Long time integrations of a convective PDE on the sphere by RBF collocation

Bengt Fornberg

University of Colorado
Department of Applied Mathematics

and

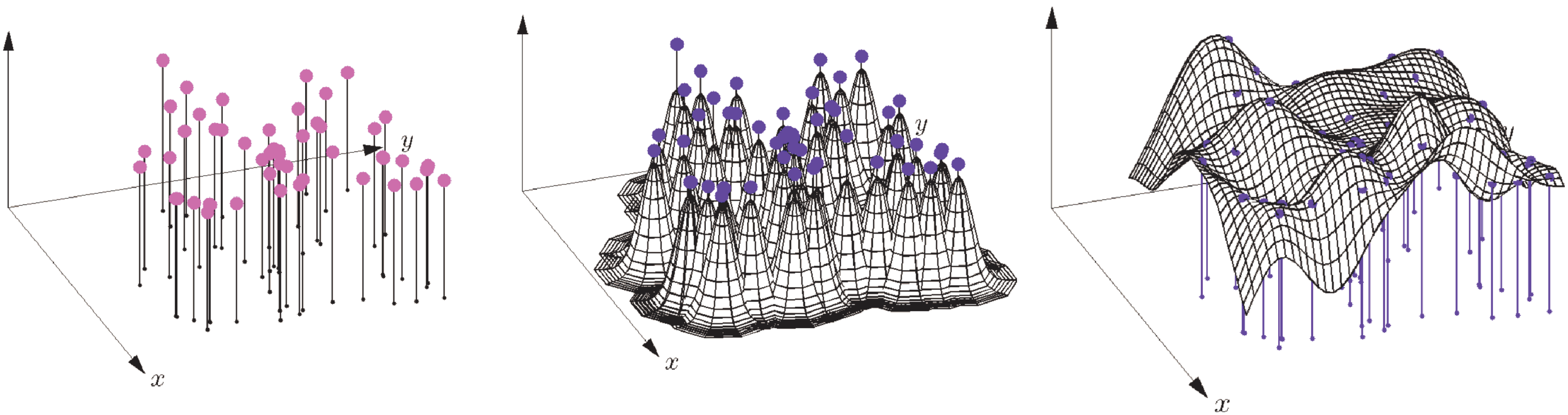
Natasha Flyer

NCAR
Institute for Mathematics Applied to Geosceinces

<u>Collaborators:</u>

Cécile Piret, Grady Wright, Erik Lehto

RBF idea, In pictures:



1970 Invention of RBFs (for application in cartography)

Some other key dates:

1940 Unconditional non-singularity for many types of radial functions
1984 Unconditional non-singularity for multiquadrics $(\phi(r) = \sqrt{1+(\varepsilon r)^2})$
1990 First application to numerical solutions of PDEs
2002 Flat RBF limit exists - generalizes all 'classical' pseudospectral methods
2004 First numerically stable algorithm in flat basis function limit
2007 First application of RBFs to geophysical test problems on a sphere

RBF idea, In formulas:

Given scattered data $(\underline{x}_k, f_k)$, $k = 1, 2, \ldots, N$, the coefficients λ_k in $s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|)$ are found by collocation: $s(\underline{x}_k) = f_k$, $k = 1, 2, \ldots, N$:

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

Key theorems:
- For 'most' $\phi(r)$, this system can never be singular.
- Spectral accuracy for smooth radial functions

Main present issues:
- Defeat numerical ill-conditioning
- Reduce the computational cost

  Most immediate algorithms (RBF-Direct):
  - Solve system above for λ_k: $O(N^3)$ operations
  - Evaluate interpolant at M locations: $O(M\,N)$ operations
  - Applying approximation of space derivatives: $O(N^2)$ operations
- Develop fast and scalable codes for large-scale parallel computers

Moving Vortices on A Sphere

(Flyer and Lehto, 2008)

Method of lines formulation: $\frac{\partial h}{\partial t} = -\left(\underline{U}(\alpha, \theta, \varphi, t) \bullet \nabla\right) h \quad \Leftrightarrow \quad \frac{\partial h}{\partial t} = -\left(\frac{u(t)}{\cos\theta} D_N^{\varphi} + v(t) D_N^{\theta}\right) h$

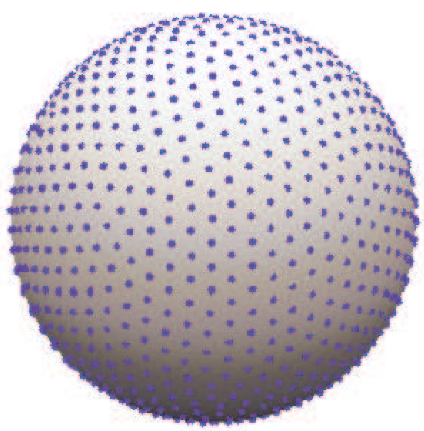
D_N^{φ} and D_N^{θ} are discrete RBF differentiation matrices:

- Free of Pole Singularities
- Error Invariant of α, angle of rotation

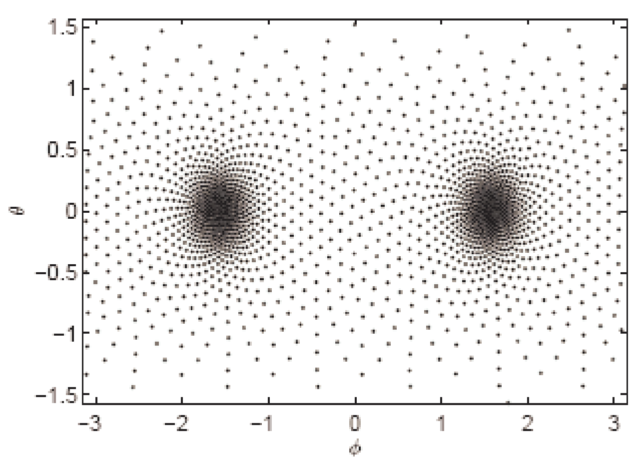
Inverse Multiquadrics RBFs; 12 Day Simulation

$N = 3136$ nodes

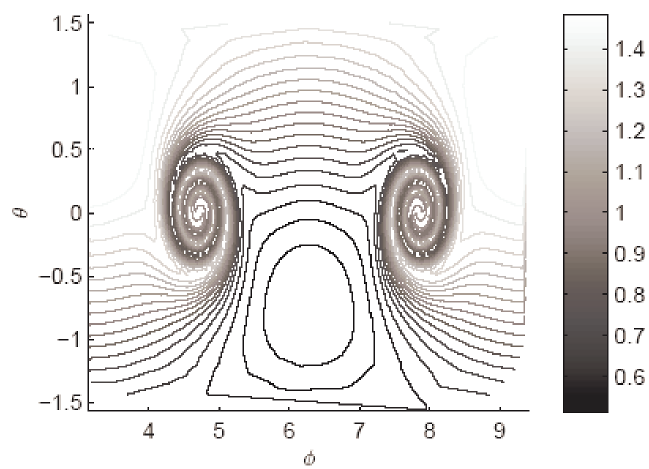
$\Delta t = 20$ minutes; 4^{th} order Runge-Kutta



ME Nodes

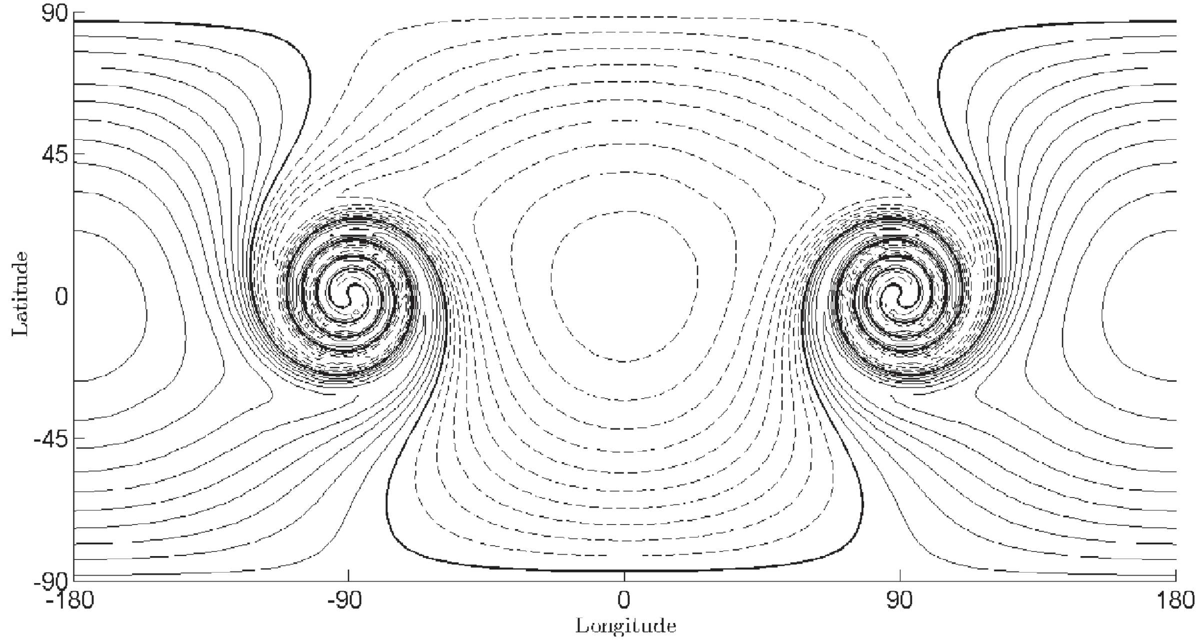


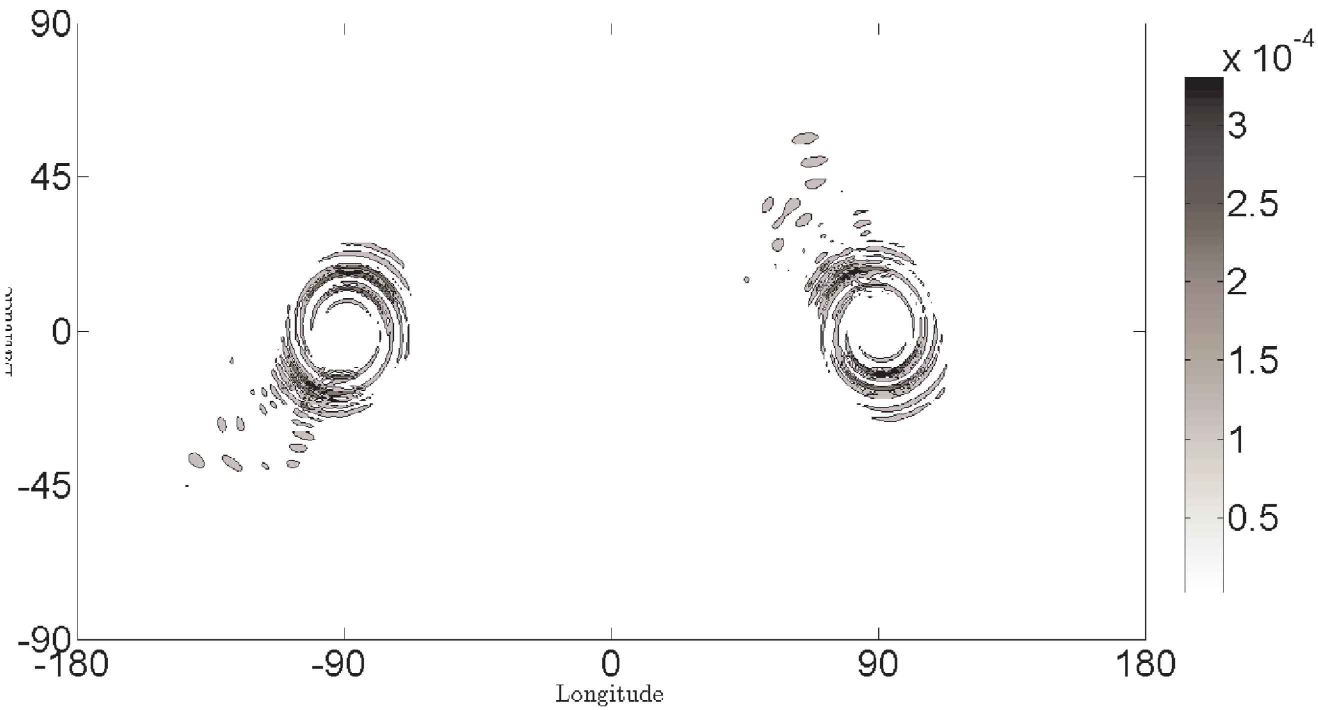
Node Refinement



Final Solution

Final Solution and Magnitude of Error





Comparison With Other Methods

Method	Resolution	Δt (mins.)	ℓ_1	ℓ_2
With local node refinement				
RBF [1]	$N = 3136$	20	$4 \cdot 10^{-5}$	$8 \cdot 10^{-5}$
Finite Volume AMR [2]	Base 5° ; 3 level adaptive	Variable	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$
With uniform node distribution				
RBF [1]	$N = 3136$, 6.4°	80	$3 \cdot 10^{-3}$	$4 \cdot 10^{-3}$
Finite Volume [2]	0.625°	10	$5 \cdot 10^{-4}$	$2 \cdot 10^{-3}$
Discontinuous Galerkin [2]	$N = 9600$	6	$2 \cdot 10^{-3}$	$7 \cdot 10^{-3}$
Semi-Lagrangian [2]	$N = 10512$	60	$4 \cdot 10^{-2}$	$5 \cdot 10^{-2}$

References:

[1] Flyer, N. and Lehto, E., A radial basis function implementation of local node refinement: Two vortex test cases on a sphere, to be submitted to Mon. Wea. Rev.

[2] Nair, R.D. and Jablonowski, C., Moving vortices on the sphere: A test case for horizontal advection problems, Mon. Wea. Rev. 136 (2008), 699-711.

Full Nonlinear Unsteady Shallow Water Equations

(Flyer and Wright, 2008)

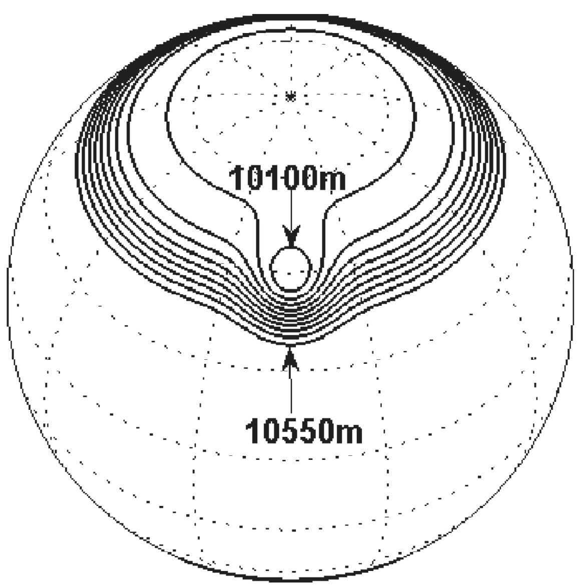
Description: Forcing terms added to the shallow water equations to generate a flow that mimics a short wave trough embedded in a westerly jet

$N = 3136$

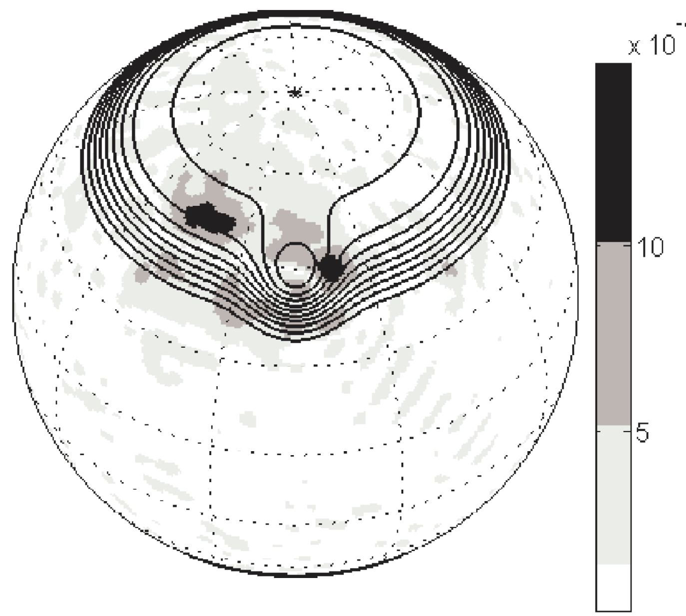
$\Delta t = 10$ minutes

RK4 time-stepping; 5 day run

Geopotential height, 50m contour intervals



|Exact-Numerical| Error



1.4 (10^{-4})

(white$<10^{-5}$)

Comparison with Other Methods

Method	Number of grid points		Time step	Relative l_2 error in h
RBF [1]	748	(28^2)	20 minutes	$4.96 \cdot 10^{-1}$
	1849	(43^2)	12 minutes	$3.47 \cdot 10^{-3}$
	3136	(56^2)	10 minutes	$8.91 \cdot 10^{-6}$
	4096	(64^2)	8 minutes	$2.57 \cdot 10^{-7}$
	5041	(71^2)	6 minutes	$3.84 \cdot 10^{-8}$
Spherical Harmonics [2]	8192	(*T*42)	20 minutes*	$2 \cdot 10^{-3}$
Double Fourier [3]	2048		6 minutes	$3.9 \cdot 10^{-1}$
	8192		3 minutes	$8.2 \cdot 10^{-3}$
Spectral Elements [4]	6144		90 seconds	$6.5 \cdot 10^{-3}$
	24576		45 seconds	$4 \cdot 10^{-5}$

* semi-implicit time stepping

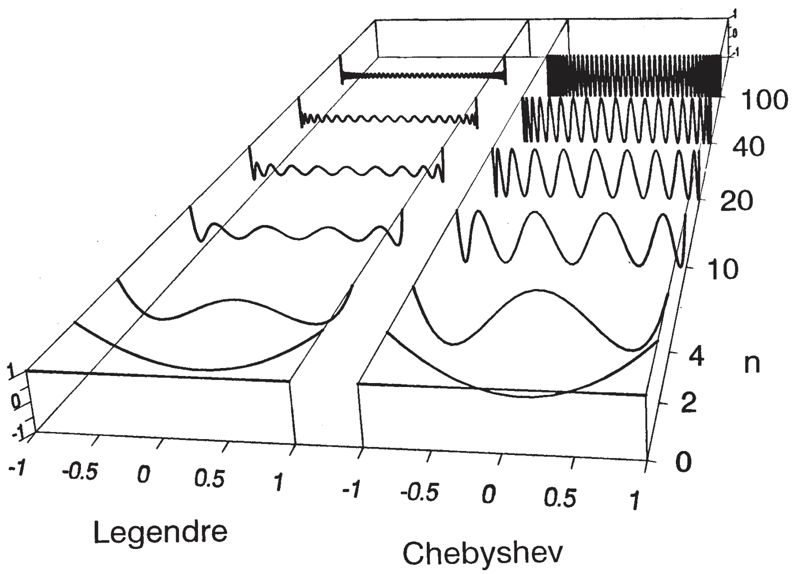
References:

[1] Flyer, N. and Wright, G.B., Solving the shallow water equations on a sphere using radial basis functions, to be submitted to JCP.
[2] Jacob-Chien, R., Hack, J.J. and Williamson, D.L., Spectral transform solutions to the shallow water test set, JCP 119 (1995), 164-187.
[3] Spotz, W.F., Taylor, M.A. and Swarztrauber, P.N., Fast shallow water equation solvers in latutude-longitude coordinates, JCP 145 (1998), 432-444.
[4] Taylor, M., Tribbia, J. and Iskadrarani, M., The spectral element method for the shallow water equations on the sphere, JCP 130 (1997), 92-108.

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

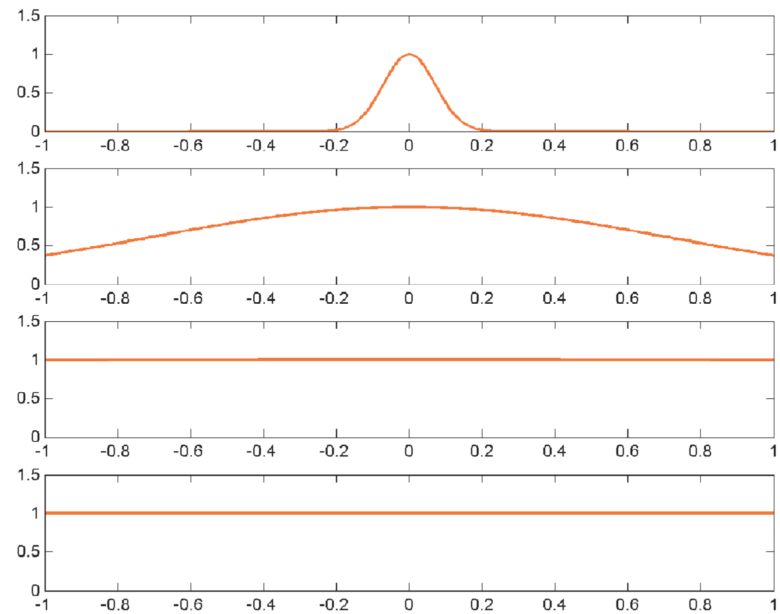


ε = 10

ε = 1

ε = 0.1

ε = 0.01



Condition number of RBF matrix $O(\varepsilon^{-\alpha(N)})$; Exact values are available for $\alpha(N)$:

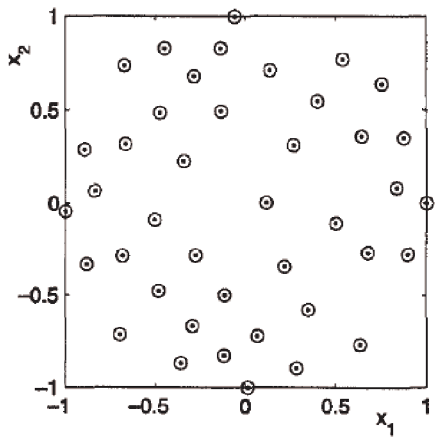
(Fornberg and Zuev, 2007)

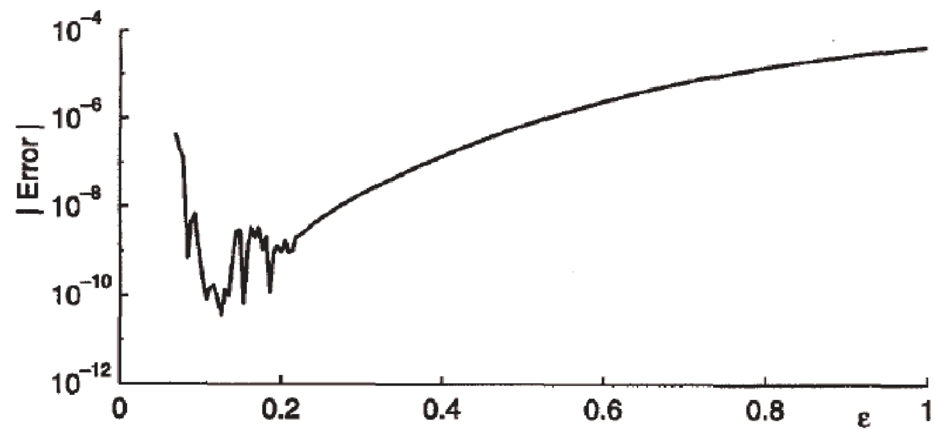
		$\alpha(n)$		$\alpha(n)$		$\alpha(n)$
1-D (non-periodic)	$n = 10^1$	18	$n = 10^2$	198	$n = 10^3$	1,998
2-D (non-periodic)	$n = 10^2$	26	$n = 10^4$	280	$n = 10^6$	2,826
3-D (non-periodic)	$n = 10^3$	34	$n = 10^6$	360	$n = 10^9$	3,632
Resolves in each direction	about 5 modes		about 50 modes		about 500 modes	

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)

- In any number of dimensions, the $\varepsilon \to 0$ limit reduces to 'classical' PS methods if used on tensor type grids.

- The RBF approach generalize PS methods in many ways:
  - Guaranteed nonsingular also for scattered nodes on irregular geometries
  - Allow spectral accuracy to be combined with mesh refinement
  - Best accuracy often obtained not in the $\varepsilon \to 0$ limit, but for larger 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

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

- **High precision arithmetic** — It is known exactly how the condition number varies with domain type, N, ε . Approach often costly.

- **Algorithms that completely bypass ill-conditioning all the way into $\varepsilon \to 0$ limit, while using only standard precision arithmetic:**

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

  - **Contour-Padé algorithm** — First algorithm of its kind; established that concept is possible; limited to relatively small N-values (Fornberg and Wright, 2004) Simplified version Contour-SVD under development.

  - **RBF-QR method** — So far developed only for nodes scattered over the surface of a sphere (Fornberg and Piret, 2007). No limit on N; cost about five times that of RBF-Direct (even as $\varepsilon \to 0$).

  Probably many more genuinely stable algorithms to come...

Background to RBF-QR for spheres: Spherical Harmonics (SPH)

Spherical harmonics: Restriction to surface of unit sphere of simple polynomials in *x*, *y*, *z*:

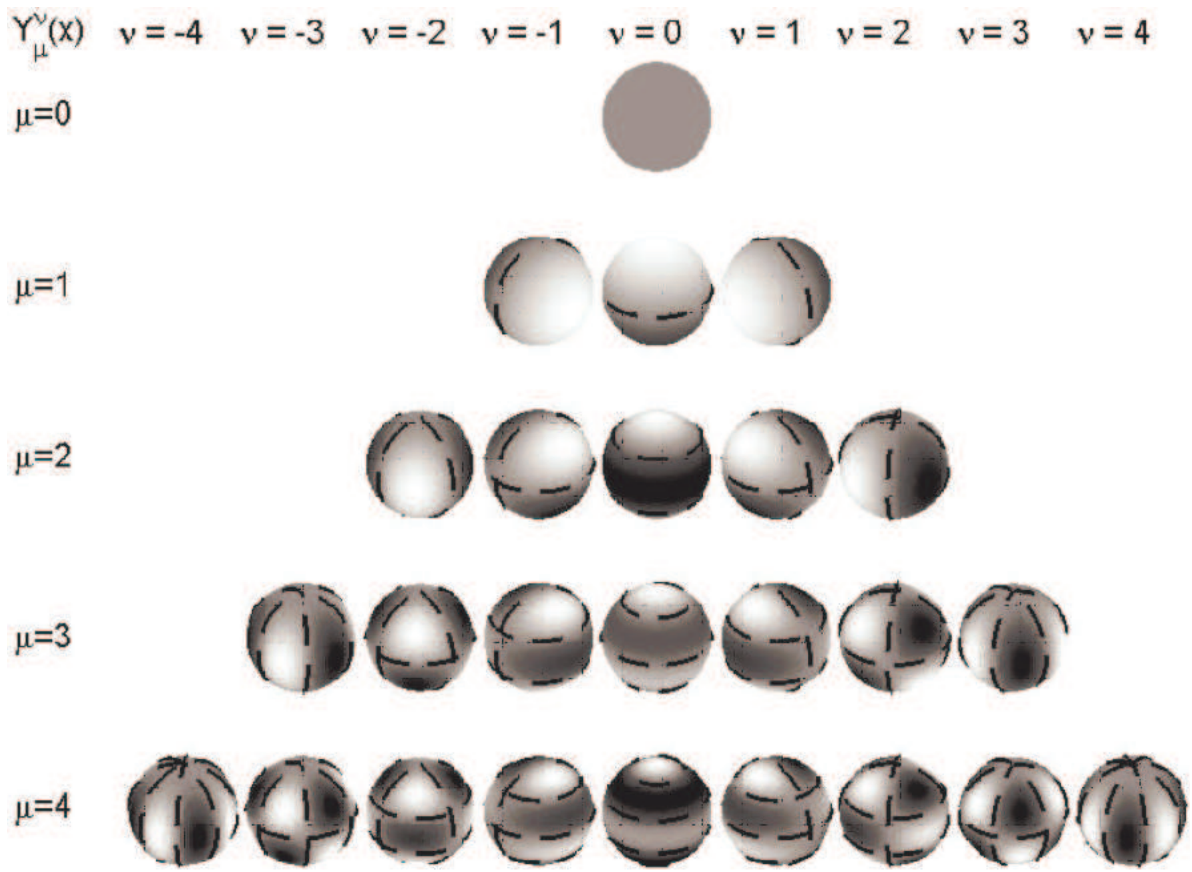
$Y_\mu^\nu(x,y,z)$

$\nu =$	...	-2	-1	0	1	2	...
$\mu = 0$				$\frac{1}{2\sqrt{\pi}} 1$			
1			$-\frac{1}{2}\sqrt{\frac{3}{2\pi}}\, y$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\, z$	$-\frac{1}{2}\sqrt{\frac{3}{2\pi}}\, x$		
2		$\frac{1}{2}\sqrt{\frac{15}{2\pi}}\, xy$	$-\frac{1}{2}\sqrt{\frac{15}{2\pi}}\, zy$	$\frac{1}{4}\sqrt{\frac{5}{\pi}}\,(3z^2-1)$	$-\frac{1}{2}\sqrt{\frac{15}{2\pi}}\, zx$	$\frac{1}{4}\sqrt{\frac{15}{2\pi}}\,(x^2-y^2)$	
...	...	...	...	...	...	...	...

- Counterpart to Fourier modes around periphery of unit circle
- Orthogonal
- Uniform resolution over surface
- Spectral accuracy for PDEs

but

- Not associated with any particular node set
- No clear counterpart to FFT
- No opportunities for variable resolution



Expansions of RBFs in terms of SPH

RBFs, centered on the surface of the unit sphere, can be expanded in SPH as follows:

$$\phi(\|\underline{x}-\underline{x}_i\|) = \sum_{\mu=0}^{\infty} \sum_{\nu=-\mu}^{\mu} \left\{\varepsilon^{2\mu} c_{\mu,\varepsilon}\, Y_\mu^\nu(\underline{x}_i)\right\} Y_\mu^\nu(\underline{x})$$

where, for example

MQ: $\phi(r) = \sqrt{1+(\varepsilon r)^2}$ $\qquad c_{\mu,\varepsilon} = \dfrac{-2\pi(2\varepsilon^2+1+(\mu+\frac{1}{2})\sqrt{1+4\varepsilon^2})}{(\mu+\frac{3}{2})(\mu+\frac{1}{2})(\mu-\frac{1}{2})}\left(\dfrac{2}{1+\sqrt{4\varepsilon^2+1}}\right)^{2\mu+1}$

IMQ: $\phi(r) = \dfrac{1}{\sqrt{1+(\varepsilon r)^2}}$ $\qquad c_{\mu,\varepsilon} = \dfrac{4\pi}{(\mu+\frac{1}{2})}\left(\dfrac{2}{1+\sqrt{4\varepsilon^2+1}}\right)^{2\mu+1}$

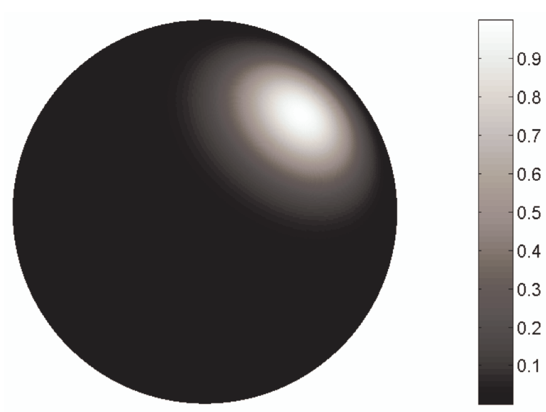
Key points of the RBF-QR algorithm (Fornberg and Piret, 2007):

- There is no loss of accuracy in computing $c_{\mu,\varepsilon}\, Y_\mu^\nu(\underline{x}_i)$, even if $\varepsilon \to 0$.
- The factors $\varepsilon^{2\mu}$ contain all the ill-conditioning, and they can be *analytically* kept out of the numerical algorithm in going from data values to interpolant values.
- Algorithm involves, among other steps, a QR factorization.
- The algorithm proves that, as $\varepsilon \to 0$, the RBF interpolant (usually) converges to the SPH interpolant

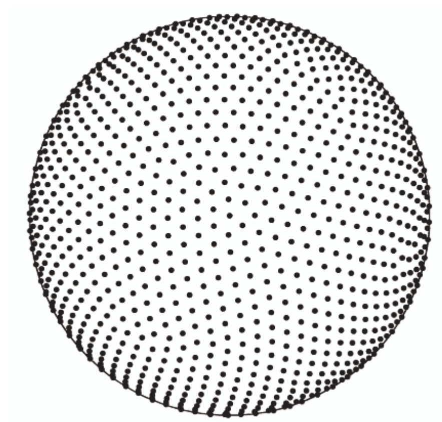
Test case for interpolation

Test function:

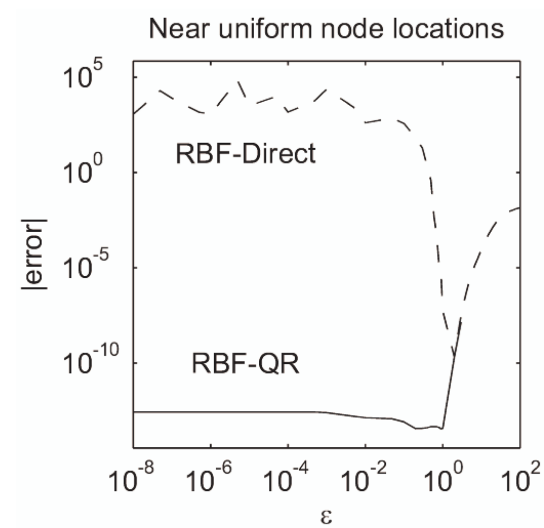
$$f(\underline{x}) = e^{-7(x+\frac{1}{2})^2-8(y+\frac{1}{2})^2-9(z-\frac{1}{\sqrt{2}})^2}$$



1849 minimal energy nodes



Error: RBF-Direct vs. RBF-QR



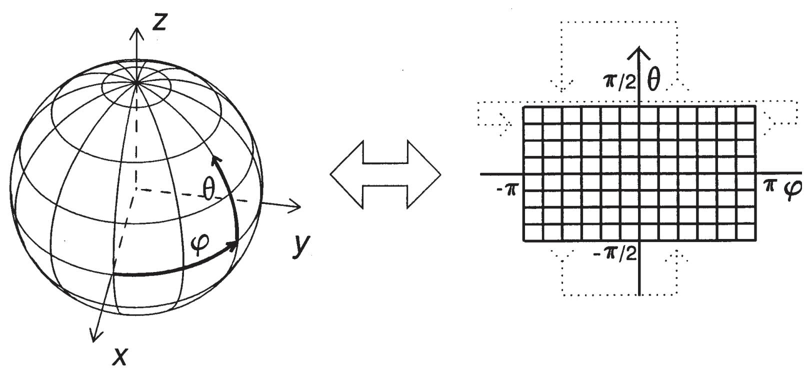
RBF-Direct: cond(A) = $O(\varepsilon^{-84})$; each 16 extra decimal digits of arithmetic precision lowers the onset of ill-conditioning by a factor of 0.65 for ε.

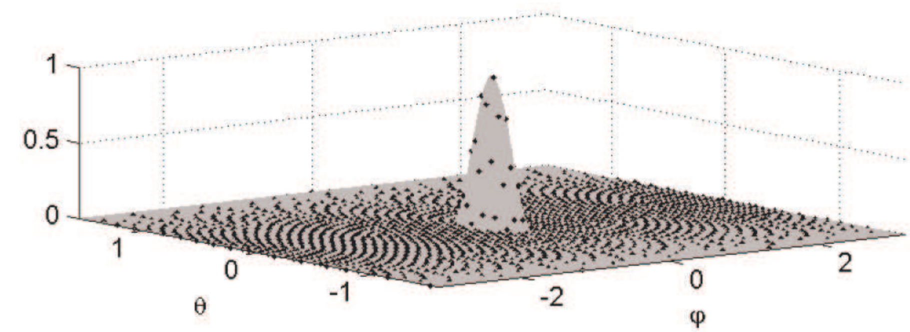
Since RBF $\varepsilon \to 0$ limit agrees with the SPH interpolant, why not just use the latter?

- The error often increases in the last stages of $\varepsilon \to 0$
- The SPH interpolant can be singular for certain node distributions - the RBF interpolant can never be singular
- RBFs offer opportunities for local node refinement

Long time integration of convective flow over a sphere

(Fornberg and Piret, 2008) - follow-up on shorter-time integration with GA and RBF-Direct by Flyer and Wright (2007)

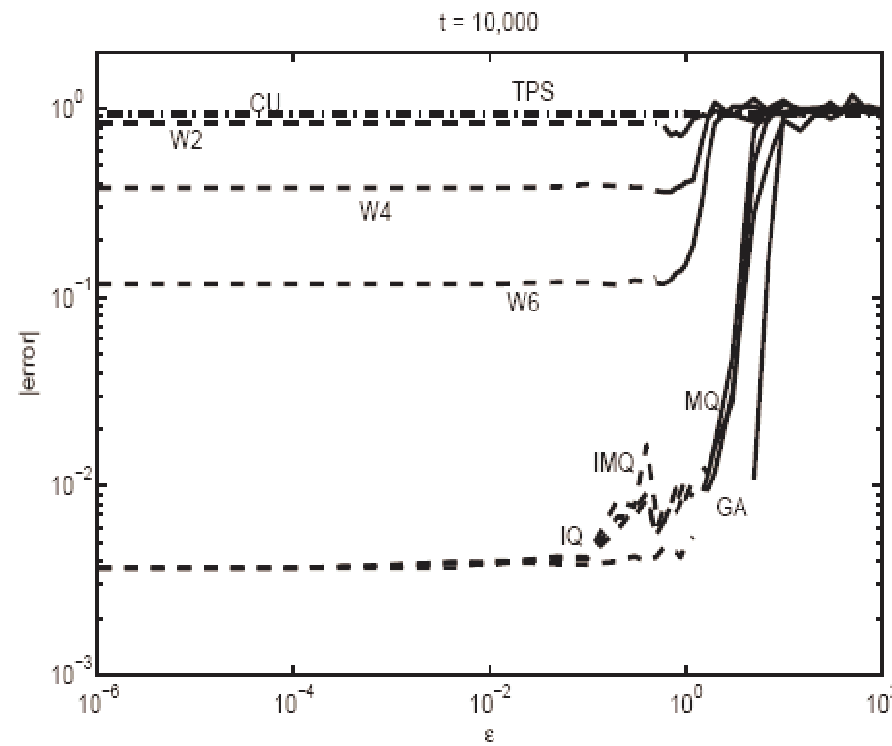




'Unrolled' spherical coordinate system

Initial condition: Cosine bell, discretized at n = 1849 'minimal energy' nodes

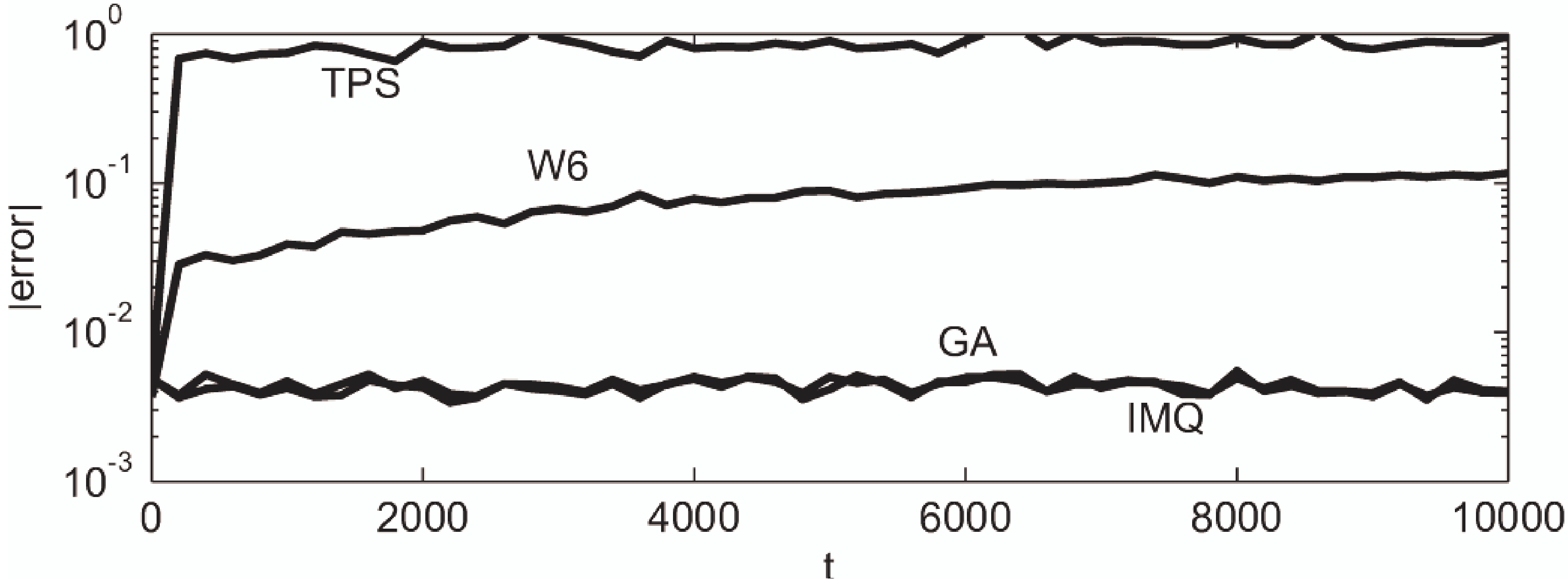
One full rotation corresponds to $t = 2\pi$



Some observations:

- Smooth global RBF types give almost identical results once ε is small enough.
- Smooth RBFs important even if the convected solution is not smooth.
- Robust results require ε some two orders of magnitude below what RBF-Direct provides.

Error evolution up to time t = 10,000



Error for smooth RBF types does not increase with time (no trailing dispersive wake)

Operation counts for the RBF-Direct algorithm

Three main tasks (in case of RBF-Direct):

1. Given data ($\underline{x}_k$, f_k), k = 1, 2, ... , N, solve linear systems

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

$O(N^3)$ operations

2. Given λ_k, evaluate $s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|)$ at M different locations. $O(M\ N)$ operations

3. Perform matrix - vector multiplications $\begin{bmatrix} Lu \end{bmatrix} = \begin{bmatrix} & D & \end{bmatrix} \begin{bmatrix} u \end{bmatrix}$. $O(N^2)$ operations

All steps of very simple 'structure' (quite straightforward parallelization), but:

A wealth of opportunities are available for algorithms which both:

- reduce operation count
- reduce memory requirement

Fast RBF algorithms in cases of large ε

Surveyed for ex. in Fasshauer: Meshfree Approximation Methods with Matlab (World Scientific, 2007)

1. Non-uniform Fast Fourier Transform
2. Fast multipole method
3. Fast tree codes
4. Domain decomposition methods
5. Krylov-type iterations
6. Fast Gauss transform
7. The BFGP algorithm
8. Sparse matrix approaches based on compact RBFs

: ??????? (more algorithms are bound to be discovered)

Stable RBF algorithms in cases of small ε

1. Contour-Padé — Severe limitation on number of nodes ($N \lesssim 20$ in 1-D, $N \lesssim 200$ in 2-D)
2. RBF-QR — Works for thousands of nodes on the sphere

: ??????? (more algorithms are bound to be discovered)

Challenge: Find an algorithm that combines high speed with numerically stability

RBF-generated Finite Differences (FD)

- Resolves cost and conditioning issues
- All approximations 'local' - much less message passing in parallel computing environments

but

- Algebraic instead of spectral accuracy

Conclusions

Established:

- RBFs can be seen as a generalization of PS methods to arbitrarily shaped domains.
- RBFs can offer excellent accuracy also over very long integration times.
- The near-flat basis function regime (ε small) is found to be of particular interest, and the first genuinely stable numerical algorithms for this case are emerging.
- After ill-conditioning has been eliminated, the next accuracy-limiting factor has been identified (found to be related to the polynomial Runge phenomenon).
- Many types of fast algorithms exist - however so far only for large ε.

Current research issues:

- Compare RBFs against alternative methods for standard test problems.
- Explore further the combination of spectral accuracy with local node refinement.
- Find RBF algorithms that combine high speed with numerical stability (for small ε).
- Develop further the concept of RBF-generated FD formulas.

If you had access to a peta-scale computing system, what would you do with it?