UPPSALA UNIVERSITET

RBF–QR



# The RBF-QR method and its applications—A tutorial in two parts

Elisabeth Larsson

*with thanks to numerous co-investigators and colleagues*

Division of Scientific Computing
Department of Information Technology
Uppsala University

Dolomites Research Week on Approximation 2015

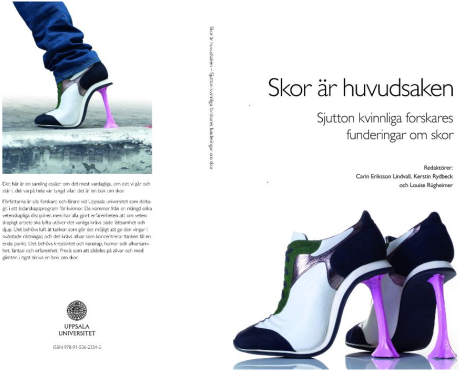
UPPSALA
UNIVERSITET

RBF–QR



# Inofficial competition

Produce the *most beautiful picture* by modifying the RBF-QR demo MATLAB codes. The winner can get a copy of the English version of this book or eternal glory...




Head over Heels—Seventeen women scientist's thoughts on shoes.

# Outline

# Short introduction to (global) RBF methods

UPPSALA UNIVERSITET

RBF–QR

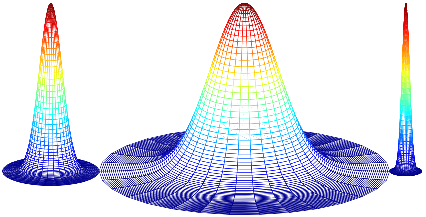


Basis functions: $\phi_j(\underline{x}) = \phi(\|\underline{x} - \underline{x}_j\|)$. Translates of one single function rotated around a center point.

Example: Gaussians
$\phi(\varepsilon r) = \exp(-\varepsilon^2 r^2)$

Approximation:
$s_\varepsilon(\underline{x}) = \sum_{j=1}^{N} \lambda_j \phi_j(\underline{x})$

Collocation:
$s_\varepsilon(\underline{x}_i) = f_i \;\Rightarrow\; A\underline{\lambda} = \underline{f}$




Advantages:

- Flexibility with respect to geometry.
- As easy in $d$ dimensions.
- Spectral accuracy / exponential convergence.
- Continuosly differentiable approximation.

# `demo1.m`

(RBF interpolation in 1-D)

UPPSALA UNIVERSITET

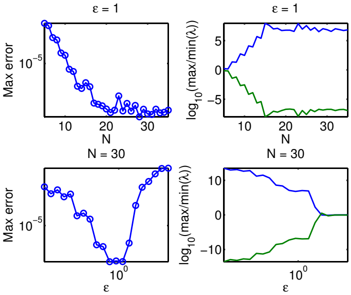
RBF–QR



# Observations from the results of `demo1.m`

- As $N$ grows for fixed $\varepsilon$, convergence stagnates.
- As $\varepsilon$ decreases for fixed $N$, the error blows up.
- $\lambda_{\text{min}} = -\lambda_{\text{max}}$ means cancellation.
- Coefficients $\lambda \to \infty$ means that $\text{cond}(A) \to \infty$.

- For small $\varepsilon$, the RBFs are nearly flat, and almost linearly dependent. That is, they form a **bad basis**.



UPPSALA
UNIVERSITET



# Why is it interesting to use small values of $\varepsilon$?

## Driscoll & Fornberg 2002

Somewhat surprisingly, in 1-D for small $\varepsilon$

$$s(x,\varepsilon) = P_{N-1}(x) + \varepsilon^2 P_{N+1}(x) + \varepsilon^4 P_{N+3}(x) + \cdots,$$

where $P_j$ is a polynomial of degree $j$ and $P_{N-1}(x)$ is the Lagrange interpolant.

## Implications

- It can be shown that $\mathrm{cond}(A) \sim \mathcal{O}(N\varepsilon^{-2(N-1)})$, but the limit interpolant is well behaved.
- It is the intermediate step of computing $\underline{\lambda}$ that is ill-conditioned.
- By choosing the corresponding nodes, the flat RBF limit reproduces pseudo-spectral methods.
- This is a **good** approximation **space**.

UPPSALA UNIVERSITET

RBF–QR



# The multivariate flat RBF limit

## Larsson & Fornberg 2005, Schaback 2005

In n-D the flat limit can either be

$$s(\underline{x},\varepsilon) = P_K(\underline{x}) + \varepsilon^2 P_{K+2}(\underline{x}) + \varepsilon^4 P_{K+4}(\underline{x}) + \cdots,$$

where $\begin{pmatrix} (K-1)+d \\ d \end{pmatrix} < N \leq \begin{pmatrix} K+d \\ d \end{pmatrix}$ and $P_K$ is *a* polynomial interpolant or

$$\begin{aligned} s(\underline{x},\varepsilon) &= \varepsilon^{-2q} P_{M-2q}(\underline{x}) + \varepsilon^{-2q+2} P_{M-2q+2}(\underline{x}) + \cdots \\ &+ P_M(\underline{x}) + \varepsilon^2 P_{M+2}(\underline{x}) + \varepsilon^4 P_{M+4}(\underline{x}) + \cdots. \end{aligned}$$

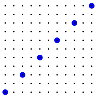
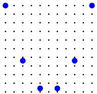
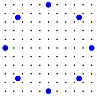
The questions of uniqueness and existence are connected with multivariate polynomial uni-solvency.

## Schaback 2005

Gaussian RBF limit interpolants always converge to the de Boor/Ron least polynomial interpolant.

# The multivariate flat RBF limit: Divergence

Necessary condition: $\exists\ Q(\underline{x})$ of degree $N_0$ such that $Q(\underline{x}_j)=0,\ j=1,\ldots,N$.
Then divergence as $\varepsilon^{-2q}$ may occur, where
$q=\lfloor (M-N_0)/2 \rfloor$ and $M=$ min non-degenerate degree.

| Points | $Q$ | $N_0$ | Basis | $M$ | $q$ |
|---|---|---|---|---|---|
|  | $x-y$ | 1 | $1,\ x,\ x^2,$ $x^3,\ x^4,\ x^5$ | 5 | 2 |
|  | $x^2-y-1$ | 2 | $1,\ x,\ y,\ xy,$ $y^2xy^2$ | 3 | 0 |
|  | $x^2+y^2-1$ | 2 | $1,\ x,\ y,\ x^2,\ xy,$ $x^3,\ x^2y,\ x^4$ | 4 | 1 |

Divergence actually only occurs for the first case as $\varepsilon^{-2}$.

# The multivariate flat RBF limit, contd

## Schaback 2005, Fornberg & Larsson 2005

**Example:** In two dimensions, the eigenvalues of $A$ follow a pattern: $\mu_1 \sim \mathcal{O}(\varepsilon^0)$, $\mu_{2,3} \sim \mathcal{O}(\varepsilon^2)$, $\mu_{4,5,6} \sim \mathcal{O}(\varepsilon^4)$,...

In general, there are $\begin{pmatrix} k+n-1 \\ n-1 \end{pmatrix} = \frac{(k+1)\cdots(k+n-1)}{(n-1)!}$ eigenvalues $\mu_j \sim \mathcal{O}(\varepsilon^{2k})$ in $n$ dimensions.

## Implications

- There is an opportunity for pseudo-spectral-like methods in $n$-D.
- There is no amount of variable precision that will save us.
- For "smooth" functions, a small $\varepsilon$ can lead to very high accuracy.

UPPSALA UNIVERSITET

RBF–QR

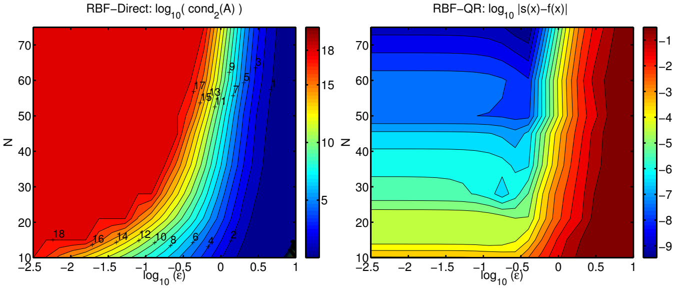


# `demo2.m`

(Conditioning and errors)

# Comments on the results of `demo2`

- Error is small where condition is high and vice versa.
- Interesting region only reachable with stable method.
- Best results for small $\varepsilon$.




*Teaser: Conditioning for RBF–QR is perfect...*

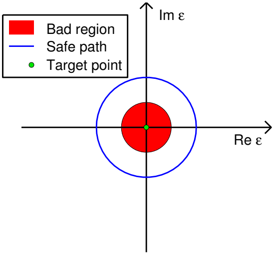
# The Contour-Padé method

## Fornberg & Wright 2004

- Think of $\varepsilon$ as a complex variable.
- The limit $\varepsilon = 0$ is a removable singularity.
- Complex $\varepsilon$ for which $A$ is singular lead to poles.
- Pole location only depend on the location of nodes.

## Example

- Evaluate $f(\varepsilon) = \frac{1-\cos(\varepsilon)}{\varepsilon^2}$
- Numerically unstable.
- Removable singularity at 0.
- Compute $f(0)$ as average of $f(\varepsilon)$ around "safe path".



# The Contour-Padé method: Algorithm

- Compute $s(\underline{x},\varepsilon) = A_e A^{-1} f$ at $M$ points around a "safe path" (circle).
- Inverse FFT of the $M$ values gives a Laurent expansion

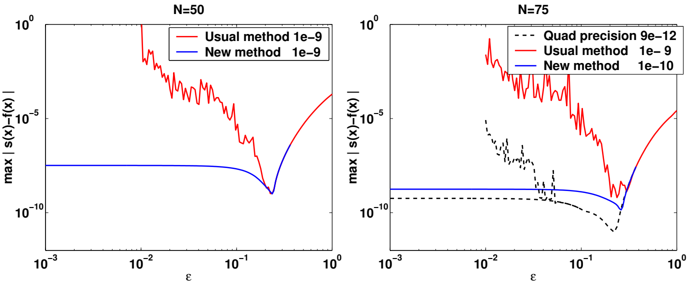
$$u(\underline{x}) = \underbrace{\ldots + s_{-2}(\underline{x})\varepsilon^{-4} + s_{-1}(\underline{x})\varepsilon^{-2}}_{\text{Needs to be converted}} + s_0(\underline{x}) + s_1(\underline{x})\varepsilon^2 + s_2(\underline{x})\varepsilon^4 + \ldots$$

- Convert the negative power expansion into Padé form and find the correct number of poles and their locations

$$s_{-1}\varepsilon^{-2} + s_{-2}\varepsilon^{-4} + \ldots = \frac{p_1\varepsilon^{-2} + \cdots + p_m\varepsilon^{-2m}}{1 + q_1\varepsilon^{-2} + \cdots + q_n\varepsilon^{-2n}}.$$

- Evaluate $u(\underline{x})$ using Taylor + Padé for any $\varepsilon$ inside the circle.

# The Contour-Padé method: Results




- Stable computation for all $\varepsilon$ with Contour-Padé.
- Limited number of nodes, otherwise general.
- Expensive to compute $A^{-1}$ at $M$ points.
- Tricky to find poles.
- Modern efficient version RBF-RA, see Grady Wright.

# Expansions of (Gaussian) RBFs

## On the surface of the sphere

*Hubbert & Baxter 2001*

For different RBFs there are expansions

$$\phi(\|\underline{x}-\underline{x}_k\|)=\sum_{j=0}^{\infty}\varepsilon^{2j}\sum_{m=-j}^{j}c_{j,m}Y_j^m(\underline{x})$$

## Cartesian space, polynomial expansion

For Gaussians

$$\begin{aligned}\phi(\|\underline{x}-\underline{x}_k\|) &= e^{-\varepsilon^2(\underline{x}-\underline{x}_k)\cdot(\underline{x}-\underline{x}_k)}\\ &= e^{-\varepsilon^2(\underline{x}\cdot\underline{x})}e^{-\varepsilon^2(\underline{x}_k\cdot\underline{x}_k)}e^{2\varepsilon^2(\underline{x}\cdot\underline{x}_k)}\\ &= e^{-\varepsilon^2(\underline{x}\cdot\underline{x})}e^{-\varepsilon^2(\underline{x}_k\cdot\underline{x}_k)}\sum_{j=0}^{\infty}\varepsilon^{2j}\frac{2^j}{j!}(\underline{x}\cdot\underline{x}_k)^j\end{aligned}$$

# Expansions of (Gaussian) RBFs contd

## Mercer expansion (Mercer 1909)

For a positive definite kernel $K(\underline{x}, \underline{x}_k) = \phi(\|\underline{x} - \underline{x}_k\|)$, there is an expansion

$$\phi(\|\underline{x} - \underline{x}_k\|) = \sum_{j=0}^{\infty} \lambda_j \varphi_j(\underline{x}) \varphi_j(\underline{x}_k),$$

where $\lambda_j$ are positive eigenvalues, and $\varphi_j(\underline{x})$ are eigenfunctions of an associated compact integral operator.

UPPSALA UNIVERSITET

RBF–QR



# The RBF-QR method on the sphere

*Fornberg & Piret 2007*

$$\phi(\|\underline{x}-\underline{x}_k\|)=\sum_{j=0}^{\infty}\varepsilon^{2j}\sum_{m=-j}^{j}c_{j,m}Y_j^m(\underline{x})$$

The number of SPH functions/power matches the RBF eigenvalue pattern on the sphere.

If we collect RBFs and expansion functions in vectors, and coefficients in the matrix $B$, we have a relation

$$\Phi(\underline{x})=B\cdot Y=Q\cdot E\cdot R\cdot Y(\underline{x})$$

The new basis $\Psi(\underline{x})=R\cdot Y(\underline{x})$ spans the same space as $\Phi(\underline{x})$, but the ill-conditioning has been absorbed in $E$.

# The RBF-QR method in Cartesian space

*Fornberg, Larsson, Flyer 2011*

## The expansion of the Gaussian

$$\phi(\|\underline{x}-\underline{x}_k\|) = e^{-\varepsilon^2(\underline{x}\cdot\underline{x})}e^{-\varepsilon^2(\underline{x}_k\cdot\underline{x}_k)}\sum_{j=0}^{\infty}\varepsilon^{2j}\frac{2^j}{j!}(\underline{x}\cdot\underline{x}_k)^j$$

+ The number of expansion functions for each power of $\varepsilon$ matches the eigenvalue pattern in $A$.
– The expansion functions are the monomials.

## Better expansion functions in 2-D

- Change to polar coordinates.
- Trigs in the angular direction are perfect.
- Necessary to preserve powers of $\varepsilon$ $\Rightarrow$ Partial conversion to Chebyshev polynomials.

# The RBF-QR method in Cartesian space contd



## New expansion functions

$$\begin{cases} T^c_{j,m}(\underline{x}) = e^{-\varepsilon^2 r^2} r^{2m} T_{j-2m}(r) \cos((2m+p)\theta), \\ T^s_{j,m}(\underline{x}) = e^{-\varepsilon^2 r^2} r^{2m} T_{j-2m}(r) \sin((2m+p)\theta), \end{cases}$$

## Matrix form of factorized expansion

Express $\Phi(\underline{x}) = (\phi(\|\underline{x}-\underline{x}_1\|), \ldots, \phi(\|\underline{x}-\underline{x}_N\|))^T$ in terms of expansion functions $T(\underline{x}) = (T^c_{0,0}, T^c_{1,0}, \ldots)^T$ as.

$$\Phi(\underline{x}) = C \cdot D \cdot T(\underline{x}),$$

where $c_{ij}$ is $\mathcal{O}(1)$ and $D = \mathrm{diag}(\mathcal{O}(\varepsilon^0, \varepsilon^2, \varepsilon^2, \varepsilon^4, \ldots))$.

Note that $C$ has an infinite number of columns etc.

# The RBF-QR method in Cartesian space contd

UPPSALA UNIVERSITET

RBF–QR



## The QR part

The coefficient matrix $C$ is QR-factorized so that $\Phi(\underline{x}) = Q \cdot \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \cdot T(\underline{x})$, where $R_1$ and $D_1$ are of size $(N \times N)$.

## The change of basis

Make the new basis (same space) close to $T$

$$\Psi(\underline{x}) = D_1^{-1} R_1^{-1} Q^H \Phi(\underline{x}) = \begin{bmatrix} I & \tilde{R} \end{bmatrix} \cdot T(\underline{x}).$$

## Analytical scaling of $\tilde{R} = D_1^{-1} R_1^{-1} R_2 D_2$

Any power of $\varepsilon$ in $D_1 \leq$ any power of $\varepsilon$ in $D_2 \Rightarrow$
Scaling factors $\mathcal{O}(\varepsilon^0)$ or smaller, truncation is possible.

# `demo3.m`

(RBF interpolation in 2-D with and without RBF–QR)

# Stable computation as $\varepsilon \to 0$ and $N \to \infty$

The RBF-QR method allows stable computations for small $\varepsilon$. *(Fornberg, Larsson, Flyer 2011)*

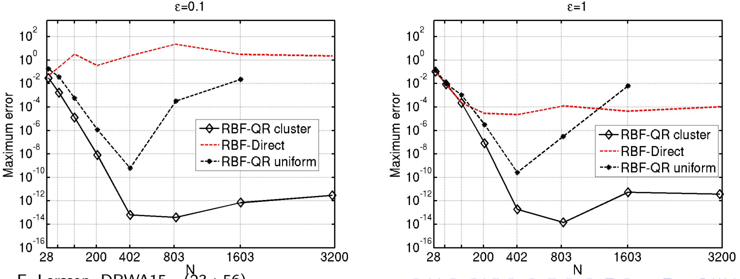
Consider a finite non-periodic domain.

## Theorem (Platte, Trefethen, and Kuijlaars 2010):

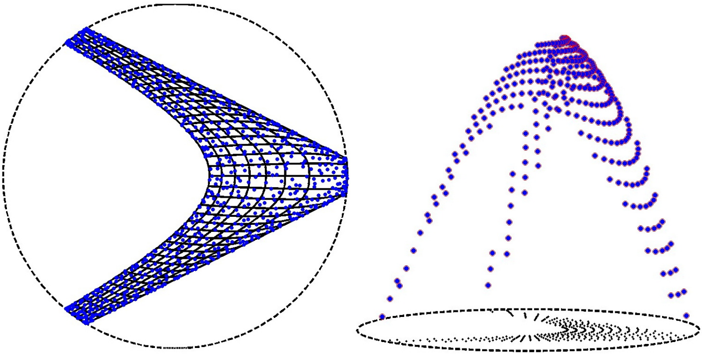
Exponential convergence on equispaced nodes $\Rightarrow$ exponential ill-conditioning.

## Solution #1:

Cluster nodes towards the domain boundaries.



# An RBF-QR example with clustered nodes in a non-trivial domain



$f(x, y) = \exp(-(x - 0.1)^2 - 0.5y^2)$
N=793 node points
Cosine-stretching towards each boundary
Maximum error 2.2e-10

# `demo4.m`

(RBF interpolation in 2-D with clustered nodes)

# Non-unisolvent nodes

- The expansion functions in the RBF-QR method are at the bottom polynomials.
- QR-factorization in the non-unisolvent case will find columns that are linearly dependent.
- Solved by 'selective pivoting' in the RBF-QR method. *(Larsson,Lehto,Heryudono, Fornberg 2013)*
- Sensitive to nearly non-unisolvent cases.
- Cannot always recover the true Gaussian limit.
- However, whatever limit is produced is well-behaved.

*This works in most cases, but it is not perfect.*

# Summary so far of the RBF–QR methods properties

- Special expansion functions needed. Natural for the sphere. Done in 1-D, 2-D, 3-D in Cartesian space.
- Works for small $\varepsilon$ (in relation to the domain size).
- Provides significant improvements in accuracy.
- Clustering needed for $N > 20$, 200, 2000 depending on dimensions.
- Sensitive to regular node layouts.
- Complexity $\mathcal{O}(N^3)$ as RBF-Direct.
- Gets more expensive for larger $\varepsilon$.

# The RBF–GA method

*Fornberg, Lehto, Powell 2013*

- Related approach
- Different expansion of the Gaussian RBF with remainder
- Does not have problems with regular node layouts.
- Only accurate for fairly small node sets.
- 2–4 times faster than RBF–QR.

# Brief survey of Mercer based methods

## Fasshauer & McCourt 2012

Eigenvalues and eigenfunctions in 1-D can be chosen as

$$\lambda_n = \sqrt{\frac{\alpha^2}{\alpha^2+\delta^2+\varepsilon^2}}\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{n-1},$$

$$\phi_n = \gamma_n e^{-\delta^2 x^2} H_{n-1}(\alpha\beta x),$$

where $\beta = \left(1+\left(\frac{2\varepsilon}{\alpha}\right)^2\right)^{\frac{1}{4}}$, $\gamma_n = \sqrt{\frac{\beta}{2^{n-1}\Gamma(n)}}$, $\delta^2 = \frac{\alpha^2}{2}(\beta^2-1)$.

- Eigenfunctions are orthogonal in a weighted norm.
- The QR-step is similar to that of previous methods.
- Tensor product form is used in higher dimensions $\Rightarrow$ The powers of $\varepsilon$ do not match the eigenvalues of $A$.
- New parameter $\alpha$ to tune.

UPPSALA UNIVERSITET

RBF–QR



# Brief survey of Mercer based methods contd

## De Marchi & Santin 2013

- Discrete numerical approximation of eigenfunctions.
- $W$ diagonal matrix with cubature weights. Perform SVD $\sqrt{W} \cdot A \cdot \sqrt{W} = Q \cdot \Sigma^2 \cdot Q^T$. The eigenbasis is given by $\sqrt{W^{-1}} \cdot Q \cdot \Sigma$.
- Rapid decay of singular values $\Rightarrow$ Basis can be truncated $\Rightarrow$ Low rank approximation of $A$.

## De Marchi & Santin 2014

- Faster: Lanczos algorithm on Krylov space $\mathcal{K}(A, f)$.
- Eigenfunctions through SVD of $H_m$ from Lanczos.
- Computationally efficient.
- Basis depends on $f$. Potential trouble for $f \notin \mathcal{N}_K(X)$

*For details it is a good idea to ask the authors :-)*

# Differentiation matrices and RBF-QR

*Larsson, Lehto, Heryudono, Fornberg 2013*

Let $\underline{u}_X$ be an RBF approximation evaluated at the nodes.

To compute $\underline{u}_Y$ evaluated at the set of points $Y$, we use

$$A\underline{\lambda} = \underline{u}_X \quad \Rightarrow \quad \underline{\lambda} = A^{-1}\underline{u}_X \text{ to get}$$

$$\underline{u}_Y = A_Y\underline{\lambda} = A_Y A^{-1}\underline{u}_X$$

where $A_Y(i,j) = \phi_j(y_i)$.

To instead evaluate a differential operator applied to $\underline{u}$,

$$\underline{u}_Y = A_Y^{\mathcal{L}} A^{-1}\underline{u}_X,$$

where $A_Y^{\mathcal{L}}(i,j) = \mathcal{L}\phi_j(y_i)$.

To do the same thing using RBF–QR, replace $\phi_j$ with $\psi_j$.

UPPSALA UNIVERSITET

RBF–QR



# Solving PDEs with RBFs/RBF-QR

Domain defined by: $r_b(\theta) = 1 + \frac{1}{10}(\sin(6\theta) + \sin(3\theta))$.

PDE: $\begin{cases} \Delta u = f(\underline{x}), & \underline{x} \in \Omega, \\ u = g(\underline{x}), & \underline{x} \text{ on } \partial\Omega, \end{cases}$

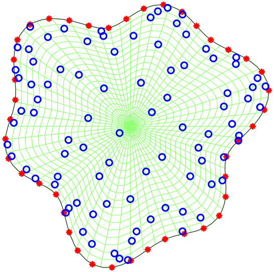
Solution: $u(\underline{x}) = \sin(x_1^2 + 2x_2^2) - \sin(2x_1^2 + (x_2 - 0.5)^2)$.

Collocation:

$$\begin{pmatrix} A^{\Delta}_{X^i} A_X^{-1} \\ I \end{pmatrix} \begin{pmatrix} \underline{u}^i_X \\ \underline{u}^b_X \end{pmatrix} = \begin{pmatrix} \underline{f}^i_X \\ \underline{g}^b_X \end{pmatrix}$$

Evaluation:

$\underline{u}_Y = A_Y A_X^{-1} \underline{u}_X$



Domain + nodes

# `demo5.m`

(Solving the Poisson problem in 2-D using RBFs)

# Cost of global method

Global RBF approximations of smooth functions are very efficent.

A small number of node points per dimension are needed.

However $N = 15$ in 1-D becomes $N = 50\,625$ in 4-D.

Up to three dimensions can be handled on a laptop, but not more.

Furthermore, for less smooth functions, the number of nodes per dimension grows quickly.

For a dense linear system: Direct solution $\mathcal{O}(N^3)$, storage $\mathcal{O}(N^2)$.

$\Rightarrow$ Move to localized methods.

UPPSALA UNIVERSITET

RBF–QR

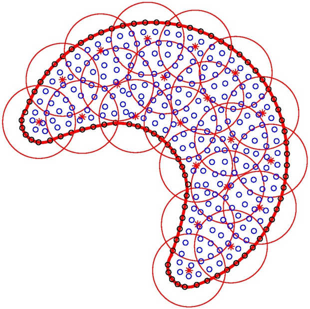


# RBF partition of unity methods for PDEs

## Global approximant

$$s(\underline{x}) = \sum_{i=1}^{M} w_i(\underline{x}) s_i(\underline{x}),$$

$w_i(\underline{x})$ are weight functions.



## Local RBF approximants

$s_i(\underline{x}) = \sum_{j=1}^{N_i} \lambda_j^{(i)} \phi_j(\underline{x}).$

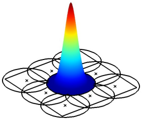
## Objectives for the RBF partition of unity approach

- Leverage spectral convergence properties.
- Retain geometric flexibility (also in high dimensions).
- Overcome conditioning and cost issues.
- Facilitate adaptive approximations.

*Interpolation: Wendland 2002, Fasshuer 2007, Cavoretto, De Rossi, Perracchione 2014. PDEs: Larsson, Heryudono 2012,. . .*

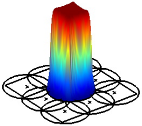
# Constructing weight functions and covering the domain

## Wendland functions + Shepard's method



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

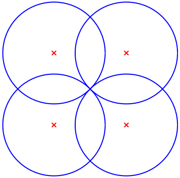
$$\psi(\rho) = (4\rho + 1)(1 - \rho)_+^4$$



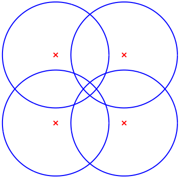
using Shepard's method $w_i(\underline{x}) = \dfrac{\psi_i(\underline{x})}{\sum_{j=1}^{M} \psi_j(\underline{x})}$.

## Disc coverings

Bad cover



Good cover

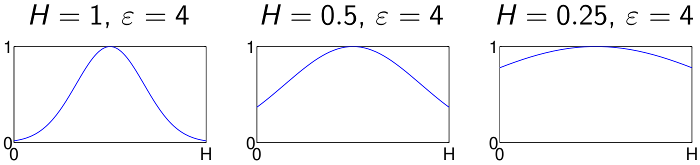


# Why do we need RBF–QR for RBF–PUM?

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, keep $\varepsilon$ fixed.

## The effect of patch refinement




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

Patch refinement is not a problem. $N$ cannot be increased to infinity, but to reasonable numbers. Clustering may or may not be needed at the exterior boundary.

UPPSALA UNIVERSITET

RBF–QR



# `demo6.m`

(Solving a Poisson problem in 2-D with RBF–PUM)

UPPSALA UNIVERSITET

RBF–QR



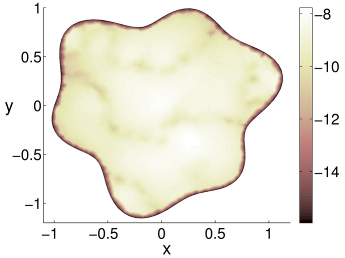
# Poisson test problem

*Larsson, Heryudono 2016*

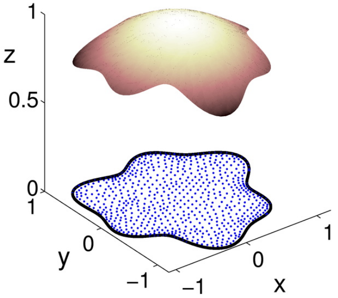
Domain defined by: $r_b(\theta) = 1 + \frac{1}{10}(\sin(6\theta) + \sin(3\theta))$.

PDE: $\begin{cases} \Delta u = f(\underline{x}), & \underline{x} \in \Omega, \\ u = g(\underline{x}), & \underline{x} \text{ on } \partial\Omega, \end{cases}$ with $u(r,\theta) = \frac{1}{0.25r^2+1}$.

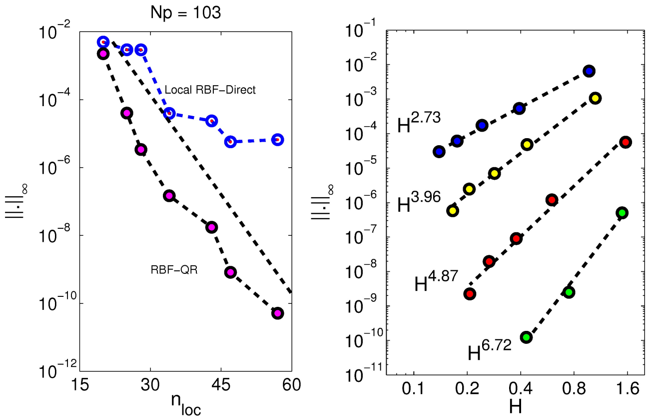
$\log_{10}(\text{error})$




RBF-PU solution



# RBF-PUM results for the elliptic PDE




Increasing the number of local points for fixed number of partitions ⇒ Spectral convergence.

Increasing the number of partitions for fixed $n_{\text{loc}}$ (21, 28, 45, 66) ⇒ Algebraic convergence (th. 3, 4, 6, 8).

UPPSALA UNIVERSITET

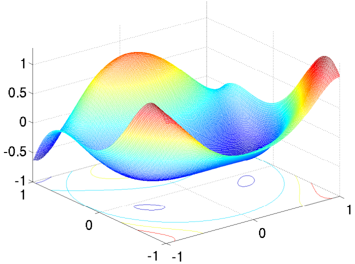
RBF–QR

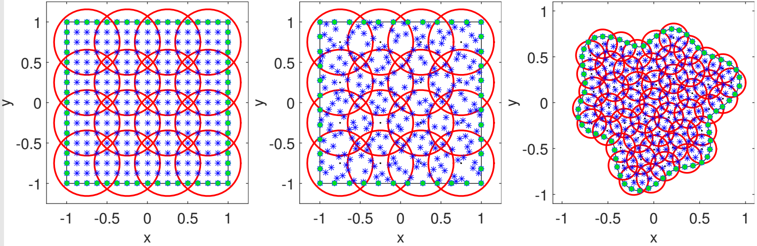


# Problems used for convergence and solver tests

Poisson problem with solution

$$u(\underline{x}) = \sin(x_1^2 + 2x_2^2) - \sin(2x_1^2 + (x_2 - 0.5)^2)$$





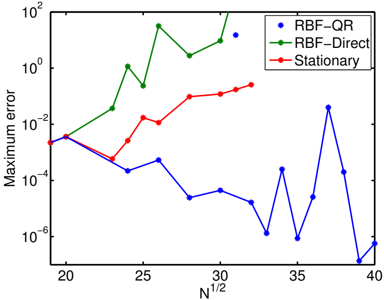
Fully structured Unstructured nodes Fully unstructured

# Poisson: Errors with and without RBF–QR

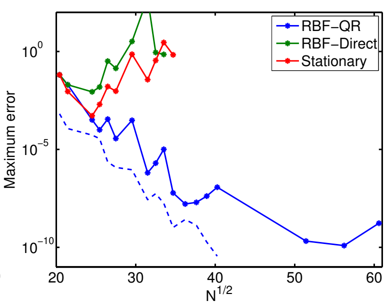
## Setting

$N$ nodes, $5 \times 5$ patches, (except dashed line with $4 \times 4$)
$\varepsilon = 1.2$ or scaled such that $\varepsilon h \approx \frac{\varepsilon}{\sqrt{N}} = \mathrm{const.}$

Square, Cartesian nodes




Square, Halton nodes




- RBF–QR is needed for convergence.
- Cartesian nodes are sub-optimal with RBF–QR

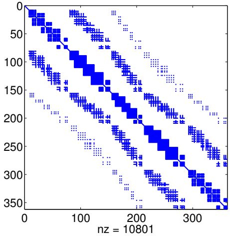
UPPSALA UNIVERSITET



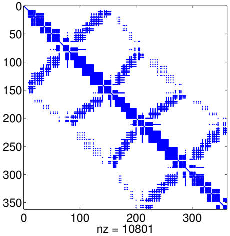
# RBF–PUM: Iterative solver

Question: Is there a structure in the unstructured case?

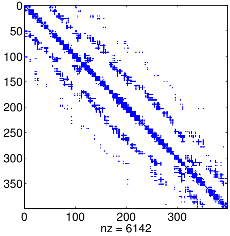
Cartesian, vertical



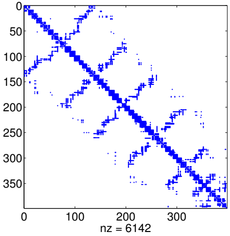

Cartesian, snake




Unstructured, vertical




Unstructured, snake

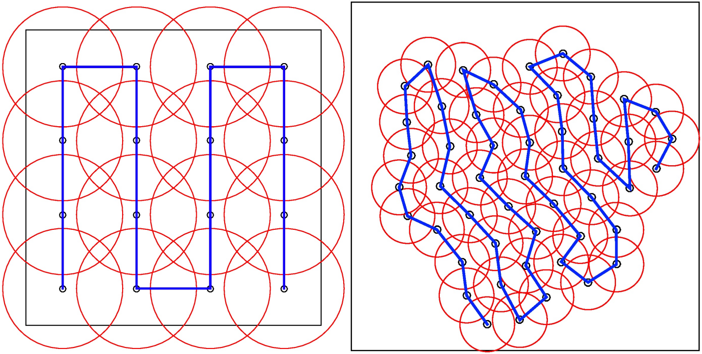


# What is snake ordering?

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 secondary patch.



*Heryudono, Larsson, Ramage, and von Sydow, 2015*

# Preconditioned iterative solution

- Right preconditioned GMRES, $LM^{-1}y = f$, $Mu = y$.
- Preconditioner ILU(0) of central band.
- Stopping criterion, residual reduction of $10^{-8}$.

## Results for the square with Cartesian nodes

| $N$ | # it no prec | # it ILU(0) | Time gain |
|---|---|---|---|
| 400 | 32 | 21 | 2.0 |
| 576 | 127 | 38 | 4.5 |
| 676 | 165 | 43 | 5.8 |
| 900 | 170 | 49 | 2.8 |
| 1089 | 180 | 53 | 4.3 |

UPPSALA UNIVERSITET

RBF–QR



# Results for the iterative method contd.

## Results for the square with Halton nodes

| $N$ | # it no prec | # it ILU(0) | Time gain |
|---|---|---|---|
| 436 | 189 | 72 | 3.1 |
| 583 | 209 | 91 | 2.4 |
| 681 | 231 | 112 | 2.7 |
| 884 | 262 | 125 | 2.3 |
| 1090 | 295 | 135 | 3.0 |

## Results for the unstructured case

| $N$ | # it no prec | # it ILU(0) | Time gain |
|---|---|---|---|
| 398 | 207 | 68 | 3.6 |
| 695 | 235 | 78 | 5.6 |
| 994 | 279 | 119 | 3.6 |
| 1094 | 304 | 120 | 4.3 |
| 1292 | 322 | 149 | 3.3 |

# Solving time-dependent PDEs

## Before: Time-independent PDE

Continuous

$$\begin{cases} \mathcal{L}u = f(\underline{x}), & \underline{x} \in \Omega, \\ u = g(\underline{x}), & \underline{x} \text{ on } \partial\Omega, \end{cases}$$

RBF collocated

$$\begin{cases} A^{\mathcal{L}}_{X^i} A_X^{-1} \underline{u}_X = \underline{f}^i_X \\ \underline{u}^b_X = \underline{g}^b_X, \end{cases}$$

## Time-dependent PDE

Continuous

$$\begin{cases} \frac{\partial u}{\partial t} = \mathcal{L}u - f(\underline{x}, t), \\ u = g(\underline{x}, t), \end{cases}$$

RBF collocated

$$\begin{cases} \frac{\partial}{\partial t} \underline{u}^i_X = A^{\mathcal{L}}_{X^i} A_X^{-1} \underline{u}_X - \underline{f}^i_X(t) \\ \underline{u}^b_X = \underline{g}^b_X(t), \end{cases}$$

## Time evolution

We have mostly used a version of BDF-2 (second order, implicit) for parabolic PDEs. Also built in solvers from MATLAB.

UPPSALA UNIVERSITET

RBF–QR



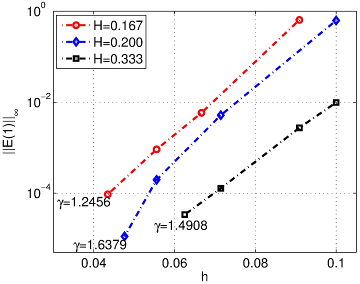
# `demo7.m`

(Solving the heat equation in 2-D)

# Convergence results for convection-diffusion

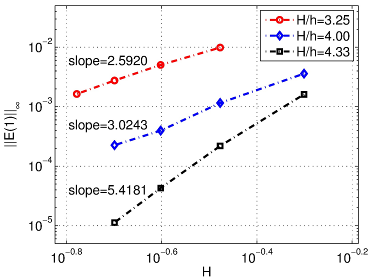
*Safdari-Vaighani, Heryudono, Larsson, 2104*

## Spectral case, $H$ fixed




$\kappa = 1, \ v = (1, 1)$

## Algebraic, $H/h$ fixed

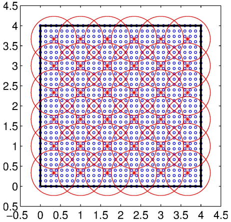



Expected rates $p = 2, 3, 4$

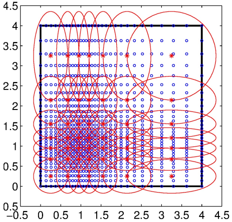
- Convergence as expected also in practice.
- Range could be extended with RBF-QR.

# Comparisons for American option problem

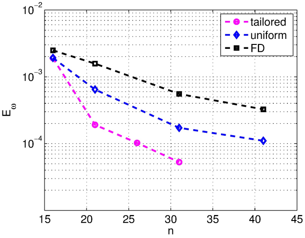
## Uniform nodes



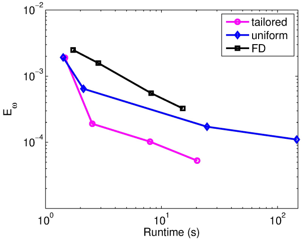
## Non-uniform nodes



## Accuracy comparison




## Run-time comparison




Reference: Uniform FD-operator splitting method.

# Some results for option pricing

BENCHOP—The BENCHmarking project in Option Pricing
http://www.it.uu.se/research/project/compfin/benchop

Radial basis function partition of unity methods for pricing vanilla basket options
Shcherbakov, Larsson 2015(?)

RBF–PUM operator splitting method for pricing multi-asset American options
Shcherbakov, submitted

# Stabilization for hyperbolic PDEs

*Fornberg, Lehto 2011*
For hyperbolic (purely convective) PDEs, local scattered node RBF discretizations typically lead to unstable eigenvalues.

For global RBFs, add term $-\gamma A^{-1}\underline{u}$ to ODE-system.

For RBF–FD add $-\gamma\Delta^k\underline{u}$ to ODE-system.

Fast computation with RBF–QR: Larsson, Lehto, Heryudono, Fornberg 2013

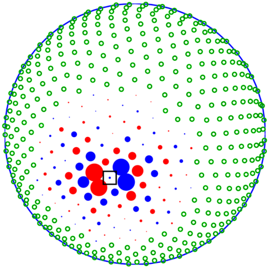
# RBF-generated finite differences RBF–FD

*Larsson, Lehto, Heryudono, Fornberg 2013*

- Approximate $\mathcal{L}u(\underline{x}_c)$ using the $n$ nearest nodes by

$$\mathcal{L}u(\underline{x}_c) \approx \sum_{k=1}^{n} w_k u(\underline{x}_k)$$

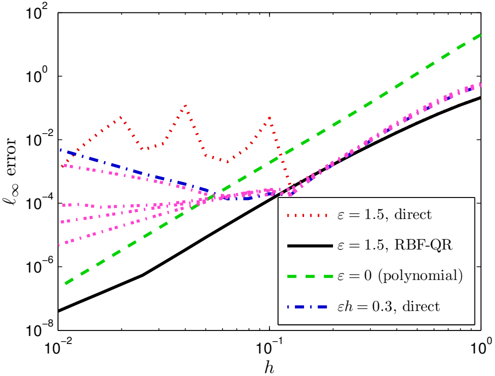
- Find weights $w_k$ by asking exactness for RBF-interpolants.



$$\begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \cdots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n(\mathbf{x}_1) & \phi_n(\mathbf{x}_2) & \cdots & \phi_n(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \mathcal{L}\phi_1(\mathbf{x}_c) \\ \mathcal{L}\phi_2(\mathbf{x}_c) \\ \vdots \\ \mathcal{L}\phi_n(\mathbf{x}_c) \end{bmatrix}.$$

# Is RBF-QR needed with RBF–FD?

Approximation of $\Delta u$ with $n = 56$. Magenta lines are with added polynomial terms $p = 0, \ldots, 3$.




- Scaled $\varepsilon$: No ill-conditioning, but saturation/stagnation. *(See Kindelan et al.)*
- Fixed $\varepsilon$: RBF-QR is needed.
- Added terms: Compromise with partial recovery.

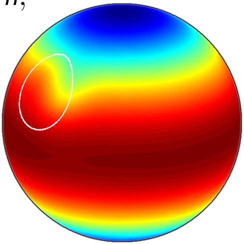
# Shallow water simulation

*Tillenius, Larsson, Lehto, Flyer 2015*

## The shallow water equations

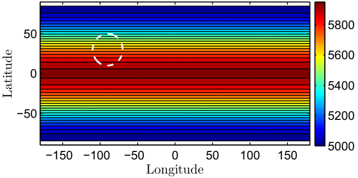
$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - f(\mathbf{x}\times\mathbf{u}) - g\nabla h,$$

$$\frac{\partial h}{\partial t} = -\nabla\cdot(h\mathbf{u})$$

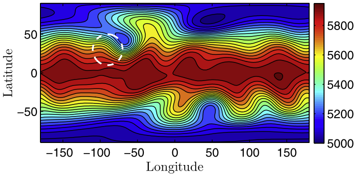


## Test cases

- Flow over an isolated mountain
- Highly non-linear wave



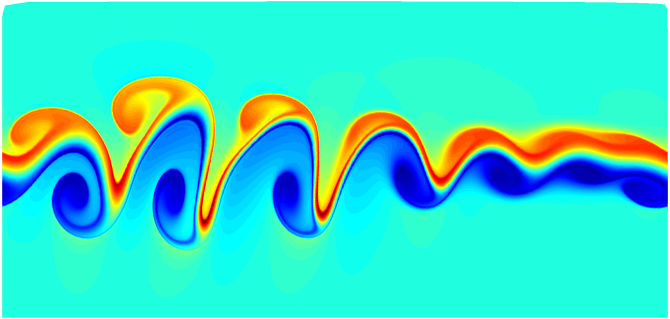

Day 0




Day 15

# Results shallow water

The highly non-linear wave with 612 346 nodes on the sphere.



Some problems with stability. Did not use RBF–QR. Would need adaptivity.