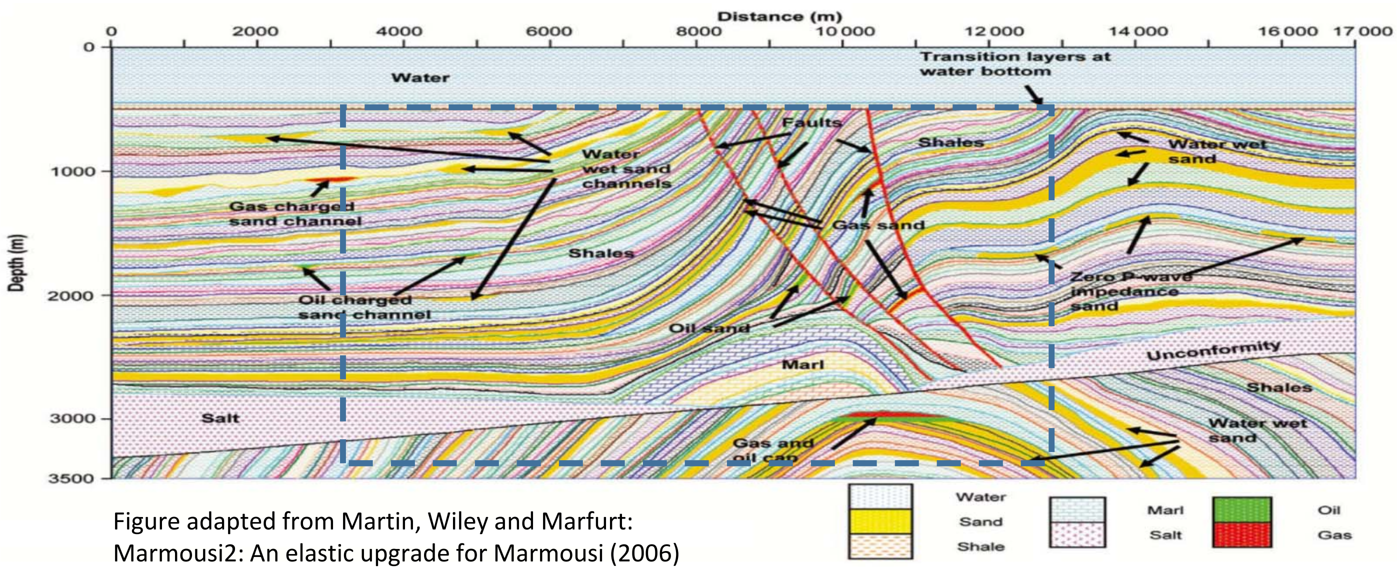


Figure adapted from Martin, Wiley and Marfurt: Marmousi2: An elastic upgrade for Marmousi (2006)

Region inside dashed rectangle simplified to form standardized Marmousi test problem

(shown on next slide)

Forward modeling

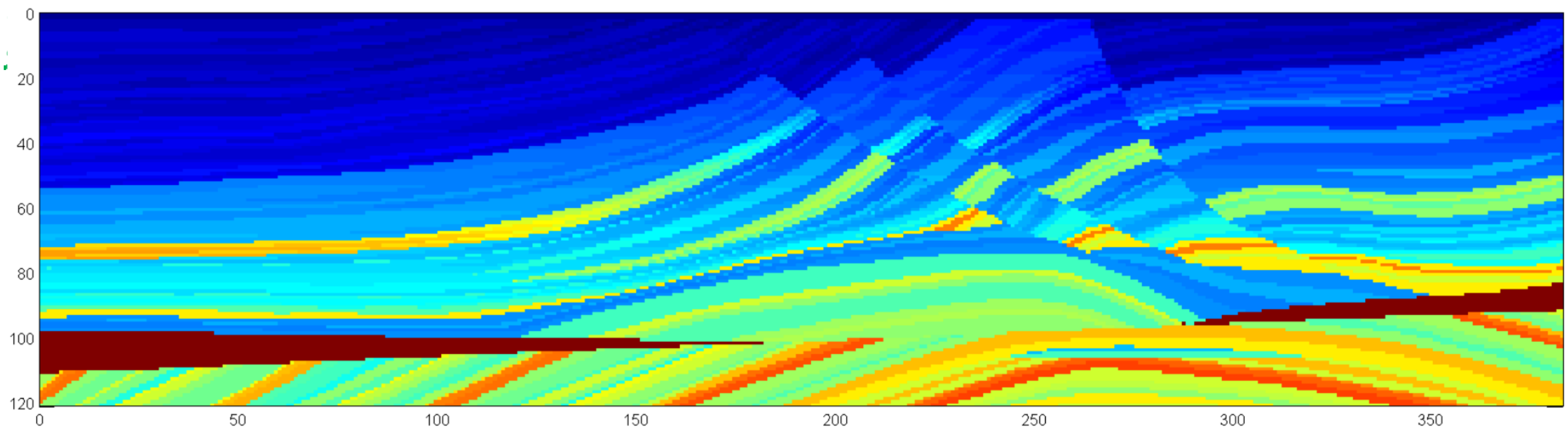
Assume subsurface structures known, then simulate the propagation of elastic waves

Inverse modeling

Adjust the subsurface assumptions to reconcile forward modeling with seismic data.

Requires fast and accurate solution of a vast number of forward modeling problems.

Governing equations for elastic wave propagation in 2-D



Acoustic (pressure wave) velocities ↑

Elastic wave equation in 2-D

$$\begin{cases} \rho u_t &= f_x + g_y \\ \rho v_t &= g_x + h_y \\ f_t &= (\lambda + 2\mu) u_x + \lambda v_y \\ g_t &= \mu (u_x + v_y) \\ h_t &= (\lambda + 2\mu) v_y + \lambda u_x \end{cases}$$

Dependent variables:

u, *v* horizontal and vertical velocities

f, *g*, *h* components of the symmetric stress tensor

Material parameters:

ρ density

λ, μ Lamé parameters (compression and shear)

Wave types:

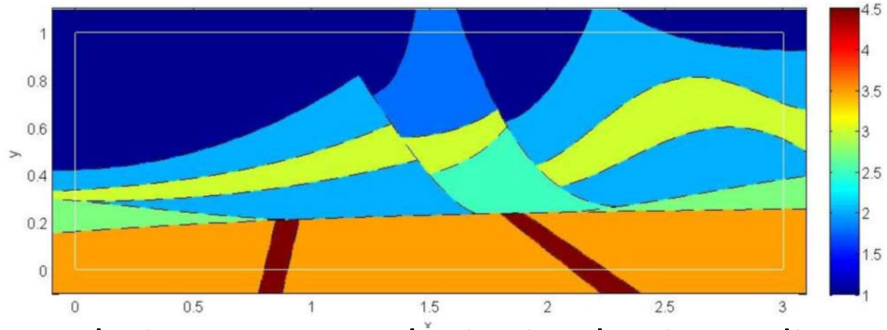
Pressure $c_p = \sqrt{(\lambda + 2\mu)/\rho}$, Shear $c_s = \sqrt{\lambda / \rho}$

Also: Rayleigh, Love, and Stonley waves

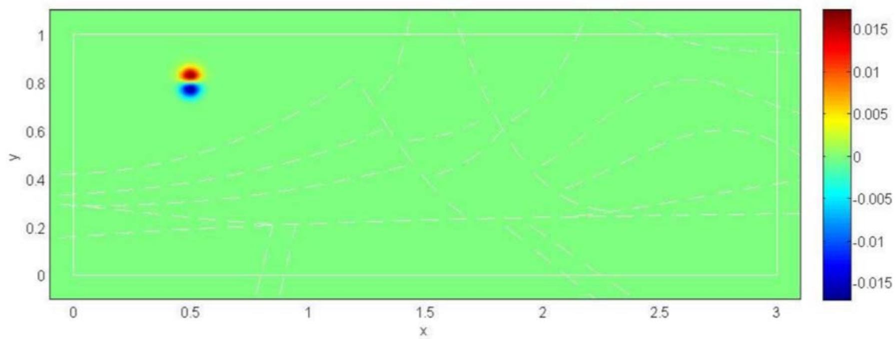
(Martin, Fornberg, St-Cyr, 2015)

Region Type	Dominant Errors	Computational Remedies
Smoothly variable medium	Dispersive errors	High order approximations 1980's From 2^{nd} order to 4^{th} order FD (or FEM) 2010's 20^{th} order (or higher still) FD
Interfaces	Reflection and transmission of pressure and shear waves	<u>Analysis based interface enhancements on grids:</u> Very limited successes reported in the literature in cases of complex geometries <u>Industry standard:</u> Refine and 'hope for the best' (typically 1^{st} order) **<u>Present novelties:</u>** - Distribute RBF-FD nodes to align with all interfaces (suffices for 2^{nd} order) - Modify basis functions to analytically correct for interface conditions (RBF-FD/AC) (high order possible also for curved interfaces)

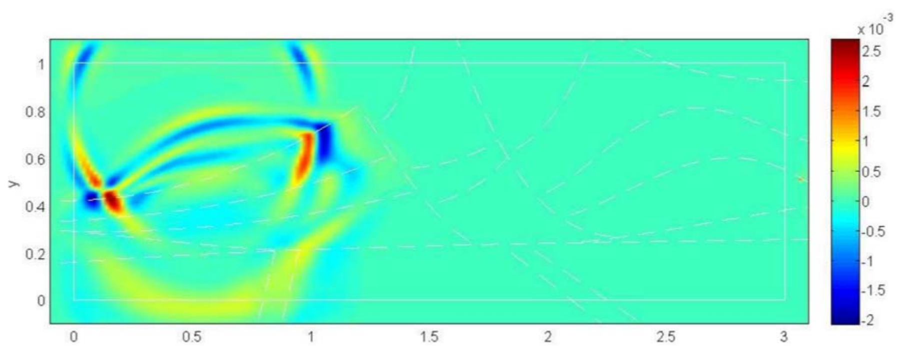
‘Mini-Marmousi’ test case



Relative *p*-wave velocity in elastic medium

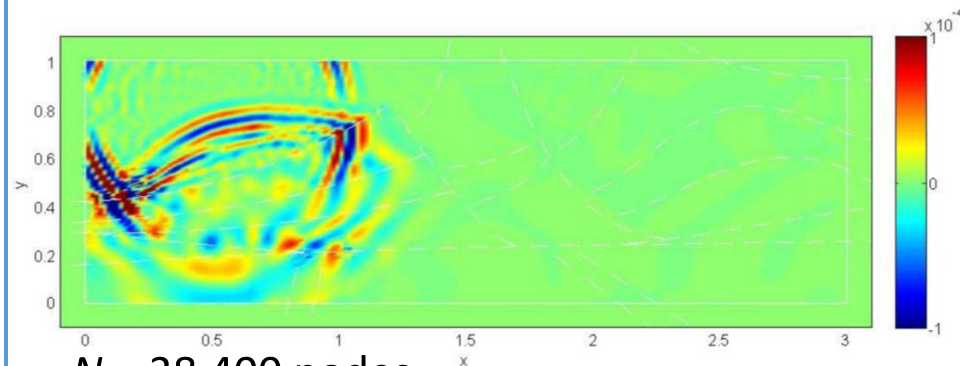


Initial condition for v at time $t = 0$

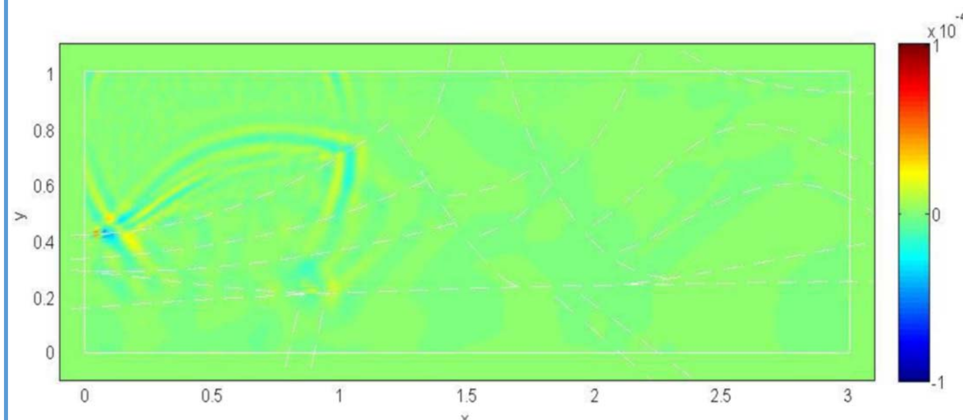


Accurate solution for v at time $t = 0.3$

Errors with RBF-FD/AC discretization,
at $t = 0.3$, using $n = 19$ node RBF-FD stencil



N = 38,400 nodes

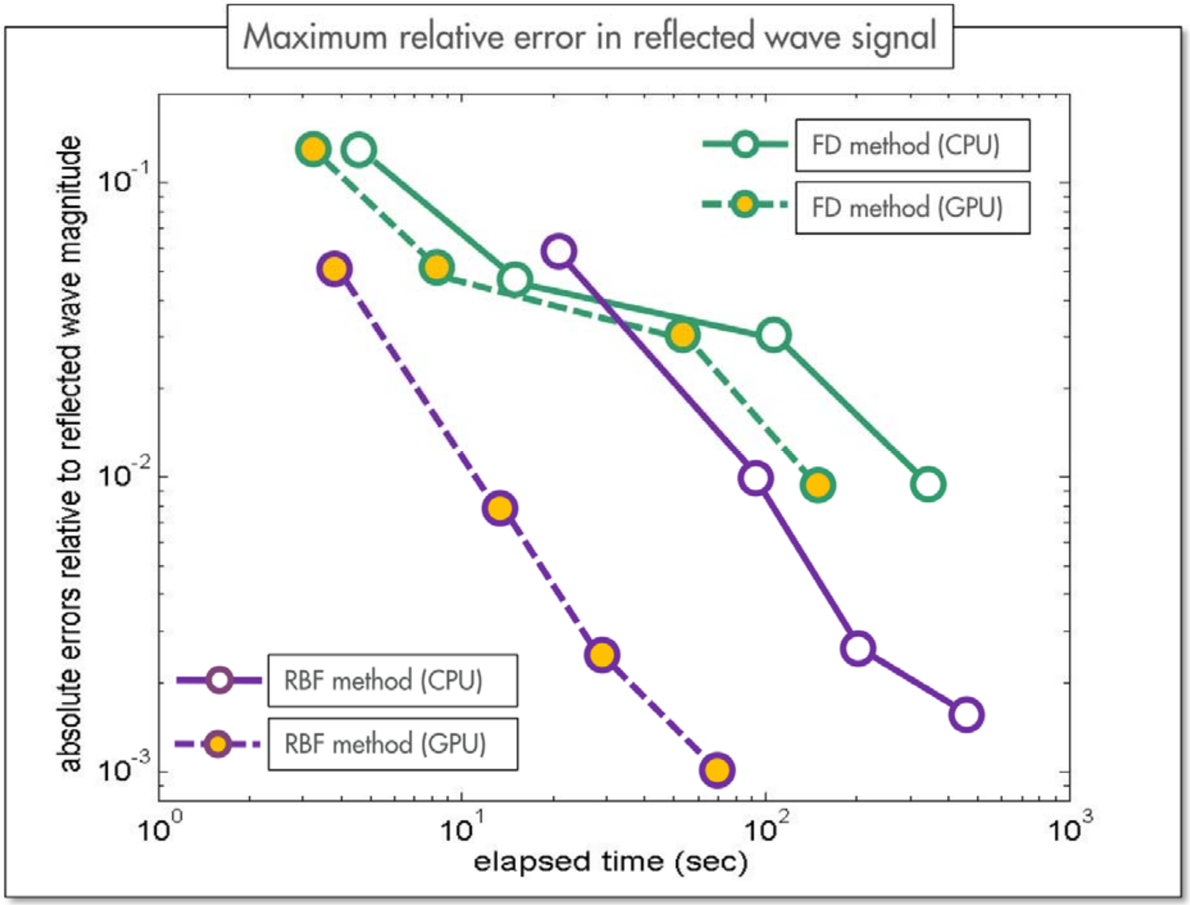


N = 153,600 nodes

Typical node separation reduced by factor of two; error reduced by factor of 10, indicating better than 3rd order in all regions

Timing comparison against FD20 (FD of 20th order of accuracy)

3-D acoustic test problem



CPU vs. GPU:

FD20: Very wide stencils; large domain overlaps ; lots of communications

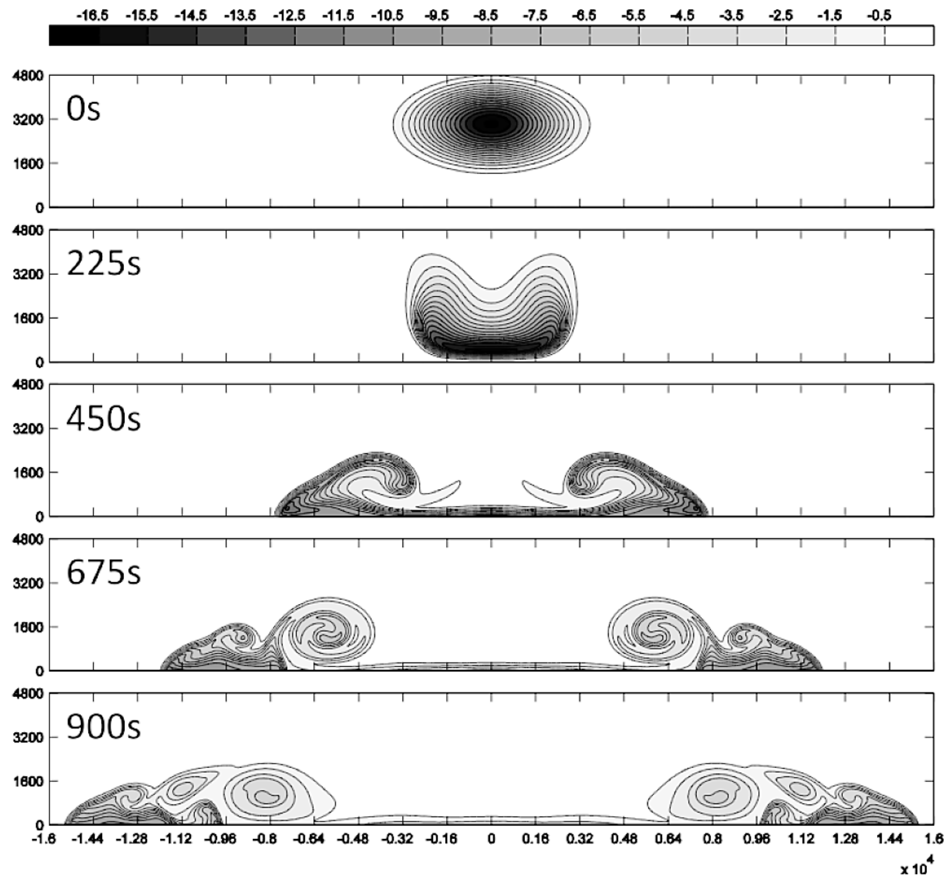
RBF-FD: The opposite in all regards; utilizes GPUs more effectively (in spite of scattered nodes)

Modeling 2D nonhydrostatic compressible Navier-Stokes

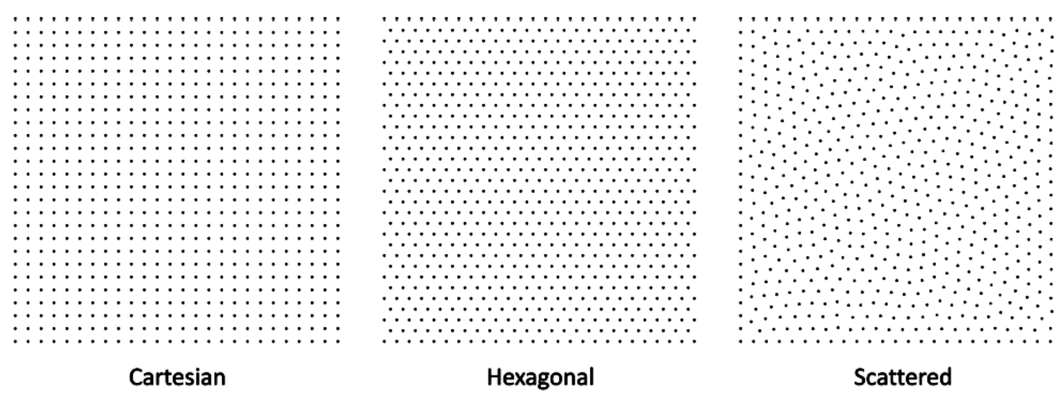
(Flyer, Barnett, Wicker, 2016)

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\,\mathbf{u} - c_p\theta\nabla P - g\mathbf{k} + \mu\Delta\mathbf{u}, \quad \text{momentum}$$
$$\frac{\partial \theta}{\partial t} = -(\mathbf{u}\cdot\nabla)\,\theta + \mu\Delta\theta, \quad \text{energy}$$
$$\frac{\partial P}{\partial t} = -(\mathbf{u}\cdot\nabla)\,P - \frac{R}{c_v}(\nabla\cdot\mathbf{u})\,P, \quad \text{mass}$$

Accurate time evolution $\Rightarrow$

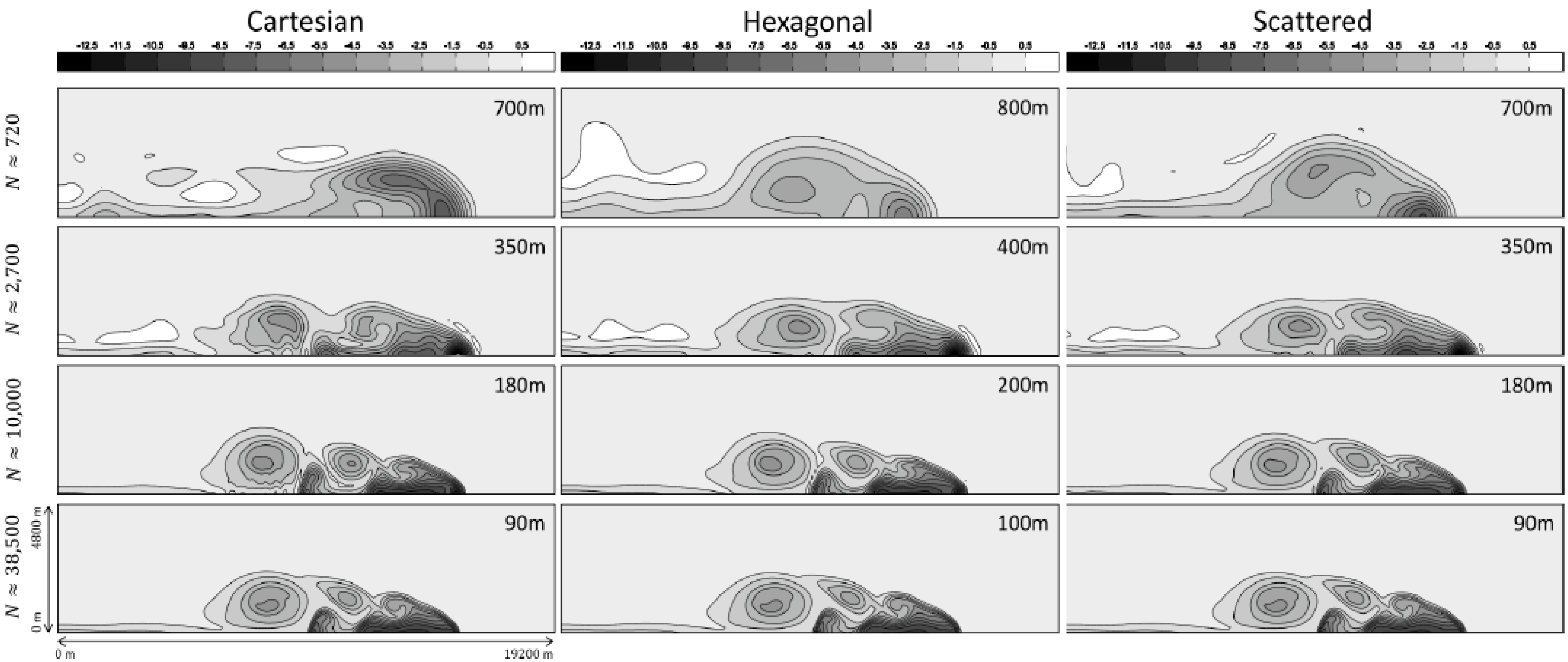


With RBF-FD, it becomes easy to explore the intrinsic capabilities of different node layouts.



Hexagonal have along history, but have never become ‘mainstream’ due to implementation complexities (especially beyond 2-D).

Comparisons on different node layouts



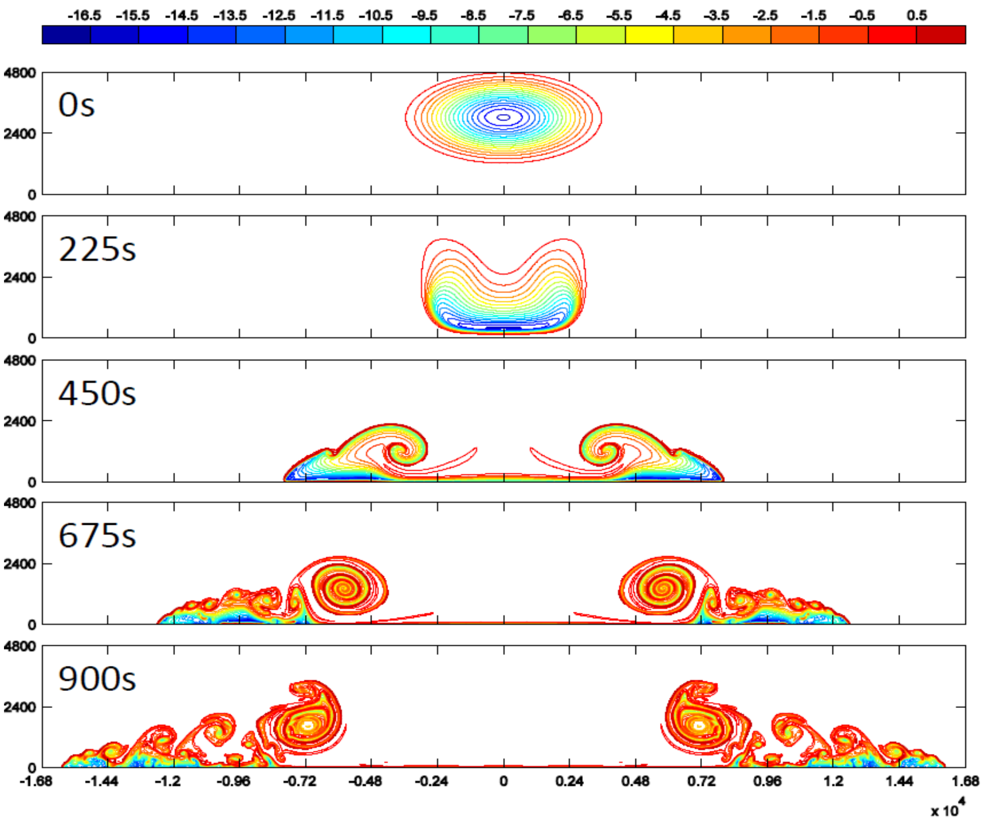
RBFs : r^7 with **4th** degree polynomial support, n = **37**, Δ^3-type hyperviscosity

For comparable node numbers:

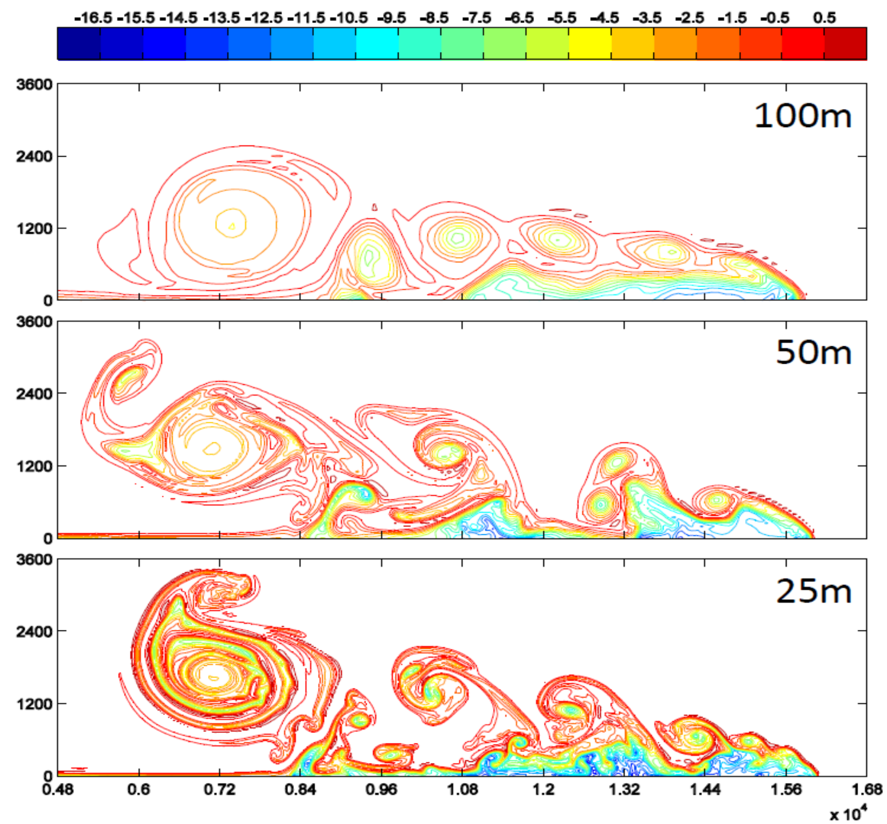
- Cartesian node layout gives rise to the most amount of unphysical artifacts
- Hexagonal nodes excellent (in the past, too complex to be used routinely – now similar concept easily used also in 3-D)
- No detectable performance penalty when going to quasi-uniformly scattered (but have then gained great geometric flexibility).

Same test problem, but with physical viscosity removed altogether

Modeling 2D nonhydrostatic compressible Euler equations – 25m resolution (RBF-FD, hex nodes)



Details when using different resolutions



Calculate weights for numerical integration over a sphere

(Reeger, Fornberg, 2016)

Algorithm steps:

1. Given nodes on the sphere, create a spherical Delaunay triangularization

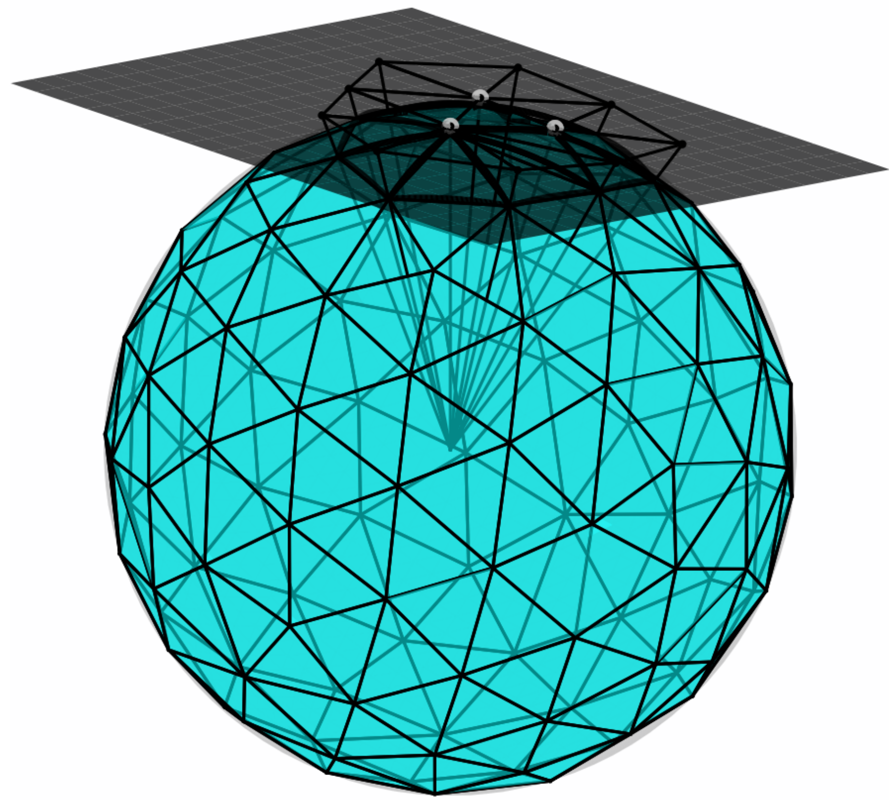
2. For each surface triangle, project it together with some nearby nodes to a tangent plane

3. Find quadrature weights over the local tangent plane node set for the central planar triangle

4. Convert weights from the tangent plane case to corresponding weights on sphere surface

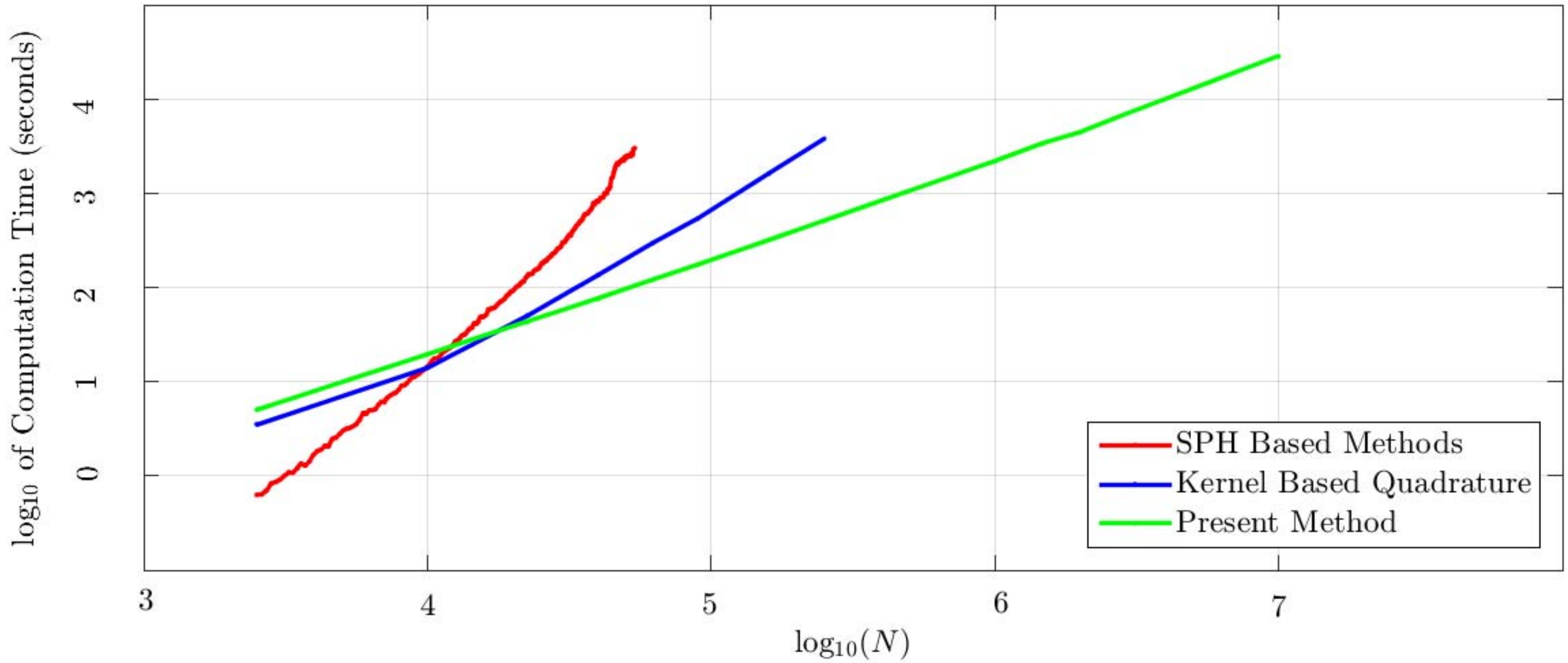
5. Add together the weights for the individual triangles to obtain the full weight set for the sphere

Concept illustration:



Timing results

Methods:	Cost for *N* nodes:	Reference:
Spherical harmonics (SPH)	$O(N^3)$	Womersley and Sloan, 2003, Fornberg and Martel, 2014
Lagrange-RBF	$O(N^2 \log N)$	Fuselier et.al. (2014)
Present RBF-FD	$O(N \log N)$	Reeger and Fornberg (2016)



Timings above on a standard PC. Present method 'embarrassingly parallel' - speeds up proportionally to the number of cores available.

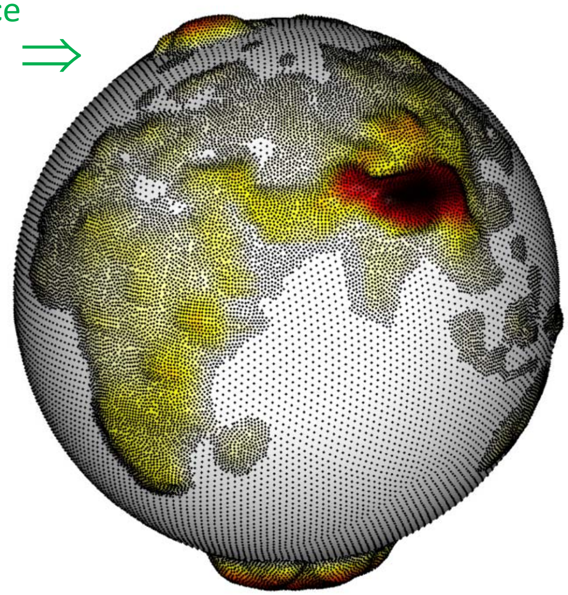
Generalization to arbitrarily curved surfaces in progress Reeger, Fornberg and Watts (2016)

Some examples of recent RBF-FD applications not described here:

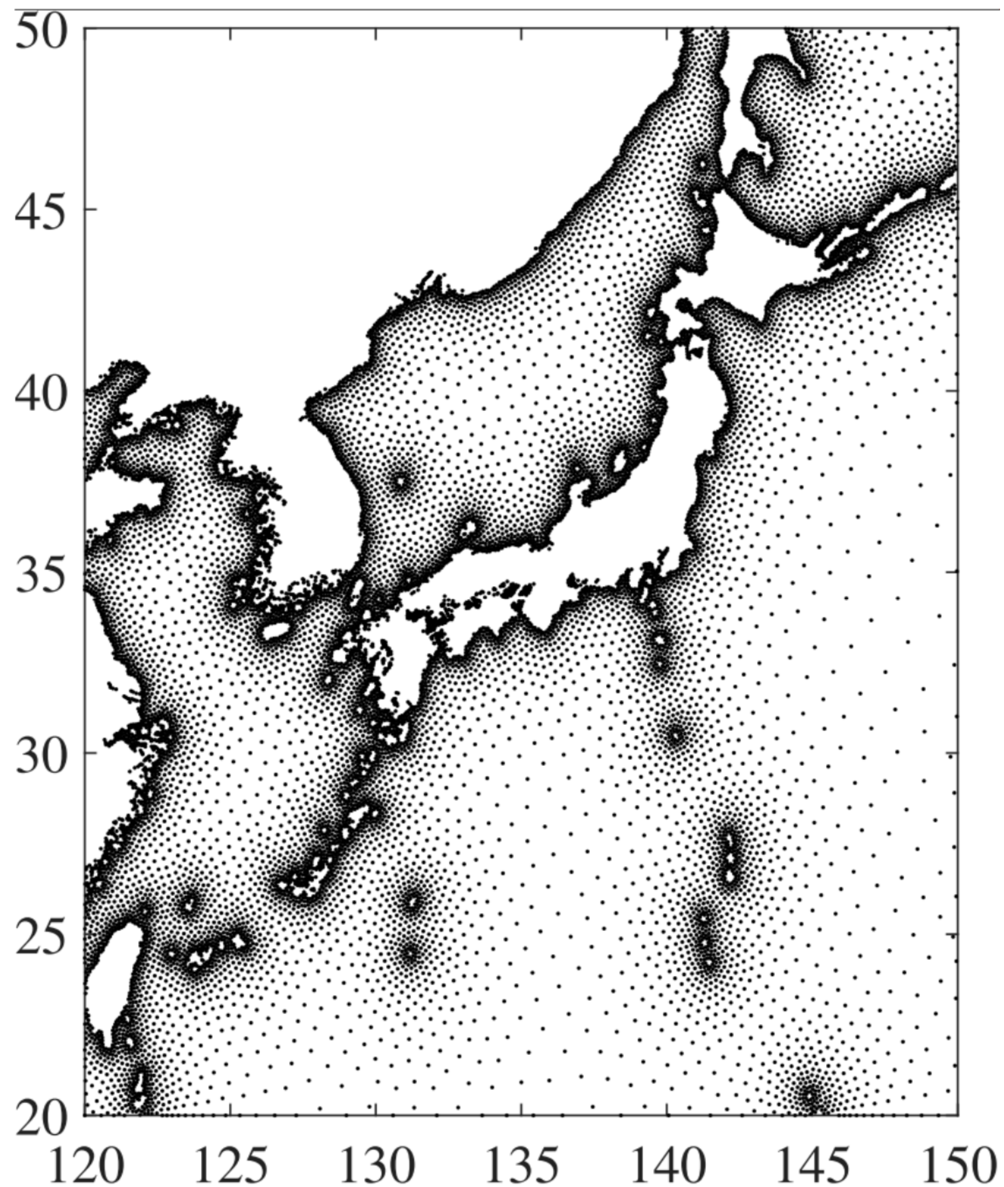
- Global electric circuit:

  Nonlinear elliptic system of PDEs. A recent fully 3-D RBF-FD calculation is the first with any method to use the actual earth topography as its bottom boundary (Bayona, Flyer et.al. 2015).

  Example of a 3-D node set as it touches the earth's surface ⇒



- **Tsunami modeling (beginning collaboration with Japanese scientists)** ⇒



- **Quantum mechanics**

- **Many further applications in elasticity, fluid mechanics**, etc.

Some conclusions

- There is a natural method evolution: FD $\Rightarrow$ PS $\Rightarrow$ RBF $\Rightarrow$ RBF-FD

- RBF and RBF-FD methods combine high accuracy with great flexibility for handling intricate geometries and local refinement

- RBF-FD methods compete very favorably against previous methods on a large number of applications, many with established benchmark problems

- RBF-FD is particularly effective on GPUs and other massively parallel hardware

- RBF-FD is the most effective approach known for quadrature (integration) over general smooth surfaces

SIAM book published November 2015

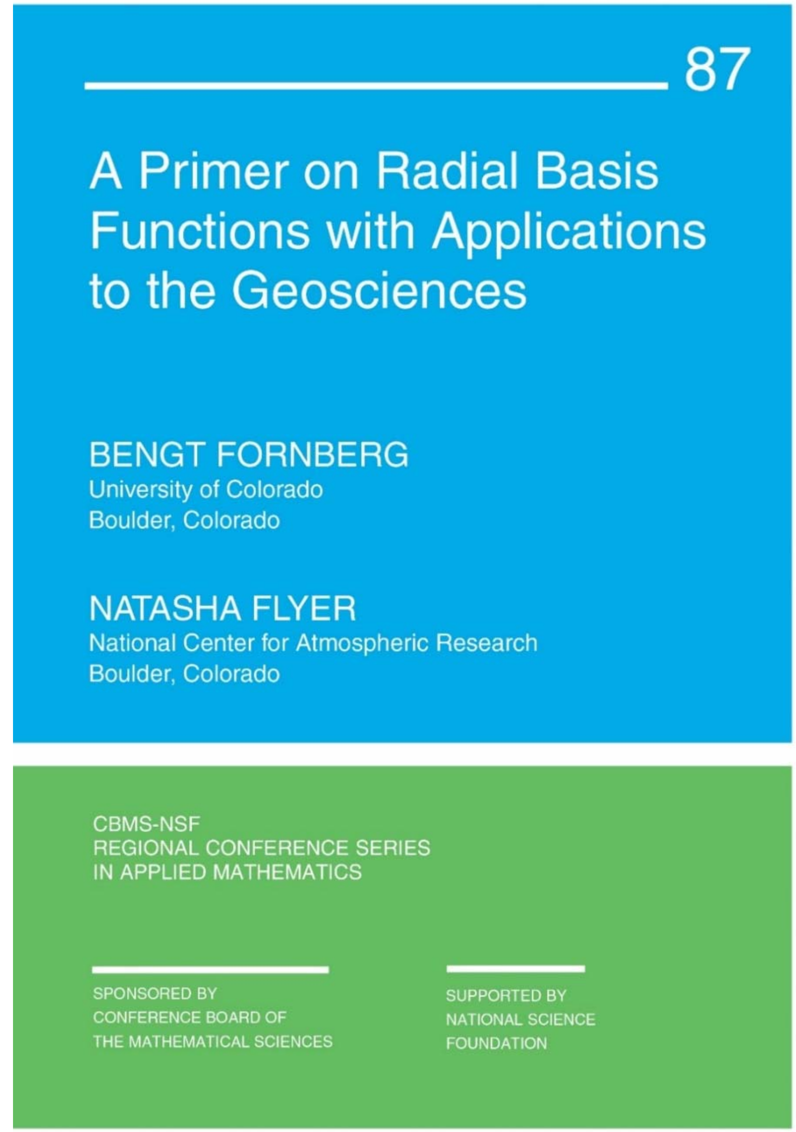
Summarizes the evolution FD $\Rightarrow$ PS $\Rightarrow$ RBF $\Rightarrow$ RBF-FD

Surveys global RBFs

First book format overview of RBF-FD

Geophysics applications include:

- Exploration for oil and gas,
- Weather and climate modeling,
- Electromagnetics, etc.



RBF–FD: LEAVE THE MESH BEHIND !

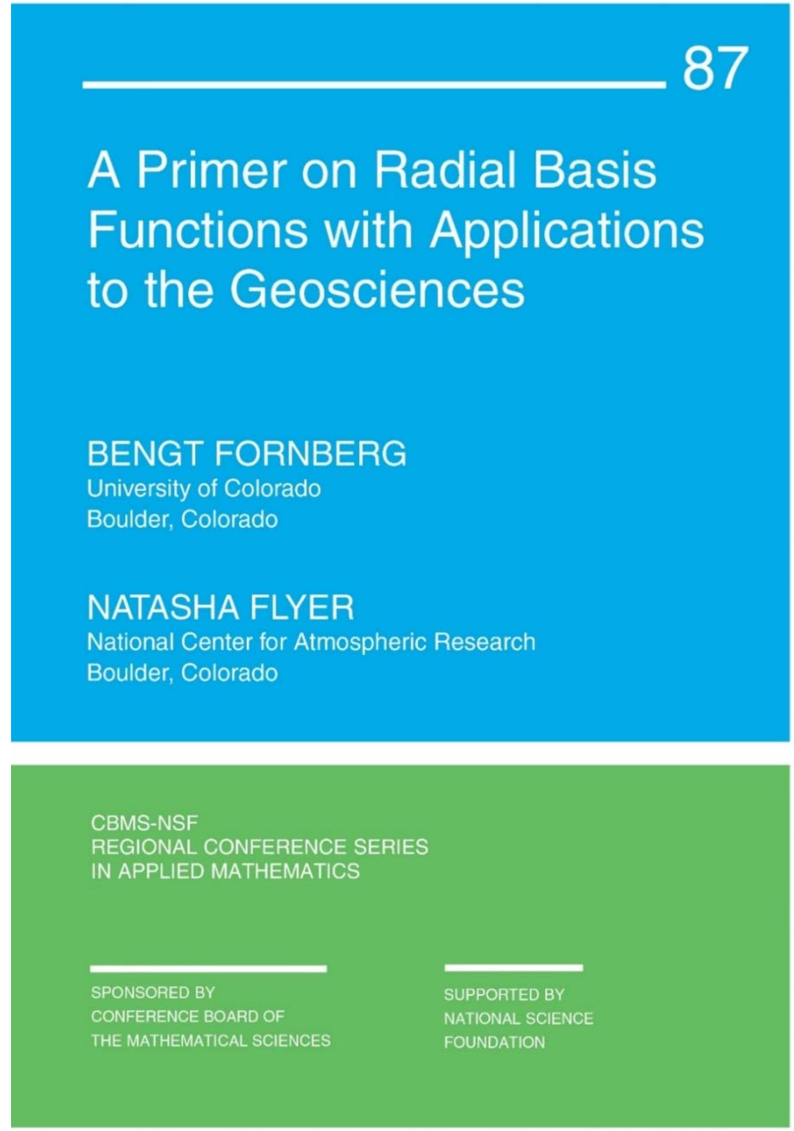
SIAM book published November 2015

Summarizes the evolution FD $\Rightarrow$ PS $\Rightarrow$ RBF $\Rightarrow$ RBF-FD

Surveys global RBFs

First book format overview of RBF-FD

Geophysics applications include:
- Exploration for oil and gas,
- Weather and climate modeling,
- Electromagnetics, etc.



RBF–FD: LEAVE THE MESS BEHIND !