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

Bengt Fornberg
University of Colorado
Department of Applied Mathematics

Natasha Flyer
NCAR
Institute for Mathematics Applied to Geosciences

Collaborators:
Cécile Piret, Grady Wright, Erik Lehto
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 \( \phi(r) = \sqrt{1 + (\epsilon r)^2} \)
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
Given scattered data \((x_k, f_k), k = 1, 2, \ldots, N\), the coefficients \(\lambda_k\) in 
\[
s(x) = \sum_{k=1}^{N} \lambda_k \phi(\|x - x_k\|)
\]
are found by collocation: 
\[
s(x_k) = f_k, \quad k = 1, 2, \ldots, N:
\]

\[
\begin{bmatrix}
\phi(\|x_1 - x_1\|) & \phi(\|x_1 - x_2\|) & \cdots & \phi(\|x_1 - x_N\|) \\
\phi(\|x_2 - x_1\|) & \phi(\|x_2 - x_2\|) & \cdots & \phi(\|x_2 - x_N\|) \\
\vdots & \vdots & \ddots & \vdots \\
\phi(\|x_N - x_1\|) & \phi(\|x_N - x_2\|) & \cdots & \phi(\|x_N - x_N\|)
\end{bmatrix}
\begin{bmatrix}
\lambda_1 \\
\vdots \\
\lambda_N
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
\vdots \\
f_N
\end{bmatrix}
\]

**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 \(\lambda_k\): \(O(N^3)\) operations
    - Evaluate interpolant at \(M\) locations: \(O(MN)\) 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: \[
\frac{\partial h}{\partial t} = -\left( U(\alpha, \theta, \varphi, t) \cdot \nabla \right) h
\]
\[
\Leftrightarrow \quad \frac{\partial h}{\partial t} = -\left( \frac{u(t)}{\cos \theta} D_N^\theta + v(t)D_N^\theta \right) h
\]

\(D_N^\theta\) and \(D_N^\theta\) are discrete RBF differentiation matrices:
- Free of Pole Singularities
- Error Invariant of \(\alpha\), 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

![Diagram showing a contour plot with latitude and longitude scales. The plot contains a pattern with two large structures in the center, surrounded by concentric lines representing different values. The color bar on the right indicates magnitude, with values ranging from $10^{-4}$ to 3.]}
## Comparison With Other Methods

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

References:


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

<table>
<thead>
<tr>
<th>Exact-Numerical</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4 $\times$ 10^{-4}</td>
<td></td>
</tr>
</tbody>
</table>

(white $< 10^{-5}$)
## Comparison with Other Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of grid points</th>
<th>Time step</th>
<th>Relative $l_2$ error in $h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF [1]</td>
<td>748 $(28^2)$</td>
<td>20 minutes</td>
<td>$4.96 \cdot 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>1849 $(43^2)$</td>
<td>12 minutes</td>
<td>$3.47 \cdot 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>3136 $(56^2)$</td>
<td>10 minutes</td>
<td>$8.91 \cdot 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>4096 $(64^2)$</td>
<td>8 minutes</td>
<td>$2.57 \cdot 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td>5041 $(71^2)$</td>
<td>6 minutes</td>
<td>$3.84 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>Spherical Harmonics [2]</td>
<td>8192 $(T42)$</td>
<td>20 minutes*</td>
<td>$2 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>Double Fourier [3]</td>
<td>2048</td>
<td>6 minutes</td>
<td>$3.9 \cdot 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>3 minutes</td>
<td>$8.2 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>Spectral Elements [4]</td>
<td>6144</td>
<td>90 seconds</td>
<td>$6.5 \cdot 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>24576</td>
<td>45 seconds</td>
<td>$4 \cdot 10^{-5}$</td>
</tr>
</tbody>
</table>

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

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

(Fornberg and Zuev, 2007)

<table>
<thead>
<tr>
<th>(non-periodic)</th>
<th>(n = 10^1)</th>
<th>(n = 10^2)</th>
<th>(n = 10^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>18</td>
<td>198</td>
<td>1,998</td>
</tr>
<tr>
<td>2-D</td>
<td>26</td>
<td>280</td>
<td>2,826</td>
</tr>
<tr>
<td>3-D</td>
<td>34</td>
<td>360</td>
<td>3,632</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(non-periodic)</th>
<th>(n = 10^1)</th>
<th>(n = 10^2)</th>
<th>(n = 10^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolves in each direction</td>
<td>about 5 modes</td>
<td>about 50 modes</td>
<td>about 500 modes</td>
</tr>
</tbody>
</table>
Why are flat (or near-flat) RBFs interesting?

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

'Toy-problem' example: 41 node MQ interpolation of

$$f(x_1, x_2) = \frac{59}{67 + (x_1 + \frac{1}{7})^2 + (x_2 - \frac{1}{11})}$$

- 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 $\varepsilon$.

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

**Numerical computations for small values of $\varepsilon$**

- **High precision arithmetic**
  
  It is known exactly how the condition number varies with domain type, $N$, $\varepsilon$. 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 $f$ to $s(x, \varepsilon)$ that does not go via the ill-conditioned $\hat{\lambda}$.

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

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

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>$\mu = 0$</th>
<th>$\mu = 1$</th>
<th>$\mu = 2$</th>
<th>$\mu = 3$</th>
<th>$\mu = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>-2</td>
<td>$-\frac{1}{2}\sqrt{\frac{3}{2\pi}}y$</td>
<td>$\frac{1}{2}\sqrt{\frac{3}{2\pi}}z$</td>
<td>$-\frac{1}{2}\sqrt{\frac{3}{2\pi}}x$</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>-1</td>
<td>$\frac{1}{2}\sqrt{\frac{15}{2\pi}}xy$</td>
<td>$-\frac{1}{2}\sqrt{\frac{15}{2\pi}}zy$</td>
<td>$\frac{1}{4}\sqrt{\frac{5}{2\pi}}(3z^2 - 1)$</td>
<td>$-\frac{1}{2}\sqrt{\frac{15}{2\pi}}zx$</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- 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:

\[ \phi(||x - x_i||) = \sum_{\mu=0}^{\infty} \sum_{v=-\mu}^{\mu} \left[ \varepsilon^{2\mu} c_{\mu,\varepsilon} Y_\mu^v(x_i) \right] Y_\mu^v(x) \]

where, for example

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

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

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

- There is no loss of accuracy in computing \( c_{\mu,\varepsilon} Y_\mu^v(x_i) \), 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: 1849 minimal energy nodes  Error: RBF-Direct vs. RBF-QR

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

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

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

One full rotation corresponds to \( t = 2\pi \)

Some observations:

- Smooth global RBF types give almost identical results once \( \epsilon \) is small enough.

- Smooth RBFs important even if the convected solution is not smooth.

- Robust results require \( \epsilon \) 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 \((x_k, f_k), k = 1, 2, \ldots, N\), solve linear systems
   \[
   \begin{bmatrix}
   \phi(||x_1 - x_1||) & \phi(||x_1 - x_2||) & \cdots & \phi(||x_1 - x_N||) \\
   \phi(||x_2 - x_1||) & \phi(||x_2 - x_2||) & \cdots & \phi(||x_2 - x_N||) \\
   \vdots & \vdots & \ddots & \vdots \\
   \phi(||x_N - x_1||) & \phi(||x_N - x_2||) & \cdots & \phi(||x_N - x_N||)
   \end{bmatrix}
   \begin{bmatrix}
   \lambda_1 \\
   \vdots \\
   \lambda_N
   \end{bmatrix}
   =
   \begin{bmatrix}
   f_1 \\
   \vdots \\
   f_N
   \end{bmatrix}
   \]

   \(O(N^3)\) operations

2. Given \(\lambda_k\), evaluate \(s(x) = \sum_{k=1}^{N} \lambda_k \phi(||x - x_k||)\) at \(M\) different locations.

   \(O(M N)\) operations

3. Perform matrix - vector multiplications
   \[
   \begin{bmatrix}
   Lu \\
   D
   \end{bmatrix}
   =
   \begin{bmatrix}
   u
   \end{bmatrix}
   \]

   \(O(N^2)\) 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 $\varepsilon$

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 $\varepsilon$

1. Contour-Padé
   Severe limitation on number of nodes ($N \leq 20$ in 1-D, $N \leq 200$ 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 ($\varepsilon$ 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 $\varepsilon$.

**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 $\varepsilon$).
- 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?*