Radial Basis Functions:

Freedom from meshes in scientific computations

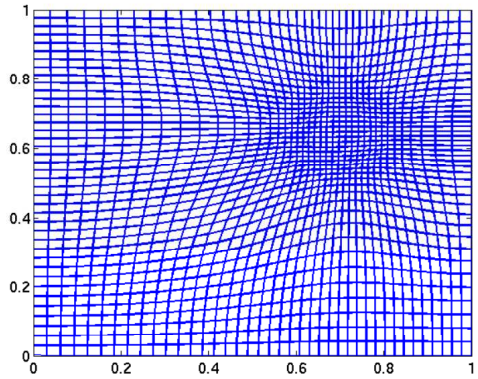
Bengt Fornberg University of Colorado at Boulder,
Department of Applied Mathematics

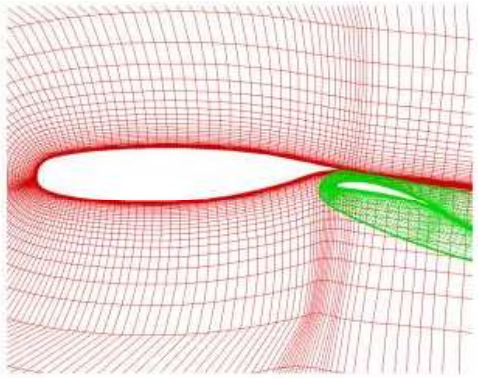
in collaboration with

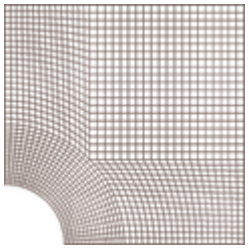
Natasha Flyer NCAR: National Center for Atmospheric Research
IMAGe: Institute for Mathematics Applied to the Geosciences

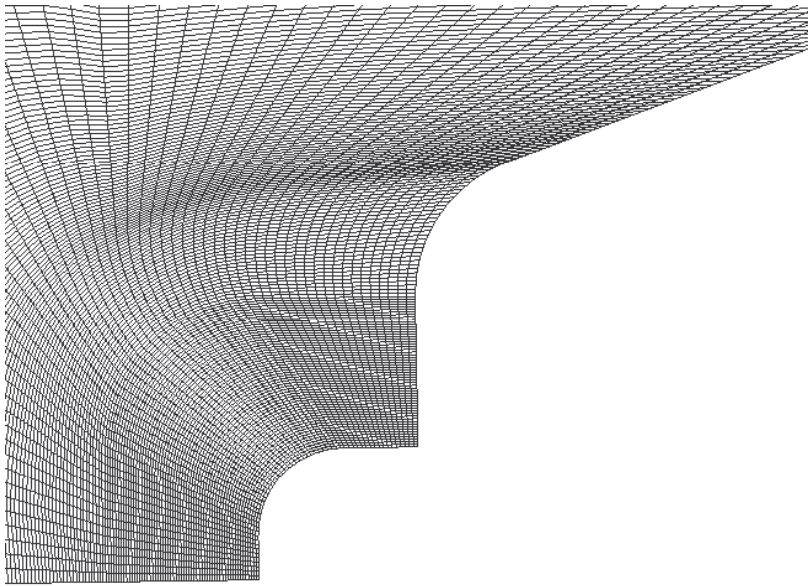
Examples of Structured Meshes in the plane

(from the web)

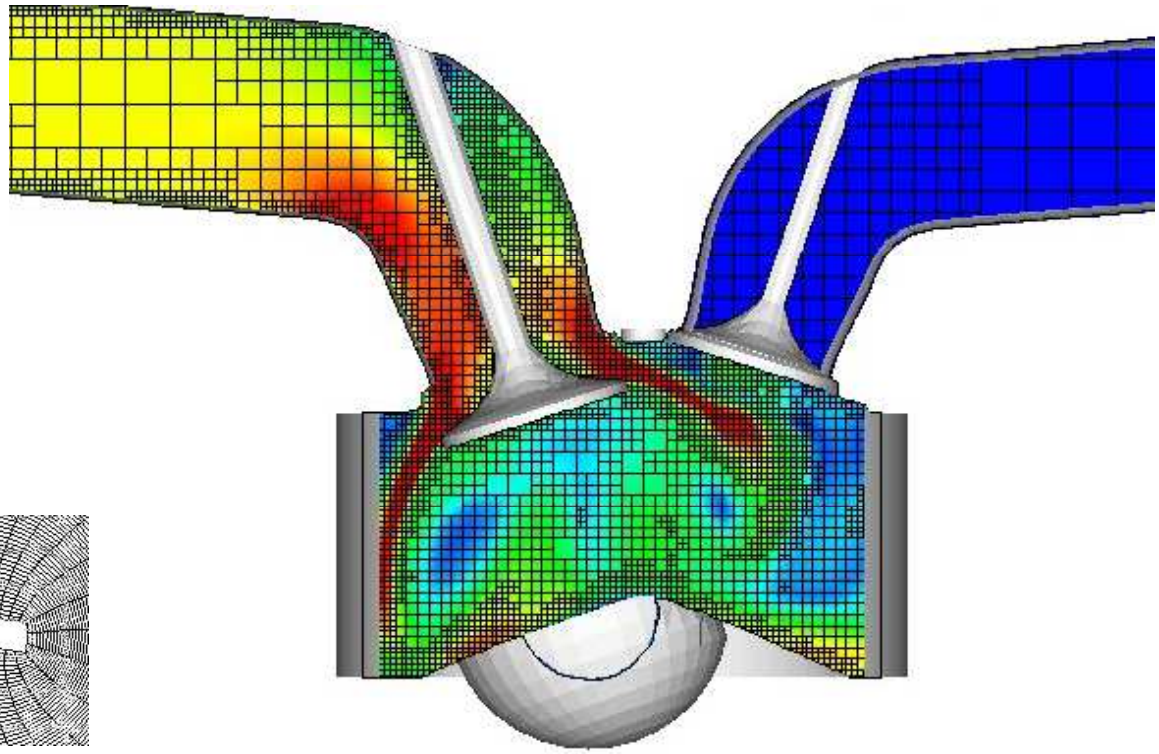


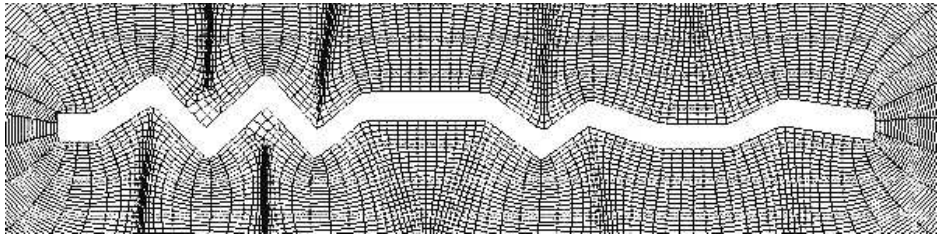






... and in 3D

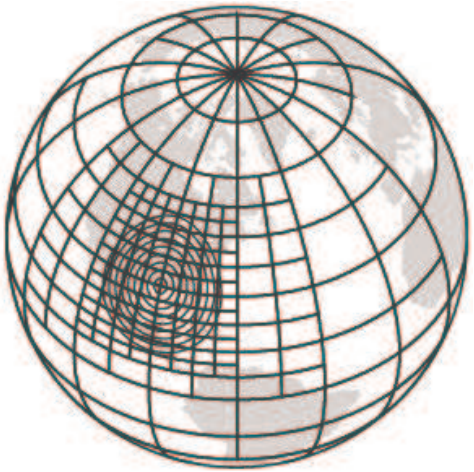




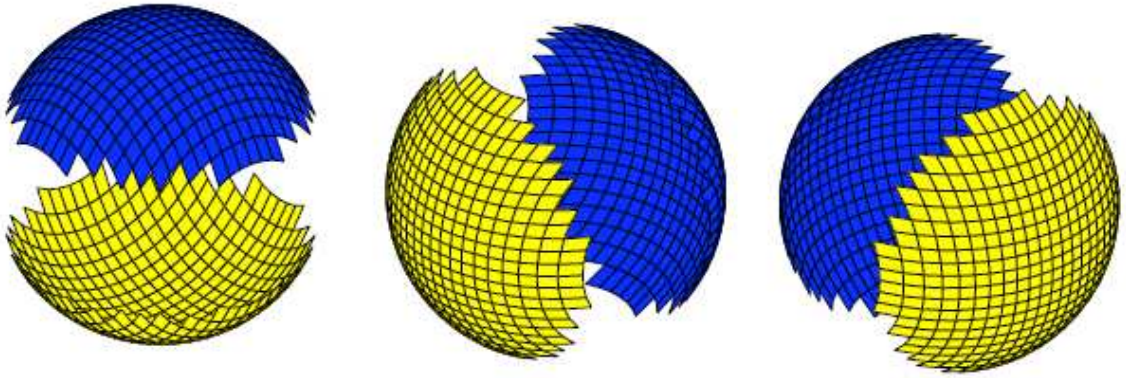
Examples of Structured Meshes on the surface of a sphere

(from the web)

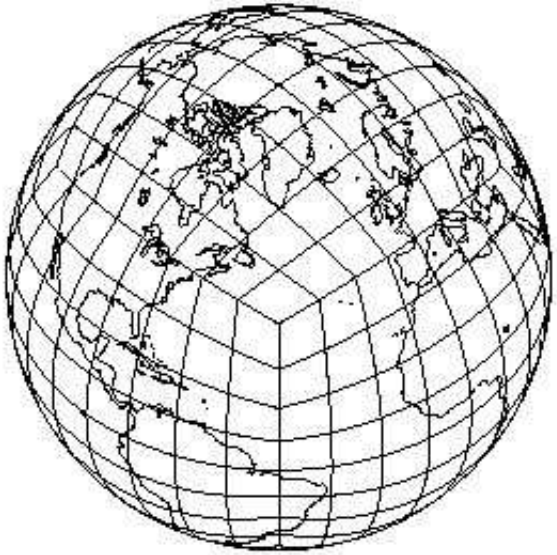
Lat-Long grids, with refinement

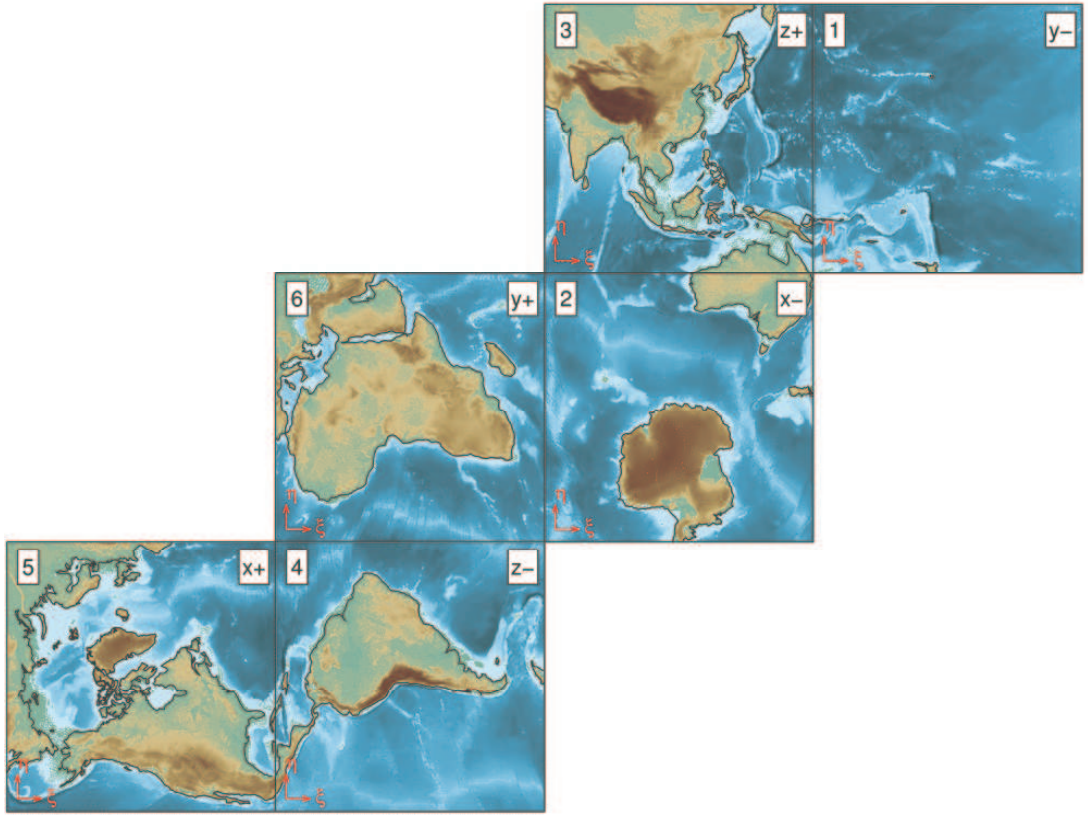


Spherical 'caps'; generalizes to arbitrary smooth surfaces



'Cubed sphere'

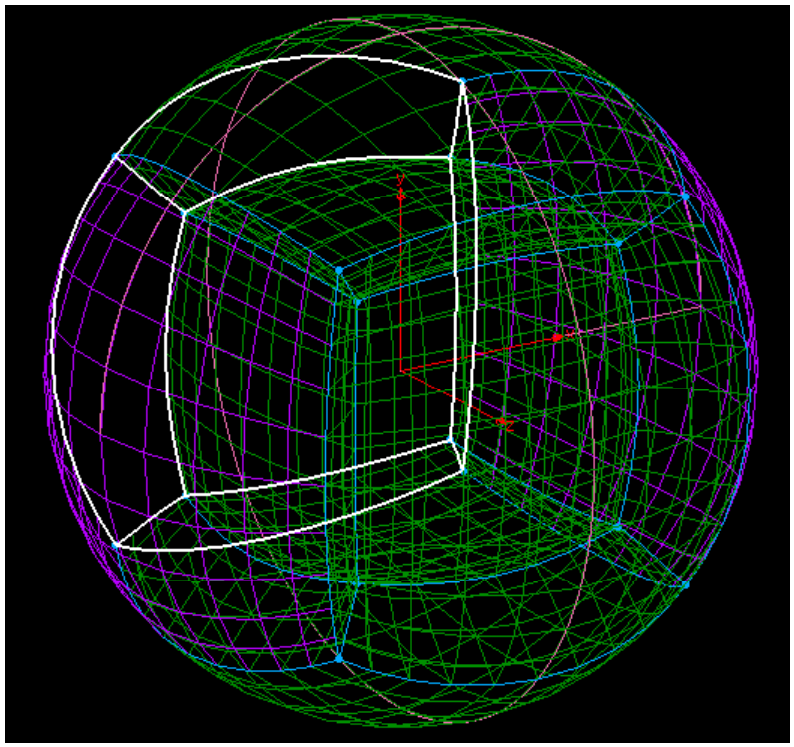




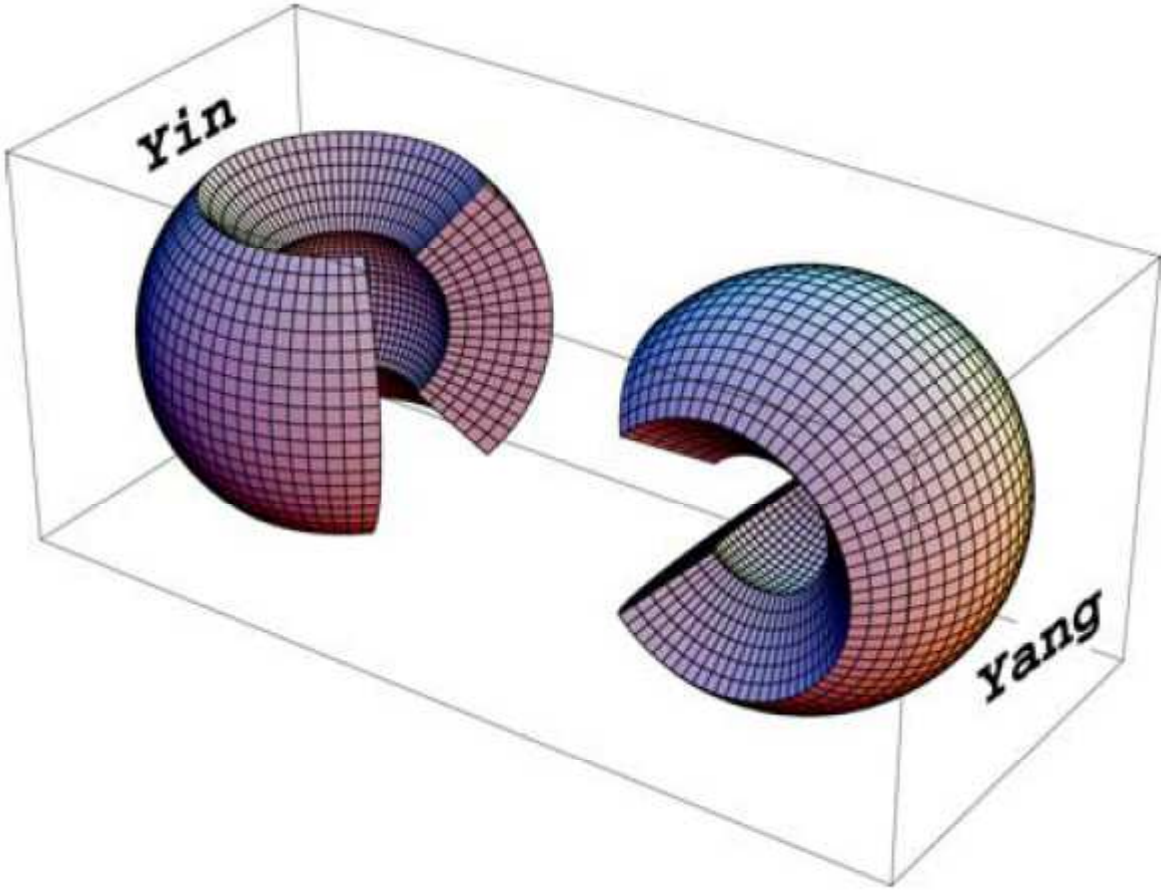
Examples of Structured Meshes within a sphere / spherical shell

(from the web)

'Cubed sphere'

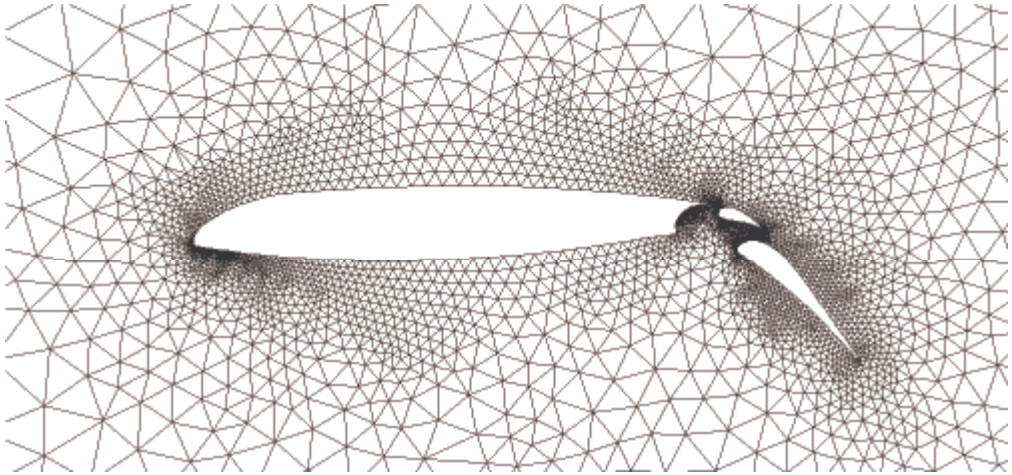


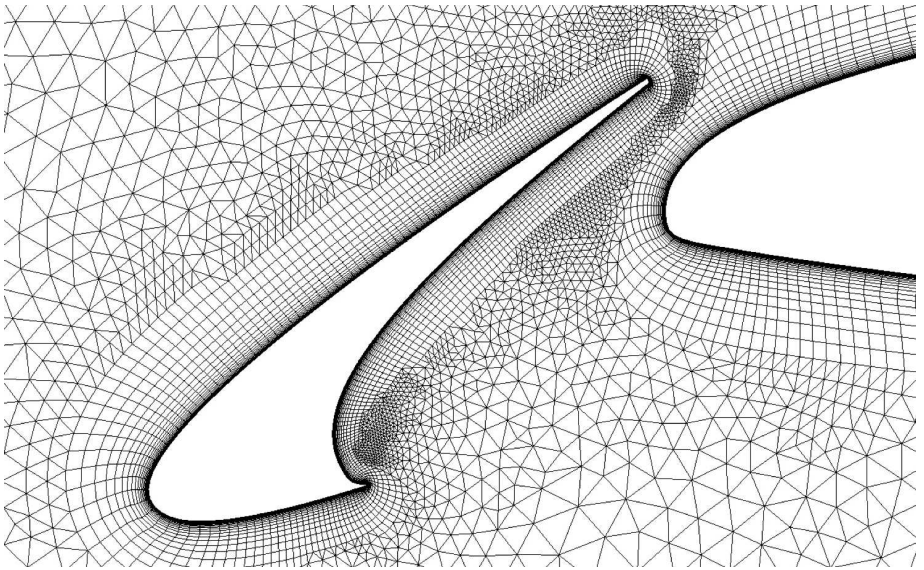
Yin-Yang grid



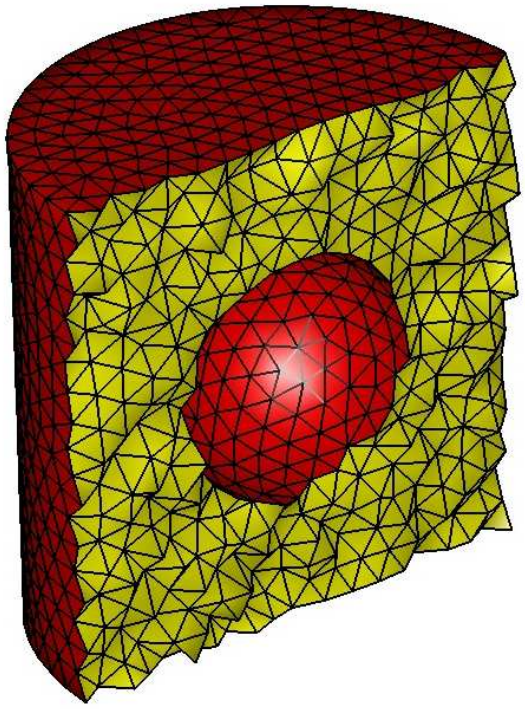
Examples of Unstructured Meshes in the plane

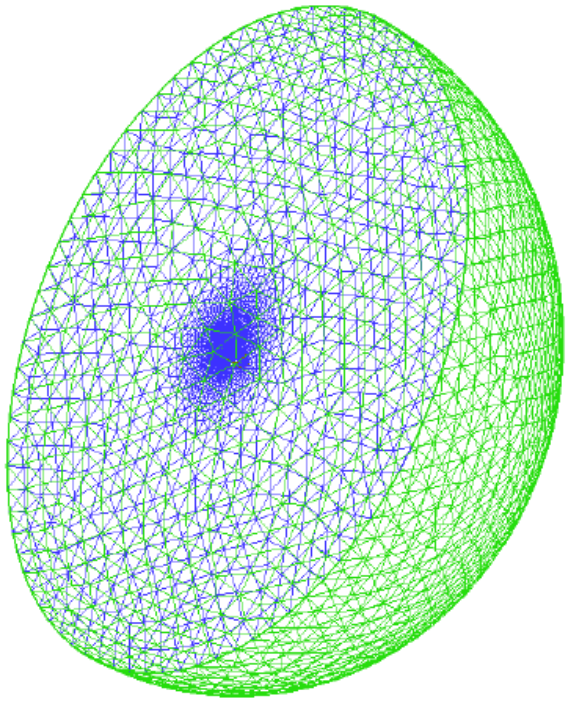
(from the web)



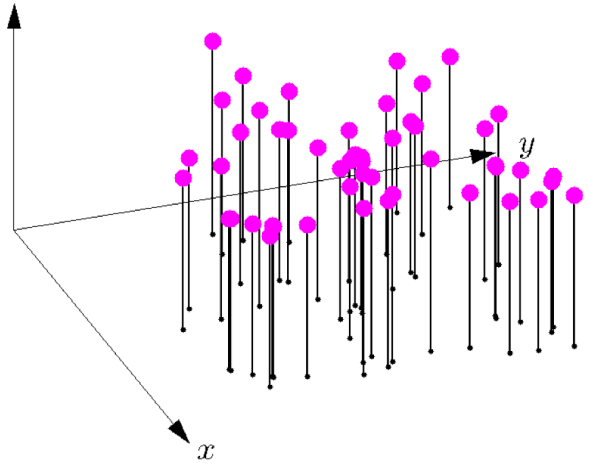


... and within / surrounding a sphere

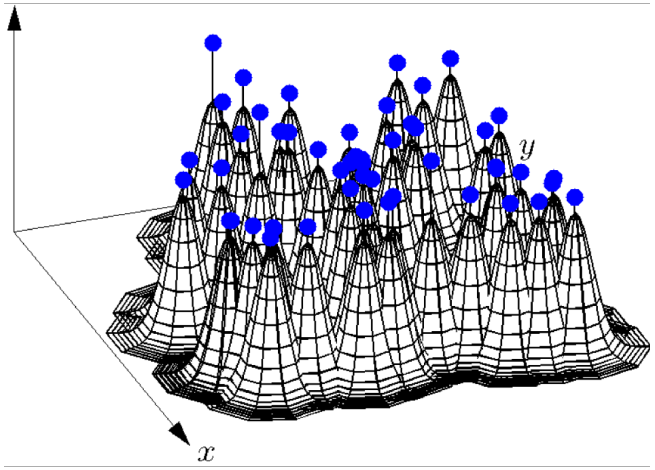




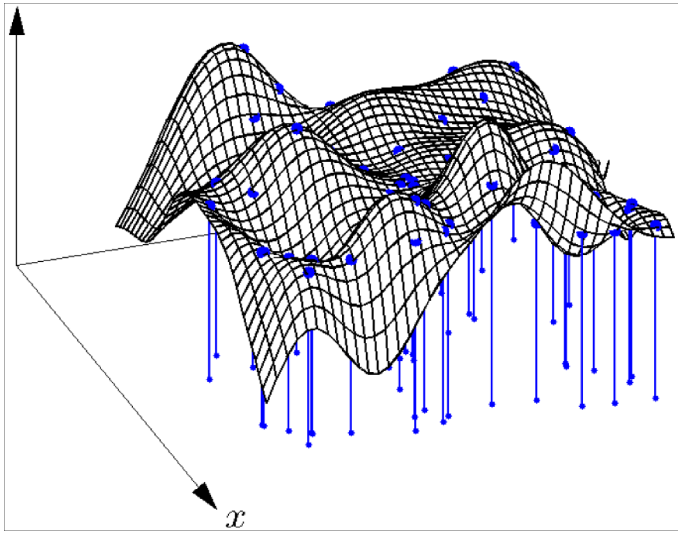
RBF idea, In pictures:



Scattered data within a 2-D region



Radial basis functions - here 'rotated Gaussians'

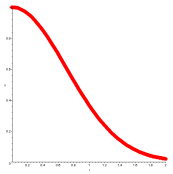


Linear combination of the basis functions that fits all the data

Many types of RBFs are available; Three examples:

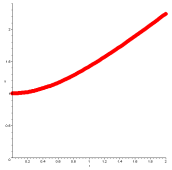
Gaussian (GA)

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



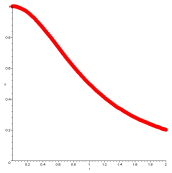
Multiquadric (MQ)

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



Inverse Quadratic (IQ)

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



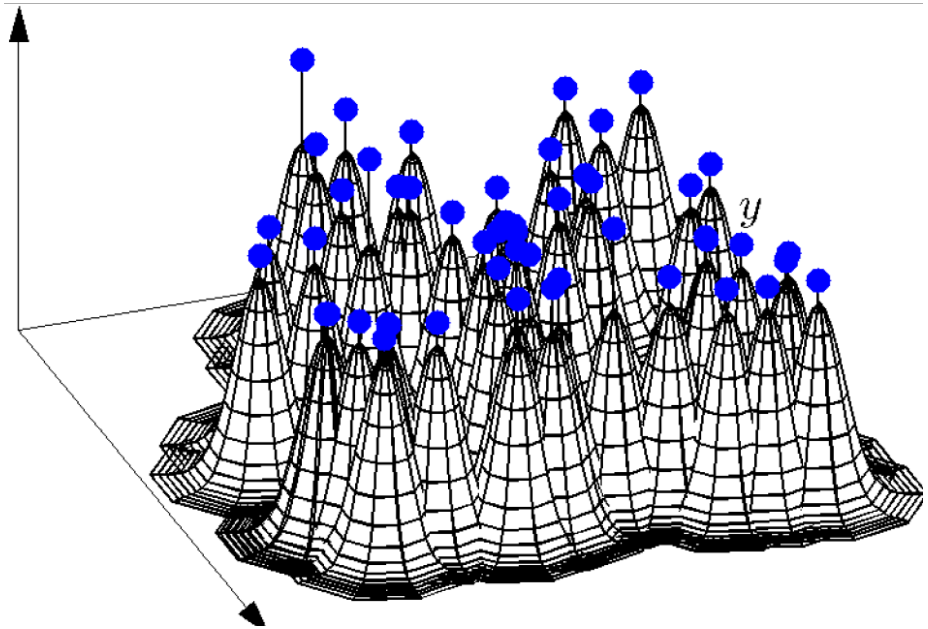
Infinitely smooth RBFs give spectral accuracy for interpolation and derivative approximations

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 :

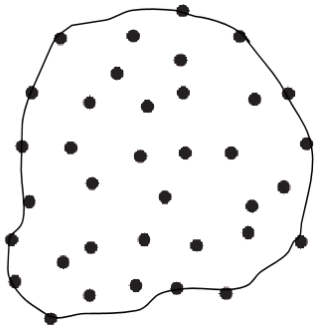


$$\begin{bmatrix} \phi(\|\underline{x}_1 - \underline{x}_1\|) & \phi(\|\underline{x}_1 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_1 - \underline{x}_N\|) \\ \phi(\|\underline{x}_2 - \underline{x}_1\|) & \phi(\|\underline{x}_2 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_2 - \underline{x}_N\|)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

What is 'special' about expanding in RBFs?

NO set of pre-specified basis functions (say based on Fourier, Chebyshev, Spherical Harmonics, etc.) can guarantee a nonsingular system in the case of scattered nodes

Scattered nodes



Interpolant $s(\underline{x}) = \sum_{k=0}^{N} c_k \, \Psi_k(\underline{x})$

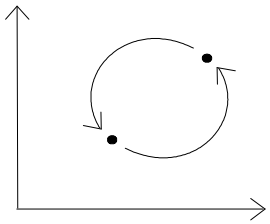
System that determines the expansion coefficients:

$$\begin{bmatrix} \Psi_0(\underline{x}_0) & \Psi_1(\underline{x}_0) & \cdots & \Psi_N(\underline{x}_0) \\ \Psi_0(\underline{x}_1) & \Psi_1(\underline{x}_1) & \cdots & \Psi_N(\underline{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\underline{x}_N) & \Psi_1(\underline{x}_N) & \cdots & \Psi_N(\underline{x}_N) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{bmatrix}$$

Move two nodes so they exchange locations:

Two rows become interchanged, determinant changes sign

$\Rightarrow$ determinant is zero somewhere along the way.



What is different with using RBFs?

Two rows -
and also two columns -
become interchanged.

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

Key theorem: This system can never be singular (for most standard RBF types, such as GA, MQ, IQ), no matter how any number of nodes are scattered in any number of dimensions.

Some history:

Global RBF methods:

1970 Invention of RBFs (for an application in cartography)

Some other key dates:

1940 Unconditional non-singularity shown for many types of radial functions
1990 First application of RBFs to numerical solutions of PDEs

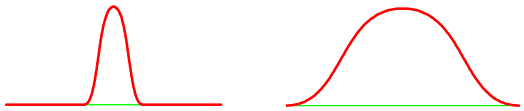
2002 Flat RBF limit exists: all 'classical' pseudospectral methods can be seen as RBF special cases
2004 First RBF algorithm that remains numerically stable in the flat basis function limit
2007 First application of RBFs to large-scale geophysical test problems
2009 A new physical phenomenon (a certain mantle convection instability) initially discovered in an RBF calculation

RBF generated finite difference (RBF-FD) methods:

2000 First proposals to use RBFs instead of polynomials to create scattered-node FD formulas
2003 Three independent major studies on RBF-FD methods
2008 Beginning of RBF-FD implementations for CFD (computational fluid dynamics)
2011 Numerical stability achieved also for very long time integration of purely convective equations
2012 First use of RBF-FD for large-scale geophysical test problems

Some technical issues to be concerned about when using RBFs

- **Value of 'shape parameter'** ε For example:



ε large vs. ε small

- **Computational cost**

  Reason for concern: Might lead to large and full rather than to large and sparse matrices.
  RBF-FD (RBF-generated finite differences) use 'local' stencils.

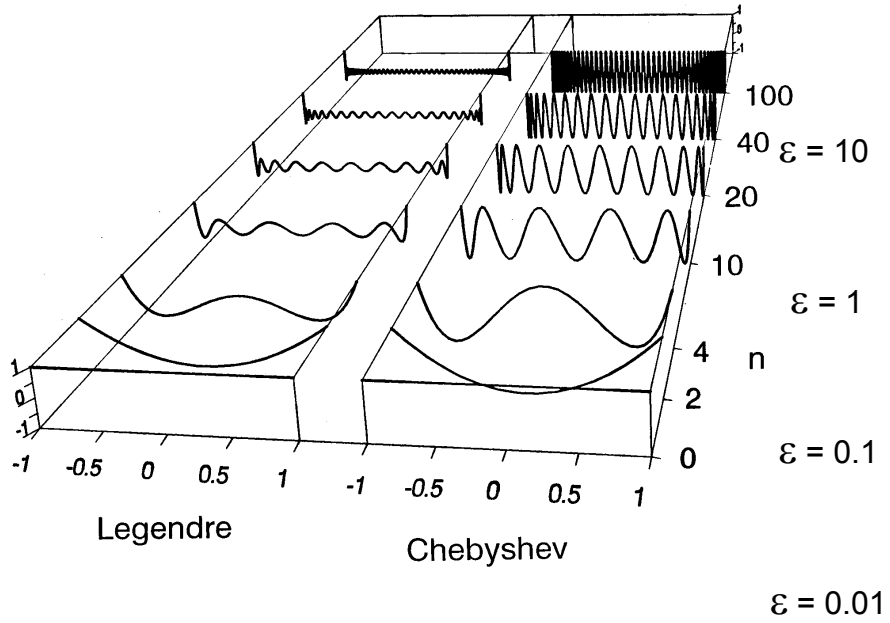
- **Numerical stability** - in connection with (explicit) time stepping of convection-type PDEs.

- **Accuracy at / near domain boundaries**

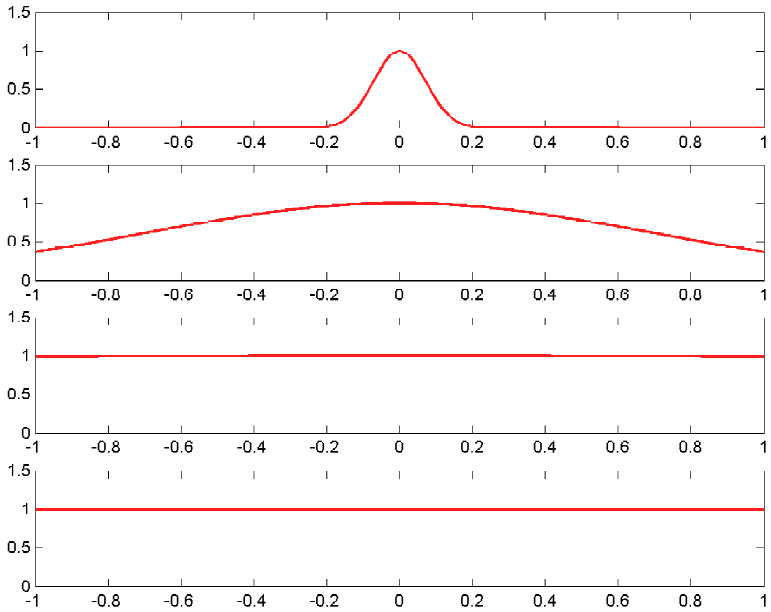
- **Opportunities for implementation on distributed computer systems.**

Value of ε: Numerical conditioning, and the flat RBF limit ($\varepsilon \to 0$)

Classical basis functions are usually highly oscillatory



RBFs are translates of one single function - here $\phi(r) = e^{-(\varepsilon r)^2}$



In case of 41 scattered nodes in 1-D: $\text{cond}(A) = O(\varepsilon^{-80})$, $\det(A) = O(\varepsilon^{1640})$.

2-D: $\text{cond}(A) = O(\varepsilon^{-16})$, $\det(A) = O(\varepsilon^{416})$.

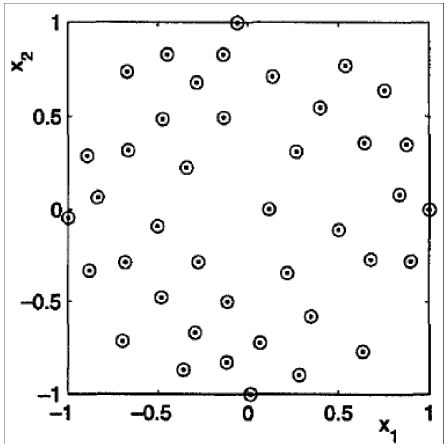
Exact formulas available for any number of nodes in any number of dimensions
(Fornberg and Zuev, 2007)

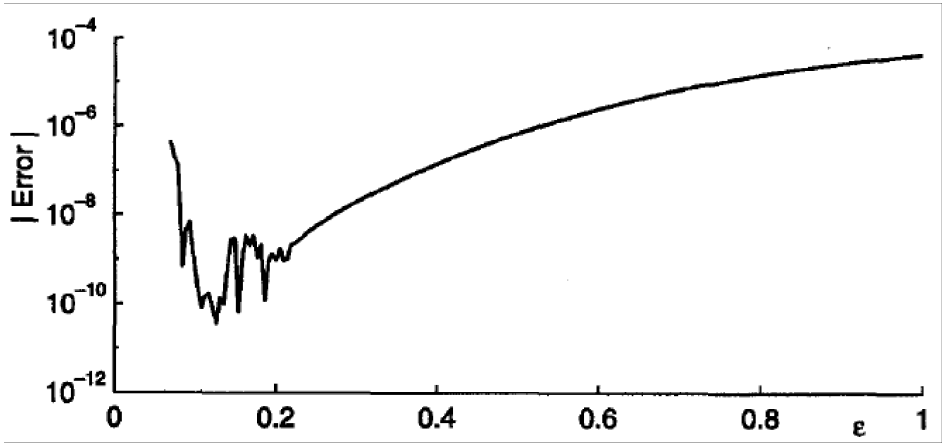
Extreme ill-conditioning typical as $\varepsilon \to 0$.

Why are flat (or near-flat) RBFs interesting ?

- Intriguing error trends as $\varepsilon \to 0$

  'Toy-problem' example: 41 node MQ interpolation of $f(x_1, x_2) = \dfrac{59}{67 + (x_1 + \frac{1}{7})^2 + (x_2 - \frac{1}{11})}$





- RBF interpolant in 1-D reduces to Lagrange's interpolation polynomial
  (Driscoll and Fornberg, 2002)

- The $\varepsilon \to 0$ limit reduces to 'classical' PS methods if used on tensor type grids. Best accuracy often obtained for non-zero ε.

Solving $A\underline{\lambda} = \underline{f}$ followed by evaluating $s(\underline{x}, \varepsilon) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|)$ is merely an unstable algorithm for a stable problem

Numerical computations for small values of ε (near-flat RBFs)

It is possible to create algorithms that completely bypass ill-conditioning all the way into $\varepsilon \to 0$ limit, while using only standard precision arithmetic:

Concept: Find a computational path from $\underline{f}$ to $s(\underline{x},\varepsilon)$ that does not go via the ill-conditioned expansion coefficients $\underline{\lambda}$.

- **Contour-Padé algorithm** First algorithm of its kind; established that the concept is possible
  Based on contour integration in a complex ε-plane.
  Limited to relatively small N-values
  First version (Fornberg and Wright, 2004).

- **RBF-QR method** Initially developed for nodes scattered over the surface of a sphere
  No limit on N ; cost about 5 - 10 times that of RBF-Direct
  Original version (for nodes on sphere) (Fornberg and Piret, 2007).
  Versions for 1-D, 2-D, and 3-D (Fornberg, Larsson, and Flyer, 2011).

- **RBF-GA method** Exploits some incomplete gamma function identities
  Works in any number of dimensions (Fornberg. Lehto, and Powell, 2013).

Probably many more stable algorithms to come.

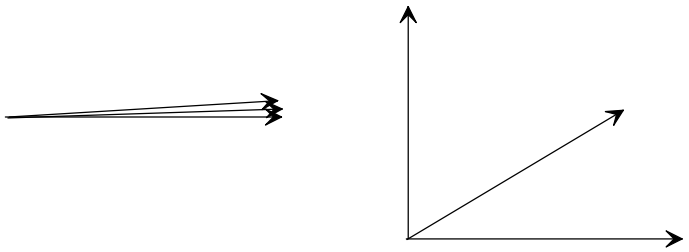
By using RBFs, one completely gives up orthogonality, but keeps spectral accuracy, and gains total geometric flexibility,

Then, by using a stable algorithm, one keeps the benefits and regains good numerical conditioning.

The concept for the RBF-QR method

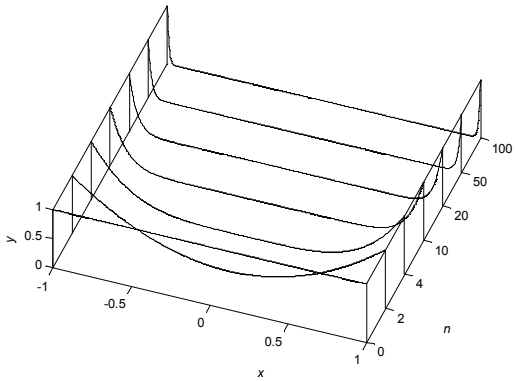
Recognize that the existence of an ill-conditioned **basis** does not imply that the spanned **space** is bad.

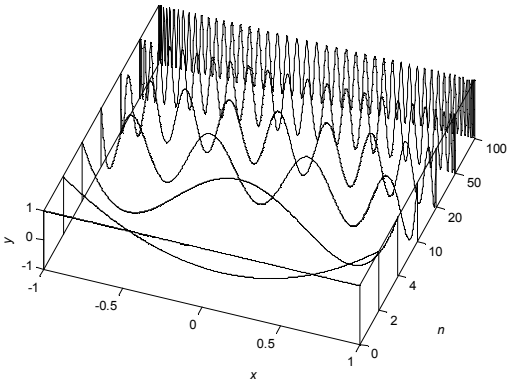
Ex. 1: 3-D space



Bad Basis Good Basis

Ex. 2: Polynomials of degree ≤ 100

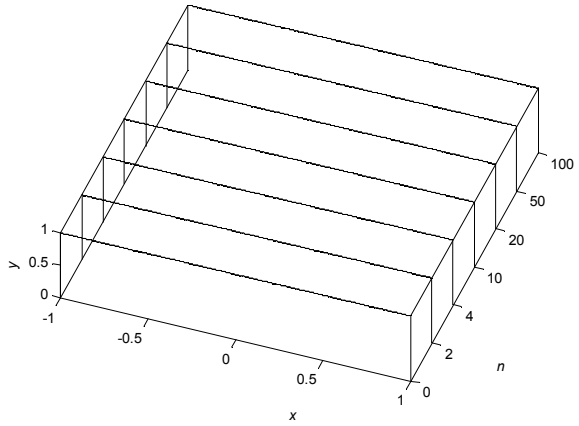




x^n , $n = 0, 1, \ldots, 100$
Bad Basis

$T_n(x)$, $n = 0, 1, \ldots, 100$
Good Basis

Ex. 3: Space spanned by RBFs in their flat $\varepsilon \to 0$ limit



Clearly a Bad Basis

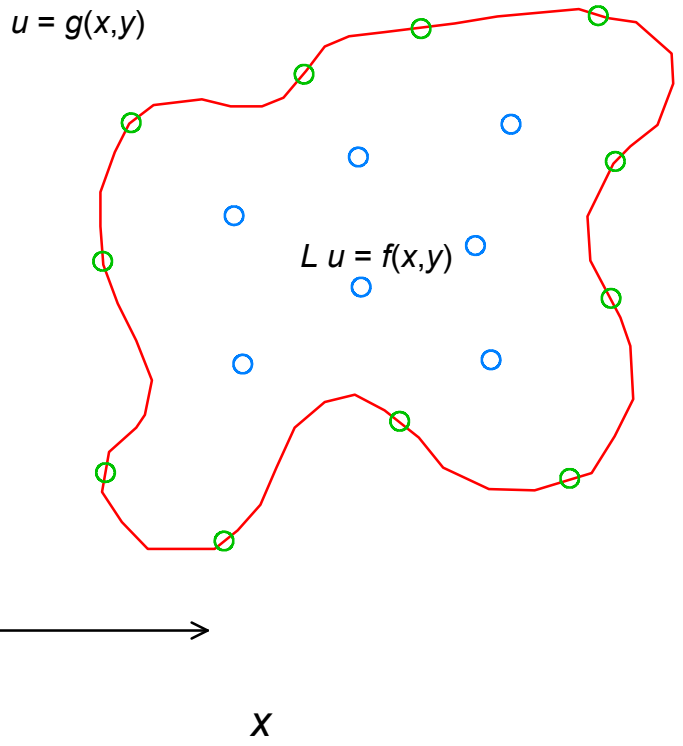
- The spanned space turns out to be excellent for computational work - just the basis that is bad.
- Is there any Good Basis in exactly the same space?
- RBF-QR finds such a basis through some analytical expansions, leading to numerical steps that all remain completely stable even in the flat basis function limit.

Poisson's equation: Simplicity of RBF implementation Illustration of benefits with small ε-regime

(Larsson and Fornberg, 2003)

On Boundary: $u = g(x, y)$

In interior: $L\,u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$



Direct collocation (Method by Kansa):

Let $u(\underline{x}) = \sum_{j=1}^{n} \lambda_j \phi(||\underline{x} - \underline{x}_j||)$

Then:

$$\begin{bmatrix} \phi(||\underline{x} - \underline{x}_j||)|_{x=x_i} \\ - \; - \; - \; - \; - \; - \; - \; - \\ L\phi(||\underline{x} - \underline{x}_j||)|_{x=x_i} \end{bmatrix} \begin{bmatrix} \\ \lambda_j \\ \\ \end{bmatrix} = \begin{bmatrix} g \\ - \\ f \end{bmatrix}$$

Key features:

- Spectral accuracy (if smooth RBFs)
- Completely arbitrary geometry
- Code maybe 15 lines... (in Matlab, if using a direct solver)

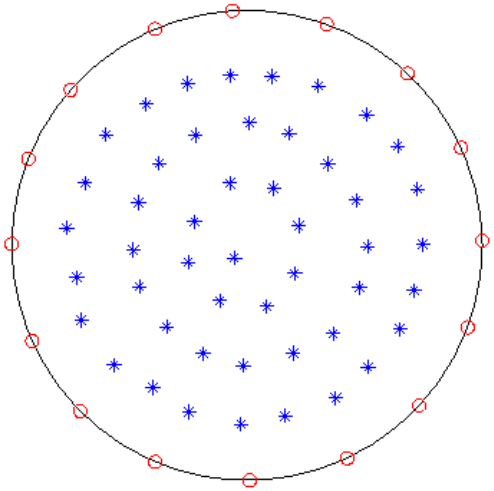
Numerical tests for Poisson's equation

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x,y) & \text{in interior,} \\ u = g(x,y) & \text{on boundary} \end{cases}$$

The RHS functions are chosen so that $u(x,y) = \dfrac{100}{100 + (x-0.2)^2 + 2y^2}$

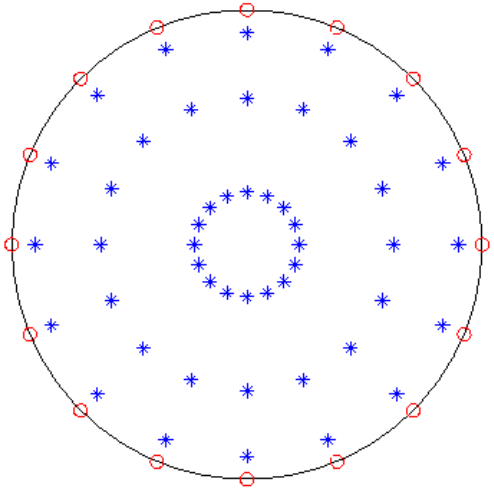
Direct RBF collocation

Test with different $\phi(r)$ and ε



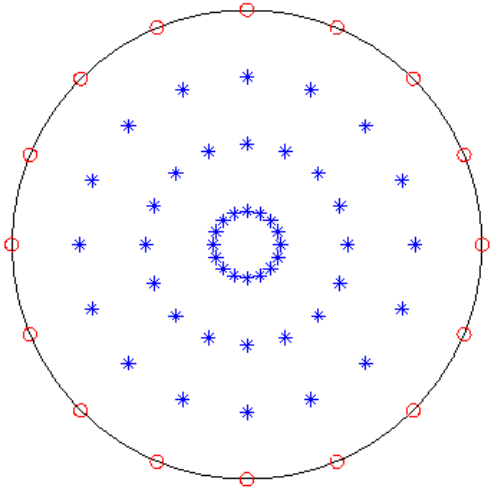
Pseudospectral (PS)

Fourier angularly
Chebyshev radially



Finite differences (FD2)

Centered FD2 in both directions



In all cases 48 nodes in the interior, 16 nodes on the boundary

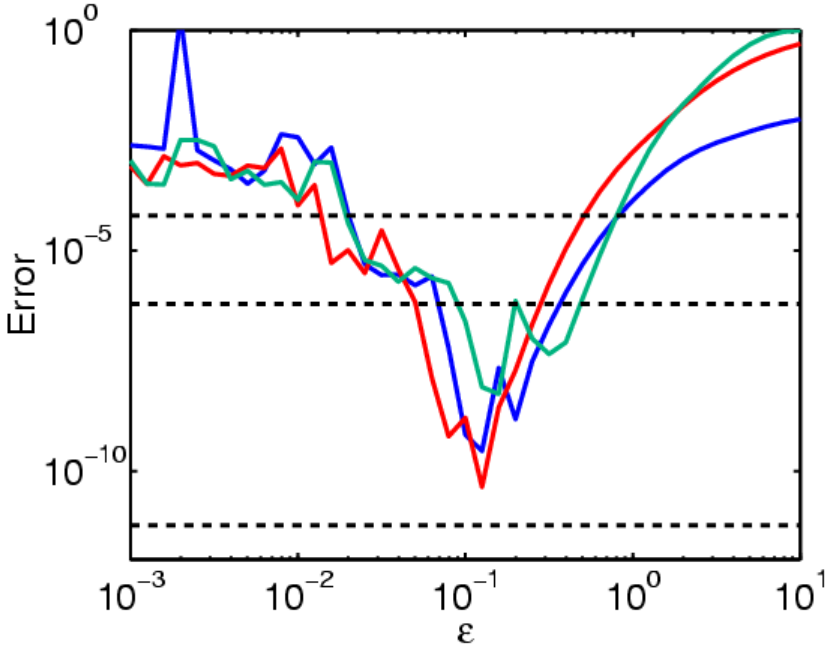
Comparison between RBF, PS, and FD2 approximations

RBF approximations:

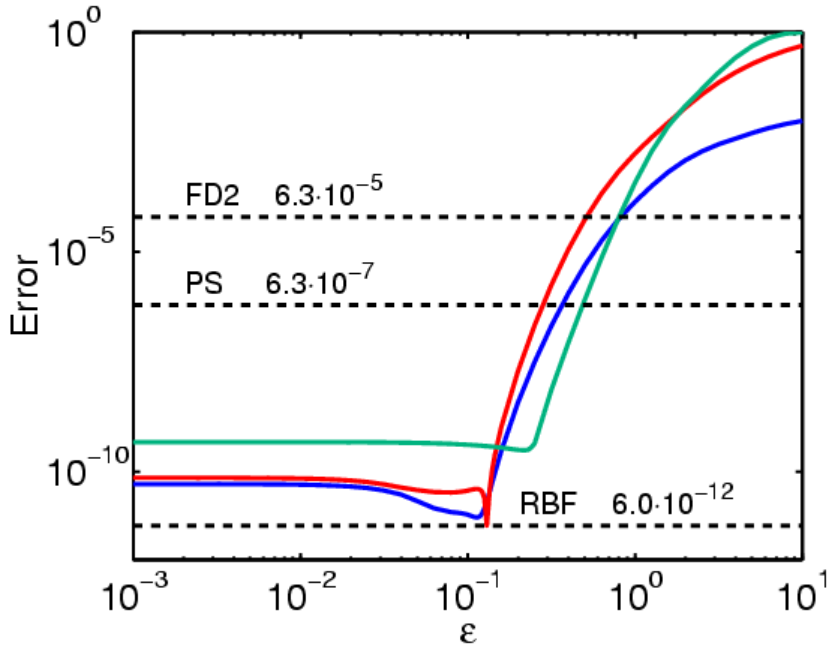
Gaussians	(GA)	$\phi(r) = e^{-(\varepsilon r)^2}$
Inverse quadratics	(IQ)	$\phi(r) = 1/(1+(\varepsilon r)^2)$
Multiquadrics	(MQ)	$\phi(r) = \sqrt{1+(\varepsilon r)^2}$

Errors (max norm) vs. ε:

Using direct computation



Using Contour-Padé algorithm



For a second order accurate scheme in 2-D: Error is inversely proportional to the total number of node points $\Rightarrow$ RBF with 64 nodes is here similar in accuracy to what would be expected from FD2 with 64 million nodes

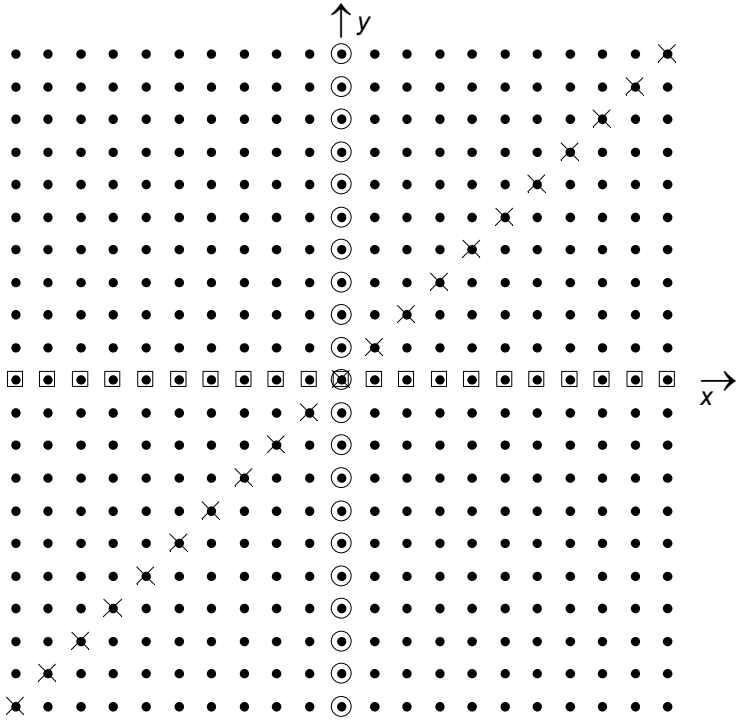
RBFs can be very effective also on meshes

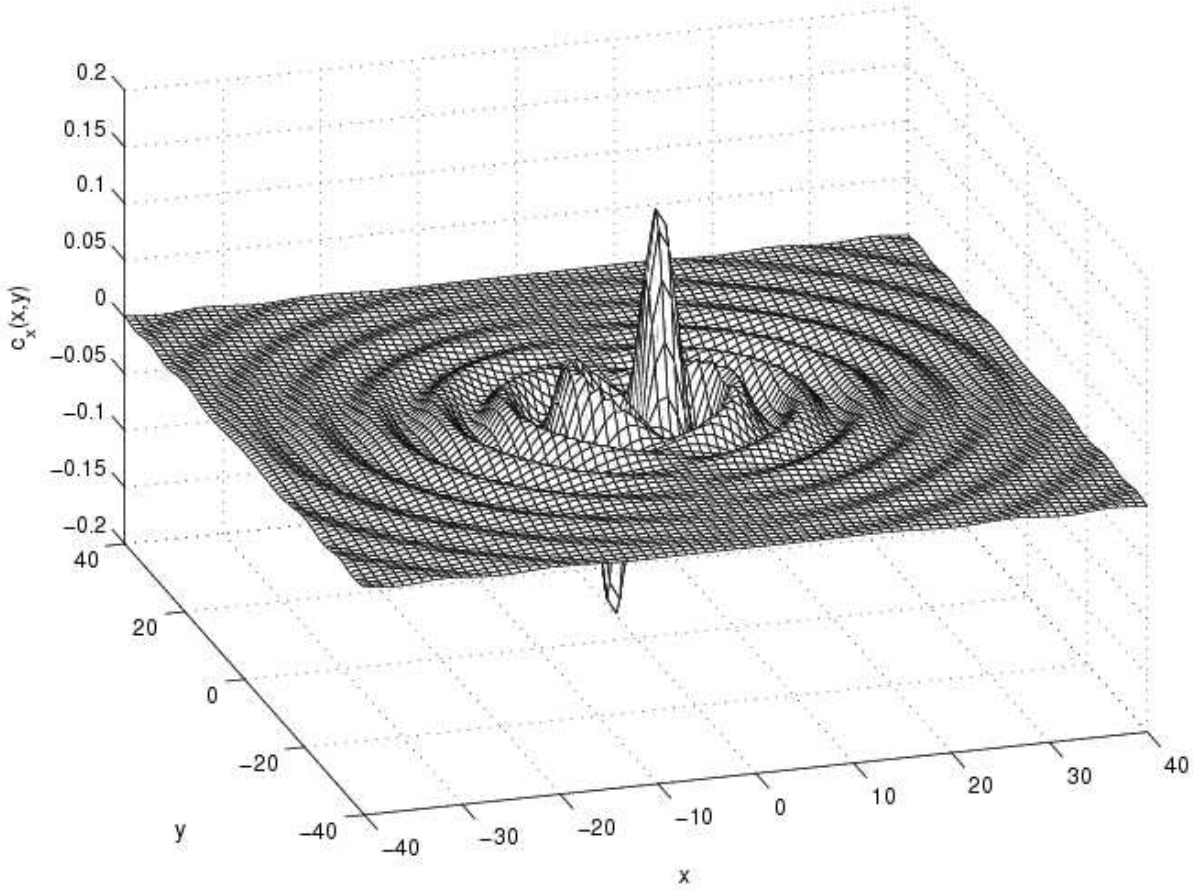
(Fornberg, Flyer, Russell, 2008)

There is something strange about how FD and PS methods approximate $\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)$

Where should $\frac{\partial}{\partial x}$ pick up its data from?

Answer: $\frac{x}{8}\,{}_0F_1(3,-\frac{1}{4}(x^2+y^2))$

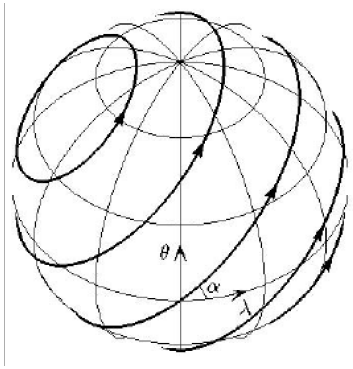




Turns out:

- RBFs can improve on FD and PS also on Cartesian meshes,
- Hexagonal or irregular node sets can be better still than Cartesian lattices.

Solving a convective PDEs on a sphere



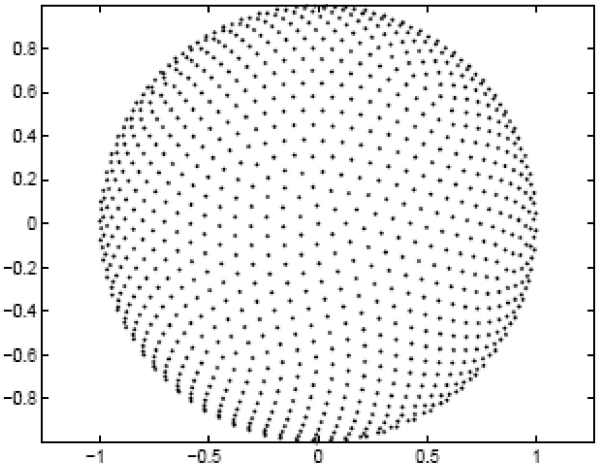
Governing PDE:

$$\frac{\partial u}{\partial t} + (\cos\alpha - \tan\theta \sin\varphi \sin\alpha)\frac{\partial u}{\partial \varphi} - \cos\varphi \sin\alpha \frac{\partial u}{\partial \theta} = 0 \quad \Rightarrow$$

RBF collocation of PDE in spherical coordinates eliminates all coordinate-based singularities

Spectrally accurate numerical methods:

RBF: **Radial Basis Functions**

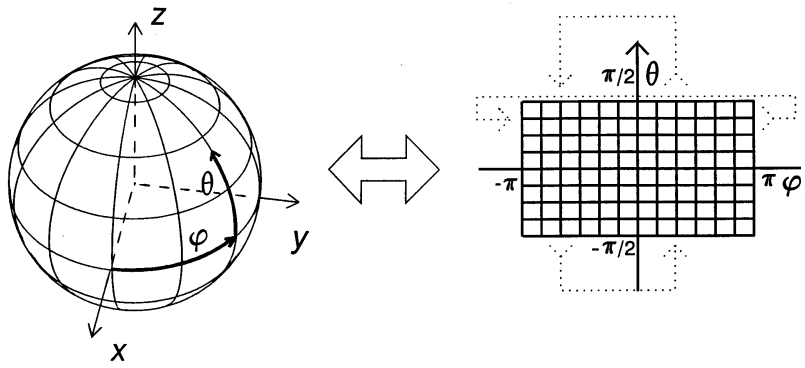


No coordinate system-induced singularities anywhere

SPH: **Spherical Harmonics**

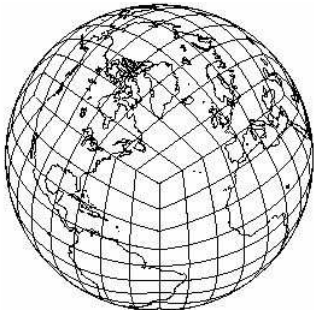
Collocation with orthogonal set of trig-like basis functions, which lead to entirely uniform resolution over surface of sphere

DF: **Double Fourier series**



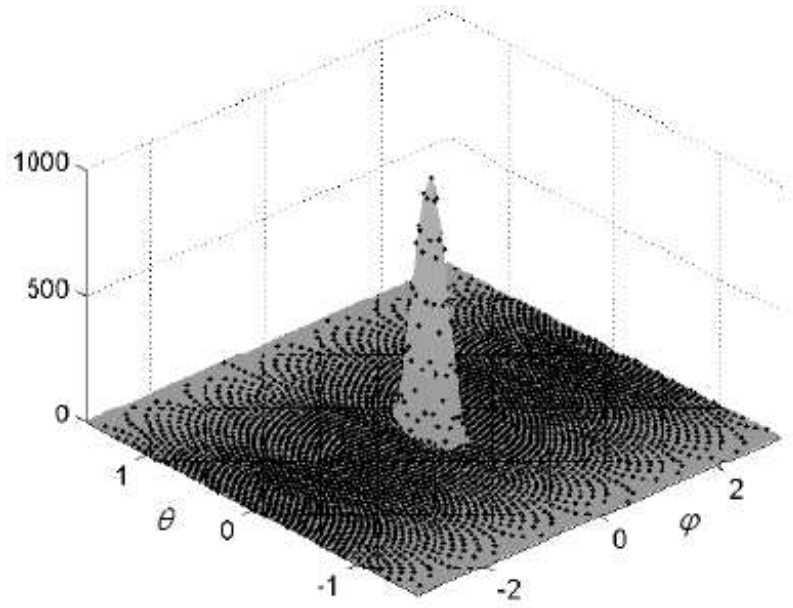
SE: **Spectral Elements**

Implemented by means of *cubed sphere*

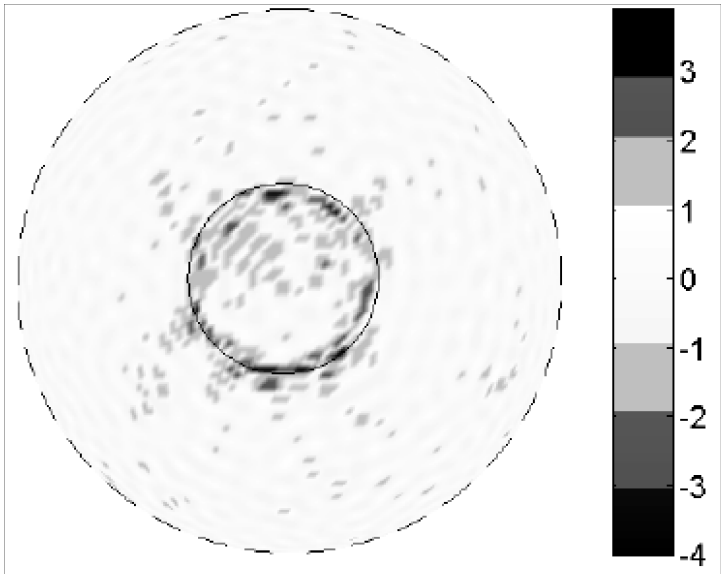


Convective flow over a sphere - Comparison between methods

(Flyer and Wright, 2007)



Snapshot of 4096 node RBF calculation after 12 days (1 full revolution) of a cosine bell



(10^{-3})

Error (parts per thousand)

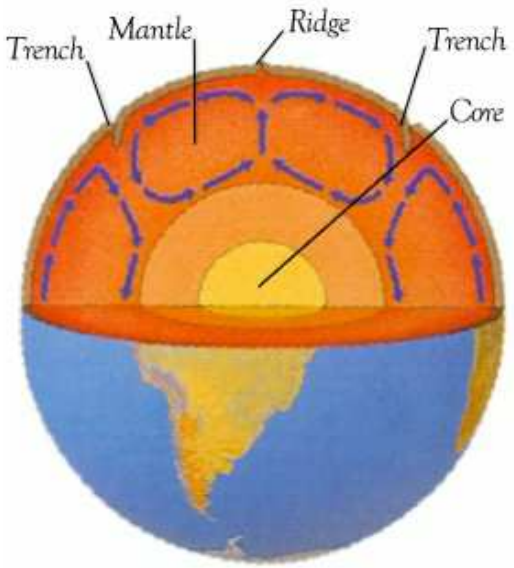
	Method	l_2 error	Number of node points/ free parameters	Time step (RK4)	Code length (lines)	Local refine-ment feasible
RBF	Radial basis functions	0.006	4,096	1/2 hour	< 40	yes
SPH	Spherical harmonics	0.005	32,768	90 seconds	> 500	no
DF	Double Fourier	0.005	32,768	90 seconds	> 100	no
SE	Spectral elements	0.005	7,776	6 minutes	> 1000	yes

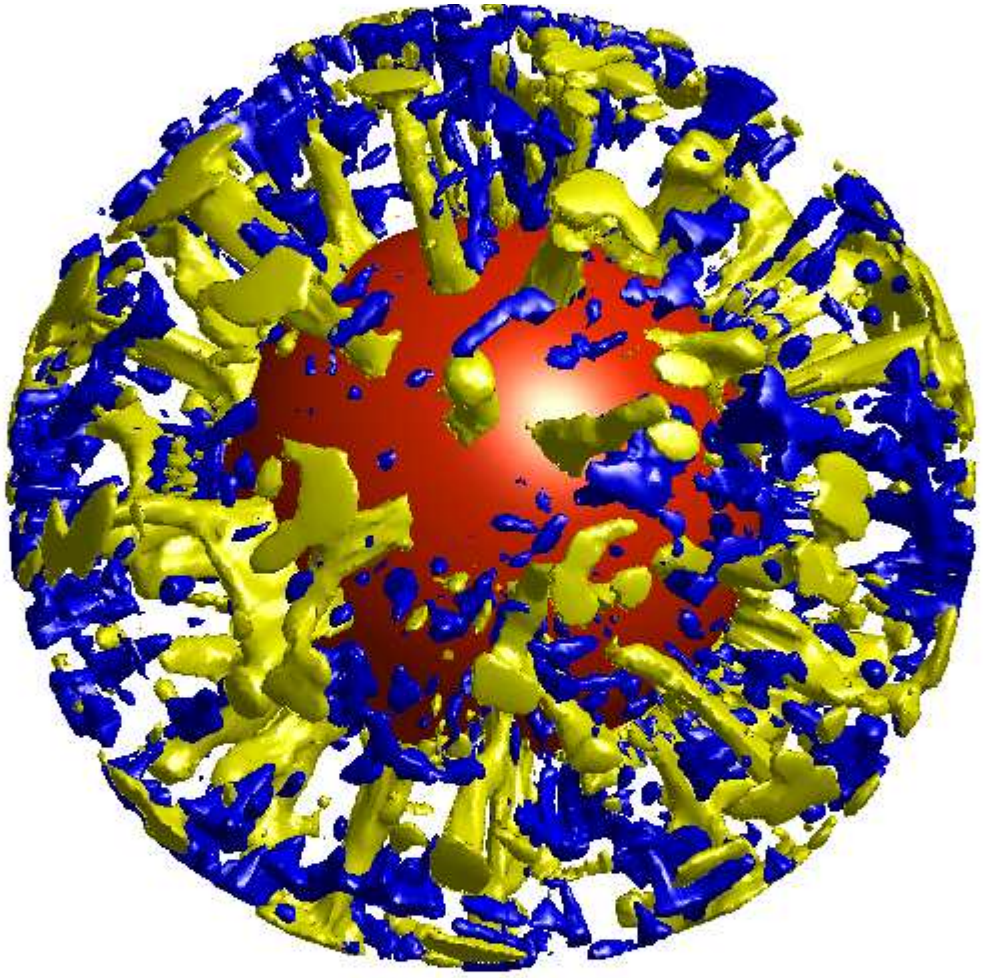
(Fornberg and Piret, 2008): Time stepping thousands of full revolutions and using RBF-QR - still no significant error growth or trailing dispersive waves.

Examples of two recent RBF applications

1. Thermal Convection in a 3-D Spherical Shell

(Flyer and Wright, 2009)





Example of computed solution for Ra = 500,000;

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

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

2. Solution of a reaction-diffusion PDE over a general surface

Relation to pattern formation on animals (Turing, 1952)

Brusselator (Prigogine, 1971)

$$\begin{cases} \dfrac{\partial u}{\partial t} = a - (b+1)u + u^2 v + D_u \Delta u \\ \dfrac{\partial v}{\partial t} = bu - u^2 v + D_v \Delta v \end{cases}$$

Illustration on right from web site of Nonlinear Dynamics Group at Brandeis University

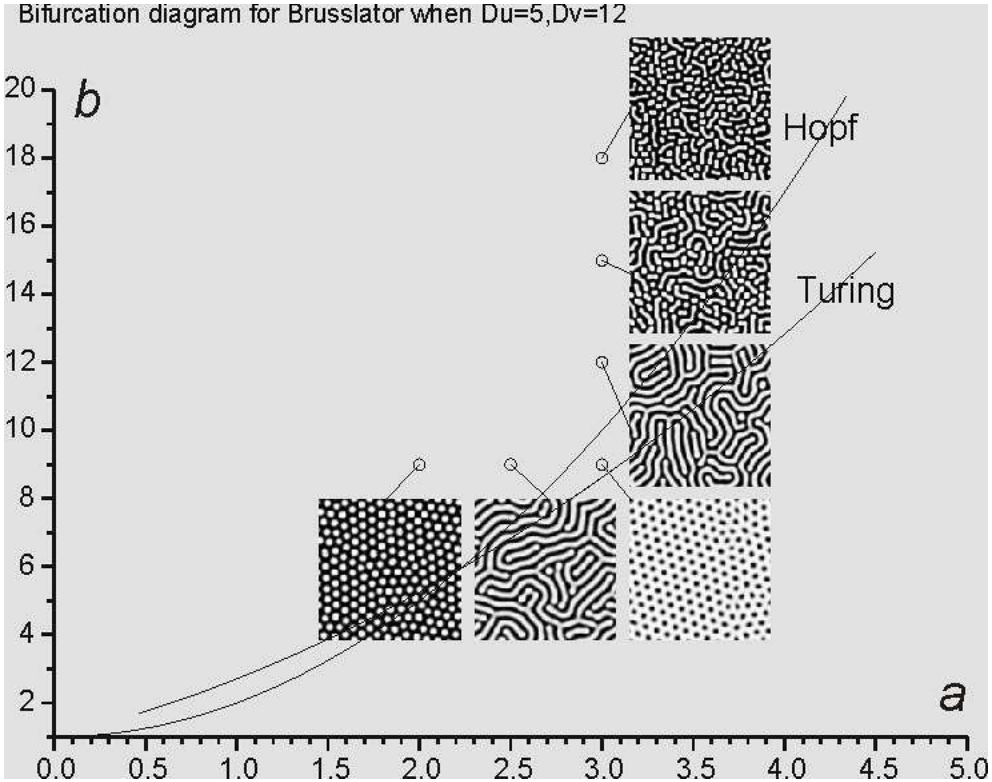


Illustration shows different patterns emerging on a flat 2-D plane for when $D_u = 5, D_v = 12.$

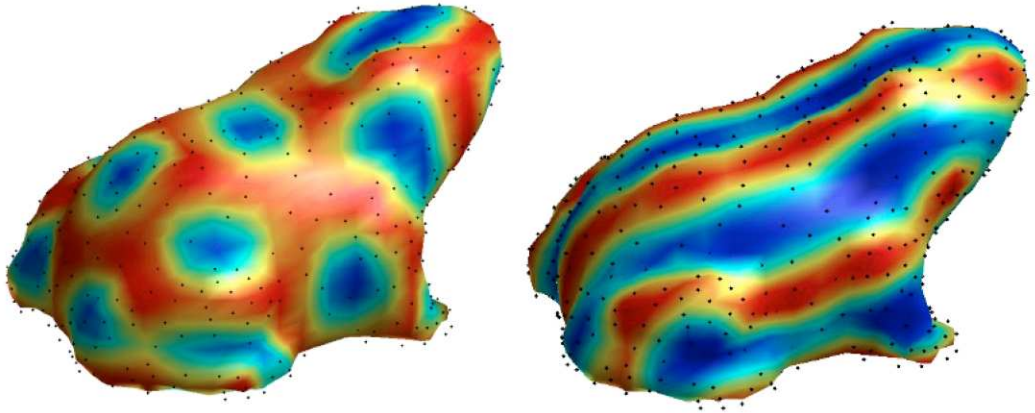
Example: RBF solution over the surface of a frog

(Piret, 2012)

The Brusselator equation, solved using global RBFs over a curves surface defined by 560 scattered nodes, again for $D_u = 5, D_v = 12.$

The scattered nodes provide
both computational node points
and define the surface geometry.

Left: $a = 4,\ b = 14$
Right: $a = 3,\ b = 11$

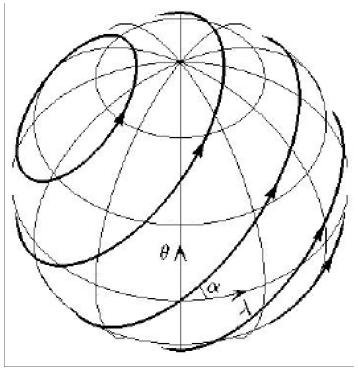


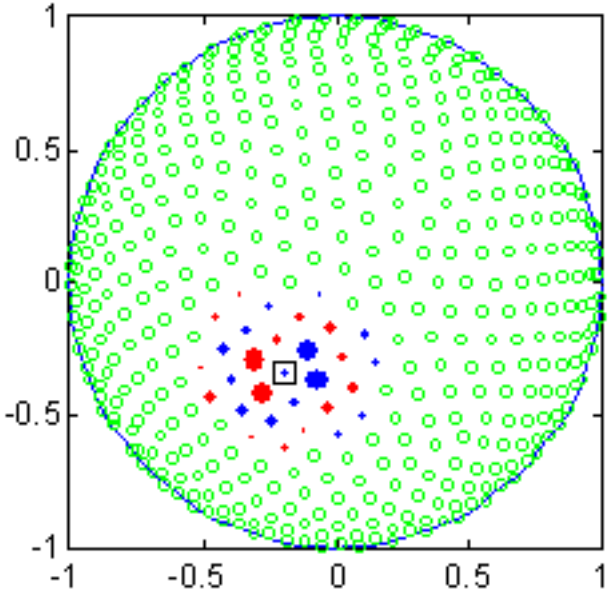
Left: Tabasara rain frog
Right: Poison dart frog

RBF-generated finite difference (RBF-FD) approximations

Shown to far right: Sphere with N (= 800) ME (minimum energy) nodes ⇒ ⇒ ⇒ ⇒

- At each node, find n (= 30) nearest neighbors
- Approximate PDE operator, here 'solid body rotation', ⇒ ⇒ ⇒ ⇒
  but use RBFs instead of polynomials to obtain the weights
- Carry out time stepping, for ex. with RK4.



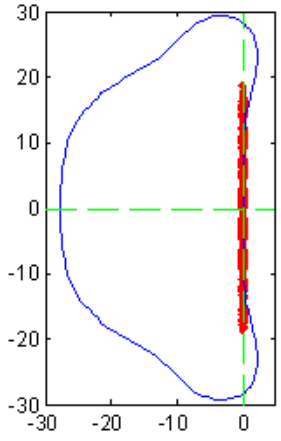


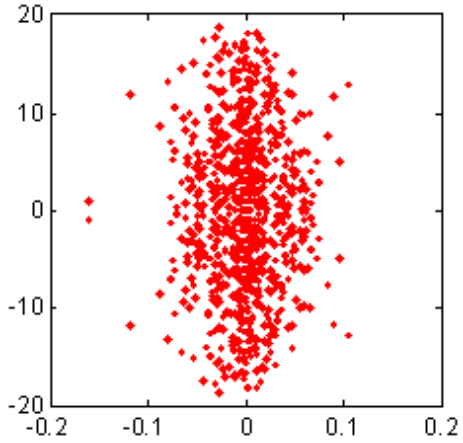
Issues:

- **Effective way to finding n nearest neighbors**
  Direct search: $O(N^2)$ operations; kd-tree $O(N \log N)$; *knnsearch* in Matlab's statistics tool box

- **Time stepping stability**
  To right: Eigenvalues of discrete spatial operator:

  Hyperviscosity leaves physically relevant eigenvalues intact, but shifts spurious ones from the right half-plane to the left one.
  (Fornberg and Lehto, 2011)



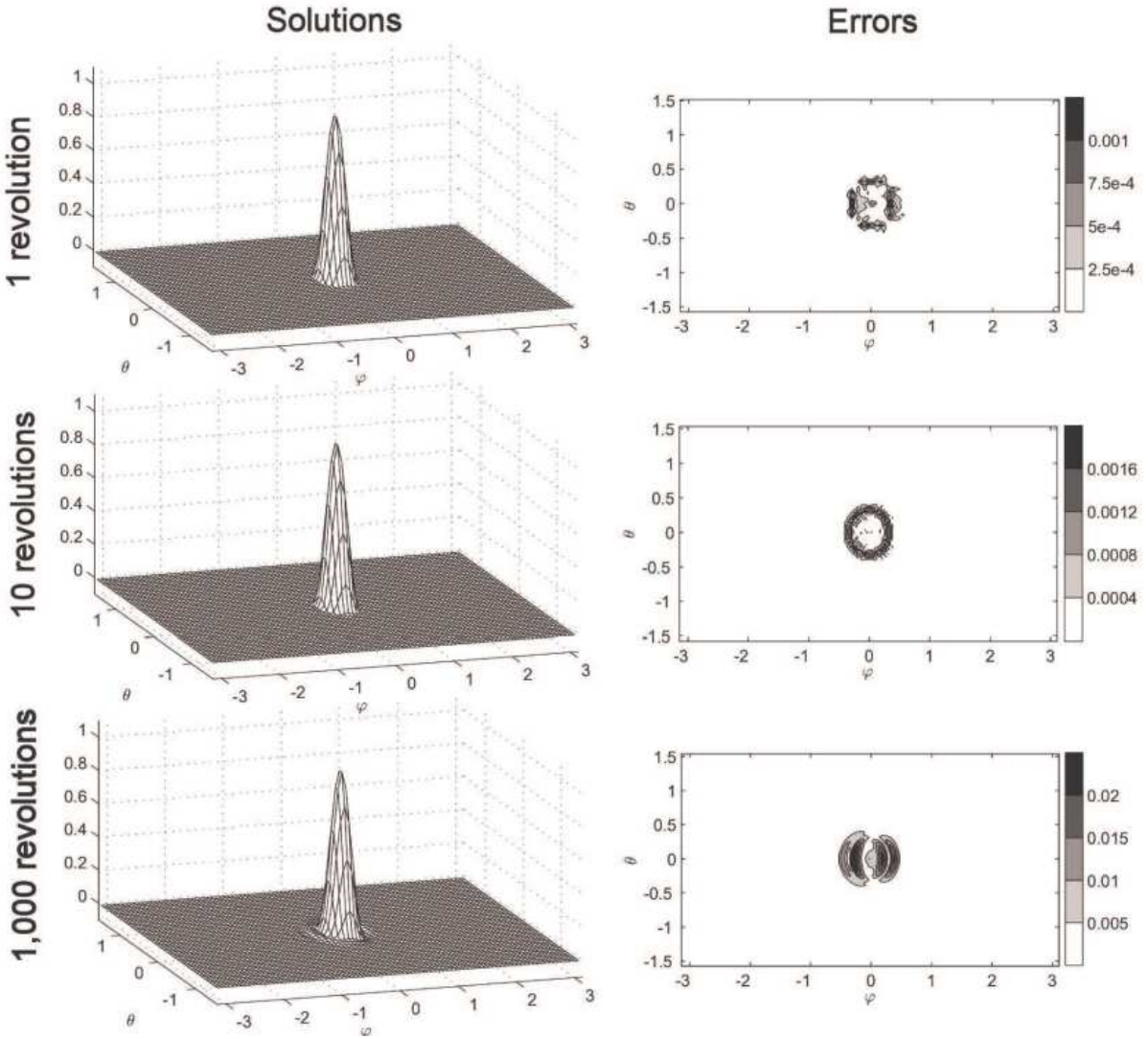


Revisit convective flow around sphere with RBF-FD method

$N = 25{,}600$, $n = 74$

Even after 1000 revolutions, no signs of either loss of peak height or of trailing wave trains

For the same level of accuracy, improves over global RBF approximations by about an order of magnitude in both computer time and memory.



Conclusions

Established:

- 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 local node refinement.
- RBF and RBF-FD methods compete very favorably against existing methods, and are particularly advantageous over long integration times.
- The near-flat basis function regime (ε small) is of particular interest, and stable numerical algorithms for it have been developed.

Some current research issues:

- Improve the understanding of RBF-methods at boundaries - want to apply to media with internal interfaces
- Compare both global RBFs and RBF-FD implementations against best previous methods in more major applications.
- Use RBF-FD methods to solve PDEs over general surfaces / manifolds - with Math. Biology applications.
- Develop effective implementations of RBF-FD methods on GPUs and on massively parallel computer hardware.

RBFs - LEAVE THE MESH BEHIND !

Conclusions

Established:

- 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 local node refinement.
- RBF and RBF-FD methods compete very favorably against existing methods, and are particularly advantageous over long integration times.
- The near-flat basis function regime (ε small) is of particular interest, and stable numerical algorithms for it have been developed.

Some current research issues:

- Improve the understanding of RBF-methods at boundaries - want to apply to media with internal interfaces
- Compare both global RBFs and RBF-FD implementations against best previous methods in more major applications.
- Use RBF-FD methods to solve PDEs over general surfaces / manifolds - with Math. Biology applications.
- Develop effective implementations of RBF-FD methods on GPUs and on massively parallel computer hardware.

RBFs - LEAVE THE MESS BEHIND !