Fast direct solvers for RBF interpolation problems

Sabine Le Borne

INSTITUT FÜR MATHEMATIK

TUHH
Technische Universität Hamburg

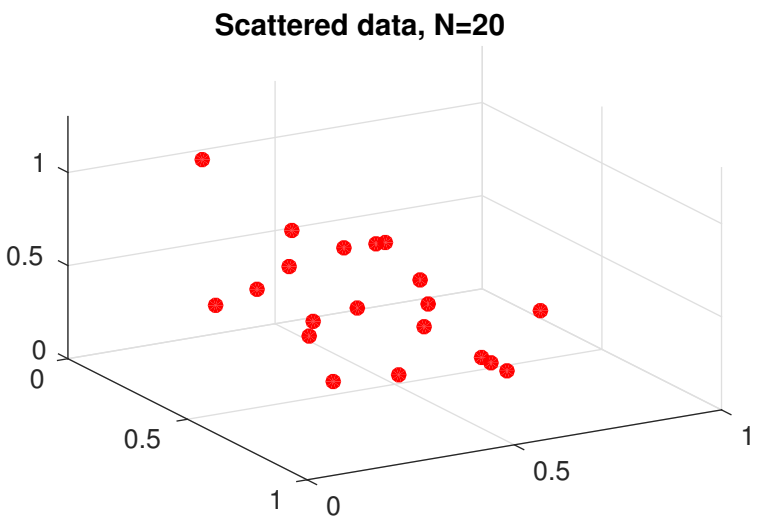
Conference on Fast Direct Solvers

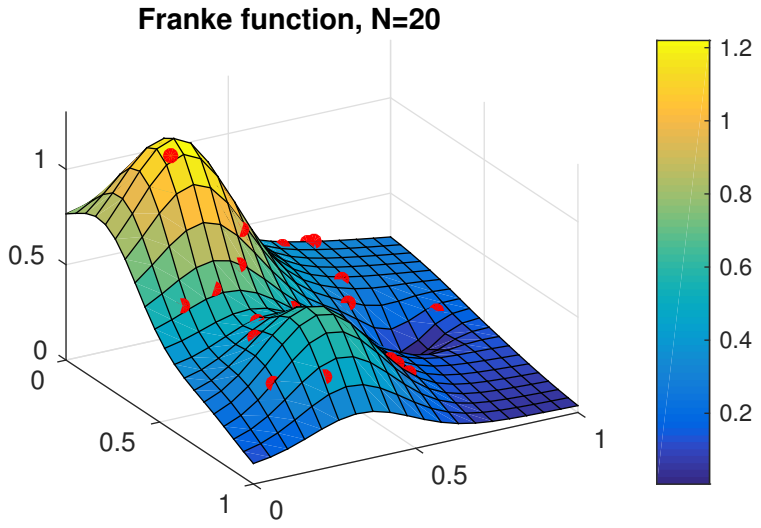
Purdue University, Nov 9-11, 2018

Scattered data interpolation problem

Given: (scattered) data $\{(\mathbf{x}_i, y_i) \mid 1 \leq i \leq N\}$ with

- data sites $\mathbf{x}_i \in \mathbb{R}^d$ and
- measurements $y_i \in \mathbb{R}$ and
- spatial dimension $d \in \mathbb{N}$ (often $d \in \{2, 3\}$),
- number of data $N \in \mathbb{N}$.





Some applications

Some applications

- surface reconstruction
- image compression/reconstruction (incl. movies)

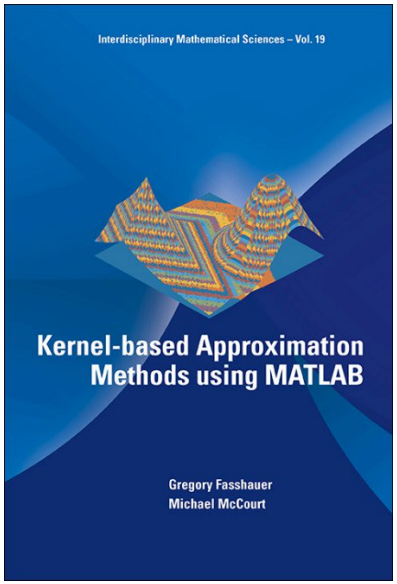
(a) Original: $256 \cdot 384 = 98304$ pixels.

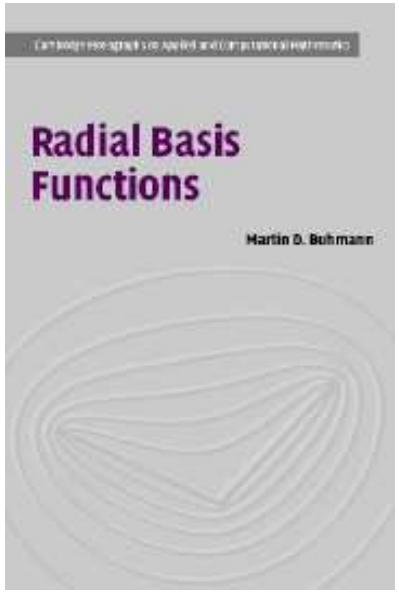
(b) Corrupted: half of all pixels removed.

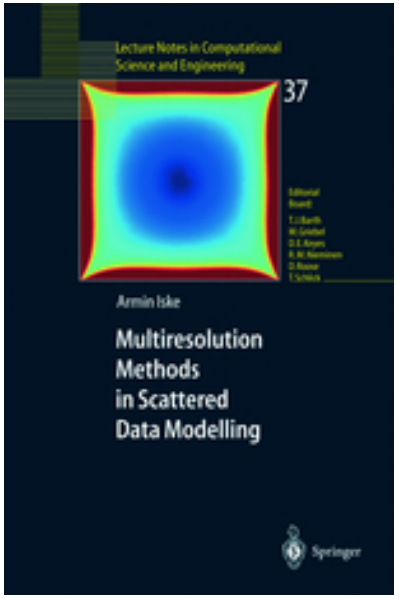
(c) Reconstructed: IMQ interpolation

- machine learning
- neural networks
- (PDEs in) fluid mechanics, wave motions, astro- and geosciences, biology, etc.

Some textbooks







Scattered data interpolation problem

Needed for interpolation: an N-dimensional, data-dependent, linear space

$$S_X = \text{span}\{b_1(\cdot), \ldots, b_N(\cdot)\}$$

of functions that depends on the data sites $X = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$.

Radial basis functions (RBFs):

$$b_i(\mathbf{x}) = \phi_\varepsilon(\|\mathbf{x} - \mathbf{x_i}\|_2)$$

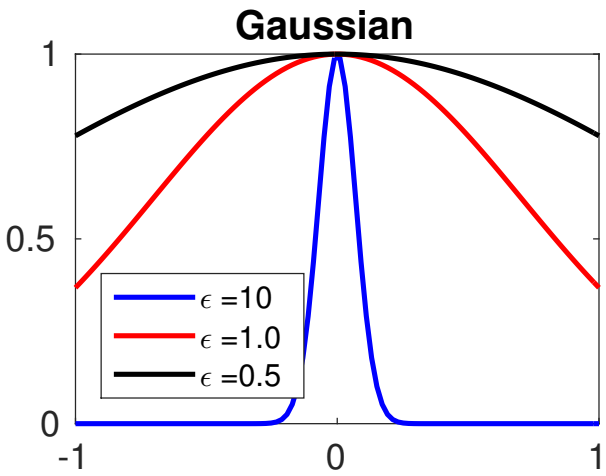
for a given (continuous) function $\phi_\varepsilon : \mathbb{R} \to \mathbb{R}$ (*kernel*), i.e.,

- basis functions depend on (distance to) data sites $\mathbf{x_i}$,
- basis functions depend on a shape factor ε,
- basis functions often have global support.

Examples for radial basis functions

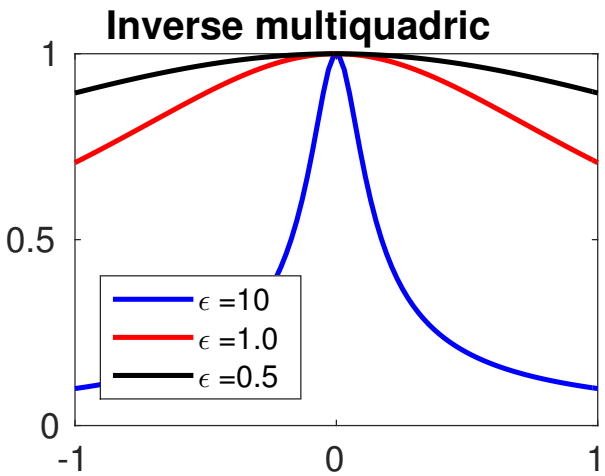
Gaussian kernel

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

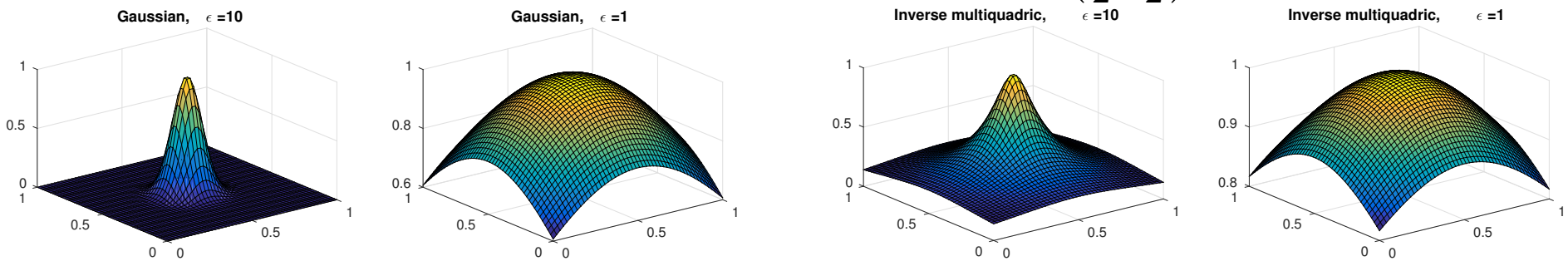


Inverse multiquadric kernel

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



$$\mathbf{b_i}(\mathbf{x}) = \phi_\varepsilon(\|\mathbf{x} - \mathbf{x_i}\|_2) \text{ for } \mathbf{x_i} = \left(\tfrac{1}{2}, \tfrac{1}{2}\right)$$



Remarks: no grid required, smooth, global support

Scattered data interpolation problem

Find a function $s \in S_X$ that satisfies the interpolation conditions

$$s(\mathbf{x_i}) = y_i \quad \forall i = 1, \ldots, N.$$

Find a coefficient vector $c = (c_1 \ldots c_N)^T$ such that

$$s(\mathbf{x}) = \sum_{j=1}^{N} c_j b_j(\mathbf{x})$$

satisfies $s(\mathbf{x_i}) = y_i \quad \forall i = 1, \ldots, N.$

Find the solution $\mathbf{c} \in \mathbb{R}^N$ of the linear system of equations

$$Ac=y$$

with $A = (b_{ij}) = (b_j(\mathbf{x_i}))$ and $y = (y_1 \ldots y_N)^T$.

RBF interpolation problem

Solve $Ac = y$ with $A = (b_{ij}) = (b_j(\mathbf{x_i}))$ and $y = (y_1 \dots y_N)^T$.

Properties of the matrix A:

- A is symmetric ($b_j(\mathbf{x_i}) = \phi_\varepsilon(\|\mathbf{x_i} - \mathbf{x_j}\|_2) = b_i(\mathbf{x_j})$.)
- A is positive definite (Gaussian and IMQ RBFs are positive definite on $\mathbb{R}^d$).
- A is dense (Gaussian, IMQ RBFs have global support).

- Condition number of A wrt. N (number of data) and ε (shape factor):

  cond(A) grows as N grows or as b_i gets "flat".

- Approximation error wrt. N (number of data) and ε (shape factor):

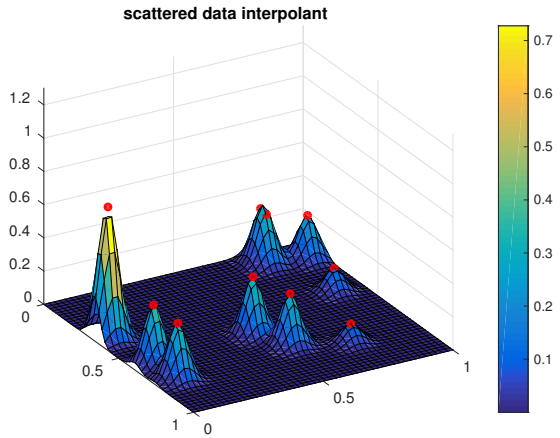
  $\|f - s\|_\infty$ decreases as N grows or as b_i gets "flat".

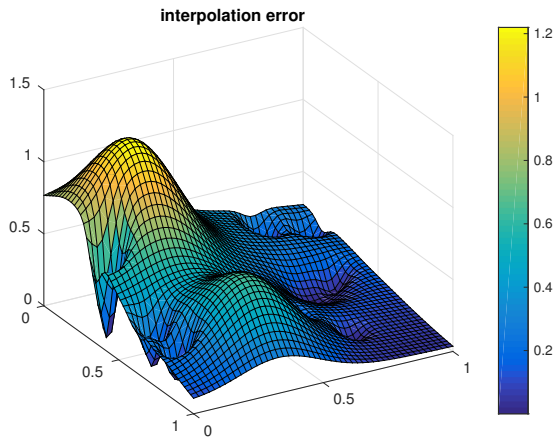
  $\Longrightarrow$ uncertainty/trade-off principle

- A is ill-conditioned.

Illustration of the scaling factor ε $(N = 10)$

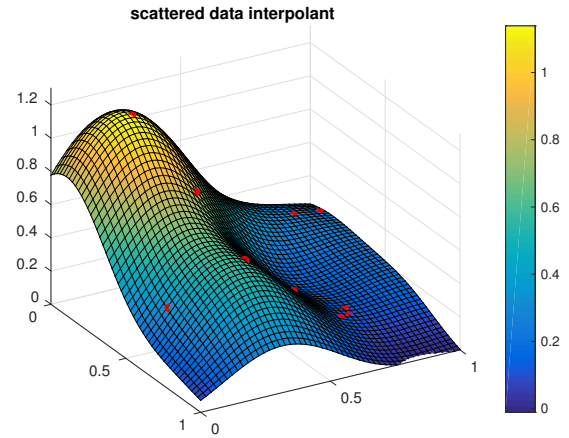
$\varepsilon = 10$

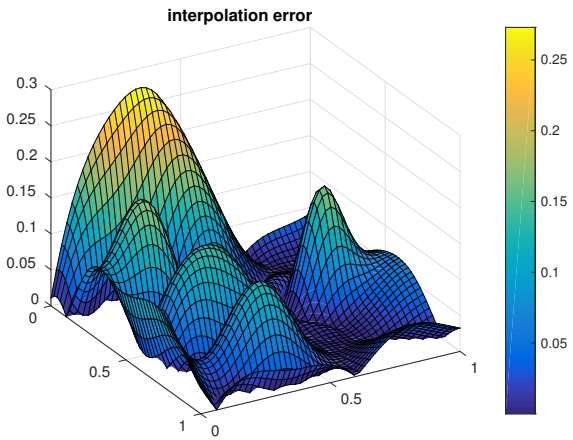




$\kappa(A) \approx 2.8$

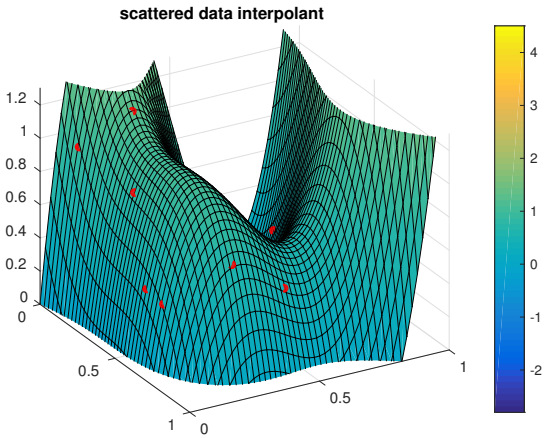
$\varepsilon = 2.5$

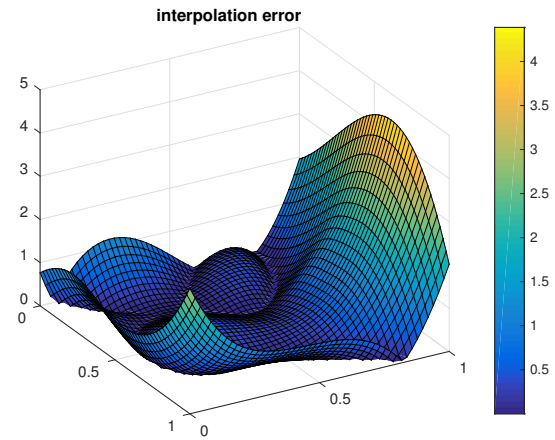




$\kappa(A) \approx 7 \cdot 10^3$

$\varepsilon = 0.625$

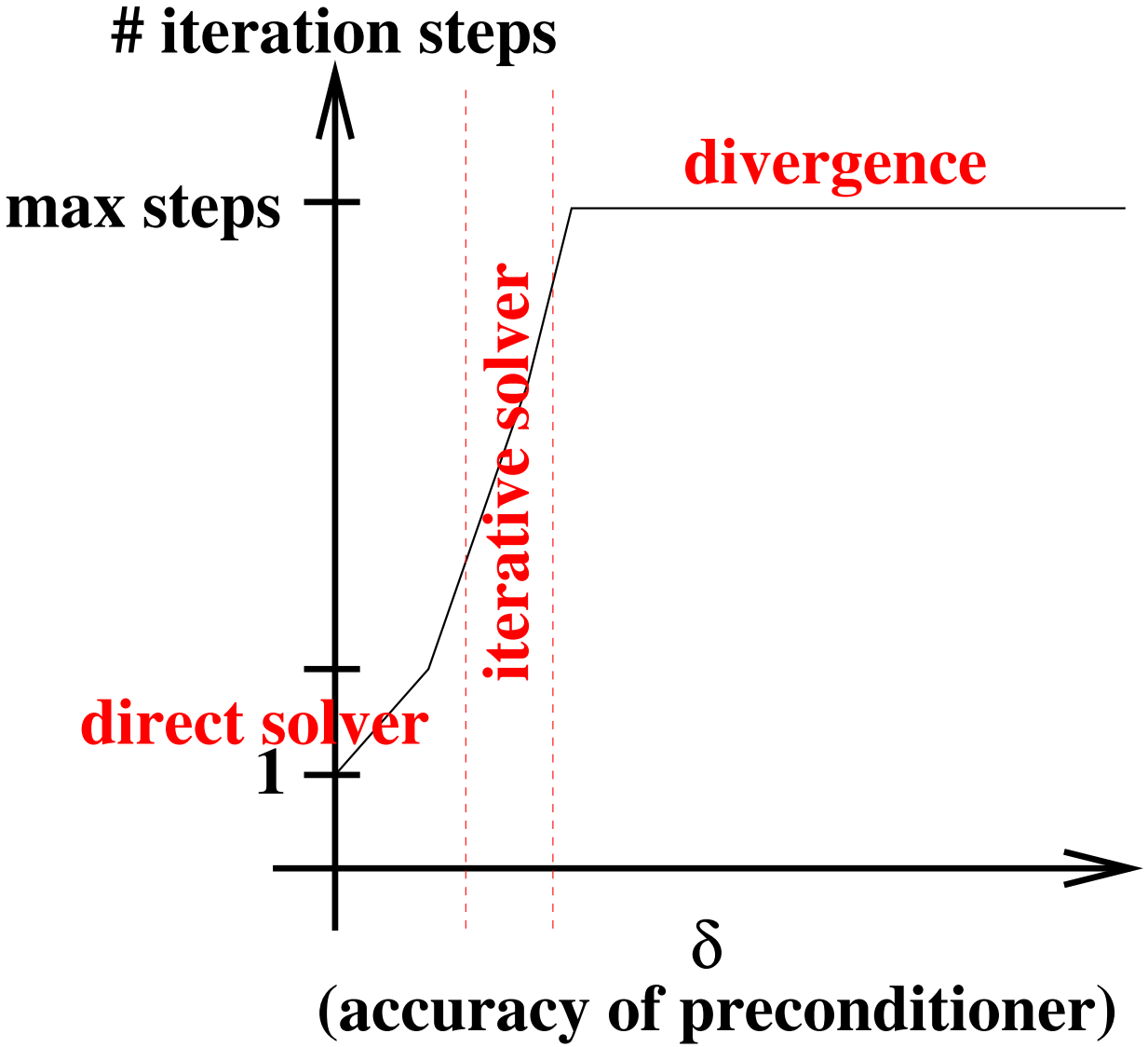




$\kappa(A) \approx 5 \cdot 10^7$

Precond. iterative solvers for highly ill-cond. A

δ measures the quality of the preconditioner P_δ (with $P_0 = A^{-1}$).



Recap

This talk so far:

Problem setting: RBF interpolation $\Longrightarrow$ $Ac = y$ with a dense, SPD, highly ill-conditioned matrix A

Fast Direct Solver: required since iterative solvers (often) fail to converge,

needs data-sparse approximation $\widetilde{A}$ to A and highly accurate preconditioner $P \approx \widetilde{A}^{-1}$

Recap

Remainder of this talk: **Fast Direct Solvers** using

1. **Structured/$\mathcal{H}$-matrices:**
   - Approximation $\widetilde{A}$ of A
   - Preconditioner P through $\mathcal{H}$-Cholesky/$\mathcal{H}$-LU factors
2. **Domain decomposition method**
3. **Change of basis:** Analytic right preconditioning
4. **Change of function space**
   - multilevel RBFs
   - variable shape parameter
   - polynomial augmentation
5. **Nullspace method/Lagrangian augmentation** for saddle point problems resulting from conditionally positive RBFs

Recap

Remainder of this talk: **Fast Direct Solvers** using

1. **Structured/$\mathcal{H}$-matrices:**
   - Approximation $\widetilde{A}$ of A
   - Preconditioner P through $\mathcal{H}$-Cholesky/$\mathcal{H}$-LU factors
2. **Domain decomposition method**
3. **Change of basis:** Analytic right preconditioning
4. **Change of function space**
   - multilevel RBFs
   - variable shape parameter
   - polynomial augmentation
5. **Nullspace method/Lagrangian augmentation** for saddle point problems resulting from conditionally positive RBFs

Structured/$\mathcal{H}$-matrices: Approximation of A

Steps for an $\mathcal{H}$-matrix construction

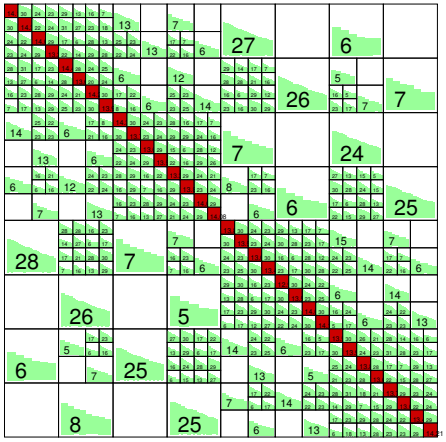
1. Construction of a nested block structure of the index set via geometric bisection of the interpolation centers
2. Standard geometric admissibility condition based on diameter and distance of clusters
3. Adaptive cross approximation (ACA) to fill the matrix with data

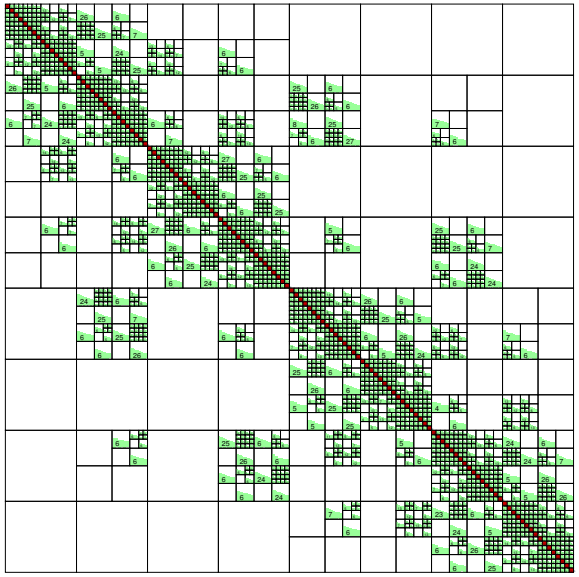
Monographs on $\mathcal{H}$-matrices: [Hackbusch (2015)], [Bebendorf (2008)]
$\mathcal{H}$-matrices for RBFs: [Iske, L., Wende (2017)]

Structured/$\mathcal{H}$-matrices: Approximation of A, $d = 2$

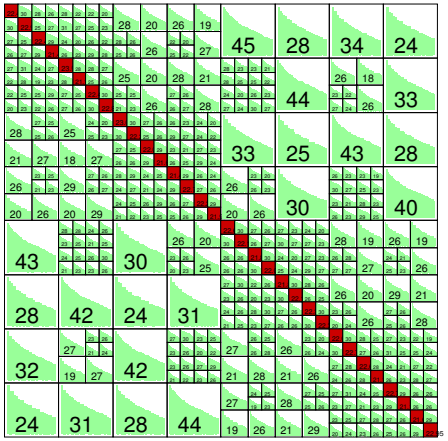
$N = 31^2 = 961$ $\qquad$ $N = 63^2 = 3969$

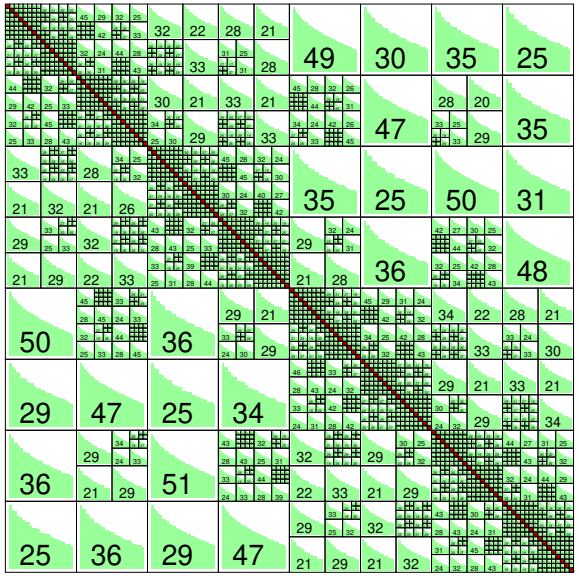
Gaussian





IMQ





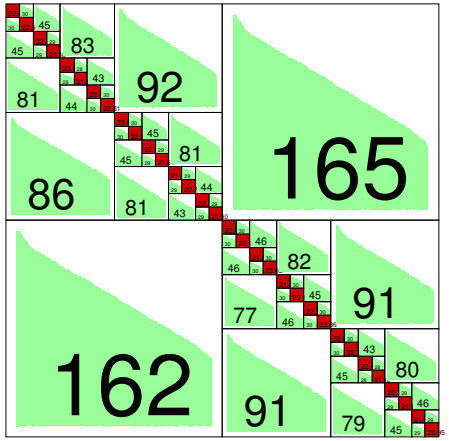
Structured/$\mathcal{H}$-matrices: Approximation of A, $d = 2$

Observations for shape parameter $\varepsilon = c\sqrt{N}$, $\mathcal{H}$-acc. $\delta = 10^{-10}$

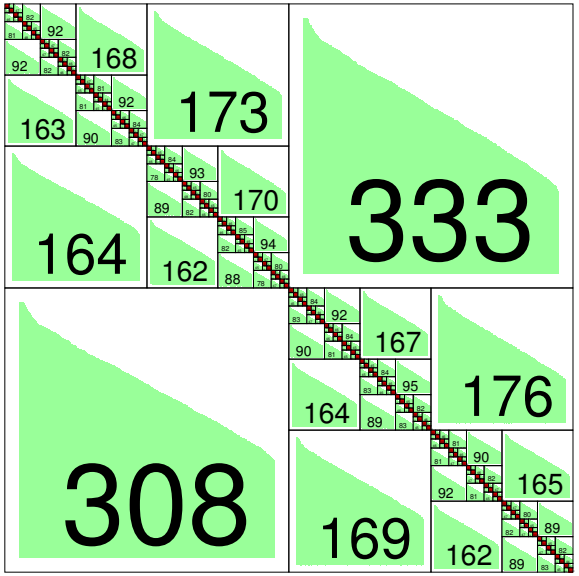
- "Standard" $\mathcal{H}$-matrix approximation applicable
- Gaussian RBF leads to numerically sparse matrix.
- IMQ RBF leads to bounded ranks within off-diagonal blocks.

Structured/$\mathcal{H}$-matrices, IMQ, weak admissbility

$N = 31^2$



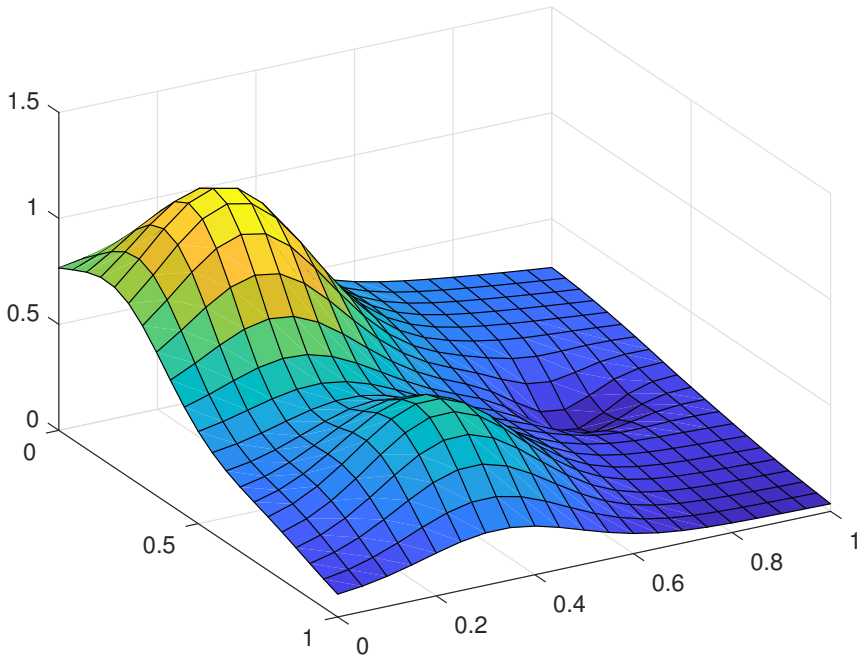
$N = 63^2$



$\mathcal{H}$-matrices: Solving the 2d interpolation problem

Franke function:

$$
\begin{aligned}
f_{2d}(x,y) \quad = \quad & \frac{3}{4}\exp\left(-\left((9x-2)^2+(9y-2)^2\right)\right) + \frac{3}{4}\exp\left(-\frac{1}{49}(9x+1)^2 - \frac{1}{10}(9y+1)\right) \\
& + \frac{1}{2}\exp\left(-\left((9x-7)^2+(9y-3)^2\right)\right) - \frac{1}{5}\exp\left(-(9x-4)^2-(9y-7)^2\right)
\end{aligned}
$$

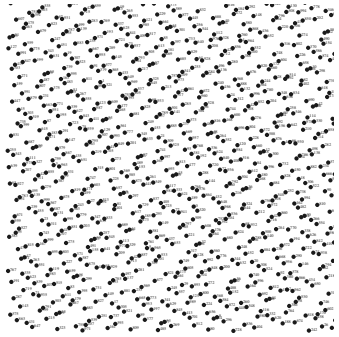


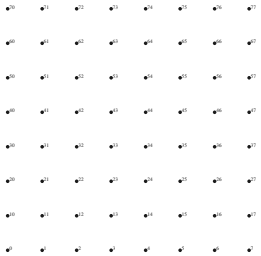
Centers:

Halton points in $[0,\ 1]^2$
(for interpolation)

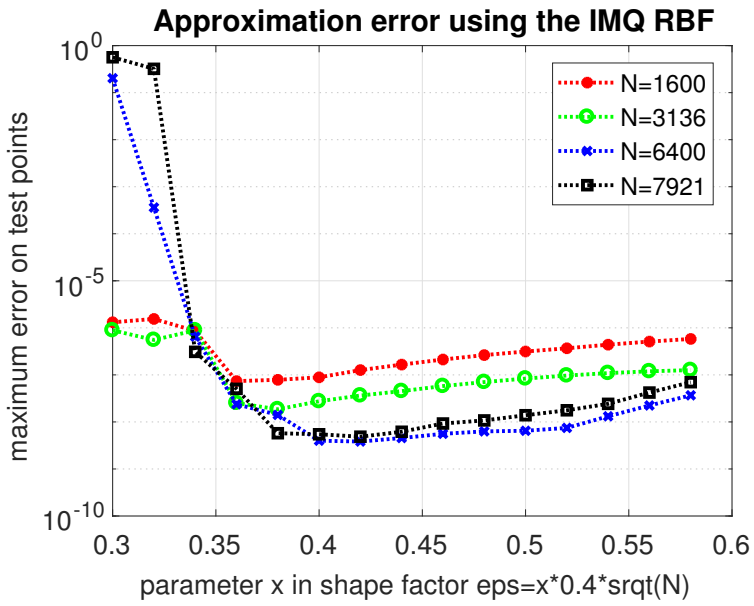
Test points:

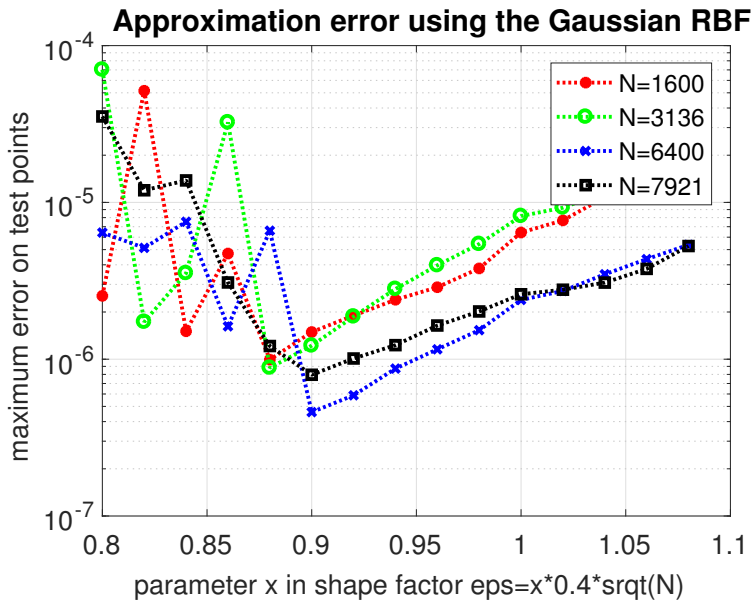
10x10 tensor grid in $[0.2;\ 0.8]^2$
(to evaluate interpolant)





$\mathcal{H}$-matrices: Solving the 2d interpolation problem





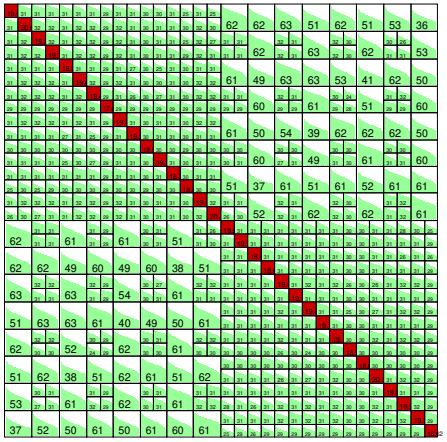
Conclusions

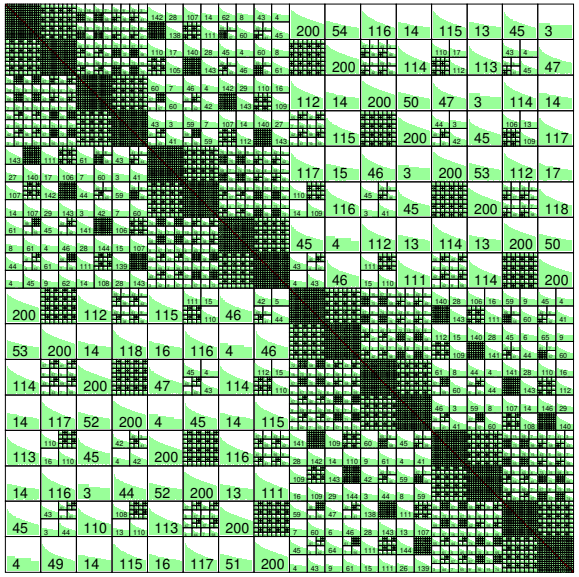
- IMQ reaches accuracy $\mathcal{O}(10^{-8})$, Gaussian reaches $\mathcal{O}(10^{-6})$
- increasing N beyond 8000 is not helpful to increase accuracy
- shape parameter ε is very important:
  too small $\Longrightarrow$ solution suffers from ill-conditioning
  (& Runge phenomenon for large N?)
  too large $\Longrightarrow$ less favorable function space

Structured/$\mathcal{H}$-matrices: Approximation of A, $d = 3$

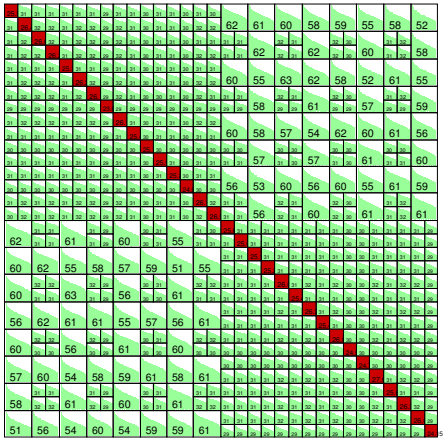
$N = 10^3 = 1000$ $\qquad$ $N = 20^3 = 8000$

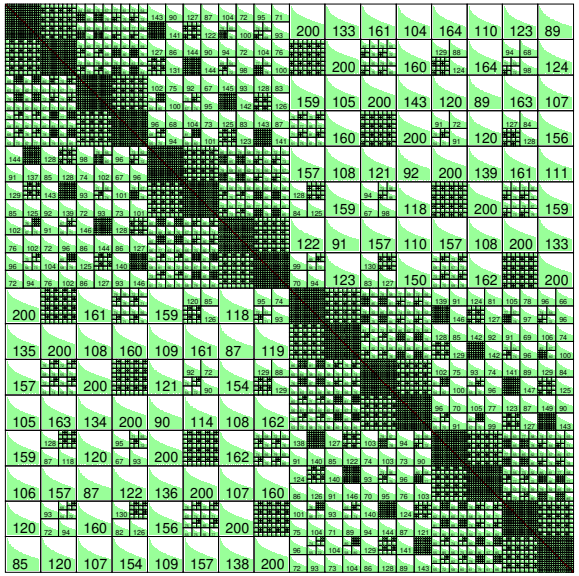
Gaussian





IMQ





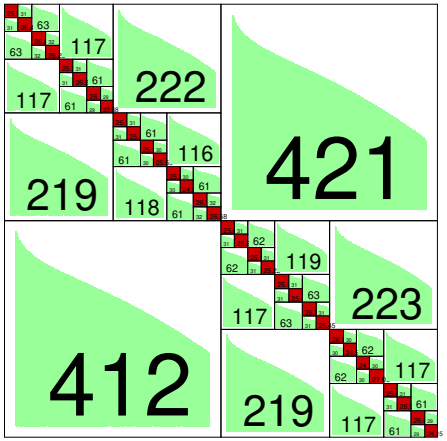
Structured/$\mathcal{H}$-matrices: Approximation of A, $d = 3$

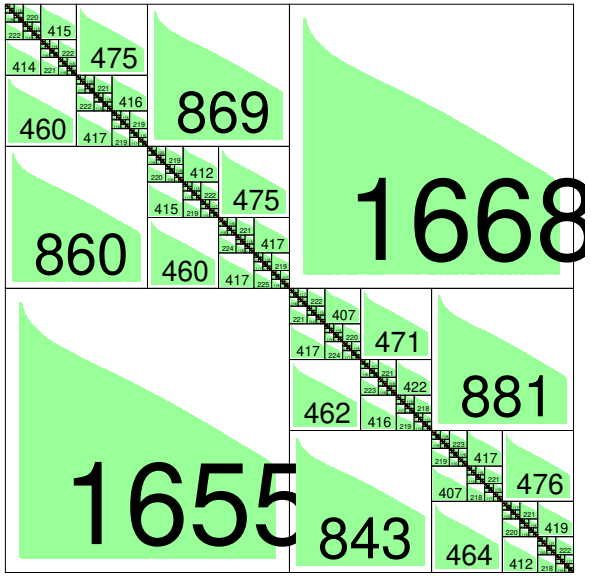
Observations for shape parameter $\varepsilon = c \cdot N^{1/3}$

- "Standard" $\mathcal{H}$-matrix approximation might pay off, but only for much larger problem sizes
- Gaussian RBF eventually leads to numerically sparse matrix.
- IMQ RBF initially leads to increasing ranks within off-diagonal blocks, suggesting the need for an even stronger admissibility condition.

Structured/$\mathcal{H}$-matrices, IMQ, weak admissbility

3d
$N = 10^3$
$N = 20^3$



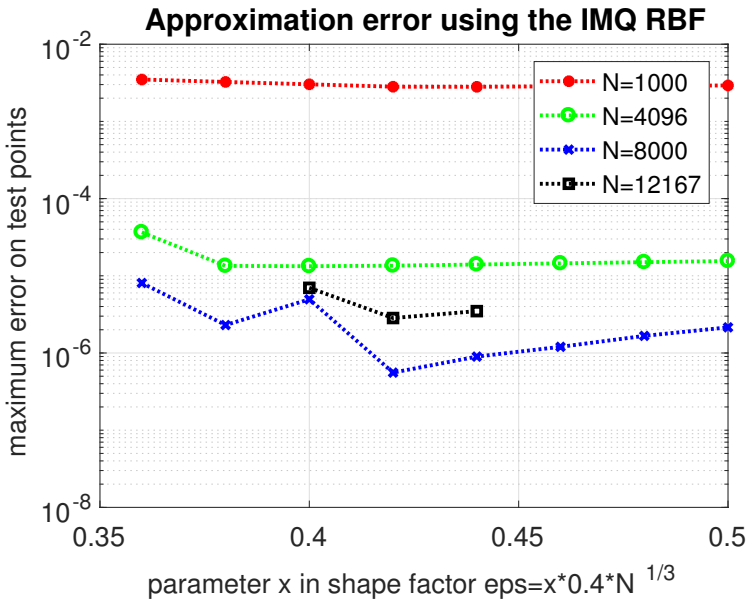


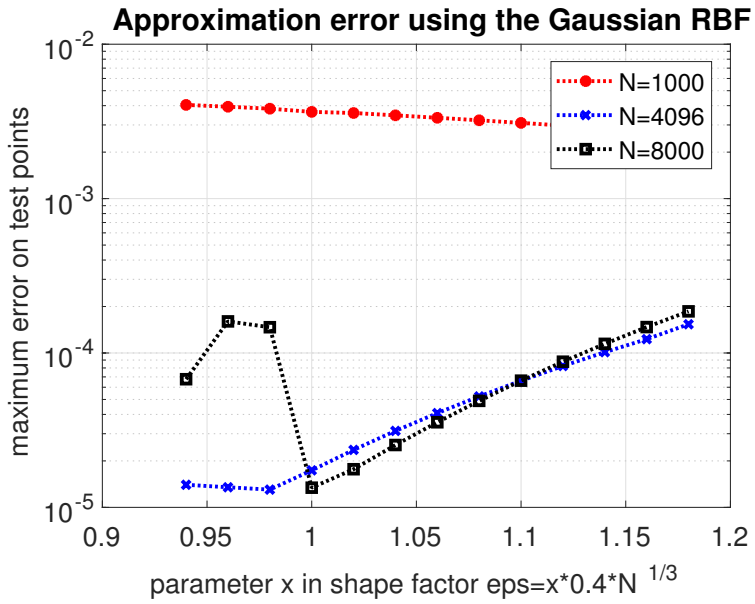
$\mathcal{H}$-matrices: Solving the 3d interpolation problem

Franke function: $f_{3d}(x, y, z) = \ldots$ (sum of four exponentials)

Centers: Halton points in $[0,\ 1]^3$ (for interpolation)

Test points: 10x10x10 tensor grid in $[0.2;\ 0.8]^3$ (to eval. interp.)





Conclusions

- IMQ reaches accuracy $\mathcal{O}(10^{-6})$, Gaussian reaches $\mathcal{O}(10^{-5})$
- increasing N beyond 8000 is not helpful to increase accuracy
- as in 2d, the shape parameter ε is very important

Recap

This talk so far:

Problem setting: RBF interpolation $\Longrightarrow$ $Ac = y$ with a dense, SPD, highly ill-conditioned matrix A

Fast Direct Solver: required since iterative solvers (often) fail to converge,
needs data-sparse approximation to A and highly accurate preconditioner $P \approx A^{-1}$

Recap

This talk so far:

Problem setting: RBF interpolation $\Longrightarrow$ $Ac = y$ with a dense, SPD, highly ill-conditioned matrix A

Fast Direct Solver: required since iterative solvers (often) fail to converge,
needs data-sparse approximation to A and highly accurate preconditioner $P \approx A^{-1}$

BUT: So far, no "fast direct solvers" were shown, (only approximation accuracies of interpolants).

Approximation errors stagnate for $N > 8,000$ centers,

$\mathcal{H}$-matrices outperform standard dense matrices only for larger N, especially in higher spatial dimensions.

Recap

This talk so far:

Problem setting: RBF interpolation $\Longrightarrow$ $Ac = y$ with a dense, SPD, highly ill-conditioned matrix A

Fast Direct Solver: required since iterative solvers (often) fail to converge,
needs data-sparse approximation to A and highly accurate preconditioner $P \approx A^{-1}$

BUT: So far, no "fast direct solvers" were shown, (only approximation accuracies of interpolants).

Approximation errors stagnate for $N > 8,000$ centers,

$\mathcal{H}$-matrices outperform standard dense matrices only for larger N, especially in higher spatial dimensions.

Is there a need for fast direct solvers for $N \gg 8,000$ centers?

(Or is this the end of the talk?)

Why this talk continues...

1. slide 14 on remainder of this talk ... **more bullets** were listed;
2. because it is an **interesting mathematical problem** to solve RBF-systems for $N \gg 8,000$, irrespectively of practical applications;
3. because **others do it**, too (looking at the literature);
4. because there are indeed **actual applications** out there that need it (beyond the interpolation of Franke's function);
5. because it has been revealed that the so-called **uncertainty/trade-off principle** is a **misconception:**

   The interpolation problem itself is well-posed, it is the choice of basis that splits it into two highly ill-conditioned subproblems (i. Computing coefficients of the interpolant;
   ii. Evaluating the interpolant).

Preface of "Fast RBFs for Engrg. Appl." (2017)

M. E. Biancolini:
This book is the result of the industrial and academic research path that I started to follow since 2007.
At that time, I had a cooperation in place with a Formula 1 top team. [...]

We found out that a good quality and long-distance interactions in 3D cases were possible employing the bi-harmonic RBF which gives a **dense matrix** to be solved.

The maximum size of the RBF was **about 10,000 points** that took approximately **a couple of hours** to fit the RBF-sought coefficients.

The need was to face RBF problems that comprised **hundreds of thousands of source points** at least, but with the target to reasonably manage **even millions of source points**.
Two orders of magnitude more!

Next:

Remainder of this talk: **Fast Direct Solvers** using

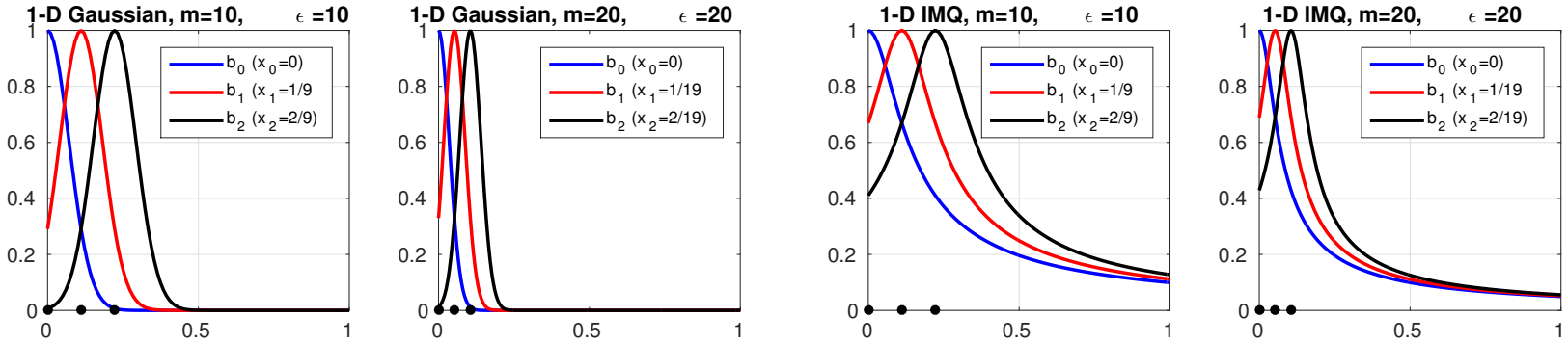
1. **Structured/$\mathcal{H}$-matrices:**
   - Approximation $\widetilde{A}$ of A
   - Preconditioner P through $\mathcal{H}$-Cholesky/$\mathcal{H}$-LU factors
2. **Domain decomposition method**
3. **Change of basis:** Analytic right preconditioning
4. **Change of function space**
   - multilevel RBFs (application where fast direct solvers are needed for $N \gg 8,000$)
   - variable shape parameter
   - polynomial augmentation
5. **Nullspace method/Lagrangian augmentation** for saddle point problems resulting from conditionally positive RBFs

Multilevel RBFs

Motivation: Shape parameter ε influences basis functions to be

- global, smooth, of low variation if ε is small;
- local, spiky, of large variation if ε is large.

1d basis functions using a fixed scaling parameter $\varepsilon = N^{1/d} = m$



"Dilemma" for the interpolation of Franke's function:

A smaller number of smooth basis functions is just as good as a larger number of spiky basis functions.

Multilevel RBFs

Motivation: Shape parameter ε influences basis functions to be

- global, smooth, of low variation if ε is small;
- local, spiky, of large variation if ε is large.

Two-level idea:

1. Use a small number of smooth basis functions to capture the smooth part within Franke's function/the given data;

   (interpolation matrix will be ill-conditioned, but small)

2. then correct the (spiky/oscillatory) residual using a larger number of spiky basis functions.

   (interpolation matrix will be large, but not too ill-conditioned)

Multilevel extension: Straight-forward. Apply two-level idea to coarse problem.

Multilevel RBF interpolation of $f : \mathbb{R}^d \to \mathbb{R}$

1. For a set X of $N = |X|$ centers in $\mathbb{R}^d$, construct a nested sequence of L subsets X_ℓ of $N_\ell = |X_\ell|$ centers,

$$\underbrace{X = X_1}_{\text{finest}} \supset X_2 \supset \ldots \supset X_{L-1} \supset \underbrace{X_L}_{\text{coarsest}} .$$

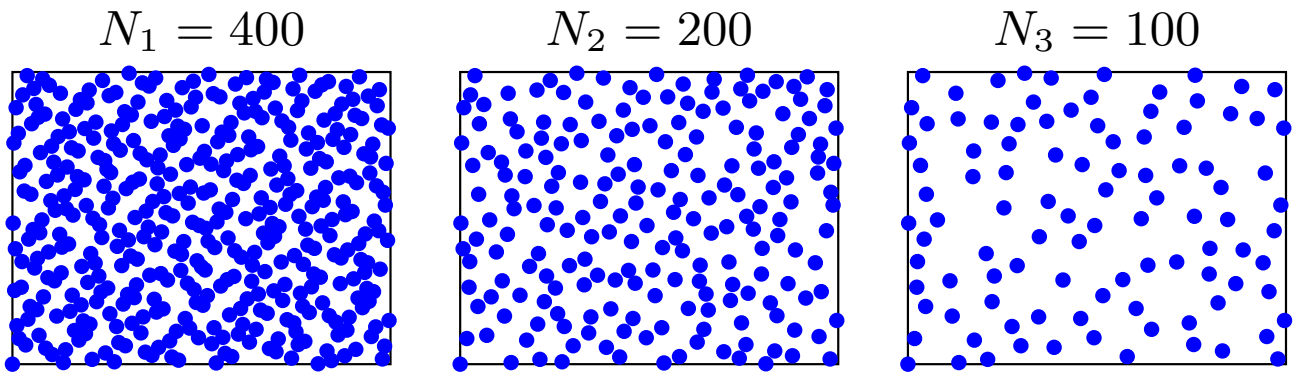


Figure: Sequence of nested sets of Halton centers in $\Omega = [0,1]^2$ using a greedy thinning algorithm with parameter $c_{\text{thin}} = 2$.

Side note: Interesting subproblem to construct suitable nested sequences with separation distances as large as possible.

Multilevel RBF interpolation of $f : \mathbb{R}^d \to \mathbb{R}$

1. For a set X of $N = |X|$ centers in $\mathbb{R}^d$, construct a nested sequence of L subsets X_ℓ of $N_\ell = |X_\ell|$ centers,

$$X = X_1 \supset X_2 \supset \ldots \supset X_{L-1} \supset X_L.$$

2. For every level ℓ, choose a shape parameter ε_ℓ based on the separation distance of X_ℓ, i. e. ideally $\varepsilon_\ell \in \mathcal{O}\left((N_\ell)^{\frac{1}{d}}\right)$, and define levelwise function spaces $S_\ell = S(X_\ell, \varepsilon_\ell)$.

3. Compute the multilevel interpolant bottom up ("c-to-f"):
   - $s_L|_{X_L} = f|_{X_L}$ (interpolant s_L on coarsest center set X_L)
   - for $\ell = (L-1)$ downto 1 do

$$s_\ell|_{X_\ell} = \underbrace{f|_{X_\ell} - \sum_{k=\ell+1}^{L} s_k|_{X_\ell}}_{\text{residual of } (\ell+1)\text{-level interpolant on } X_\ell}$$

   - The multilevel interpolant is $s := s_L + \ldots + s_1$.

Matrix representation of ML-RBF interpolation

Function spaces:

$$S_\ell = S(X_\ell, \varepsilon_\ell) = \text{span}\{\phi_{\varepsilon_\ell}(\|\mathbf{x} - \mathbf{x_i}\|) \mid \mathbf{x_i} \in X_\ell\}$$

Matrix blocks: $\mathbf{A}_{\ell,\mathbf{k}} = (\phi_{\varepsilon_k}(\|\mathbf{x_j} - \mathbf{x_i}\|)\ {}_{\substack{\mathbf{x_j} \in X_\ell \\ \mathbf{x_i} \in X_k}}$

Solve

$$\begin{pmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} & \mathbf{A_{13}} \\ & \mathbf{A_{22}} & \mathbf{A_{23}} \\ & & \mathbf{A_{33}} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \mathbf{f_1} \\ \mathbf{f_2} \\ \mathbf{f_3} \end{pmatrix}$$

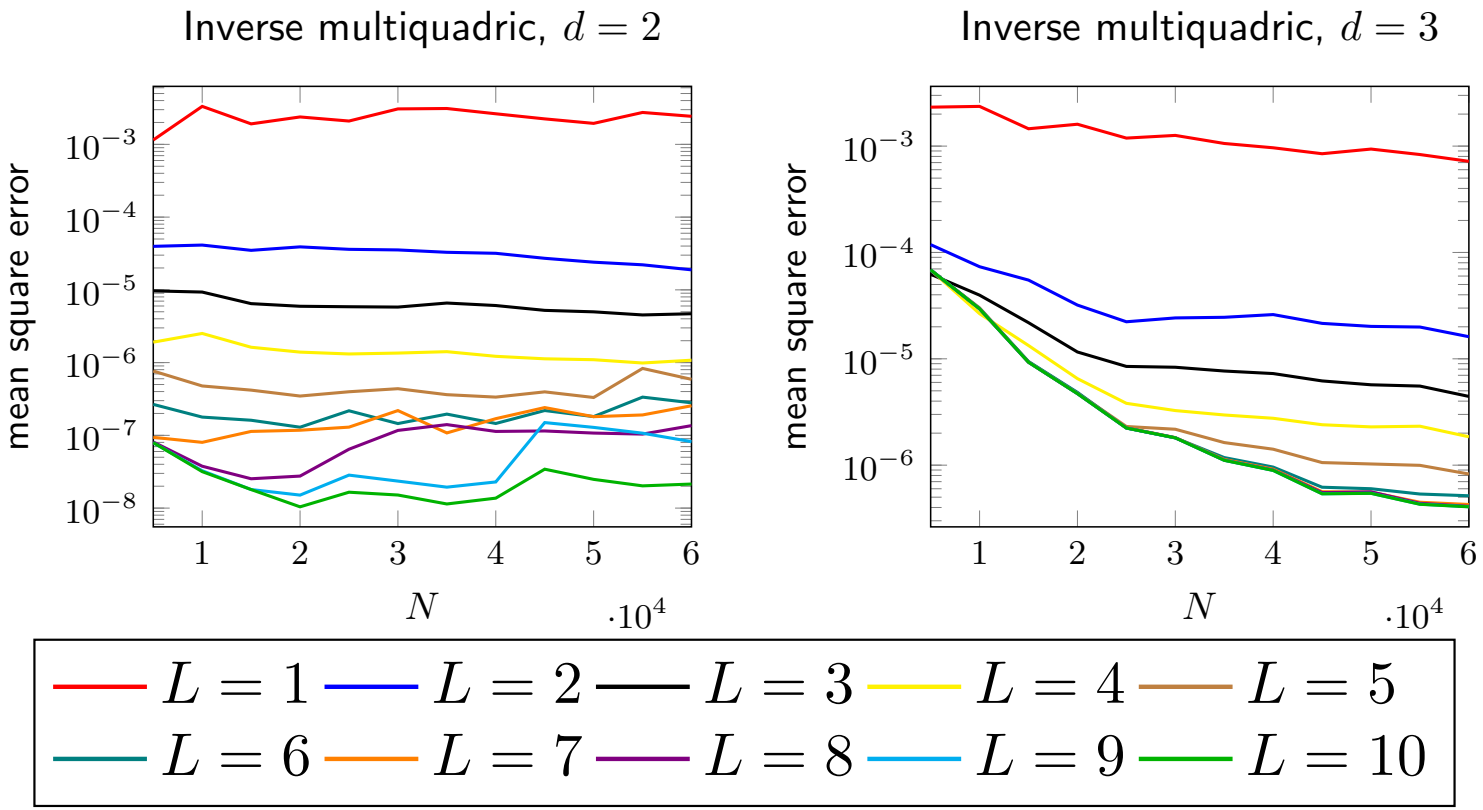
to obtain **interpolant**

$$\begin{aligned} s \;=\; & \alpha_3^T \cdot \text{ (basis functions of } S_3) \;+\; \alpha_2^T \cdot \text{ (basis functions of } S_2) \\ & + \alpha_1^T \cdot \text{ (basis functions of } S_1). \end{aligned}$$

Numerical results for Multilevel RBFs

$N_{\text{error}} \approx 10,000$ test points $X_{\text{error}} \subseteq [0.25, 0.75]^d$ (uniform grid)
mean square error when using N centers:

$$\text{error}(N) := \left(\frac{1}{N_{\text{error}}} \sum_{\mathbf{x} \in \mathbf{X}_{\text{error}}} \left(f(\mathbf{x}) - \mathbf{s}(\mathbf{x}) \right)^2 \right)^{1/2}$$



Numerical results for Multilevel RBFs

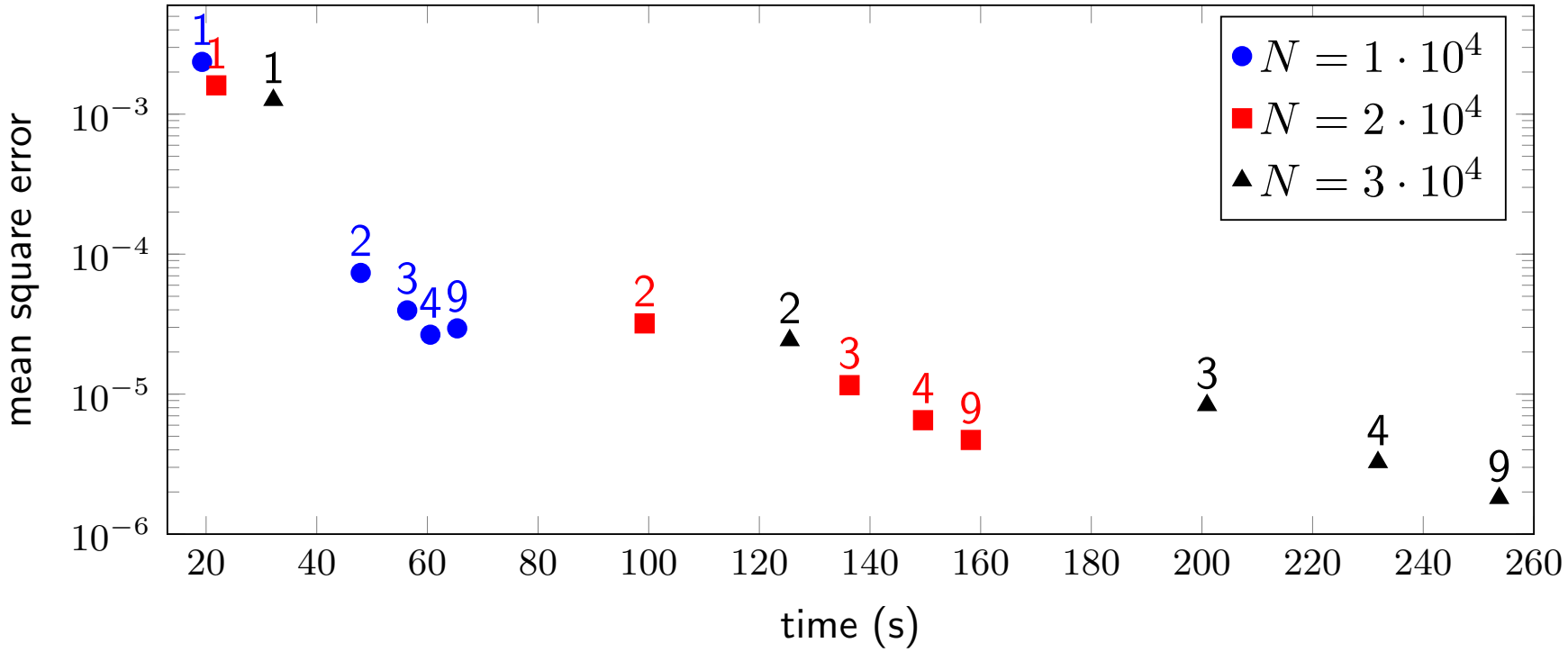


Figure: Multilevel interpolation of Franke's function using $\phi = \phi_{\mathrm{IMQ}}$ with scaling parameter $\delta_{\mathrm{ratio}} = 2$ and Halton centers in $\Omega = [0,1]^3$.

[L., Wende (2018), submitted]

Numerical results for Multilevel RBFs

Finally iteration/solver times

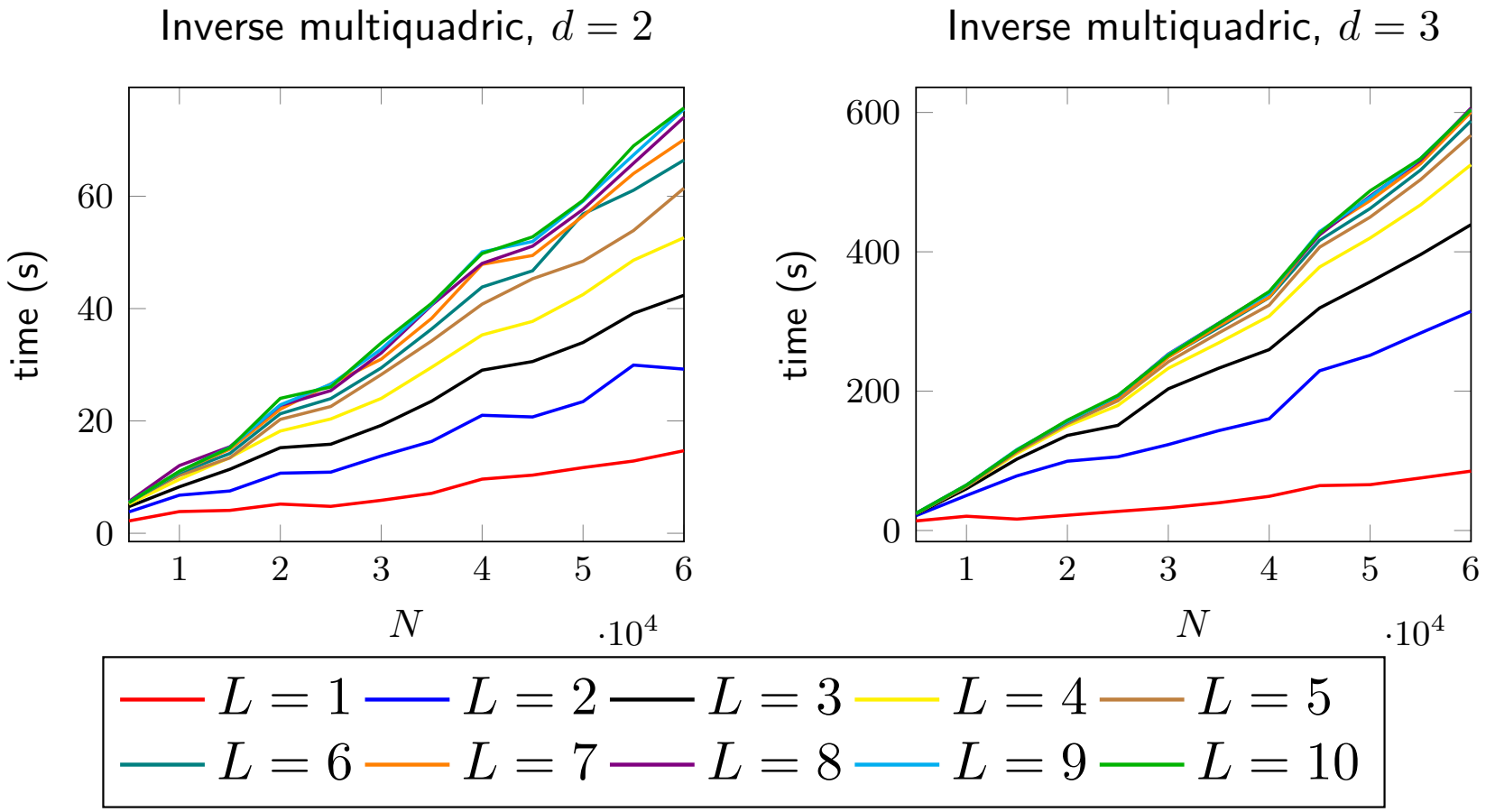


Figure: Time required for the construction of the multilevel interpolant for $d \in \{2, 3\}$ and ϕ_{IMQ}.

Next:

Remainder of this talk: **Fast Direct Solvers** using

1. **Structured/$\mathcal{H}$-matrices:**
   - Approximation $\widetilde{A}$ of A
   - Preconditioner P through $\mathcal{H}$-Cholesky/$\mathcal{H}$-LU factors
2. **Domain decomposition method**
3. **Change of basis:** Analytic right preconditioning
4. **Change of function space**
   - multilevel RBFs (application where fast direct solvers are needed for $N \gg 8,000$)

     Solution of an underdetermined problem:

     N interpolation centers, but

     $N + N_1 + \ldots + N_L \lesssim 2N$ basis functions
   - variable shape parameter
   - polynomial augmentation
5. **Nullspace method/Lagrangian augmentation** for saddle point problems resulting from conditionally positive RBFs

Next:

Remainder of this talk: **Fast Direct Solvers** using

1. **Structured/$\mathcal{H}$-matrices:**
   - Approximation $\widetilde{A}$ of A
   - Preconditioner P through $\mathcal{H}$-Cholesky/$\mathcal{H}$-LU factors
2. **Domain decomposition method**
3. **Change of basis:** Analytic right preconditioning
4. **Change of function space**
   - multilevel RBFs
   - variable shape parameter
   - polynomial augmentation (for $N \gg 8,000$)

     Solution of a saddle point problem:
     N interpolation centers,
     N radial plus M polynomial basis functions,
     M constraints
5. **Nullspace method/Lagrangian augmentation** for saddle point problems resulting from conditionally positive RBFs

Next:

Remainder of this talk: **Fast Direct Solvers** using

1. **Structured/$\mathcal{H}$-matrices:**
   - Approximation $\widetilde{A}$ of A
   - Preconditioner P through $\mathcal{H}$-Cholesky/$\mathcal{H}$-LU factors
2. **Domain decomposition method**
3. **Change of basis:** Analytic right preconditioning
4. **Change of function space**
   - multilevel RBFs
   - variable shape parameter
   - polynomial augmentation (for $N \gg 8,000$)

     Solution of a saddle point problem:

     N interpolation centers,
     N radial plus M polynomial basis functions,
     M constraints
5. **nullspace method/Lagrangian augmentation** for saddle point problems resulting from conditionally positive RBFs

Polynomial augmentation of RBFs

Motivation: Shape parameter ε influences RBFs to be

- global, smooth, of low variation if ε is small;
- local, spiky, of large variation if ε is large.

Polynomial augmentation idea:

1. Use a small number of (low degree, unisolvent) polynomial basis functions to capture the smooth part of the data;
2. Use a large number of spiky RBF basis functions to capture the oscillatory part of the data.
3. Enforce constraints to obtain a unique solution.

Polynomial augmentation of RBFs

Find RBF coeff. $\alpha_i \in \mathbb{R}$ and polynomial coeff. $\beta_j \in \mathbb{R}$ s. t.

$$\underbrace{s(\mathbf{x})}_{\text{interpolant}} = \sum_{j=1}^{N} \alpha_j \underbrace{\Phi(\mathbf{x_j}, \mathbf{x})}_{\text{RBFs}} + \sum_{j=1}^{M} \beta_j \underbrace{p_j(\mathbf{x})}_{\text{polynomials}}$$

satisfies the interpolation conditions

$$s(\mathbf{x}_i) = f(\mathbf{x}_i), \quad i = 1, \ldots, N$$

and the constraints $\displaystyle\sum_{j=1}^{N} \alpha_j p_i(\mathbf{x}_j) = 0, \quad i = 1, \ldots, M.$

In matrix form: Solve $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}.$

Side note: For conditionally positive definite RBFs, adding polynomials is necessary to ensure uniqueness of the interpolant.

Solving (cpd) RBF saddle point systems

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}.$$

Properties of the matrix $\widetilde{\mathbf{A}} := \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{O} \end{pmatrix}$:

- $\widetilde{\mathbf{A}}$ is indefinite.
- $\widetilde{\mathbf{A}}$ is symmetric.
- The main block $\mathbf{A}$ is positive definite for Gaussian and IMQ RBFs and positive definite on $\ker(B^T)$ for TPS and MQ RBFs.
- The blocks $\mathbf{A}$ (eval. of RBFs), $\mathbf{B}$ (eval. of polyn.) are dense.
- The main block $\mathbf{A}$ may be approximated by an $\mathcal{H}$-matrix.
- Storage of $\mathbf{B} \in \mathbb{R}^{N \times M}$ in $\mathcal{O}(N)$ since $M \in \mathcal{O}(1)$.

Solving (cpd) RBF saddle point systems

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathbf{T}} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}.$$

If $\mathbf{A}$ is SPD (e. g. for Gaussian or IMQ RBF):

$$\begin{aligned} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathbf{T}} & \mathbf{O} \end{pmatrix} &= \begin{pmatrix} \mathbf{I} & \\ \mathbf{B}^{\mathbf{T}}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ & \mathbf{S} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{L_A} & \\ \mathbf{B}^{\mathbf{T}}\mathbf{U_A}^{-1} & \mathbf{L_S} \end{pmatrix} \begin{pmatrix} \mathbf{U_A} & \mathbf{L_A}^{-1}\mathbf{B} \\ & \mathbf{U_S} \end{pmatrix} \end{aligned}$$

with

$$\mathbf{A} = \mathbf{L_A}\mathbf{U_A}, \qquad \mathbf{S} := -\mathbf{B}^{\mathbf{T}}\mathbf{A}^{-1}\mathbf{B} = \mathbf{L_S}\mathbf{U_S}.$$

Required: Fast solver for large, ill-conditioned, SPD $\mathbf{A}$.
(Schur complement $\mathbf{S}$ is no problem since it is small, i. e., $\mathcal{O}(1)$.)

Fast solver options: i) ($\mathcal{H}$-) Cholesky- or LU-factorization,
ii) domain decomposition & subproblem solvers for $\begin{pmatrix} \mathbf{A_i} & \mathbf{B_i} \\ \mathbf{B_i^T} & \mathbf{O} \end{pmatrix}$.

Solving (cpd) RBF saddle point systems

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathbf{T}} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}.$$

If $\mathbf{A}$ is NOT guaranteed to be SPD (e. g. for TPS or MQ RBF):

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathbf{T}} & \mathbf{O} \end{pmatrix} \overset{\lightning}{=} \begin{pmatrix} \mathbf{I} & \\ \mathbf{B}^{\mathbf{T}}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ & \mathbf{S} \end{pmatrix}, \quad \mathbf{S} := -\mathbf{B}^{\mathbf{T}}\mathbf{A}^{-1}\mathbf{B}.$$

An LU triangular factorization no longer exists.

Solving (cpd) RBF saddle point systems

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}.$$

If $\mathbf{A}$ is NOT guaranteed to be SPD (e. g. for TPS or MQ RBF):

But a **nullspace basis** $\mathbf{C}$ of $\mathbf{B^T}$ (i. e. $\mathbf{B^T C} = \mathbf{O}$ and rank($\mathbf{C}$)$=N-M$) allows an **antitriangular factorization**

$$\begin{pmatrix} [\mathbf{B}\ \mathbf{C}] & \mathbf{O} \\ \mathbf{O} & \mathbf{I_M} \end{pmatrix}^T \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{O} \end{pmatrix} \begin{pmatrix} [\mathbf{B}\ \mathbf{C}] & \mathbf{O} \\ \mathbf{O} & \mathbf{I_M} \end{pmatrix} = \begin{pmatrix} \mathbf{B^T A B} & \mathbf{B^T A C} & \mathbf{B^T B} \\ \mathbf{C^T A B} & \mathbf{C^T A C} & \mathbf{O} \\ \mathbf{B^T B} & \mathbf{O} & \mathbf{O} \end{pmatrix}.$$

Question: How can compute $\mathbf{C}$ and $\mathbf{C^T A C} \in \mathbb{R}^{(N-M)\times(N-M)}$?

Answer: Use M Householder reflections to compute $\mathbf{QB} = \mathbf{R}$, the last $N-M$ columns of $\mathbf{Q} = [\mathbf{E}\ \mathbf{C}]$ yield $\mathbf{C}$ in data-sparse form which allows the efficient computation of $\mathbf{C^T A C}$.

Question: How can we solve $\mathbf{C^T A C}x = b$? ($\mathbf{B^T B} \in \mathbb{R}^{M\times M}$ is no problem.)

Solving (cpd) RBF saddle point systems

Question: How can we solve $\mathbf{C}^{\mathbf{T}}\mathbf{A}\mathbf{C}x = b$?

Answer: $\mathbf{A}' := \mathbf{C}^{\mathbf{T}}\mathbf{A}\mathbf{C} \in \mathbb{R}^{(\text{N-M})\times(\text{N-M})}$ is symmetric positive definite,

$$\text{cond}(\mathbf{A}') \leq \text{cond}\left(\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathbf{T}} & \mathbf{O} \end{pmatrix}\right) \text{cond}(\mathbf{C})^2,$$

$\mathbf{A}'$ can be approximated by an $\mathcal{H}$-matrix,

$\mathbf{A}'$ allows $\mathcal{H}$-LU or $\mathcal{H}$-Cholesky factorizations,
domain decomposition solvers can be applied.

Condition number of nullspace projected system

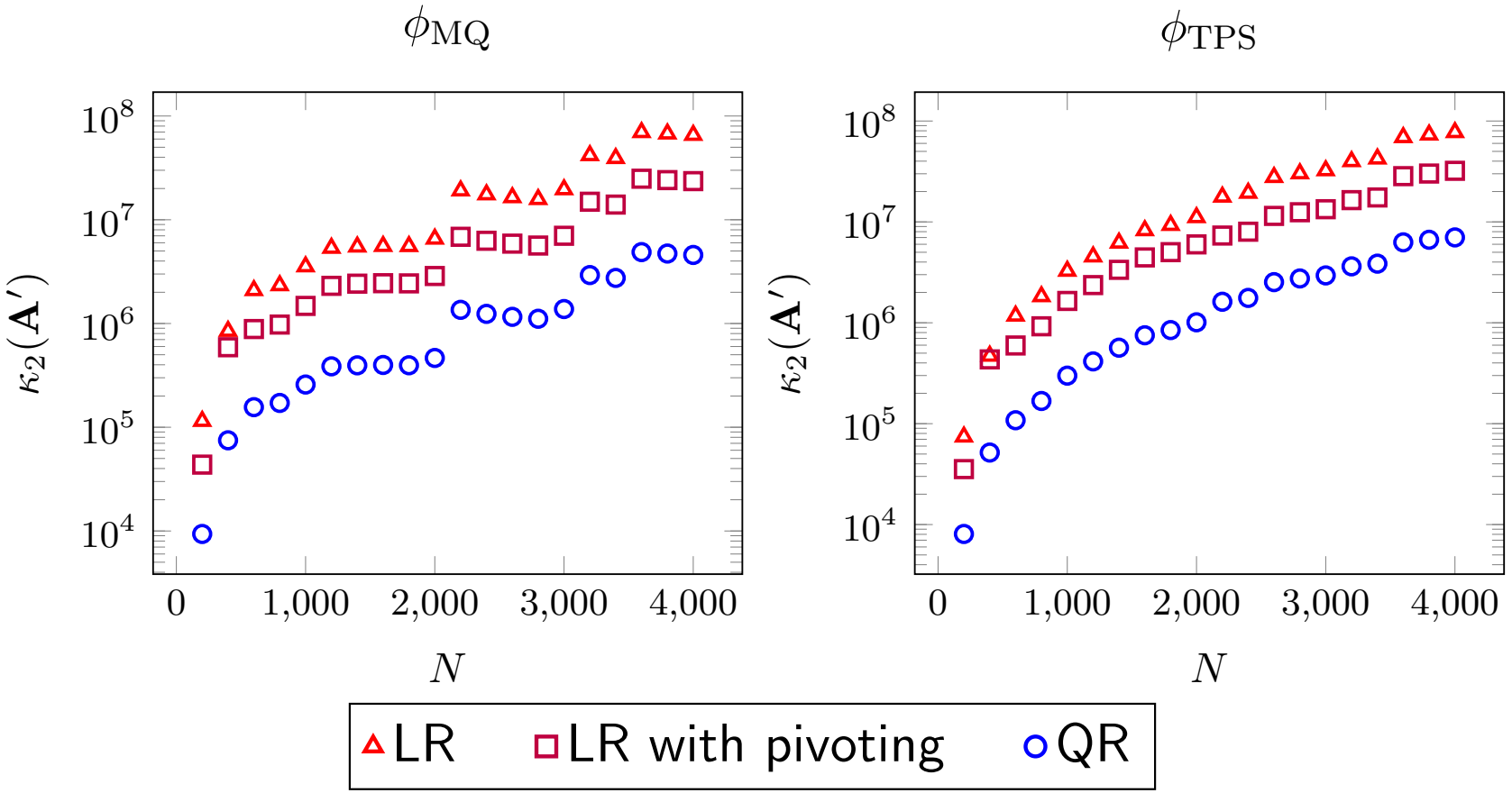


Figure: Spectral condition number of $\mathbf{A}' = \mathbf{C}^{\mathsf{T}}\mathbf{A}\mathbf{C}$ wrt. the number N of Halton centers in $[0,1]^2$ using a fixed polynomial order $m = 2$ $(M = 6)$.

Condition number of nullspace projected system

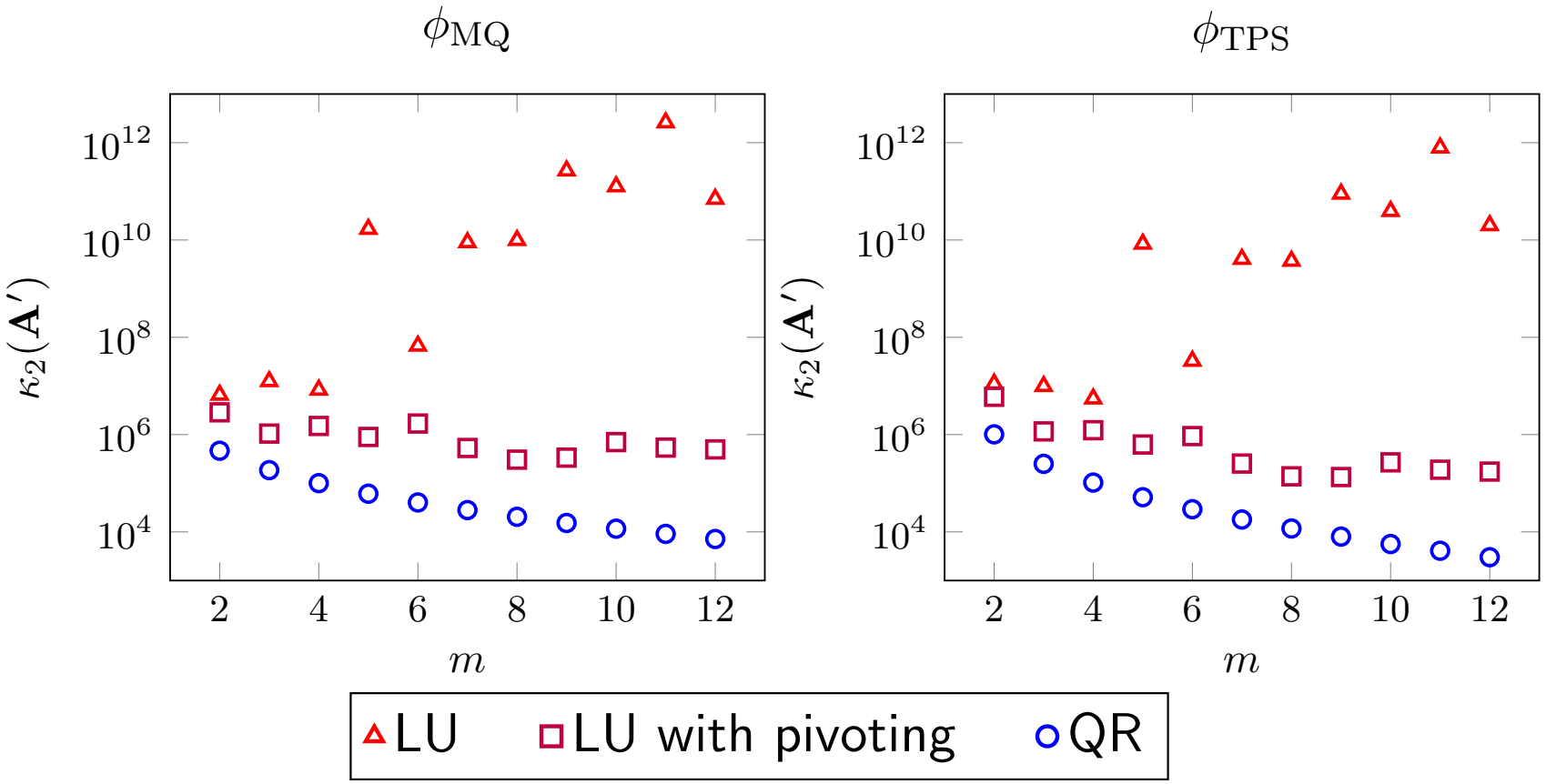


Figure: Spectral condition number of $\mathbf{A}' = \mathbf{C}^{\mathsf{T}}\mathbf{A}\mathbf{C}$ wrt. the polynomial order $m = 2, \ldots, 12$ for $N = 2000$ Halton centers in $[0, 1]^2$.

Fast solver for the nullspace projected system

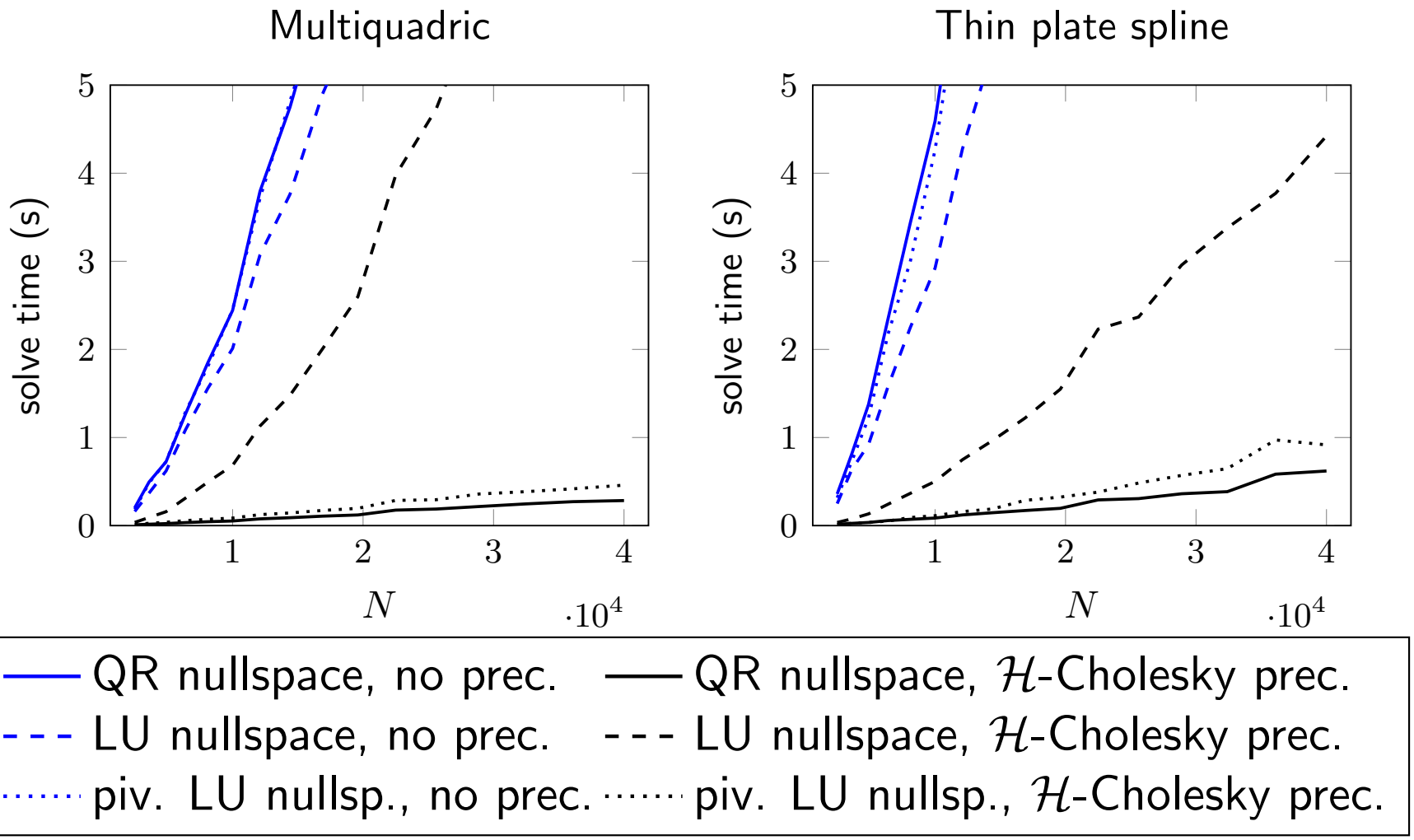


Figure: Unpreconditioned and $\mathcal{H}$-Cholesky preconditioned GMRes iteration time for up to $N = 40,000$ Halton centers in $[0,1]^2$.

Solving (cpd) RBF saddle point systems

Domain decomposition solvers

for the nullspace projected system $\mathbf{C}^{\mathbf{T}}\mathbf{A}\mathbf{C}x = b$

- allows for up to $N = 160,000$ centers,
- needs overlapping subdomains and coarse grid correction,
- typically outperforms $\mathcal{H}$-Cholesky factorization as a preconditioner,
- `[L., Wende (2018)]`, to appear.

Recap

Remainder of this talk: **Fast Direct Solvers** using

1. **Structured/$\mathcal{H}$-matrices:** ✓
   - Approximation $\widetilde{A}$ of A
   - Preconditioner P through $\mathcal{H}$-Cholesky/$\mathcal{H}$-LU factors
2. **Domain decomposition method** (✓)
3. **Change of basis:** Analytic right preconditioning ?
4. **Change of function space** ✓
   - multilevel RBFs
   - variable shape parameter
   - polynomial augmentation
5. **Nullspace method/Lagrangian augmentation** for saddle point problems resulting from conditionally positive RBFs ✓

Alternate bases/preconditioning

Ac=y	with $A = (b_{ij}) = (b_j(x_i))$,	alternate basis
Bd=y	with $B = (a_{ik}) = (a_k(x_i))$,	$a_k(\cdot) = \sum_{i=1}^{N} t_{ik} b_i(\cdot)$

Change of basis $\iff$ Right-Preconditioning

$$a_k(\cdot) = \sum_{i=1}^{N} t_{ik} b_i(\cdot)$$

$$\underbrace{AT}_{B} \underbrace{T^{-1}c}_{d} = y.$$

Challenge: Compute B directly using analytical considerations, not by matrix-matrix multiplication of two highly ill-conditioned matrices A and T.

Stable bases for RBFs

Approaches for **alternate, stable bases**:

- Contour-Padé/RBF-RA
- RBF-QR, Hilbert-Schmidt SVD
- RBF-GA (only for Gaussian RBF)

Dilemma:

- Alternate bases work only for relatively small numbers of centers ($N \ll 8,000$);
- Alternate bases have up to ten times higher work complexities (for computing the interpolation matrix).

To do:

- Multilevel RBF with stable bases on coarse levels;
- Fast methods (not just solvers) for stable bases;
- Fasshauer: "potential exists but dangers abound" (Remark 13.13)

Summary of this talk/work

- RBF interpolation leads to large, dense, ill-conditioned systems of equations.
- If not the straighforward interpolation itself, then modifications (multilevel RBFs, polynomial augmentation) or other applications (machine learning) require fast direct solvers for these systems.
- Rank-structured ($\mathcal{H}$-)matrices offer a tool to construct such solvers.
- Many intertwined subproblems to consider (coarsening of point sets, saddle point systems, etc.) that affect the approximation accuracy and the solver complexity.

Summary of this talk/work

- RBF interpolation leads to large, dense, ill-conditioned systems of equations.
- If not the straighforward interpolation itself, then modifications (multilevel RBFs, polynomial augmentation) or other applications (machine learning) require fast direct solvers for these systems.
- Rank-structured ($\mathcal{H}$-)matrices offer a tool to construct such solvers.
- Many intertwined subproblems to consider (coarsening of point sets, saddle point systems, etc.) that affect the approximation accuracy and the solver complexity.

Thank you for your attention!