Radial Basis Function Generated Finite Differences (RBF-FD): Basic Concepts and Some Applications

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

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

in collaboration with

Greg Barnett, Victor Bayona, Brad Martin, Cécile Piret 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

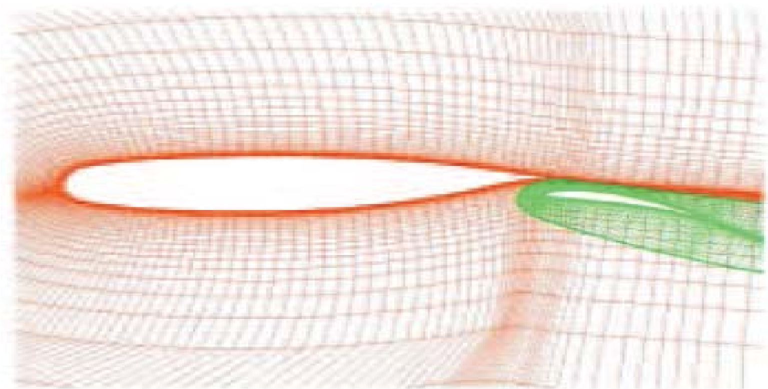
➔

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).

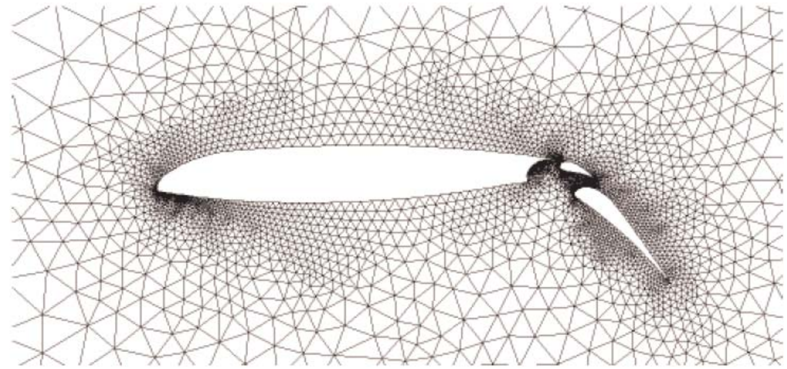
Meshes vs. Mesh-free discretizations



Structured meshes:

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

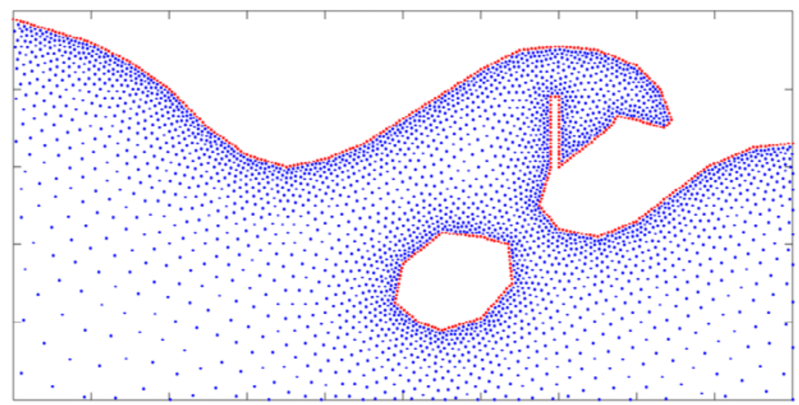
Require domain decomposition / curvilinear mappings



Unstructured meshes:

Finite Elements (FE)

Improved geometric flexibility; requires triangles, tetrahedra, etc.



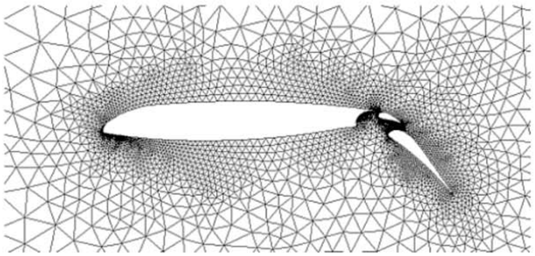
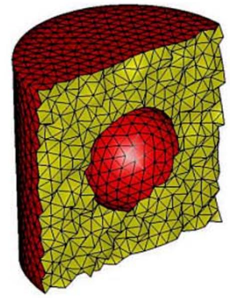
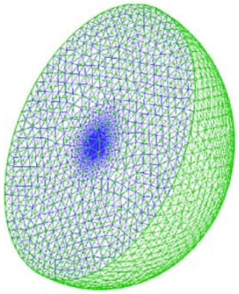
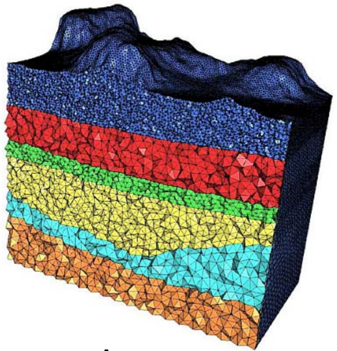
Mesh-free:

Radial Basis Function generated FD (RBF-FD)

Use RBF methods to generate weights in scattered node local FD formulas

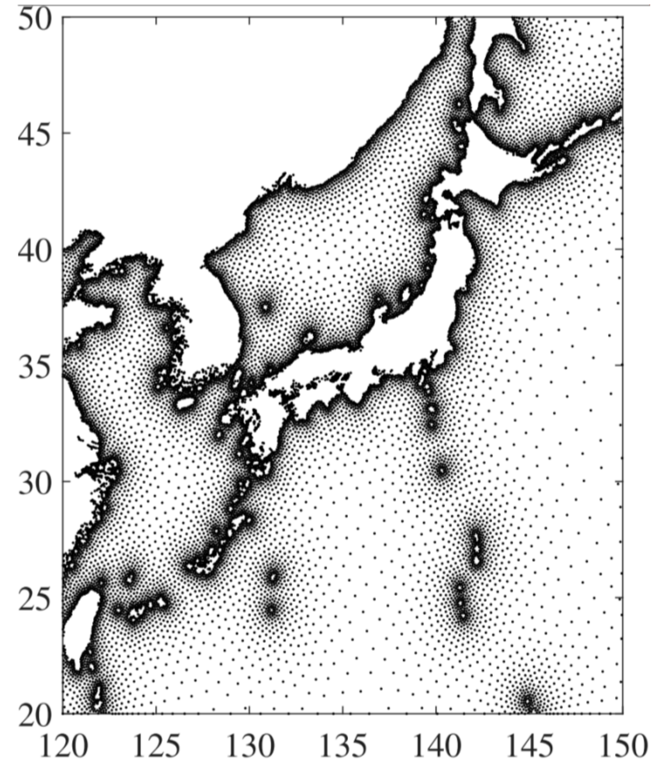
Total geometric flexibility; needs just scattered nodes, but no connectivities, e.g. no triangles or mappings

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 $O(N \log N)$ when using *kd-tree*.

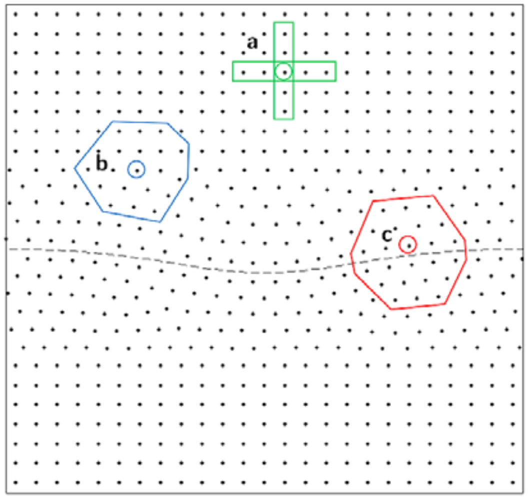
Illustration by Adrian Webb

RBF-FD stencils – Some concept illustrations

2-D planar

Hybrid Cartesian-quasi-uniform node set.

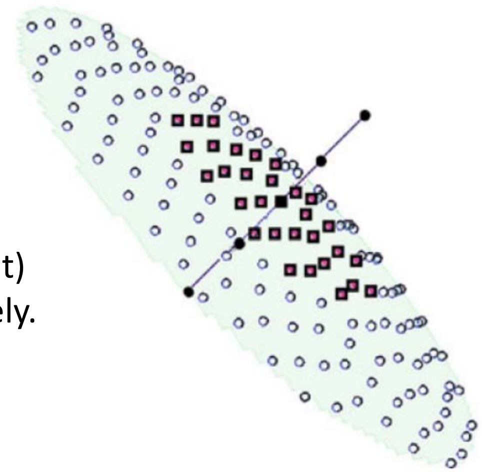
Set-up we'll use later for seismic modeling.



Surface in 3-D space

2-D-like stencil on curved surface.

Normal direction (if present) can be discretized separately.



3-D volume

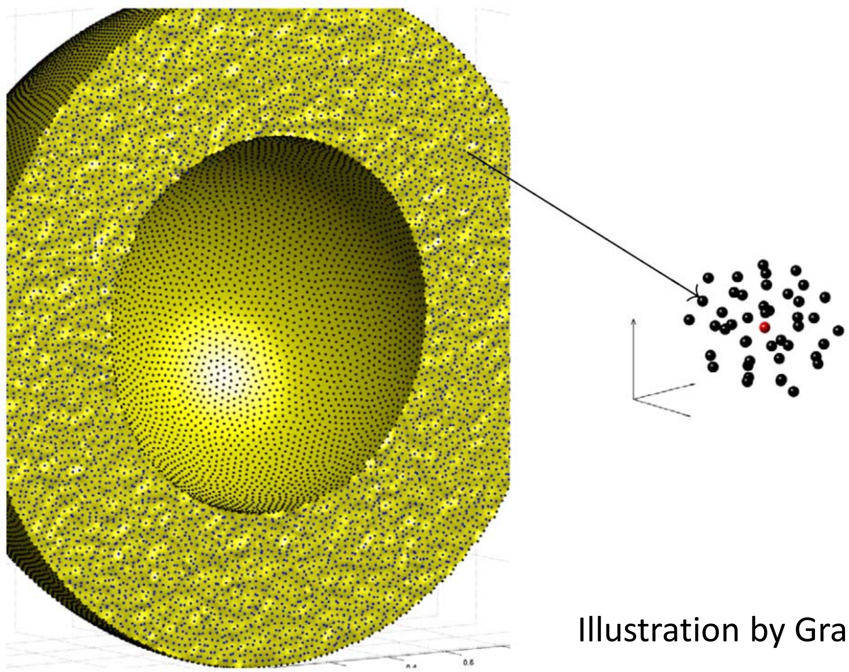


Illustration by Grady Wright

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

Strategy: Choose *weights* so the result becomes exact for all RBFs interpolants of the form

$$s(\underline{x}) = \sum_{k=1}^{n} \lambda_k \phi(\| \underline{x} - \underline{x}_k \|) + \{ p_m(\underline{x}) \} \text{ with constraints } \sum \lambda_k p_m(x_k) = 0$$

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

$$\left[\begin{array}{ccc:ccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_n & y_n \\ \hdashline 1 & \cdots & 1 & & & \\ x_1 & \cdots & x_n & & 0 & \\ y_1 & \cdots & y_n & & & \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hdashline \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{array}\right] = \left[\begin{array}{c} L\phi(\| \underline{x} - \underline{x}_1 \|) \,|_{\underline{x}=\underline{x}_c} \\ \vdots \\ L\phi(\| \underline{x} - \underline{x}_n \|) \,|_{\underline{x}=\underline{x}_c} \\ \hdashline L1 \,|_{\underline{x}=\underline{x}_c} \\ Lx \,|_{\underline{x}=\underline{x}_c} \\ Ly \,|_{\underline{x}=\underline{x}_c} \end{array}\right]$$

A-matrix entries $A_{i,j} = \phi(\| \underline{x}_i - \underline{x}_j \|)$

In result vector γ should be ignored.

Compact formulation of system:

$$\left[\begin{array}{c|c} A & P \\ \hline P^T & 0 \end{array}\right] \left[\begin{array}{c} w \\ \hline \gamma \end{array}\right] = \left[\begin{array}{c} L\phi \\ \hline Lp \end{array}\right]$$

Optimization interpretation:

The same linear system also solves $\underbrace{\min_w \frac{1}{2} w^T A w - w^T L\phi}_{\text{Keeps the } w \text{ small}}$ subject to $\underbrace{P^T w = Lp}_{\text{Exact for polynomials}}$

- Under refinement, order of convergence matches the degree of the polynomials;
- RBFs do not need to be correspondingly smooth; r^3, r^5, r^7 are good choices.

Common RBF types:

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

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

Three main choices when creating RBF-FD stencils / weights:

Smooth RBFs without poly:

Need ε small; Either choose ε so cond(A) about 10^8, or use a stable algorithm (such as RBF-QR, RBF-GA, RBF-RA; initialization cost increase about 10 times, but task 'embarrassingly parallel')

Smooth RBFs with poly:

Typically little gain, since the polynomials lie in about same space as the RBFs. However effective for PDEs at interfaces (used for seismic modeling, described below)

PHS with poly:

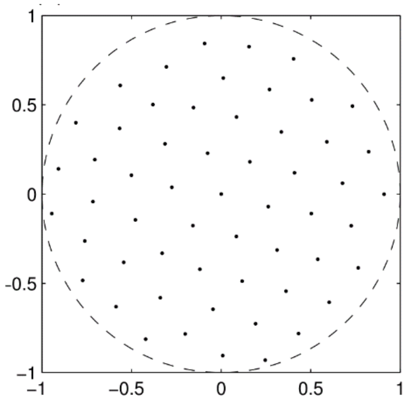
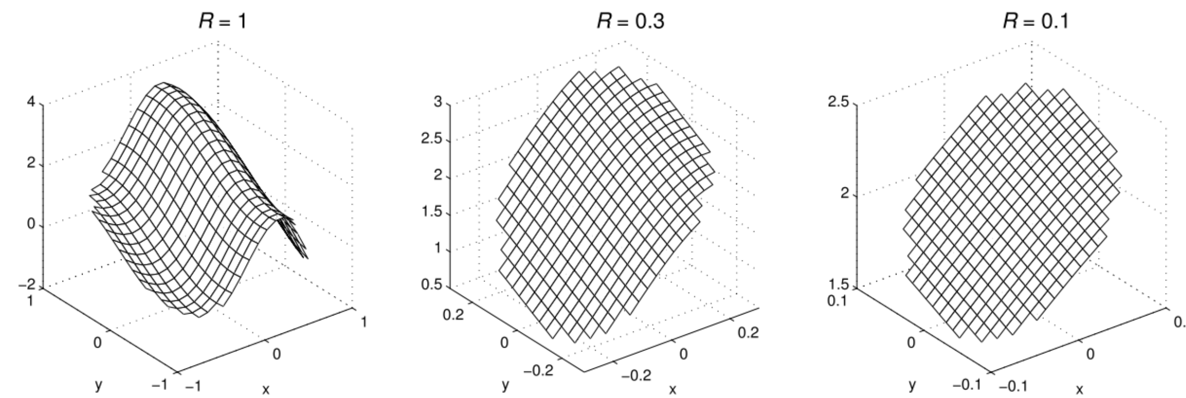
Will become our preferred choice in most cases.

Some key features illustrated in the next four slides.

Accuracy of PHS + poly RBF-FD stencils away from boundaries

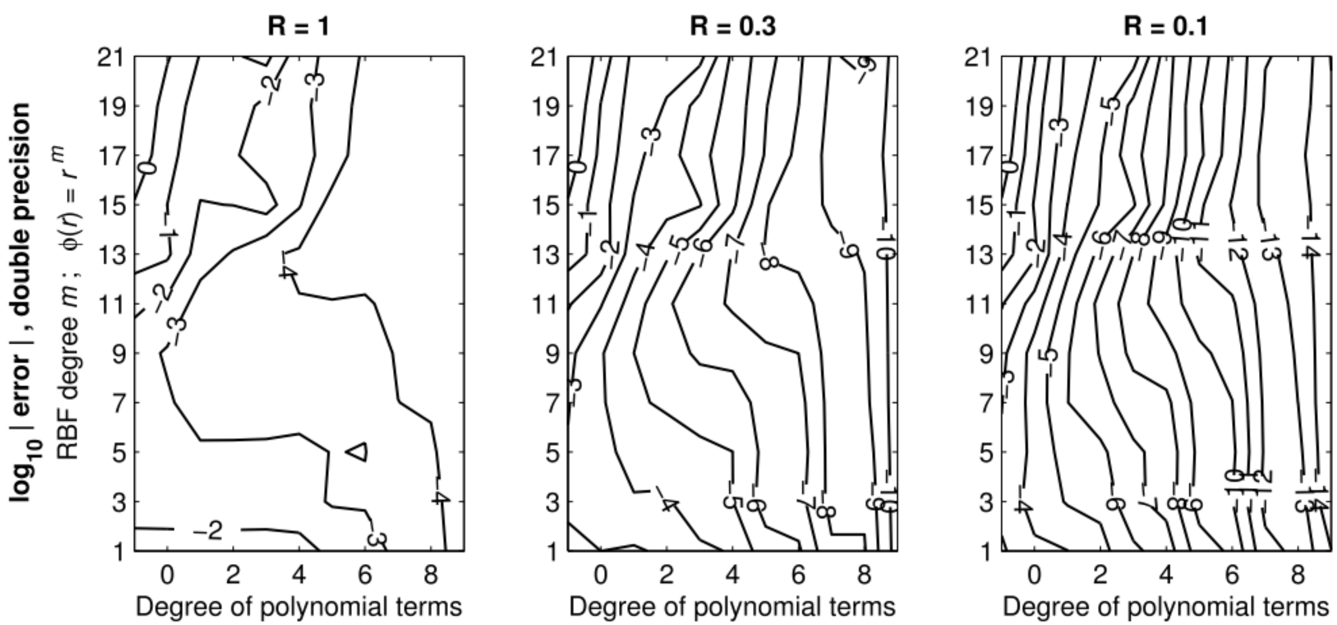
Illustrations from Fornberg and Flyer, SIAM book, 2015;
Additional description in Flyer, Fornberg, Bayona, Barnett, 2016.

Scatter n = 56 nodes to approximate, near the stencil center, three test functions of increasing smoothness

Note:

Under refinement, power in RBFs become irrelevant; accuracy order given by the polynomials



PHS + poly: Major differences between Global RBFs and RBF-FD

	Global RBFs	**RBF-FD:**
Total number of nodes	N	N
Number of nodes per stencil	N	$n << N$
Number of separate polynomials	1	N
Determines the order of convergence	PHS	poly
Main role of the PHS	Form the approximation	Improve conditioning
Main role of the ploynomials	Provide conditionally (positive or negative) definite operators	Form the approximation
Comments	Worsening conditioning for m large => keep m low (implying accuracy low)	Guideline: Bring up number of polynomial terms to around $n/2$ Under refinement (N increasing, n fixed), the PHS coefficients 'fade out'

Conditioning of linear system for creating PHS+poly RBF-FD stencils

Errors:

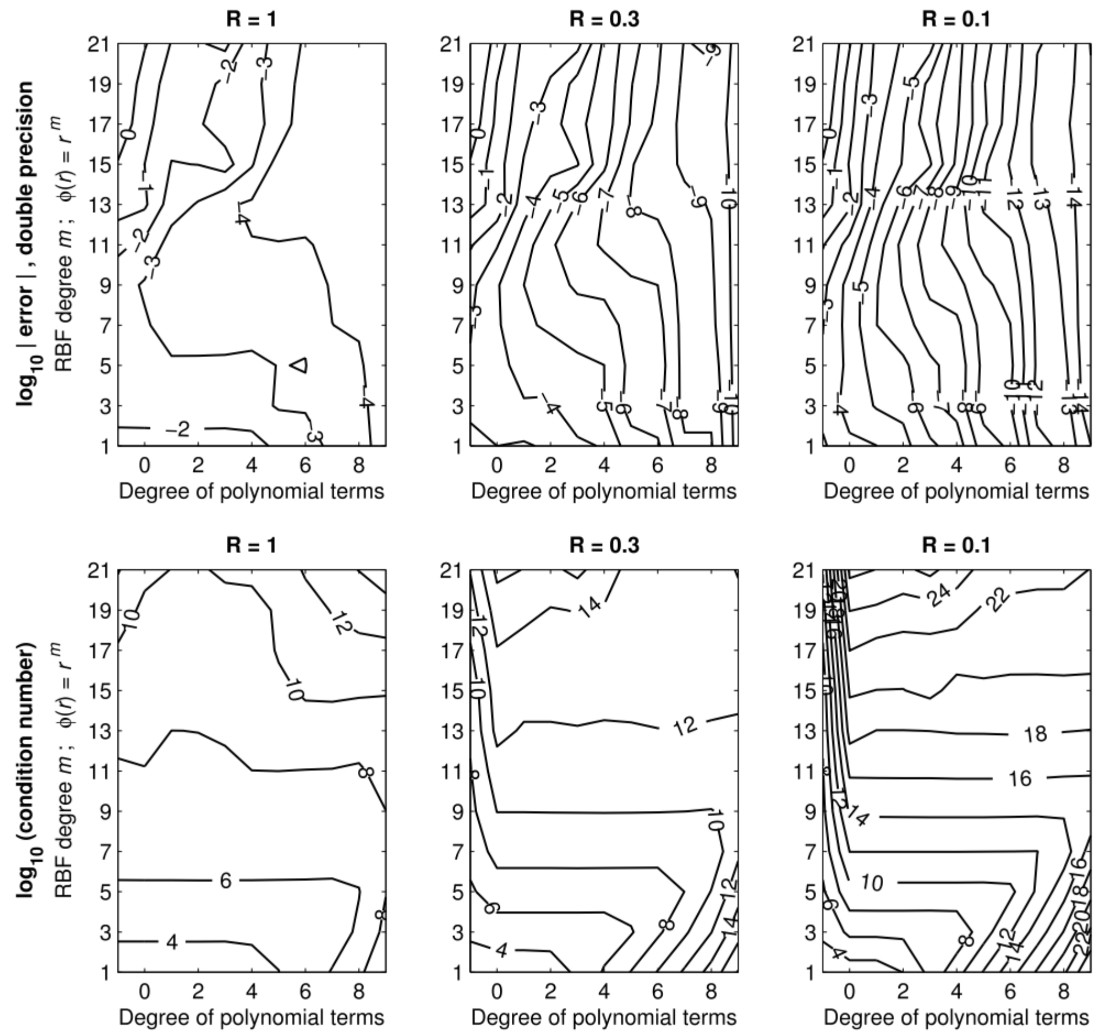
The figure to the right is identical to the one two slides ago:

Linear system conditioning:

Bottom row of figures suggest disasterously high condition numbers

This is an example where condition number is completely misleading

Simpler example still of same situation:



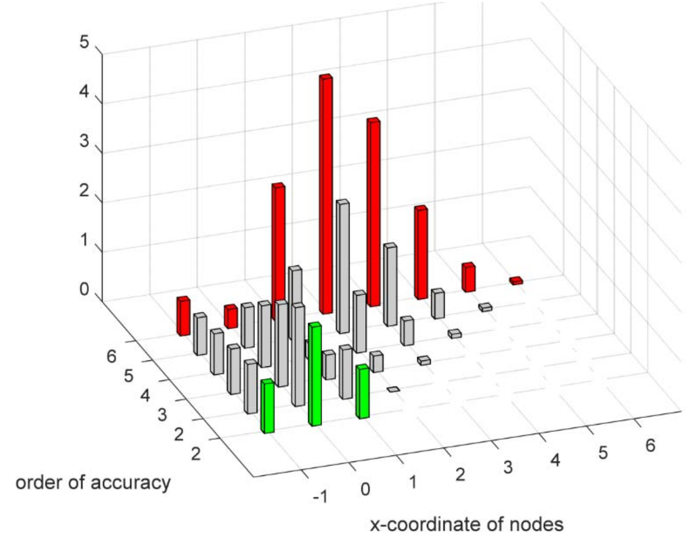
Example: $\begin{bmatrix} 10^{100} & & \\ & 10^{-100} & \\ & & 1 \end{bmatrix} \begin{bmatrix} \underline{x} \end{bmatrix} = \begin{bmatrix} \underline{b} \end{bmatrix}$; cond(A)=10^{200} . Nevertheless, NO loss in significant digits when solving by Gaussian elimination. Equivalent situation for present systems; illusionary issue only matter of scaling rows/columns.

Character of high order PHS+poly RBF-FD approximations near boundaries

(Observation described in Bayona, Flyer, Fornberg, Barnett, 2017)

Regular FD weights of increasing orders of accuracy, to approximate u''(0) one step in from a boundary:

x = -1	0	1	2	3	4	5	6
1.0000	-2.0000	1.0000					
1.0000	-2.0000	1.0000	0.0000				
0.9167	-1.6667	0.5000	0.3333	-0.0833			
0.8333	-1.2500	-0.3333	1.1667	-0.5000	0.0833		
0.7611	-0.8167	-1.4167	2.6111	-1.5833	0.5167	-0.0722	
0.7000	-0.3889	-2.7000	4.7500	-3.7222	1.8000	-0.5000	0.0611



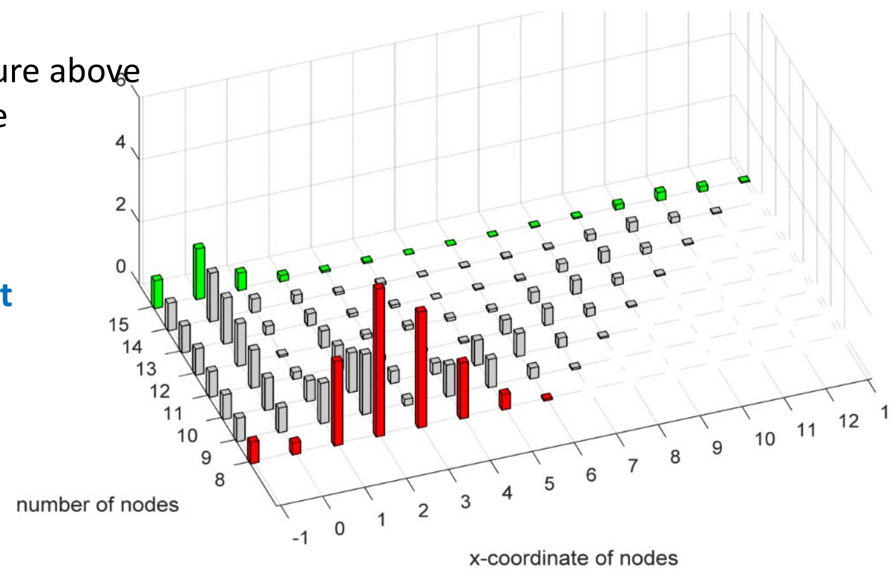
Last line shows complete loss of diagonal dominance for increasing accuracy
To the right – magnitude of the weights from 2nd to 6th order

PHS+poly generated weights, $\phi(r) = r^3$ with poly degree 7

The front row in figure to right matches the back row in the figure above
When adding still further nodes (making the stencils even more one-sided):

- **Accuracy remains locked to 6th order**
- **Stencil returns almost perfectly to perfect diagonally dominant case of [1 -2 1], centered at the node of interest**
- **Result can be deduced from optimization interpretation**

$$\underbrace{\min_{w} \frac{1}{2} w^T A w - w^T L\phi}_{\text{Keeps the } w \text{ small}} \quad \textbf{subject to} \quad \underbrace{P^T w = Lp}_{\text{Exact for polynomials}}$$

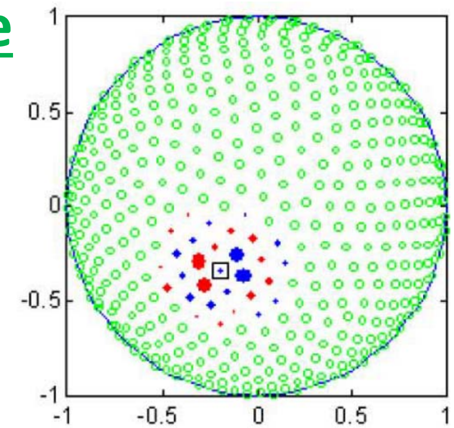


The result generalizes to more space dimensions, making PHS+poly very attractive for use in bounded domains.

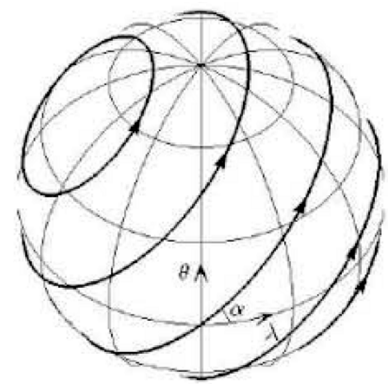
RBF-FD example: Convective flow around a sphere

(Fornberg and Lehto, 2011)

RBF-FD stencil illustration: N = 800 ME nodes, n = 30.
No surface-bound coordinate system used, implying no counterpart to pole singularities ⇒



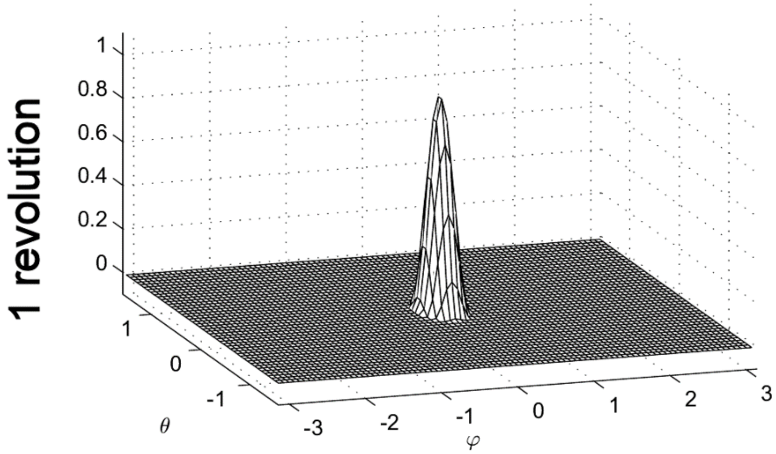
Test problem: Solid body rotation around a sphere
Initial condition: Cosine bell
Smooth (GA) RBFs, no polynomials. ⇒
N = 25,600, n = 74, RK4 in time

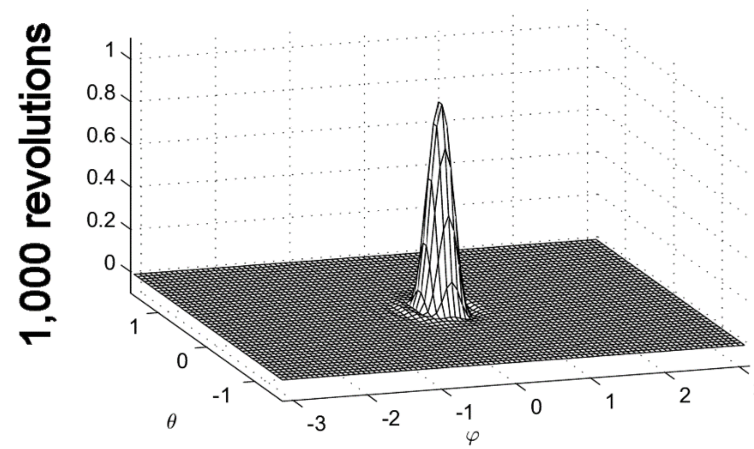


Key novelty: Stability achieved by use of hyperviscosity

Numerical solution:

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





RBF-FD Example: Seismic Exploration

Forward vs. Inverse Modeling

2-D vertical slice near Madagascar:

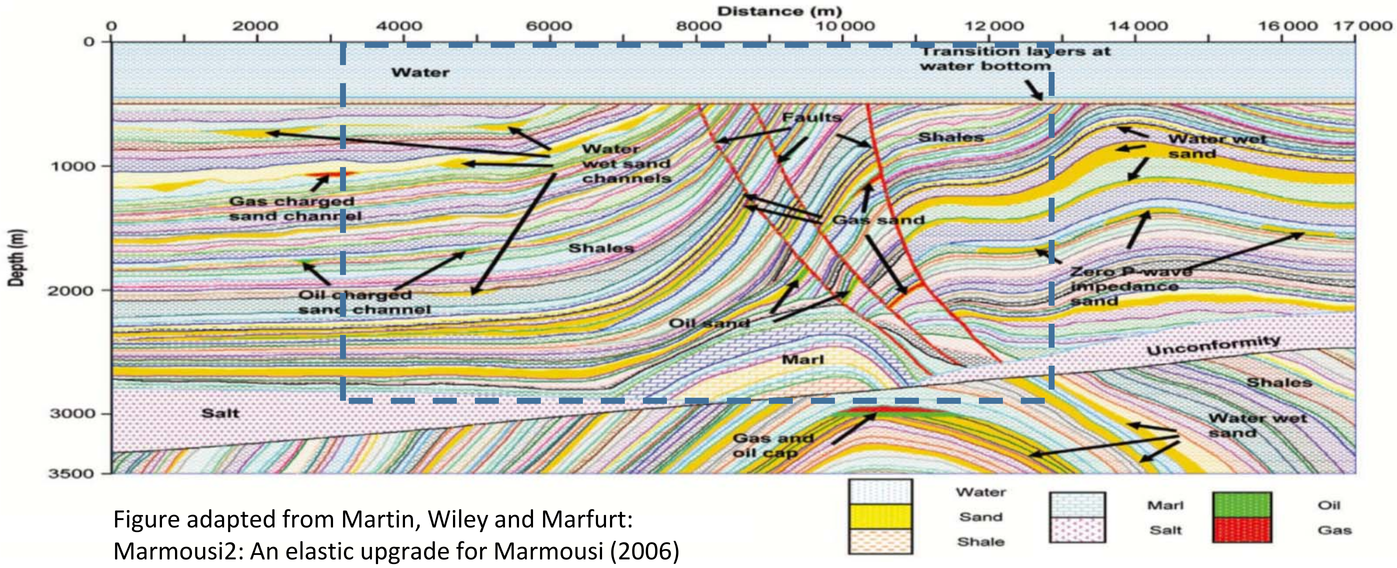


Figure adapted from Martin, Wiley and Marfurt: Marmousi2: An elastic upgrade for Marmousi (2006)

Region inside dashed rectangle simplified to form standardized Marmousi test problem

(shown on next slide)

Forward modeling

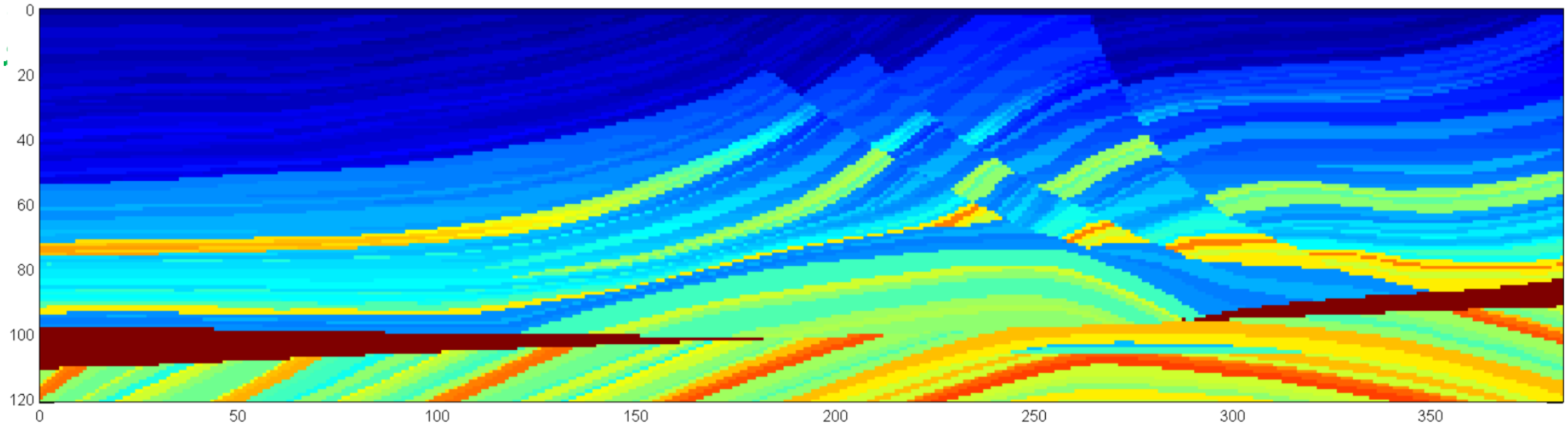
Assume subsurface structures known, then simulate the propagation of elastic waves

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

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>** (Martin, Fornberg, St-Cyr, 2015; Martin, Fornberg, 2017) - Distribute RBF-FD nodes to align with all interfaces (this alone suffices for 2^{nd} order) - Modify basis functions to analytically correct for interface conditions (RBF-FD/AC) High orders then possible also for curved interfaces

RBF-FD implementation

Regular Finite Differences (FD) can be used if:

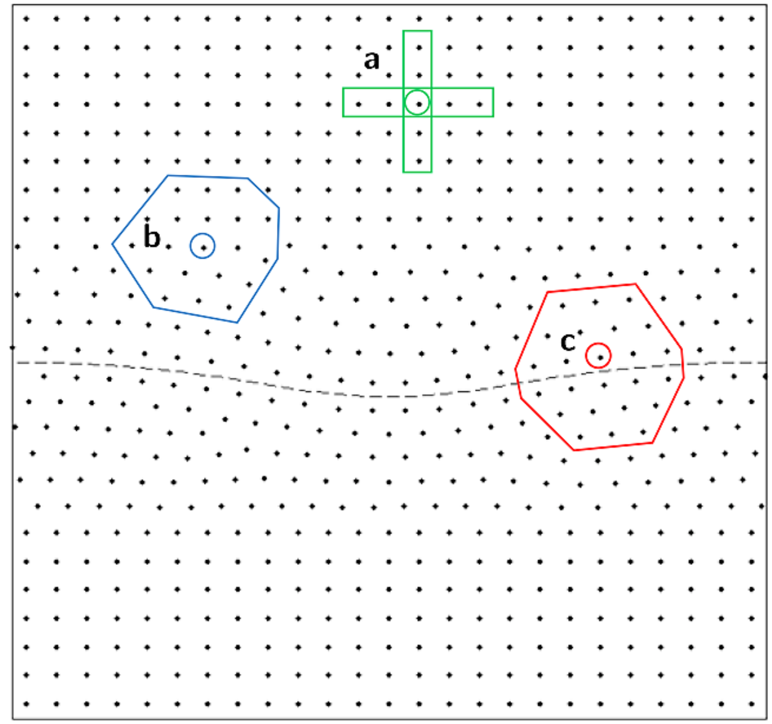
- Of high order of accuracy, and no near-by interfaces

However:

- Mapping spatial grids to align with interfaces is hopeless in realistic geometries
- Regular FD approximations are a flawed concept for mixed derivatives (such as $\frac{\partial}{\partial x}+\frac{\partial}{\partial y}$)

So instead use RBF-FD:

- Align nodes locally to each interface $\Rightarrow$
- 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).



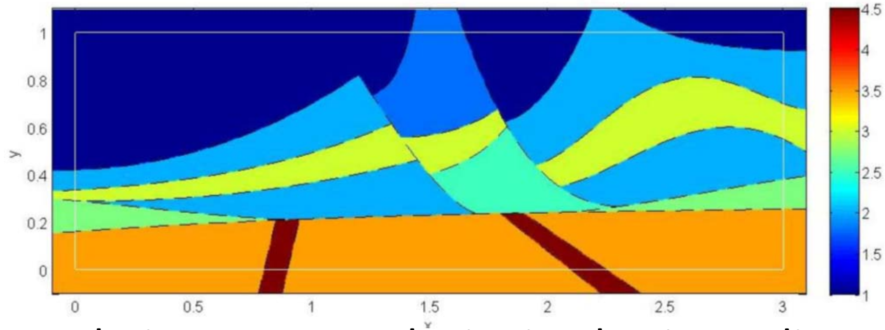
Case (b):

- Smooth RBFs work fine, as does PHS+poly. In the latter case, the poly ‘take over’ under refinement.

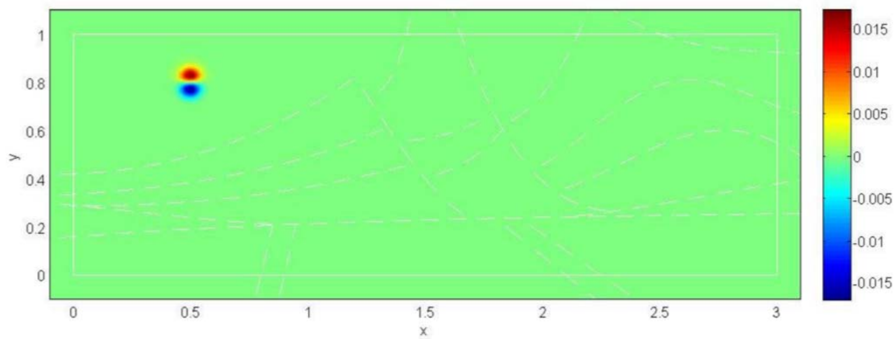
Case (c):

- Here we need RBF+poly. We can here alter the polynomials to build in the the analytic ‘kink’ information.
- In this case, the polynomials again ‘take over’ under refinement, now both at interfaces and in smooth regions, ensuring overall high orders of accuracy (with the RBFs still assisting with stability).

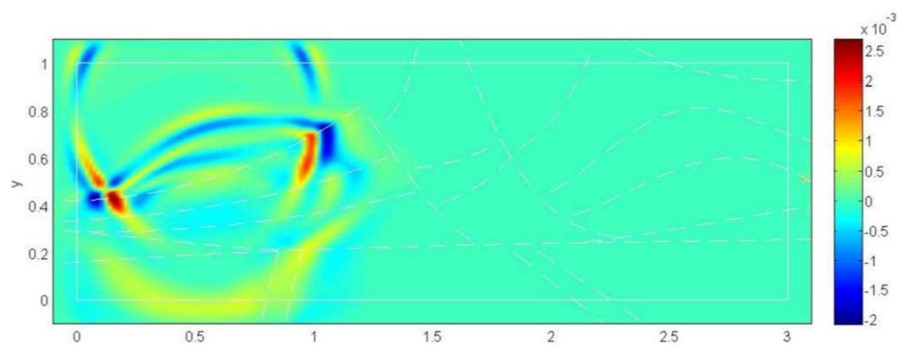
‘Mini-Marmousi’ test case



Relative *p*-wave velocity in elastic medium

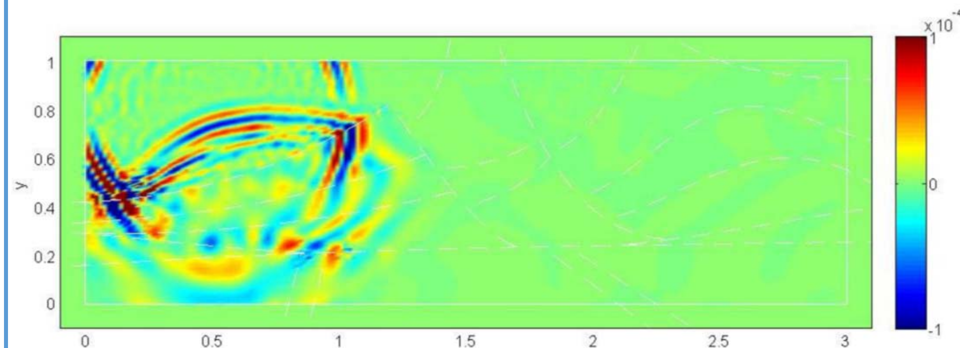


Initial condition for v at time $t = 0$

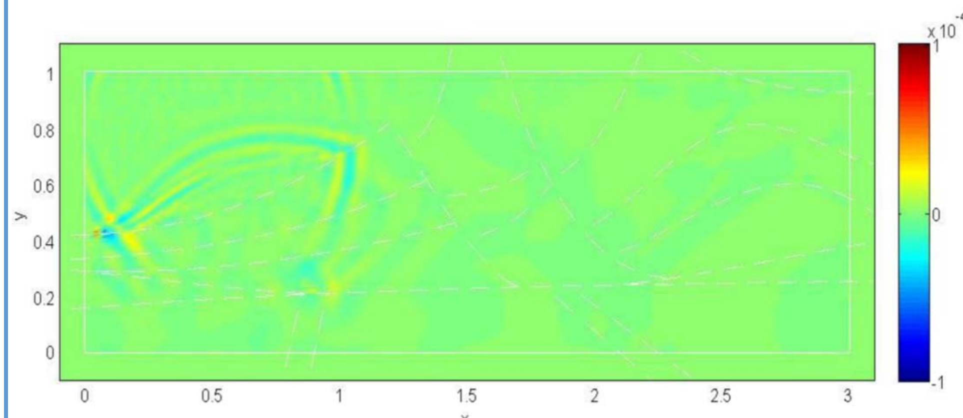


Accurate solution for v at time $t = 0.3$

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

RBF-FD Example: Calculate weights for numerical integration over a sphere

(Reeger, Fornberg, 2016)

Algorithm steps:

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

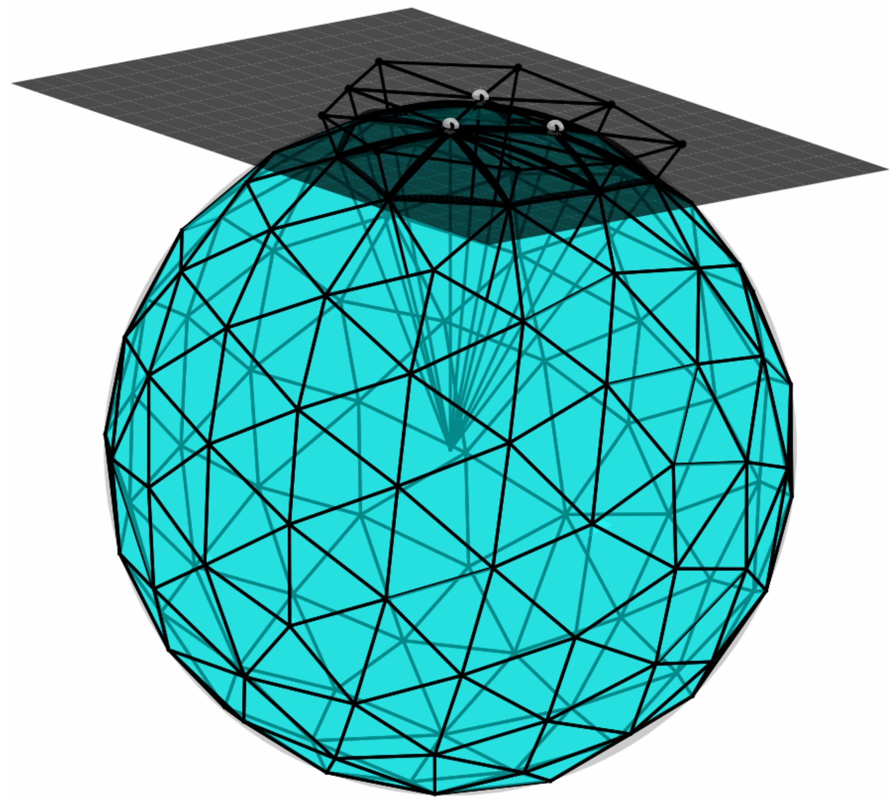
2. For each surface triangle, project it together with some nearby nodes to a tangent plane

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

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

With PHS+poly, RHS of linear system available in closed form.

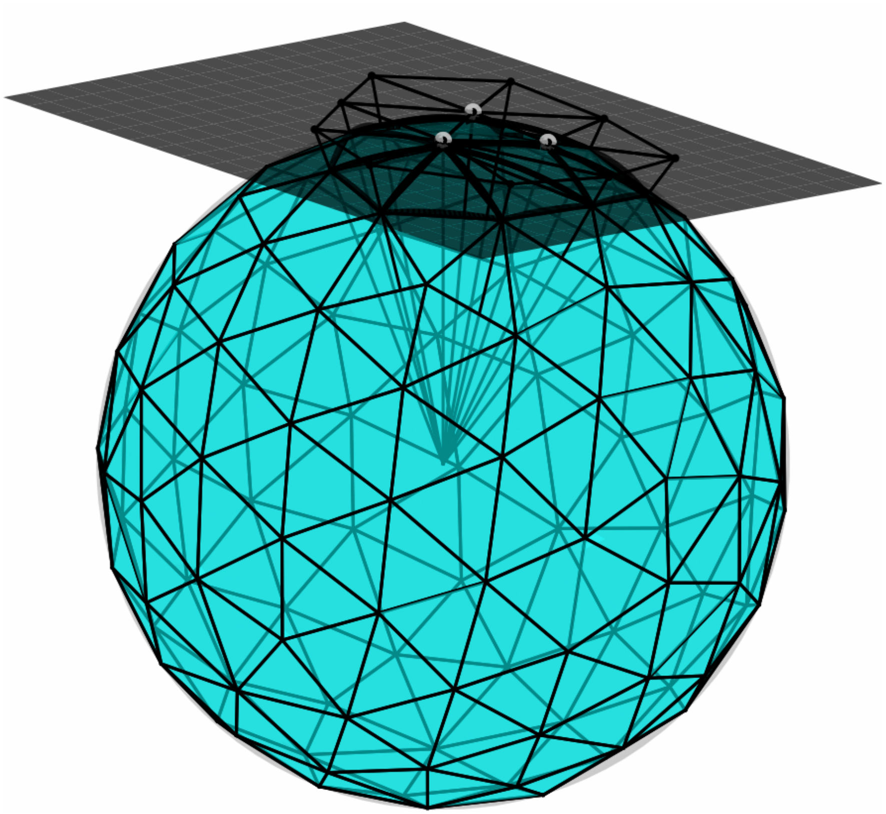
Concept illustration:



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

   Very simple, explicit conversion formula available that preserves accuracy order for any arbitrary curved smooth surface.

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



- Resulting accuracy order will match that of the supporting polynomials in the PHS+poly planar approximation (typically implemented to $O(h^7)$).
- Computational costs:
  - $O(N \log N)$ operations for N nodes for *kd*-tree to find 'nearest neighbors',
  - $O(N)$ operations to find all N weights; this task furthermore 'embarrassingly parallel'.
- Generalizations beyond sphere case:
  - Smooth closed surfaces (Reeger, Fornberg, Watts, 2016)
  - Curved surfaces with boundaries (Reeger, Fornberg, 2017)

Some conclusions

Discussed here:

- 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 also local refinement

- For RBF-FD, PHS + poly is often the preferred choice: High accuracy, no shape parameter, well conditioned, and excellent boundary accuracy (even for one-sided stencils)

- RBF-FD highly effective not only for PDEs, but also for quadrature over smooth surfaces

Additionally: (Natasha Flyer and collaborators)

- RBF-FD methods compete very favorably against *all* previous methods on a large number of applications, demonstrated especially in the geosciences

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

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.