Radial basis function partition of unity methods for PDEs

Elisabeth Larsson, Scientific Computing, Uppsala University

Credit goes to a number of RBF-PUM collaborators

Localized Kernel-Based Meshless Methods for Partial Differential Equations, ICERM, Aug 8, 2017
Outline

Introduction

RBF partition of unity methods for PDEs

Theoretical results

Numerical results
  RBF-QR
    Convergence
    Robustness
  3-D results
  Cost
  Adaptivity

Summary
Motivation for RBF-PUM

Global RBF approximation

+ Ease of implementation in any dimension.
+ Flexibility with respect to geometry.
+ Potentially spectral convergence rates.
- Computationally expensive for large problems.

RBF-PUM

▷ Local RBF approximations on patches are blended into a global solution using a partition of unity.
▷ Provides spectral or high-order convergence.
▷ Solves the computational cost issues.
▷ Allows for local adaptivity.

The RBF partition of unity method

Global approximation

\[ \tilde{u}(x) = \sum_{j=1}^{P} w_j(x) \tilde{u}_j(x) \]

PU weight functions

Generate weight functions from compactly supported $C^2$ Wendland functions

\[ \psi(\rho) = (4\rho + 1)(1 - \rho)^4 \]

using Shepard’s method

\[ w_i(x) = \frac{\psi_i(x)}{\sum_{j=1}^{M} \psi_j(x)}. \]

Cover

Each \( x \in \Omega \) must be in the interior of at least one \( \Omega_j \). Patches that do not contain unique points are pruned.
Differentiating RBF-PUM approximations

Applying an operator globally

\[
\Delta \tilde{u} = \sum_{i=1}^{M} \Delta w_i \tilde{u}_i + 2 \nabla w_i \cdot \nabla \tilde{u}_i + w_i \Delta \tilde{u}_i
\]

Local differentiation matrices

Let \( \underline{u}_i \) be the vector of nodal values in patch \( \Omega_i \), then

\[
\underline{u}_i = \underline{A} \underline{\lambda}_i, \text{ where } A_{ij} = \phi_j(x_i) \quad \Rightarrow
\]

\[
\mathcal{L} \underline{u}_i = \underline{A}^\mathcal{L} \underline{A}^{-1} \underline{u}_i, \text{ where } A_{ij}^\mathcal{L} = \mathcal{L} \phi_j(x_i).
\]

The global differentiation matrix

Local contributions are added into the global matrix.
An RBF-PUM collocation method

Choices & Implications

- Nodes and evaluation points coincide.
  
  *Square matrix, iterative solver available (Heryudono, Larsson, Ramage, von Sydow 2015).*

- Global node set.
  
  *Solutions $\tilde{u}_i(x_k) = \tilde{u}_j(x_k)$ for $x_k$ in overlap regions.*

- Patches are cut by the domain boundary.
  
  *Potentially strange shapes and lowered local order.*
An RBF-PUM least squares method

Choices & Implications

- Each patch has an identical node layout.
  
  *Computational cost for setup is drastically reduced.*

- Evaluation nodes are uniform.
  
  *Easy to generate both local and global high quality node sets.*

- Patches have nodes outside the domain.
  
  *Good for local order, but requires denser evaluation points.*

E. Larsson, Aug 8, 2017  (7 : 28)
The RBF-PUM interpolation error

\[ E_\alpha = D^\alpha (I(u) - u) = \sum_{j=1}^{M} \sum_{|\beta| \leq |\alpha|} \binom{\alpha}{\beta} D^\beta w_j D^{\alpha-\beta} (I(u_j) - u_j) \]

The weight functions

For \( C^k \) weight functions and \(|\alpha| \leq k\)

\[ \| D^\alpha w_j \|_{L_\infty(\Omega_j)} \leq \frac{C_\alpha}{H_j^{\alpha}}, \quad H_j = \text{diam}(\Omega_j). \]

The local RBF interpolants (Gaussians)

Define the local fill distance \( h_j \) (Rieger, Zwicknagl 2010)

\[ \| D^\alpha (I(u_j) - u_j) \|_{L_\infty(\tilde{\Omega}_j)} \leq c_{\alpha,j} h_j^{m_j - \frac{d}{2} - |\alpha|} \| u_j \|_{\mathcal{N}(\tilde{\Omega}_j)}, \]

\[ \| D^\alpha (I(u_j) - u_j) \|_{L_\infty(\tilde{\Omega}_j)} \leq e^{\gamma_{\alpha,j} \log(h_j)/\sqrt{h_j}} \| u_j \|_{\mathcal{N}(\tilde{\Omega}_j)}. \]
RBF-PUM interpolation error estimates

Algebraic estimate for $H_j/h_j = c$

$$\| \mathcal{E}_\alpha \|_{L_\infty(\Omega)} \leq K \max_{1 \leq j \leq M} C_j H_j^{m_j - \frac{d}{2} - |\alpha|} \| u \|_{\mathcal{N}(\tilde{\Omega}_j)}$$

- $K$ — Maximum # of $\Omega_j$ overlapping at one point
- $m_j$ — Related to the local # of points
- $\tilde{\Omega}_j = \Omega_j \cap \Omega$

Spectral estimate for fixed partitions

$$\| \mathcal{E}_\alpha \|_{L_\infty(\Omega)} \leq K \max_{1 \leq j \leq M} C e^{\gamma_j \log(h_j)/\sqrt{h_j}} \| u \|_{\mathcal{N}(\tilde{\Omega}_j)}$$

Implications

- Bad patch reduces global order.
- Two refinement modes.
- Guidelines for adaptive refinement.
Error estimate for PDE approximation

The PDE estimate

\[ \| \tilde{u} - u \|_{L^\infty(\Omega)} \leq C_P \mathcal{E}_\mathcal{L} + C_P \| L_{\cdot,\cdot} L_{Y,\cdot}^+ \|_{\infty} (C_M \delta_M + \mathcal{E}_\mathcal{L}), \]

where \( C_P \) is a well-posedness constant and \( C_M \delta_M \) is a small multiple of the machine precision.

Implications

- Interpolation error \( \mathcal{E}_\mathcal{L} \) provides convergence rate.
- Norm of inverse/pseudoinverse can be large.
- Matrix norm better with oversampling.
- Finite precision accuracy limit involves matrix norm.

*Follows strategies from Schaback (2007) and Schaback (2016)*
Does RBF-PUM require stable methods?

In order to achieve convergence we have two options

- Refine patches such that diameter $H$ decreases.
- Increase node numbers such that $N_j$ increases.
- In both cases, theory assumes $\varepsilon$ fixed.

The effect of patch refinement

\[ H = 1, \; \varepsilon = 4 \quad H = 0.5, \; \varepsilon = 4 \quad H = 0.25, \; \varepsilon = 4 \]

The RBF–QR method: Stable as $\varepsilon \to 0$ for $N \gg 1$

Effectively a change to a stable basis.

Effects on the local matrices

Local contribution to a global Laplacian

\[ L_j = (W_j^\Delta A_j + 2W_j^\nabla \odot A_j^\nabla + W_jA_j^\Delta)A_j^{-1}. \]

Typically: \( A_j \) ill-conditioned, \( L_j \) better conditioned.

RBF-QR for accuracy

- Stable for small RBF shape parameters \( \varepsilon \)
- Change of basis
  \[ \tilde{A} = AQR_1^{-T}D_1^{-T} \]
- Same result in theory
  \[ \tilde{A}^L \tilde{A}^{-1} = A^L A^{-1} \]
- More accurate in practice

Relative error in \( A_j^\Delta A_j^{-1} \) without RBF-QR

\[ 10 \quad 10^{-2} \quad 10^{-5} \quad 10^{-10} \]

\[ \varepsilon \quad 10^{-1} \quad 10^{0} \]

N=10 N=20 N=40
Poisson test problems in 2-D

Domain $\Omega = [-2, 2]^2$.

Uniform nodes in the collocation case.

$$u_R(x, y) = \frac{1}{25x^2+25y^2+1}$$

$$u_T(x, y) = \sin(2(x-0.1)^2)\cos((x-0.3)^2)+\sin^2((y-0.5)^2)$$
Error results with and without RBF–QR

- Least squares RBF-PUM
- Fixed shape $\varepsilon = 0.5$ or scaled such that $\varepsilon h = c$
- Left: $5 \times 5$ patches Right: 55 points per patch

With RBF–QR better results for $H/h$ large.
Scaled approach good until saturation.
Convergence as a function of patch size

Collocation (dashed lines) and Least Squares (solid lines).

- Points per patch $n = 28, 55, 91$.
- Theoretical rates $p = 4, 7, 10$.
- Numerical rates $p \approx 3.9, 6.9, 9.8$. 

E. Larsson, Aug 8, 2017 (15:28)
Spectral convergence for fixed patches

Collocation (dashed lines) and Least Squares (solid lines).

LS-RBF-PU is significantly more accurate due to the constant number of nodes per patch.
Robustness and large scale problems

The global error estimate
$$\|\tilde{u} - u\|_{L_{\infty}(\Omega)} \leq C_P\varepsilon_L + C_P\|L_{\cdot,\cdot}L_{Y,\cdot}^{\cdot}\|_{\infty}(C_M\delta_M + \varepsilon_L)$$

The dark horse is the 'stability matrix norm'

- The stability norm is related to conditioning.
- In the collocation case, $\|L_{X,\cdot}^{-1}\|$ grows with $N$.
- How does it behave with least squares?
The stability matrix norm: fill distance

- Fixed patch size with $10 \times 10$ patches.
- Oversampling $M/N = 1.1, 1.2, 1.5$

Collocation (dashed) and Least Squares (solid)

- Stability norm grows exponentially as $h$ decreases.
- Oversampling reduces the norm.
- Collocation is less robust.
Stability norm: Patch size

- Fixed number of points per patch $n = 28, 55, 91$
- Results as a function of patch diameter $H$

Collocation (dashed) and LS (solid)

- The norm does not grow for LS-RBF-PUM (!)
Stability norm: Oversampling

- Fixed number of local points \( n = 55 \).
- Patches: \( 5 \times 5, 10 \times 10, 15 \times 15 \)

Collocation (dashed) and LS (solid)

- Oversampling provides stability (at a cost).
- For robustness in \( h \), \( M/N \) needs to grow with \( N \).
Poisson test problems in 3-D

\[ u(x) = \sin \left( \frac{\pi(x_1 - 0.5)x_3}{\log(x_2 + 3)} \right) \]
Convergence in 3-D

- $n = 20, 35, 56, 84, 120, 165$ points per patch

- Convergence order is a bit better than expected.
- Refinement modes work as expected.
Experiments with fixed $n = 28, 55, 91$.  
For a fixed problem size, RBF-PU-LS involves more work, but yields a smaller error.  
Overall, the LS approach is the winner.  
Strategy: Points per patch adjusted to tolerance.
Solution using iterative methods

So far implemented for the collocation approach. First step: Re-ordering of the unknowns to improve structure.

**Patches:** Preceded and followed by a neighbour.

**Nodes** \( x_k \): Define home patch \( \Omega_j \) such that \( w_j \geq w_i(x_k) \).

**Within patch:** Sub-order according to node memberships.

*Heryudono, Larsson, Ramage, and von Sydow (2015)*
Matrix structures with snake order

- The patch order fills in gaps in the band.
- The node order minimizes the central band width.
- Chosen preconditioner, ILU(0) of central band.

Structure 2-D

Structure 3-D

E. Larsson, Aug 8, 2017 (25 : 28)
### Results for the iterative method contd.

#### Results in 2-D with Halton nodes

<table>
<thead>
<tr>
<th>(N)</th>
<th># it no prec</th>
<th># it ILU(0)</th>
<th>Time gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>436</td>
<td>189</td>
<td>72</td>
<td>3.1</td>
</tr>
<tr>
<td>583</td>
<td>209</td>
<td>91</td>
<td>2.4</td>
</tr>
<tr>
<td>681</td>
<td>231</td>
<td>112</td>
<td>2.7</td>
</tr>
<tr>
<td>884</td>
<td>262</td>
<td>125</td>
<td>2.3</td>
</tr>
<tr>
<td>1090</td>
<td>295</td>
<td>135</td>
<td>3.0</td>
</tr>
</tbody>
</table>

#### Results in 3-D with Halton nodes

<table>
<thead>
<tr>
<th>(N)</th>
<th># it no prec</th>
<th># it ILU(0)</th>
<th>Time gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>4206</td>
<td>100</td>
<td>47</td>
<td>1.1</td>
</tr>
<tr>
<td>9534</td>
<td>166</td>
<td>69</td>
<td>1.5</td>
</tr>
<tr>
<td>18101</td>
<td>224</td>
<td>85</td>
<td>2.5</td>
</tr>
<tr>
<td>99568</td>
<td>454</td>
<td>174</td>
<td>4.6</td>
</tr>
<tr>
<td>231284</td>
<td>755</td>
<td>417</td>
<td>2.9</td>
</tr>
</tbody>
</table>

*Memory gain not to be forgotten.*

E. Larsson, Aug 8, 2017   (26 : 28)
Current project: Adaptivity

With Alfa Heryudono, Danilo Stocchino and Stefano De Marchi

- Box structure helps book keeping.
- Overlaps complicate things.
- Evaluation points placed using node placing algorithm.

Fornberg & Flyer (2015)
Summary

Results

- RBF-PUM is a flexible tool for solving PDEs.
- By using template patches, LS RBF-PUM becomes computationally efficient.
- LS RBF-PUM is numerically stable for large scale problems.

Things to do

- Change patch type to reduce unnecessary overlap.
- Adaptive algorithms based on LS RBF-PUM.
- Time-stability for LS RBF-PUM.
  
  *For collocation and hyperbolic PDEs on the sphere, see poster by Igor Tominec et al.*