Adaptive algorithms for kernel-based collocation methods

Leevan Ling

Hong Kong Baptist University

lling@hkbu.edu.hk

International Conference of Kernel-Based Approximation Methods in Machine Learning
May 19–21, 2017

Overview

Objective: Design a block-box for solving elliptic PDEs

Kansa methods

Generating quasi-random kernels

Adaptively select trial spaces

Numerical demonstrations

Collaborators:

- Robert Schaback
- Sung Nok Chiu
- Ka Chun Cheung
- Meng Chan

Acknowledgements: This work was supported by a GRF Grant from the Hong Kong Research Grant Council and an FRG Grant from Hong Kong Baptist University.

Kansa methods

Consider a PDE, with solution u^*, given by

$$\mathcal{L}u = f \text{ in } \Omega \qquad \text{and} \qquad \mathcal{B}u = g \text{ on } \Gamma = \partial\Omega.$$

An RBF is a smooth scalar function $\phi : \mathbb{R}^+ \to \mathbb{R}$, which usually is induced from a kernel $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ in today's applications.

$$\phi(\|x - y\|_2) = \Phi(x, y)$$

Aim: Identify an approximation to u^* from the *trial space*

$$\mathcal{U}_Z = \mathcal{U}_{Z,\Omega,\Phi} := \text{span}\{\Phi(\cdot - z_j) : \ z_j \in Z\}$$

for some set of trial centers $Z = \{z_1, \ldots, z_{n_Z}\} \subset \overline{\Omega}$

Equivalently, we need to identify the unknown λ_j in

$$u = \sum_{j=1}^{n_Z} \lambda_j \phi(\|\cdot - z_j\|_2) = \sum_{j=1}^{n_Z} \lambda_j \Phi(\cdot, z_j)$$

We do so by imposing some *collocation conditions* at some sets of collocation points

$$X = \{x_1, \ldots, x_{n_x}\} \subset \Omega \quad \& \quad Y = \{y_1, \ldots, y_{n_y}\} \subset \Gamma$$

and obtain

$$\begin{aligned} \mathcal{L}u(x_i) &= \sum \lambda_j \mathcal{L}\phi(\|x_i - z_j\|) \quad \text{for } x_i \in X \text{ and } z_j \in Z \\ \mathcal{B}u(y_i) &= \sum \lambda_j \mathcal{B}\phi(\|y_i - z_j\|) \quad \text{for } y_i \in Y \text{ and } z_j \in Z \end{aligned}$$

Now we have

- if $Z = X \cup Y$, exactly determined (original) Kansa system
  - unsymmetric
  - matrix could be singular
  - works fine in practice
- if $n_Z < n_X + n_Y$, overdetermined system
  - solved by ℓ^∞-minimization (LL, Schaback 08')
  - solved by least-squares (Cheung, LL, Schaback)
  - in $\mathbb{R}^2$ or $\mathbb{R}^3$, define *weighted LS solutions* by

$$U_Z := \arg\inf_{u \in \mathcal{U}_Z} \left\{ \|\mathcal{L}u - f\|^2_{\ell^2(X)} + \left(\frac{h_Y^{d-4}}{h_X^d} \right) \|u - g\|^2_{\ell^2(Y)} \right\}$$

  - some convergence analysis available

Schematic point sets

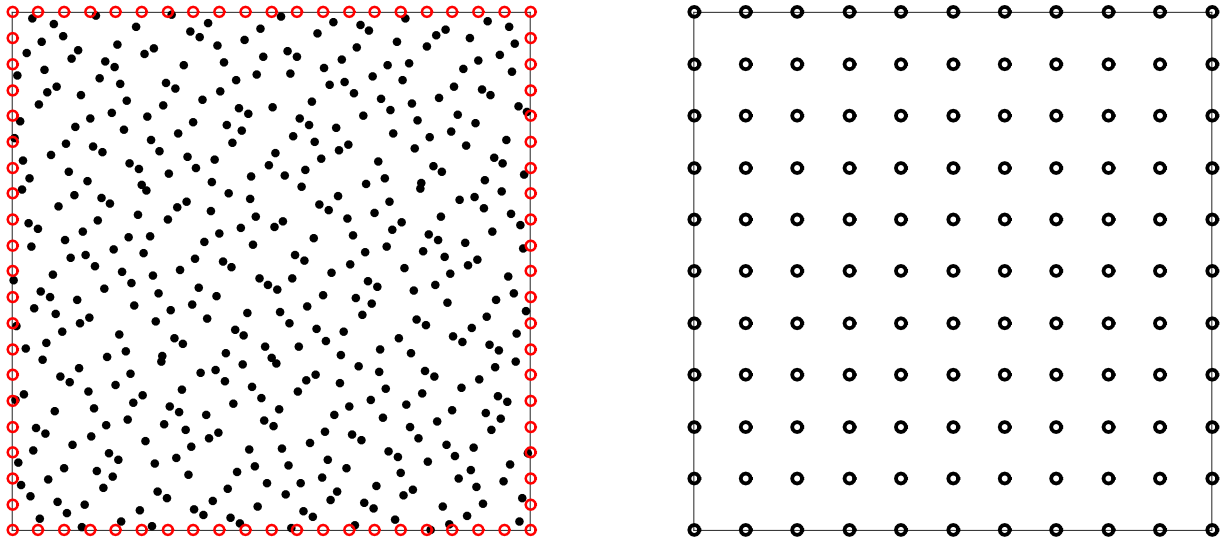


Figure: Collocations X and Y (left), and trial Z (right), used to solve various PDEs.

Theorem (Convergence estimate)

Suppose that

- *Domain Ω is nice and $\partial\Omega$ is smooth*
- *Differential operator is second order strongly elliptic and the PDE has a classical solution u^**
- *Kernel Φ reproduces $H^m(\Omega)$ with $m > 2 + d/2$*
- *Over-sampling $(h_X^{m-2} + h_Y^{m-2})/q_Z^{m-2} < C_{\Omega,\Phi,\mathcal{L}}$*

Then, for $m > 2 + \frac{d}{2}$,

$$\|U_Z - u^*\|_{2,\Omega} \quad \leq \quad C h_Z^{m-\frac{d}{2}-2} \|u^*\|_{m,\Omega}$$

and, for $m > 3 + \frac{d}{2}$,

$$\|U_Z - u^*\|_{2,\Omega} \quad \leq \quad C h_Z^{m-2} \|u^*\|_{m,\Omega}$$

with some $C = C_{\Omega,\Phi,\mathcal{L},\nu,\rho_X,\rho_Y,\rho_Z,\gamma_X,\gamma_Y}$

Example

A symmetric positive definite kernel Φ_m on $\mathbb{R}^d$ with smoothness m that satisfies

$$c_{\Phi_m}(1+\|\omega\|_2^2)^{-m} \leq \widehat{\Phi_m}(\omega) \leq C_{\Phi_m}(1+\|\omega\|_2^2)^{-m}$$

for all $\omega \in \mathbb{R}^d$ with two constants $0 < c_{\Phi_m} \leq C_{\Phi_m}$.

- For any $m > d/2$, its native space $\mathcal{N}_{\Phi_m,\mathbb{R}^d}$ is norm-equivalent to $H^m(\mathbb{R}^d)$
- Standard *Whittle-Matérn-Sobolev* kernel with exact Fourier transform $(1+\|\omega\|_2^2)^{-m}$ that takes the form

$$\Phi_m(x) := \|x\|_2^{m-d/2}\mathcal{K}_{m-d/2}(\|x\|_2) \quad \text{for all } x \in \mathbb{R}^d,$$

  where $\mathcal{K}_\nu$ is the Bessel functions of the second kind
- The compactly supported piecewise polynomial Wendland functions are another examples

Sketch of proof

- Boundary regularity estimate:

$$\|u\|_{k,\Omega} \quad \leq \quad C_{\Omega,\mathcal{L},k} \left(\|\mathcal{L}u\|_{k-2,\Omega} + \|u\|_{k-1/2,\Gamma} \right)$$

  for $2 \leq k \leq m$ and all $u \in H^m(\Omega)$

- Inverse inequality for trial functions:

$$\|u\|_{m,\Omega} \leq C_{\Omega,\Phi_m,k}\, q_Z^{-m+k} \|u\|_{k,\Omega}$$

  for $d/2 < k \leq m$ and all $u \in \mathcal{U}_Z$

- Sampling inequality $\rightarrow$ Stability

$$\|u\|_{k,\Omega} \leq C_{\Omega,\Phi_m,\mathcal{L},k} \left(h_X^{\frac{d}{2}+2-k} \|\mathcal{L}u\|_X + h_Y^{\frac{d}{2}-k} \|u\|_Y \right)$$

  for $2 \leq k \leq m$ and all $u \in \mathcal{U}_Z$ if h_X, h_Y small

- Consistency via interpolant $I_{\Phi,Z}u^*$

Kansa methods for surface PDEs

(Cheung, LL & Chen, LL)

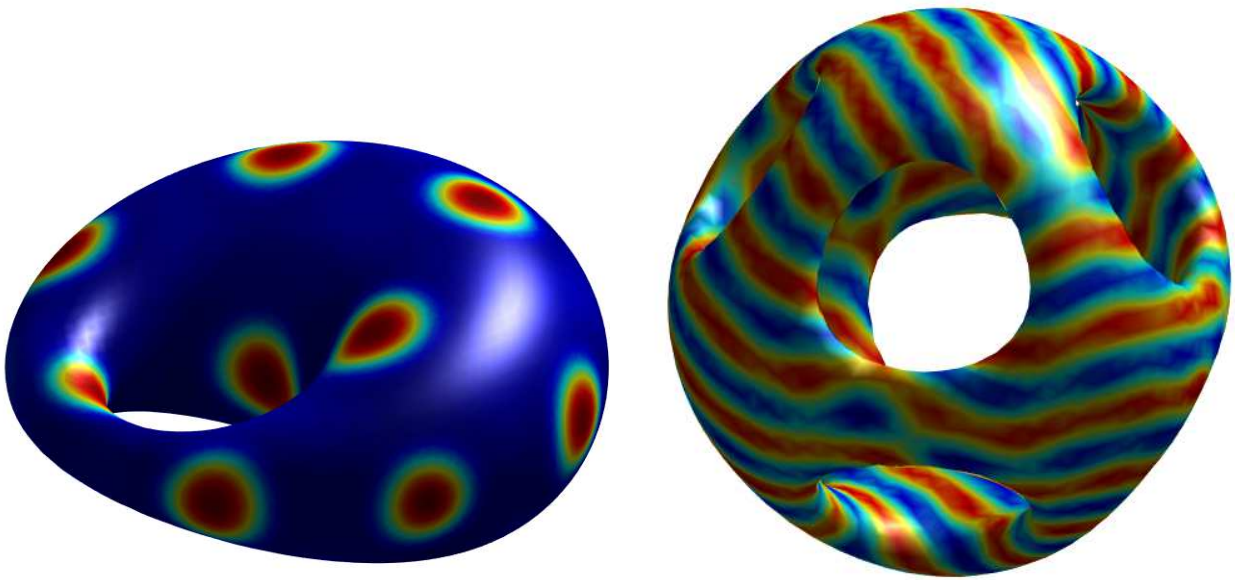


Figure: Turing spot and stripe patterns

The catch:

Still no convergence theory for Kansa methods when we use Gaussian, multiquadrics, and many other RBFs

Motivation: $(r \leftarrow \varepsilon r)$

Kernel	ε	# basis	L^∞ error
MQ	2	961	0.0039
	1	813	5.4408(-5)
	0.5	251	8.6928(-5)
GA	2	413	1.7666(-5)
	1	163	1.6022(-8)
	0.5	77	0.2747

Kernel	ε	# basis	L^∞ error
MQ	$(1/2, 2)$	859	7.3448(-7)
	$(1/3, 3)$	894	5.6504(-7)
	$(1/4, 4)$	920	1.4412(-6)
GA	$(1/2, 2)$	385	2.0364(-11)
	$(1/3, 3)$	574	2.5763(-10)
	$(1/4, 4)$	673	7.6422(-10)

Motivation: $(r \leftarrow \varepsilon r)$

Kernel	ε	# basis	L^∞ error
MQ	2	961	0.0039
	1	813	5.4408(-5)
	0.5	251	8.6928(-5)
GA	2	413	1.7666(-5)
	1	163	1.6022(-8)
	0.5	77	0.2747

Kernel	ε	# basis	L^∞ error
MQ	$(1/2, 2)$	859	7.3448(-7)
	$(1/3, 3)$	894	5.6504(-7)
	$(1/4, 4)$	920	1.4412(-6)
GA	$(1/2, 2)$	385	2.0364(-11)
	$(1/3, 3)$	574	2.5763(-10)
	$(1/4, 4)$	673	7.6422(-10)

Quasi-random kernels

To setup Kansa methods, users have to provide the following point sets:

- Interior collocation points $X = \{x_1, \ldots, x_{n_X}\} \subset \Omega$
- Boundary collocation points $Y = \{y_1, \ldots, y_{n_Y}\} \subset \Gamma$,
- Trial centers $Z = \{z_1, \ldots, z_{n_Z}\} \subset \overline{\Omega}$

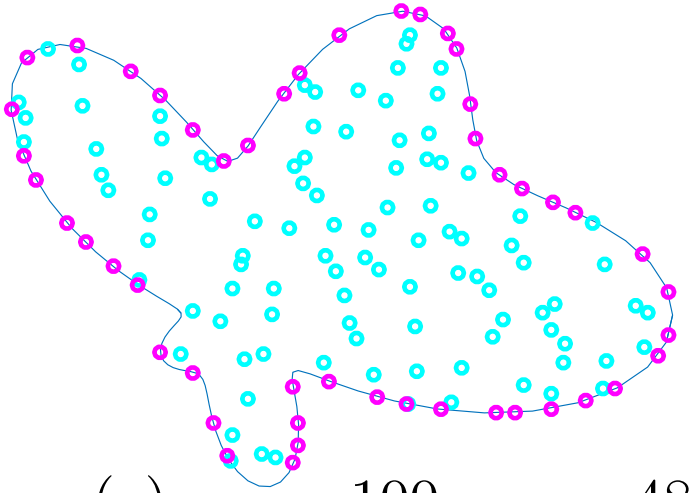
If ones adopt the variable-shape formulation, shape parameters have to be specified at each trial centers by

- Shape parameters $\mathcal{E} = \{\epsilon_1, \ldots, \epsilon_{n_Z}\} \subset \mathbb{R}_+$

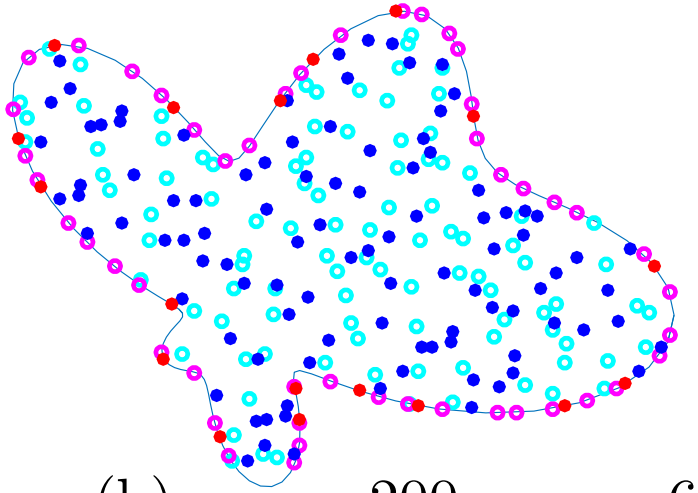
and use kernel

$$\Phi_{\epsilon_i}(\,\cdot\,) = \phi(\epsilon_j \| \,\cdot\, - z_j\|_2)$$

A schematic demonstration of data refinement.



(a) $n_X = 100$, $n_Y = 48$



(b) $n_{X'} = 200$, $n_{Y'} = 66$

Figure: (a) Request $n_X = 100$ interior point and estimate that $n_Y = 48$ is required to maintain a similar fill distance.

Figure: (b) Request an extended set of $n_{X'} = n_X + 100$ interior points that requires $n_{Y'} = 48 + 18$ boundary points to maintain $h_{X'} \sim h_{Y'}$.

- Focus on low-discrepancy sequences by quasi-Monte Carlo methods
  Examples: Faure, Halton, Sobol', etc.
- Interior points: by a quasi-random sequence in $\mathbb{R}^d$ and

$$\text{sign}(x) = \begin{cases} +1 & x \notin \overline{\Omega} \\ 0 & x \in \partial\Omega \\ -1 & x \in \Omega. \end{cases}$$

- Estimate $n_Y = n_Y(n_X, \Omega)$ s.t. $h_{X,\Omega} \sim h_{Y,\Gamma}$
- Boundary points: by a quasi-random sequence in $\mathbb{R}^d$ and by a PDF generated from the metric tensor of Γ

Adaptive trial space selection

In hand, we have

- Interior collocation: $X_1 \subset X_2 \subset \cdots \subset X_k \subset \cdots$
- Boundary collocation: $Y_1 \subset Y_2 \subset \cdots \subset Y_k \subset \cdots$
- Trial centers: $Z_1 \subset Z_2 \subset \cdots \subset Z_k \subset \cdots$

Along with sets of points, we generate a quasi-random numbers $\theta_j \in [0, 1]$ for each $z_j \in Z$

- Cheap to compute
- Contain "implicit" spatial information
- Can be used to generate shape parameter ϵ_j for kernel at z_j

Aim: Deals with the corresponding Kansa system

Basic idea

Consider the matrix system:

$$A\boldsymbol{\eta} = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 5 & 2 \\ 3 & 6 & 1 \end{bmatrix} \boldsymbol{\eta} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \mathbf{b}.$$

- Preselect 1×1 subsystem: row–3 & column–2
- To expand to 2×2, solve

$$[6]\, \boldsymbol{\eta}^{(1)} = [6]$$

  with solution $\boldsymbol{\eta}^{(1)} = 1$
- Evaluate residual $A\widehat{\boldsymbol{\eta}}^{(1)} - \mathbf{b} = [2,\, 1,\, 0]^T$
- Select row–1

To complete the subsystem expansion, we have to select a column.

$$A\boldsymbol{\eta} = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 5 & 2 \\ 3 & 6 & 1 \end{bmatrix} \boldsymbol{\eta} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \mathbf{b}.$$

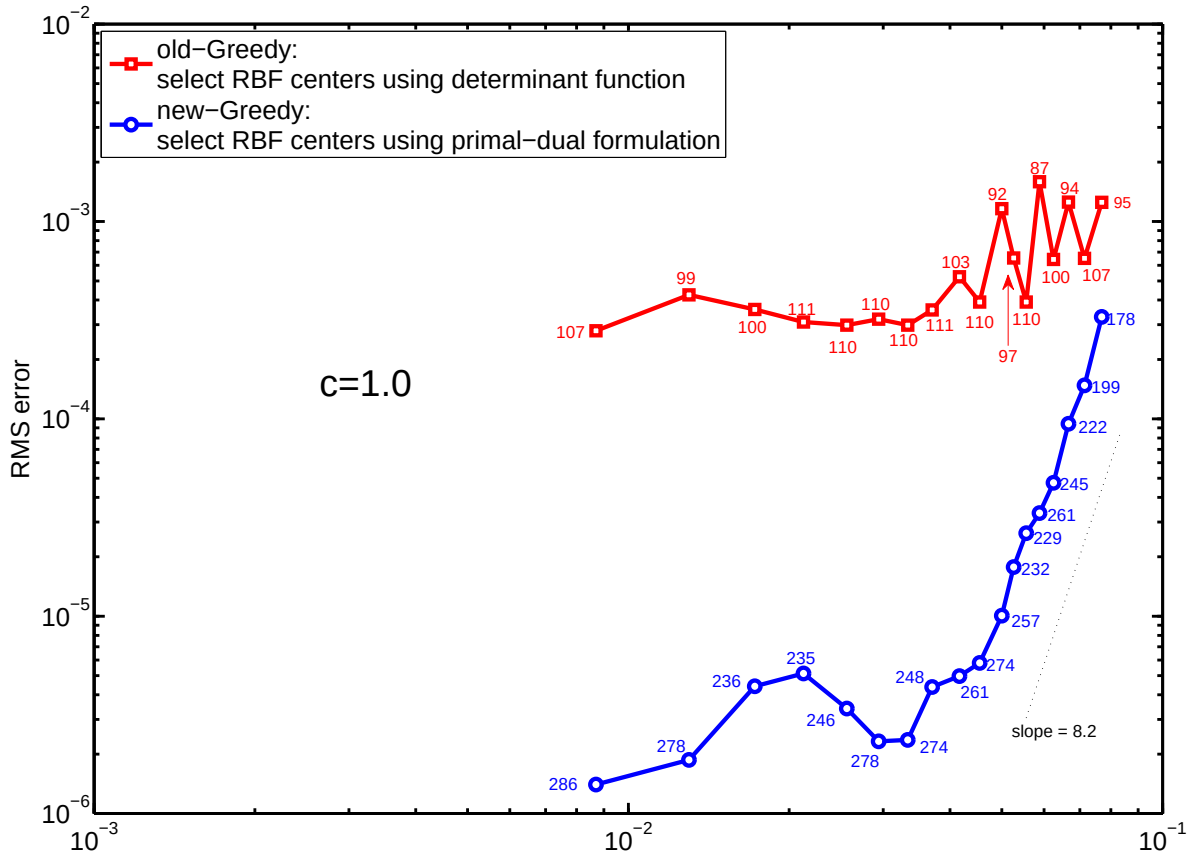
- Determinant criterion (LL, Opfer, Schaback 06'): Treat the to-be-determined 2×2 matrixas a function of column index

$$\underbrace{\begin{bmatrix} 1 & 4 \\ 3 & 6 \end{bmatrix}}_{=A_{2\times 2}(1)} \quad \text{or} \quad \underbrace{\begin{bmatrix} 4 & 3 \\ 6 & 1 \end{bmatrix}}_{=A_{2\times 2}(3)},$$

- $|det(A_{2\times 2}(1))| = 6$ and $|det(A_{2\times 2}(3))| = 14$
- Select column–1
- Latest criterion: Dual residual (LL, Schaback 09')

Example

Using unscaled multiquadrics RBF, reduce h_Z



- Sequential approach is slow....
- To speed up, we need to update in block-form (LL 16')
- By the same 4 basic steps:

  Step-1: Compute local residuals
  Step-2: Select rows to be added
  Step-3: Select columns to be added
  Step-4: Check condition number

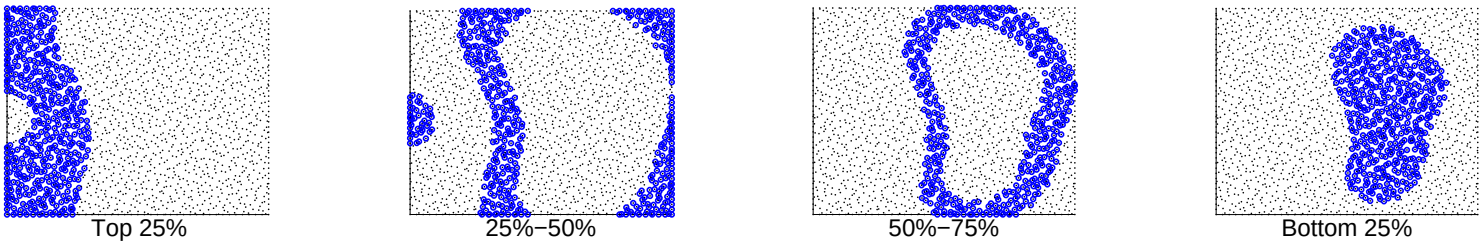
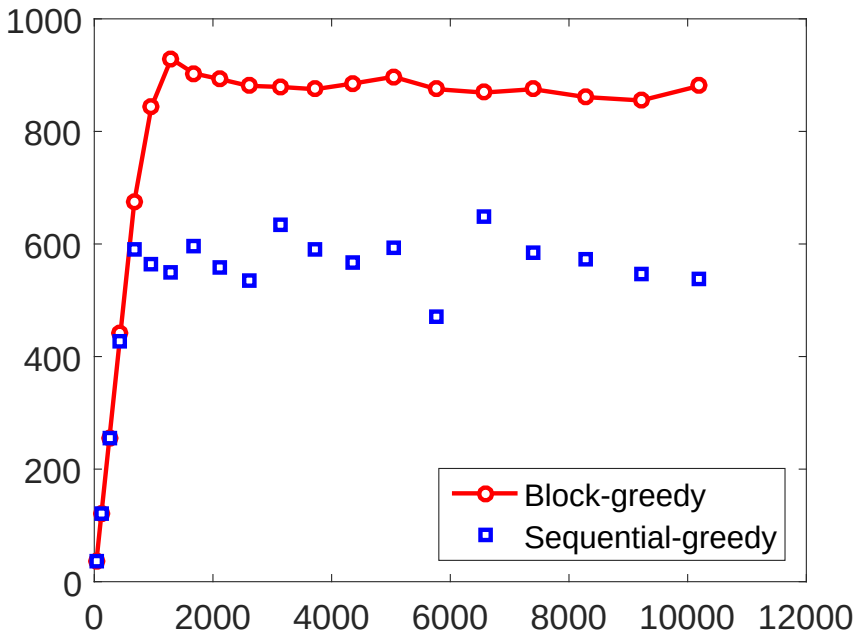


Figure: Partitioning trial centers by dual residuals.

Theorem

Let A be an $M \times N$ matrix with full rank M; choose a large $M \times K$ submatrix formed by $K \leq M$ columns of A such that it is numerically of full rank. The block-greedy algorithm can be implemented so that

$$\textit{complexity} \lesssim \mathcal{O}(NK^2) \quad \textit{and} \quad \textit{storage} \sim K(M+N)+\frac{1}{2}K^2$$

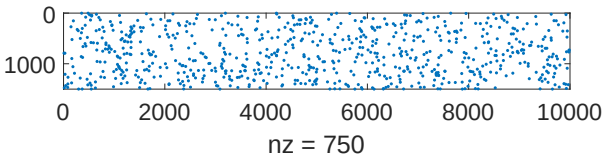


Example

Block-greedy vs permuted-QR

- Fix $M = 1500$ and let $2000 \leq N \leq 20000$
- Generate $M \times N$ random matrices
- Select $K = M$ columns

(a) Block greedy



(b) QR

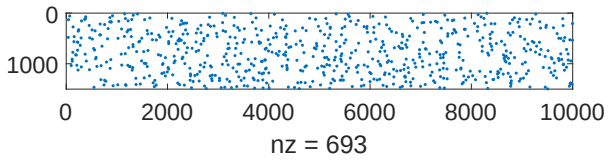
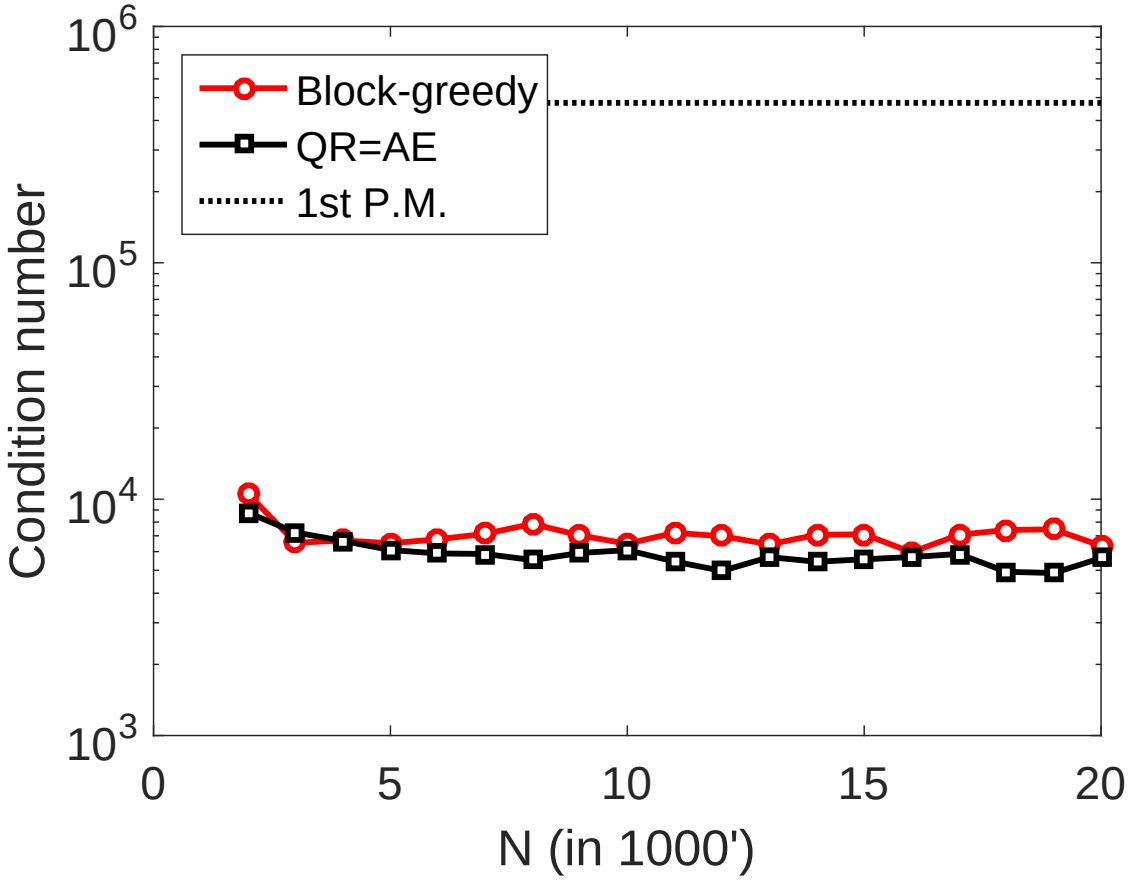
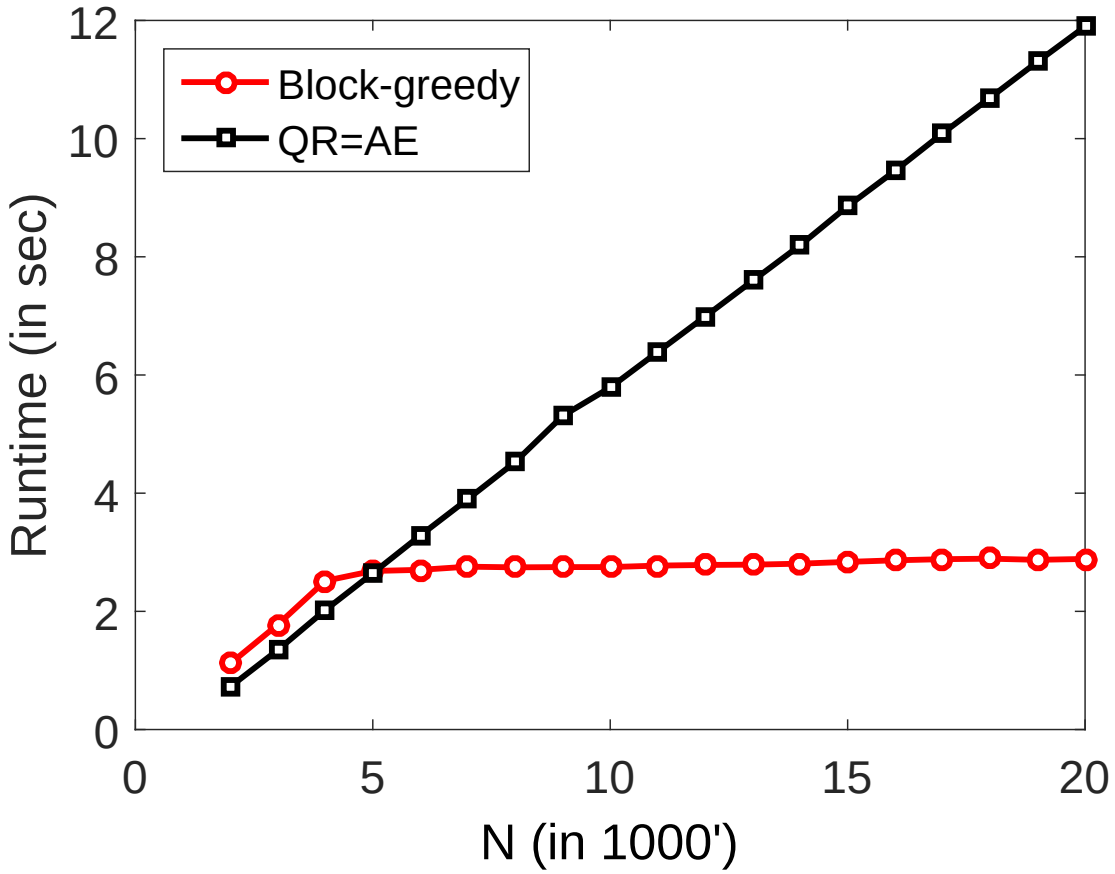


Figure: Entries of $B^{-1}A$ with absolute values greater than 1, where B is the $M \times M$ submatrix formed by the selected M columns out of $A \in \mathbb{R}^{M \times N}$ with $M = 1500$ and $N = 10000$

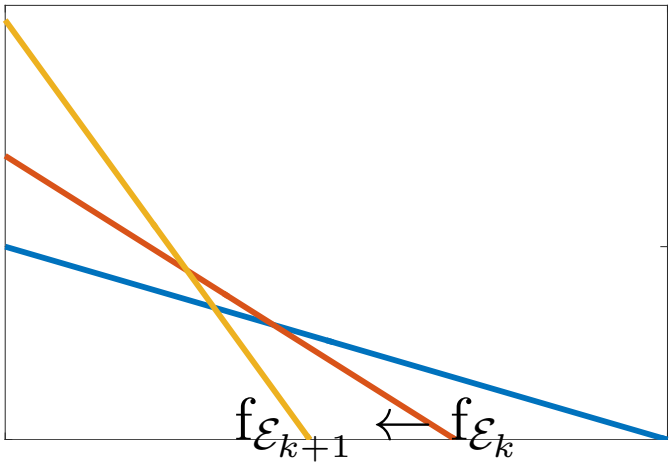

Block-greedy
QR=AE
1st P.M.
Condition number
10^6
10^5
10^4
10^3
0
5
10
15
20
N (in 1000')


Block-greedy
QR=AE
Runtime (in sec)
N (in 1000')
0
2
4
6
8
10
12
0
5
10
15
20

System-free algorithm

(Chiu, LL)

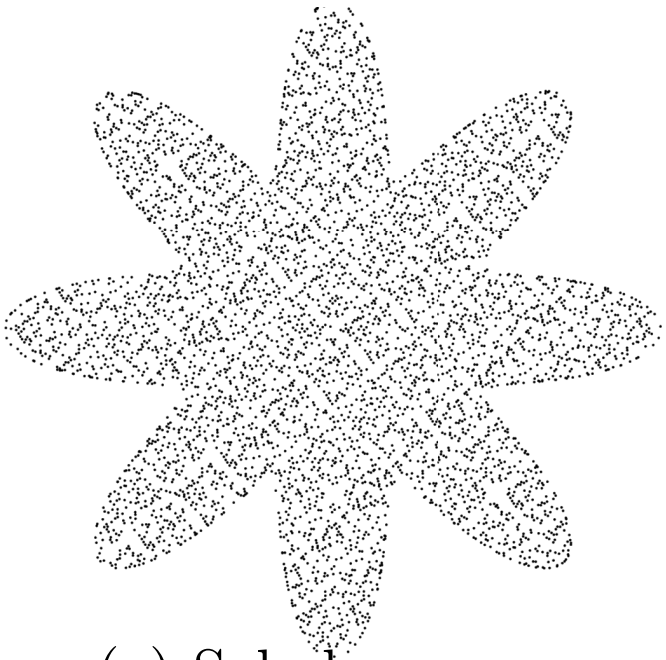
- A modification to the block-greedy algorithm
- Run on $X_k, Y_k, Z_k, \mathcal{E}_k$
- Using the selected shape parameters, we design a *distribution* for $\mathcal{E}_{k+1}$



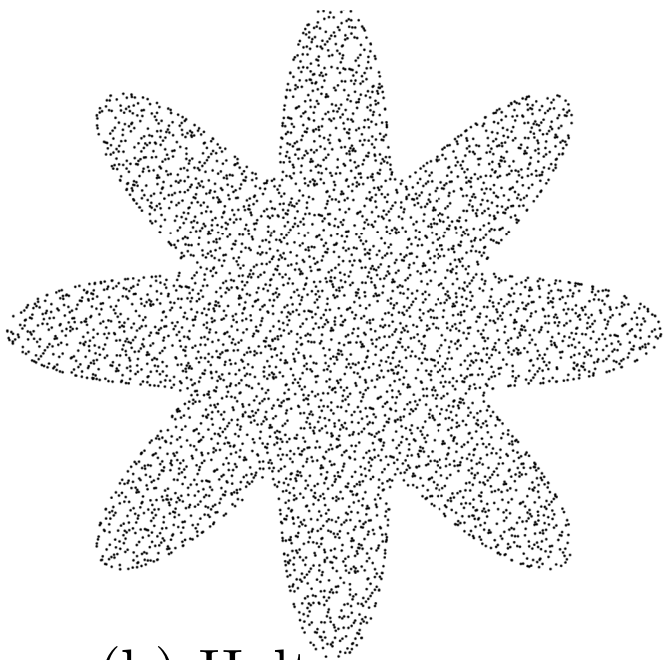
- Keep running the block-greedy algorithm until condition number is too large

Example

Quasi-random points $X \cup Y$ by different generators



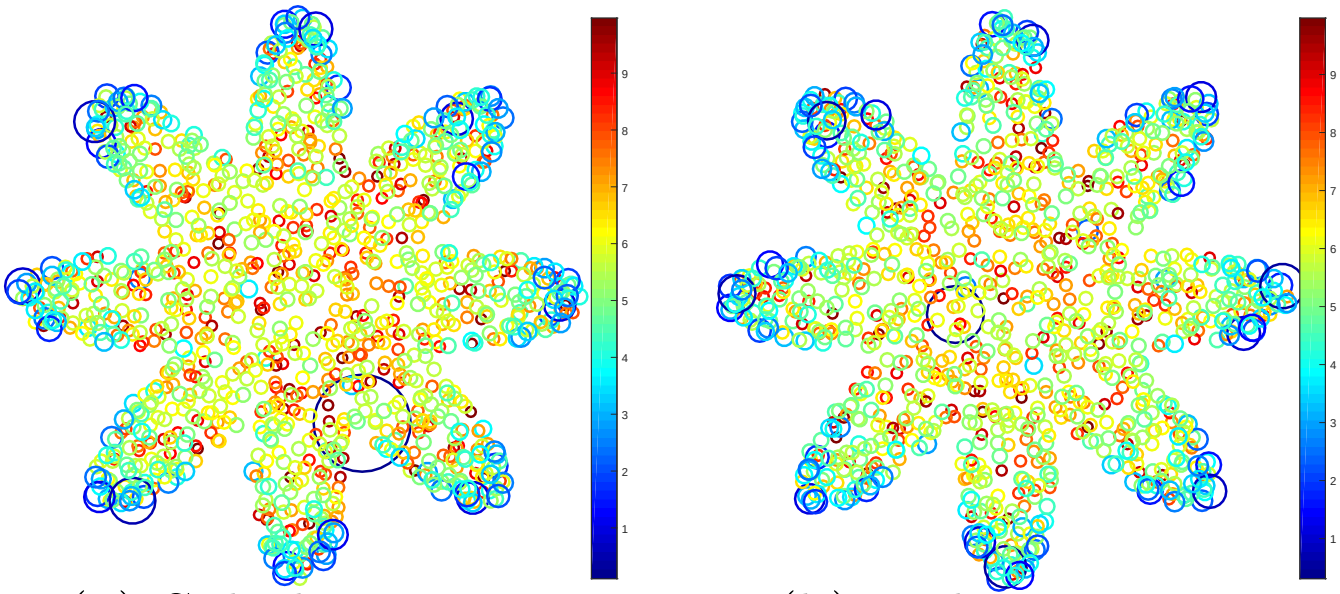
(a) Sobol sequence



(b) Halton sequence

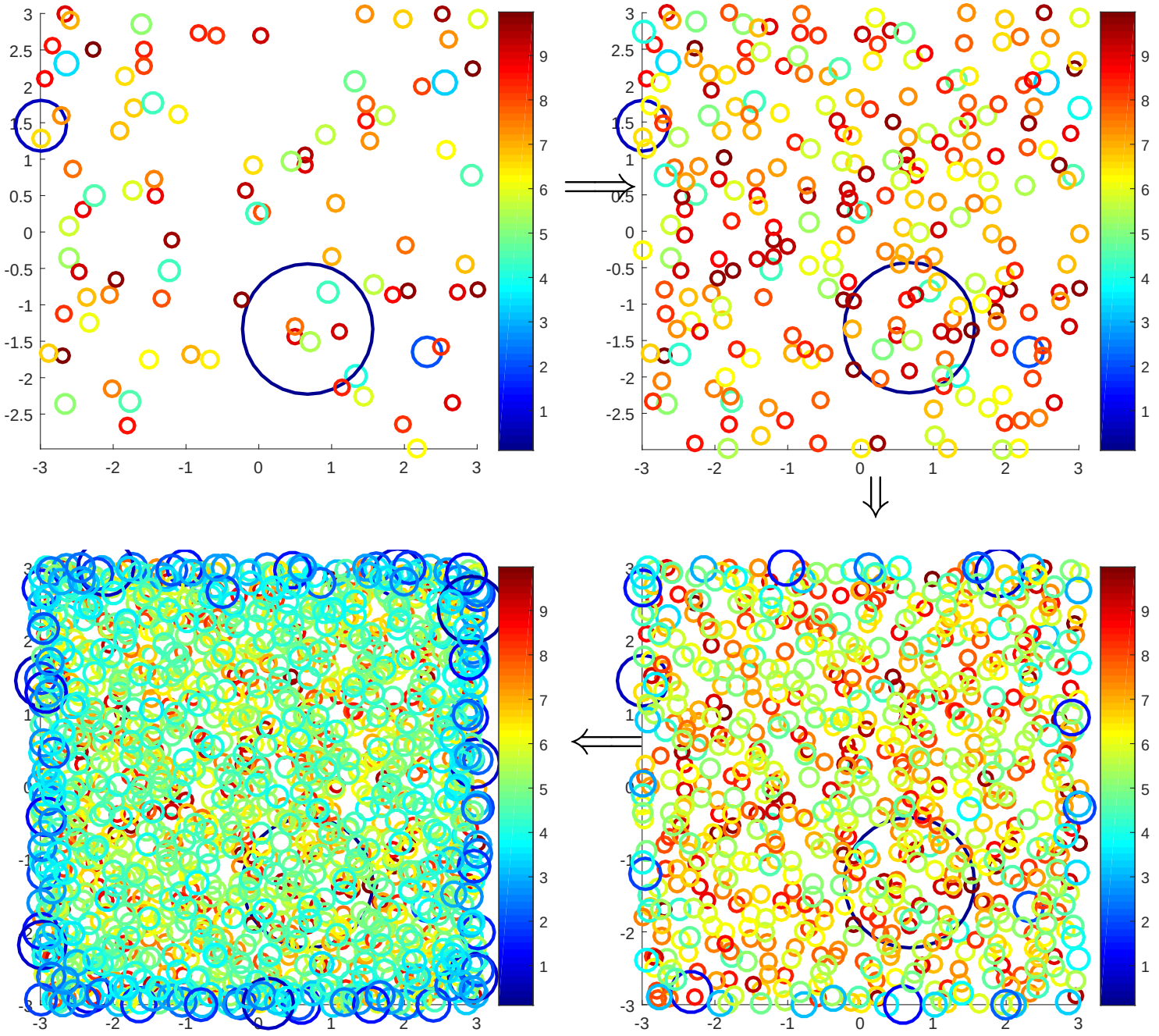
Example

System-free algorithm selected flat basis functions with shape parameters ≤ 10; peaky ones were omitted from this illustration



(a) Sobol sequence

(b) Halton sequence



peaks, 30 runs

GA $\epsilon_{\max}$	Bad Run	n_Z		$L^2(\Omega)$		$L^\infty(\Omega)$	
		Min	Max	Mean	S.D.	Mean	S.D.
25	0	635	739	2.6347e-07	5.1091e-07	8.5665e-06	1.7673e-05
50	0	887	1175	1.1854e-09	1.6752e-09	3.7677e-08	5.3122e-08
100	2	1021	1749	4.7907e-09	2.5103e-08	4.9567e-08	2.5110e-07

MQ $\epsilon_{\max}$	Bad Run	n_Z		$L^2(\Omega)$		$L^\infty(\Omega)$	
		Min	Max	Mean	S.D.	Mean	S.D.
25	0	1023	2013	1.8662e-07	9.4187e-07	1.8582e-06	9.2901e-06
50	2	2015	3909	1.7300e-08	5.6953e-08	1.7804e-07	4.5865e-07
100	1	2007	7343	5.6439e-06	1.8148e-05	6.9103e-05	2.1448e-04

MS $\epsilon_{\max}$	Bad Run	n_Z		$L^2(\Omega)$		$L^\infty(\Omega)$	
		Min	Max	Mean	S.D.	Mean	S.D.
25	0	401	429	7.2527e-05	7.4555e-05	1.7176e-03	2.4994e-03
50	0	673	745	6.5219e-07	1.0320e-06	1.6340e-05	3.1292e-05
100	0	923	1003	2.7574e-08	3.3238e-08	7.3817e-07	1.0683e-06

franke

GA $\epsilon_{\max}$	Bad Run	n_Z		$L^2(\Omega)$		$L^\infty(\Omega)$	
		Min	Max	Mean	S.D.	Mean	S.D.
25	0	655	733	9.6738e-04	1.3783e-03	2.8117e-02	4.0347e-02
50	0	823	1217	2.4058e-04	4.8471e-04	8.3423e-03	1.7349e-02
100	2	1015	1777	5.7304e-06	7.5535e-06	2.5603e-04	4.0785e-04

MS $\epsilon_{\max}$	Bad Run	n_Z		$L^2(\Omega)$		$L^\infty(\Omega)$	
		Min	Max	Mean	S.D.	Mean	S.D.
25	0	397	423	2.5519e-03	2.0556e-03	5.6153e-02	6.8083e-02
50	0	677	749	4.3585e-04	3.8856e-04	1.0987e-02	1.3409e-02
100	0	911	1227	1.5049e-04	4.6402e-04	4.8690e-03	1.6973e-02

$f(x,y) = \max(0,x)^3 - \max(0,y)^3$

GA $\epsilon_{\max}$	Bad Run	n_Z		$L^2(\Omega)$		$L^\infty(\Omega)$	
		Min	Max	Mean	S.D.	Mean	S.D.
25	0	623	757	1.4682e-05	2.5899e-05	3.7926e-04	7.6777e-04
50	0	869	1007	1.2687e-05	1.9572e-05	4.3372e-04	6.9755e-04
100	2	1023	1785	1.4470e-06	7.7554e-07	3.9665e-05	5.5451e-05

MS $\epsilon_{\max}$	Bad Run	n_Z		$L^2(\Omega)$		$L^\infty(\Omega)$	
		Min	Max	Mean	S.D.	Mean	S.D.
25	0	403	427	2.0743e-05	1.0798e-05	3.5904e-04	3.9292e-04
50	0	671	733	7.9690e-06	8.9910e-06	1.6165e-04	2.9417e-04
100	0	927	1009	4.5236e-06	4.4479e-06	1.0258e-04	1.5014e-04

The way to run is

- Give users a quick approximation
- Based on the PDE residual, readers can decide if refinement is needed

$\max n_Z$	256	512	1024	Terminate	
GA	2.1239e+00	7.2920e-01	1.1103e-03	3.8550e-09	($n_Z = 1893$)
MS	2.6413e+00	6.9293e-03	2.3652e-07	6.5507e-08	($n_Z = 1611$)

Thank You