Greedy Adaptive Local Recovery of Functions in Sobolev Spaces

Robert Schaback

Institut für Numerische und Angewandte Mathematik,
Georg-August-Universität Göttingen, Lotzestraße 16–18,
schaback@math.uni-goettingen.de

Version of July 29, 2024

Abstract
There are many ways to upsample functions from multivariate scattered data locally, using only a few neighbouring data points of the evaluation point. The position and number of the actually used data points is not trivial, and many cases like Moving Least Squares require point selections that guarantee local recovery of polynomials up to a specified order. This paper suggests a kernel-based greedy local algorithm for point selection that has no such constraints. It realizes the optimal L_∞ convergence rates in Sobolev spaces using the minimal number of points necessary for that purpose. On the downside, it does not care for smoothness, relying on fast L_∞ convergence to a smooth function. The algorithm ignores near-duplicate points automatically and works for quite irregularly distributed point sets by proper selection of points. Its computational complexity is constant for each evaluation point, being dependent only on the Sobolev space parameters. Various numerical examples are provided. As a byproduct, it turns out that the well-known instability of global kernel-based interpolation in the standard basis of kernel translates arises already locally, independent of global kernel matrices and small separation distances.

Keywords:
Interpolation, Approximation, Kernel, RBF, Algorithm, Stability, Optimality, Greedy

MSC Classification:
65D12, 65D05, 41A05, 65D25, 65D40

1 Introduction

Throughout, we shall work on a bounded domain $\Omega \subset \mathbb{R}^d$ and consider recoveries of functions f on Ω using values $f(x_i)$ on scattered locations $x_i \in X_N = \{x_1, \ldots, x_N\} \subset \Omega \subset \mathbb{R}^d$. We focus on large point sets X_N, but avoid working with the whole set. Instead, we calculate the recovery at each point $z \in \Omega$ separately, using only the points from a set $X(z) \subset X_N$, and with a fixed computational complexity for each z. This strategy is well-known from Moving Least Squares [21, 13, 4, 25, 39, 1, 15, 26], Shepard-type [35, 19, 11] and Partition-of-Unity [2, 40, 20, 24, 6] techniques, but here we ignore any tricks to ensure smoothness properties. We focus on optimal convergence

rates and minimal point sets instead, and confine ourselves to recoveries in spaces $\mathscr{H}$ like Sobolev spaces, using translates of the kernel K that is reproducing in $\mathscr{H}$.

It is well-known that using the full set X_N gives the least possible error, but how much is lost when using only the points in $X(z)$ for recovery of $f(z)$? Clearly, one loses smoothness and has a larger error while gaining much better computational complexity. This tradeoff is the basic question here. The proposed method shows optimal convergence rates in L_∞ at $\mathscr{O}(1)$ computational complexity at each evaluation point, using the minimal possible number of points for such rates.

Our point selection strategy will not be based on geometry properties like guaranteeing that sets are in general position with respect to polynomials. This is a criterion that is unstable under variation of data points. Instead, we select points by minimizing a continuous function connected to the error, aiming at error-minimal selections like in [8].

The underlying theory is kernel-based interpolation, as covered by the books [5, 41, 14]. Section 2 provides the basic notations, including the *Power Function*, i.e. the norm of the error functional in the kernel-based space considered. As a prerequisite for our algorithm, Section 3 deals with the very useful *Newton basis*. The algorithm follows in Section 4, as a stepwise adaptive greedy minimization of the Power Function in terms of the Newton basis.

Then there are various numerical examples. Sections 5 and 6 focus on the error locally and globally, while Section 7 recovers functions and shows the grades of discontinuity of the recoveries. Then Section 8 demonstrates that the optimal convergence rates are actually attained. The unexpected instabilities for large smoothness parameters m are explained in Section 9, followed by conclusions and open problems in Section 10.

2 Recovery

For a very large set $X_N = \{x_1, \dots, x_N\}$ and a point $z \notin X_N$ we pick a set of n points near z, renumbered as $X_n = \{x_1, \dots, x_n\}$. Recovery of $f(z)$ from values at points of $X_n = \{x_1, \dots, x_n\}$ can be written as

$$s_{f,X_n}(z) = \sum_{j=1}^{n} u_j(z) f(x_j),\ f \in \mathscr{H}, z \in \Omega$$

where the values $u_j(z)$ are just real numbers, because we do not vary z. Lagrange conditions $u_j(x_k) = \delta_{jk},\ 1 \le j,k \le n$ are not required. The error functional is

$$\varepsilon_{X_n,z} \,:\, f \mapsto f(z) - s_{f,X_n}(z),$$

written in terms of point evaluation functionals as

$$\varepsilon_{X_n,z} = \delta_z - \sum_{j=1}^{n} u_j(z) \delta_{x_j}.$$

Its norm in $\mathscr{H}$ is known as the *Power Function* $P_{X_n}(z)$. By the standard dual representation

$$K(x,y) = (\delta_x, \delta_y)_{\mathscr{H}^*} \text{ for all } x,y \in \Omega,$$

the squared norm is

$$\|\varepsilon_{X_n,z}\|^2_{\mathscr{H}^*} = P^2_{X_n}(z) = K(z,z) - 2\sum_{j=1}^{n} u_j(z)K(z,x_j) + \sum_{j,k=1}^{n} u_j(z)u_k(z)K(x_k,x_j).$$

This makes the Power Function computable. Due to the alternative definition

$$P_{X_n}(z) = \sup\{f(z) \,:\, \|f\|_{\mathscr{H}} \le 1,\ f(X_n) = \{0\}\},$$

the Power Function decreases at all z when the point set X_n is enlarged. Details are in basic texts on kernels, e.g. [32, 5, 41, 14].

Using the Power Function, the error of the recovery of $f(z)$ by $s_{f,X_n}(z)$ at a point z has the optimal bound

$$|f(z) - s_{f,X_n}(z)| \le P_{X_n}(z)\|f\|_{\mathscr{H}} \text{ for all } f \in \mathscr{H} \tag{1}$$

in $\mathscr{H}$, but we want to get away with fewer points forming a subset $X(z) \subset X_n$. Therefore, the goal is to find a subset $X(z)$ of $X_n \subset X_N$ such that the difference between $P_{X(z)}(z)$ and its lower bound $P_{X_n}(z)$ is small. We tacitly assume that N is too large to let the even lower bound $P_{X_N}(z)$ be computable.

Note that the error bound (1) is *local* or *pointwise*, and we shall use it after selection as

$$|f(z) - s_{f,X(z)}(z)| \le P_{X(z)}(z)\|f\|_{\mathscr{H}} \text{ for all } f \in \mathscr{H}. \tag{2}$$

We shall always know the value of $P_{X(z)}(z)$, and therefore we have an error bound that extends to an L_∞ error bound when varying z, because $P^2_{X(z)}(z) \le P^2_{\emptyset}(z) = K(z,z)$. This allows full control of the L_∞ error up to the unknown factor $\|f\|_{\mathscr{H}}$. It is an old unsolved problem to provide upper bounds on it.

We propose a greedy method to select the set $X(z) \subset X_n$ for fixed z by stepwise minimization of the Power Function as a function of $x_1, x_2, \ldots, x_n$. This is similar to, but different from the P-greedy point selection in [10]. There, x_{n+1} is picked as the argmax of $P^2_{X_n}(z)$ as a function of z. The paper [30] showed that the P-greedy point selection method is asymptotically optimal with respect to convergence rates.

Here, we proceed differently. The point z is fixed, and we select $x_1, x_2, \ldots$ sequentially to produce smallest possible values of $P^2_{\{x_1,\ldots,x_j\}}(z)$ for increasing j until the error is small enough. We can stop this at some bound on j or by a lower threshold on $P^2_{\{x_1,\ldots,x_j\}}(z)$. At any time, we have the error bound (2) where $P_{X(z)}(z)$ is explicitly known.

Of course, the final recovery $s_{f,X(z)}(z)$ will be a discontinuous function of z. But if $P_{X(z)}(z)$ is small and controllable, $s_{f,X(z)}(z)$is close to f in the L_∞ norm, and its deviation from f in L_∞ may be as small as $\|f - s_{f,X_N}\|_\infty$ if we can manage to let $P_{X(z)}(z)$ for $X(z) \subset X_N$ behave like $\|f - s_{f,X_N}\|_\infty$.

Here is an illustration for Sobolev spaces $W_2^m(\mathbb{R}^d)$ with $m > d/2$. If sets X_N have *fill distances*

$$h(X_N, \Omega) = \sup_{z \in \Omega} \inf_{x_j \in X_N} \|z - x_j\|_2$$

and *separation distances*

$$\sigma(X_N) = \frac{1}{2} \inf_{x_j \neq x_k \in X_N} \|x_k - x_j\|_2$$

with *asymptotic regularity*

$$0 < c_0 \sigma(X_N) \leq h(X_N, \Omega) \leq C_0 \sigma(X_N),$$

then by e.g. [41]

$$\|f - s_{f,X_N}\|_\infty \leq \|P_{X_n}\|_\infty \|f\|_{W_2^m(\mathbb{R}^d)} \leq C_1 h^{m-d/2}(X_N, \Omega) \|f\|_{W_2^m(\mathbb{R}^d)}.$$

Therefore our goal must be to ensure

$$P_{X(z)}(z) \leq C_2 h^{m-d/2}$$

for $X(z) \subset X_N$ and all $z \in \Omega$. This is possible for small $X(z) \neq X_N$, and the examples in later sections will show how. We shall not need asymptotic regularity for that, and we can get away with the minimal number of selected points for getting L_∞ convergence like $\mathscr{O}(h^{m-d/2})$.

The Power Function value $P_{X_n}(z)$ can be seen as a distance from z to the set X_n, and therefore the above algorithm is a greedy method to find nearest neighbours in the kernel metric. More explicitly, $P_{X_n}(z)$ is the Euclidean distance of the point evaluation functional δ_z to the space spanned by the functionals δ_{x_j}, $1 \leq j \leq n$ in the dual $\mathscr{H}^*$ of the Hilbert space $\mathscr{H}$. The problem then is to find a few functionals that already give a small distance to the space spanned by them. This, in turn, is a case of *n-term approximation* [37, 38] that has good solutions by greedy methods and a connection to sparsity techniques [7]. The Newton basis approach below will be a special implementation adapted to kernel-based spaces.

3 Newton Basis

To study the variation of $P^2_{\{x_1,\ldots,x_j\}}(z)$ as a function of x_j, we use the Newton basis representation dating back to [28], written here via recursive kernels [27]. The kernel recursion for points $x_1, x_2, \ldots$ is

$$\begin{aligned} K_1(x,y) &:= K(x,y) \\ K_{j+1}(x,y) &:= K_j(x,y) - \frac{K_j(x,x_j)K_j(x_j,y)}{K_j(x_j,x_j)}, \; j \geq 1, \; x, y \in \Omega. \end{aligned}$$

It is easy to show by induction that

$$K_{j+1}(x_k,y) = 0 = K_{j+1}(x,x_k)$$

holds for all x, y and all $1 \le k \le j$. The Newton basis function N_j then is

$$N_j(x) = \frac{K_j(x,x_j)}{\sqrt{K_j(x_j,x_j)}},\ j \ge 1,$$

satisfying

$$N_j(x_k) = 0,\ 1 \le k < j \text{ and } N_j(x_j) = \sqrt{K_j(x_j,x_j)},\ j \ge 1.$$

Now the kernel recursion takes the form

$$K_{j+1}(x,y) = K_j(x,y) - N_j(x)N_j(y) = K(x,y) - \sum_{m=1}^{j} N_m(x)N_m(y),\ j \ge 1$$

cancelling the denominators, with the final residual

$$K_{j+1}(x,y) = K(x,y) - \sum_{m=1}^{j} N_m(x)N_m(y),\ x,y \in \Omega,\ j \ge 0.$$

Inserting points for x and y, we see that

$$K(x_i,x_j) = \sum_{m=1}^{\min(i,j)} N_m(x_i)N_m(x_j)\ 1 \le i,j \le n$$

is a Cholesky factorization of the positive definite symmetric kernel matrix. It is in the background, but we prefer to work in terms of functions, not matrices. The Newton basis recursion is

$$\begin{aligned} N_j(x_j)^2 &= K_j(x_j,x_j), \\ N_j(x)N_j(x_j) &= K(x,x_j) - \sum_{m=1}^{j-1} N_m(x)N_m(x_j),\ j \ge 1. \end{aligned} \tag{3}$$

4 Greedy Point Selection Algorithm

We want an optimal point selection for recovery of $f(z)$ from values $f(x_j),\ 1 \le j \le n$ by minimizing the Power Function. The first chosen point x_1 should minimize

$$P_{x_1}^2(z) = K(z,z) - N_1^2(z) = K(z,z) - \frac{K(z,x_1)^2}{K(x_1,x_1)},$$

as a function of x_1, via

$$x_1 = \arg\max_{x_j \in X_n} \frac{K(z,x_j)^2}{K(x_j,x_j)}.$$

By positive semidefiniteness of kernel matrices,

$$K(z,x_j)^2 \le K(z,z)K(x_j,x_j)$$

holds and leads to the choice of x_j if $z = x_j$. Then the process can be stopped.

If $x_1,\dots,x_{j-1}$ are determined and x_j is still not selected, the Newton basis functions N_k are determined for $k < j$, and we can assume vectors and matrices

$$\begin{array}{rcl} \mathbf{z} & := & (K_j(z,x_k),\, 1 \le k \le n) \in \mathbb{R}^n, \\ \mathbf{d} & := & (K_j(x_k,x_k),\, 1 \le k \le n)) \in \mathbb{R}^n, \\ \mathbf{N} & := & (N_i(x_k),\, 1 \le i < j,\, 1 \le k \le n). \end{array}$$

The Power Function at z for varying x_j is

$$P^2_{x_1,\dots,x_j}(z) = K(z,z) - \sum_{k=1}^{j-1} N_k(z)^2 - N_j(z)^2 = K(z,z) - \sum_{k=1}^{j-1} N_k(z)^2 - \frac{K_j(z,x_j)^2}{K_j(x_j,x_j)}$$

and the next point should be

$$x_j = \arg\max_{x_k \in X_n, k \ge j} \frac{K_j(z,x_k)^2}{K_j(x_k,x_k)} = \arg\max_{j \le k \le n} \frac{\mathbf{z}_k^2}{\mathbf{d}_k}. \tag{4}$$

The maximum value is $N_j^2(z)$ and we can construct N_j by the recursion (3) to get a new column in $\mathbf{N}$. For upgrading our vectors, we use

$$\begin{array}{rcl} K_{j+1}(z,x_k) & = & K_j(z,x_k) - N_j(z)N_j(x_k) \\ K_{j+1}(x_k,x_k) & = & K_j(x_k,x_k) - N_j^2(x_k) \\ P^2_{j+1}(z) & = & P^2_j(z) - N_j^2(z) \end{array}$$

in terms of the Newton basis. Here, we still need to fix the sign of $N_j(z)$ to match the sign of $K_j(z,x_j) = d_j$.

The Lagrange coefficients $L_j(z)$ must recover the $N_k(z)$ exactly. They solve the triangular system

$$N_m(z) = \sum_{k=1}^{j} L_k(z) N_m(x_k),\ 1 \le m \le j \tag{5}$$

that is cheaply solvable. Then one can monitor the Lebesgue constants

$$\sum_{k=1}^{j} |L_j(z)|$$

that control the L_∞ *evaluation stability* [29] of the solution via

$$|s_{f,X(z)}(z) - s_{g,X(z)}(z)| \le \sum_{k=1}^{j} |L_j(z)||f(x_j) - g(x_j)| \le \sum_{k=1}^{j} |L_j(z)|\|f-g\|_\infty.$$

If k steps are executed, storage grows like $\mathcal{O}(kn)$, while computational complexity is $\mathcal{O}(k^2 n)$. The next section will show that one can keep k much smaller than n in most cases. In the 2D examples below we get away with $n = 5k$, letting the complexity for a single recovery be $\mathcal{O}(5k^3)$. Matrix conditions arise here only for the triangular matrices with entries $N_m(x_k)$ used in (5). This will be relevant for the stability arguments of Section 9.

5 Single Point Examples

As described in Section 2, we consider very large sets $X_N \subset \mathbb{R}^d$. For each fixed point $z \in \mathbb{R}^d$ we have a subset X_n of X_N depending on z, taking possibly nearest neighbours of z, to start the local greedy point selection on X_n to end up with an even smaller set $X(z) \subseteq X_n$. Because everything is local, we can set z to be the origin and replace X_n by $X_n - \{z\}$.

Here, for illustration, we work with $n = 100$ random points in $[-1,+1]^2$ depicted in the top left of Figure 1, and used in later cases as well. These are unduly many, but the goal of this section is to compare selections of $k << n$ points with selection of all n points. An even larger set X_N may be in the background, on a larger domain. The greedy algorithm is run up to n points to show how the error behaves for small k when compared to $k = n$. We want to find k such that selecting k points is not much different from selecting n points.

Figure 1 shows results for the kernel generating Sobolev space $W_2^3(\mathbb{R}^2)$. The decay of the squared Power Function is plotted in the centre of the top row. Large n do not pay off, error-wise, because the curve flattens dramatically. To compare with polynomial exactness orders $q = 1,2,3,\ldots$ needing $Q = \binom{q-1+d}{d} = 1,3,6,10,\ldots$ points in $\mathbb{R}^2$, these cases are marked in red circles. The match between the optimal rate $m - d/2$ in Sobolev space $W_2^m(\mathbb{R}^d)$ with polynomial exactness of order q is additionally marked here and later with a red cross. Now $m = 3$ leads to order $q = 2$ and $Q = 6$ points. This is our goal for $m = 3$.

In the plots of this section, the third plot in the first row shows the Lebesgue constants, the cases for polynomial orders marked in red circles again. The blue circles are the Lebesgue constants for using polynomials of the appropriate orders at the selected points. It can be expected that selecting the Q best points is sufficient for getting the optimal convergence rate. Taking more points does not decrease the local error substantially, see the top centre plot.

The lower plots show the point selections for different numbers of points belonging to different orders q. The central evaluation point $z = (0,0)$ is marked with a red cross. Note that the selection does not take all nearest points.

The low-regularity case $m = 1.5$ is in Figure 2, while $m = 6$ is in Figure 3. The first case should not use more than 3 points. The last case is run at a scale of $c = 0.1$, taking $K(\|x-y\|_2/c)$, and using 100 regular data locations. Here, taking 21 points is enough. Polynomial Lebesgue constants run out of hand, because the selected points are not in general position wrt. polynomials. Due to regularity of the point distribution, the greedy technique has several choices at various steps, and therefore there are plenty of equivalent point selections. For scale $c = 1$, the kernel matrix on 100 points exceeds the condition limit of $1.e14$. But if one stops when the squared Power Function is below 1.e-8, one gets away with 9 points, see Figure 4, without condition problems.

If the point is in the corner $(-1,+1)$, Figure 5 shows the results for $m = 3$ at scale 1.0, like Figure 1. The achievable squared Power Function now is about 0.004, while the central case had about 0.001. Using more than about 10 points does not help, while roughly 4 were sufficient in the central situation.

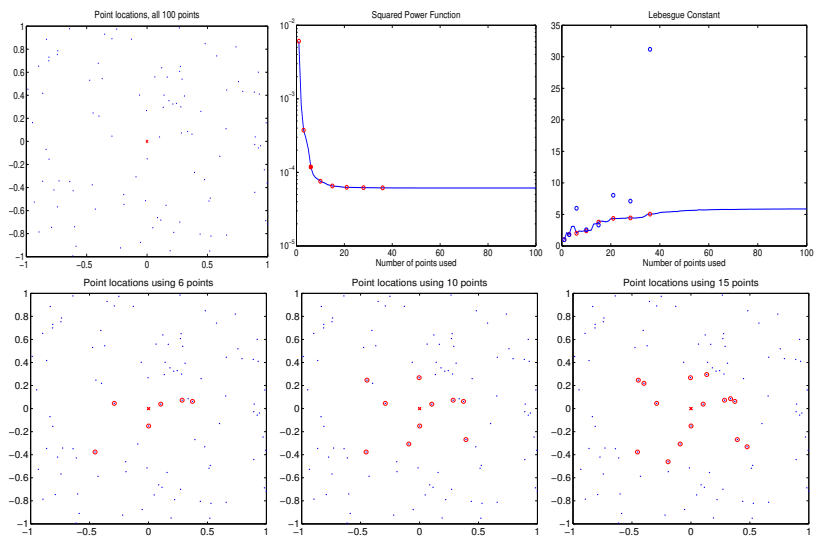

Figure 1: Local recovery in $W_2^3(\mathbb{R}^2)$ at the origin using the greedy point selection strategy.

If the Sobolev smoothness order m is fixed, users should aim at the optimal rate $h^{m-d/2}$ in terms of the fill distance h, see the end of Section 2. In the polynomial situation, one then needs reproduction of order $q = \lceil m - d/2 \rceil$ and at least $Q = \binom{q-1+d}{d}$ points [9]. But in order to allow greedy point selection it is recommended to offer more points. Since for corner points in $\mathbb{R}^2$ three quadrants are missing, one should take at least $2^d \cdot Q$ points. For $d = 2$, we shall use $n = 5 \cdot Q$ nearest neighbours in what follows, and all 2D examples work fine with this choice.

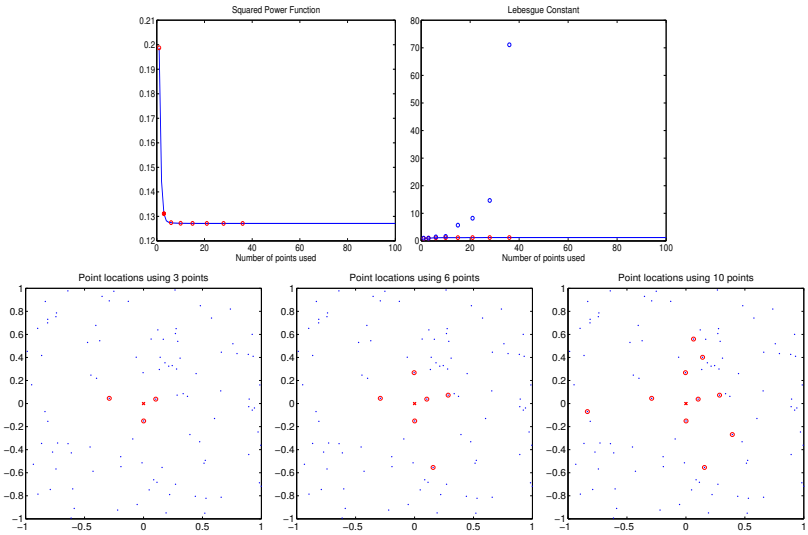

Figure 2: Local recovery in $W_2^{3/2}(\mathbb{R}^2)$ at the origin using the greedy point selection strategy. Three points are enough.

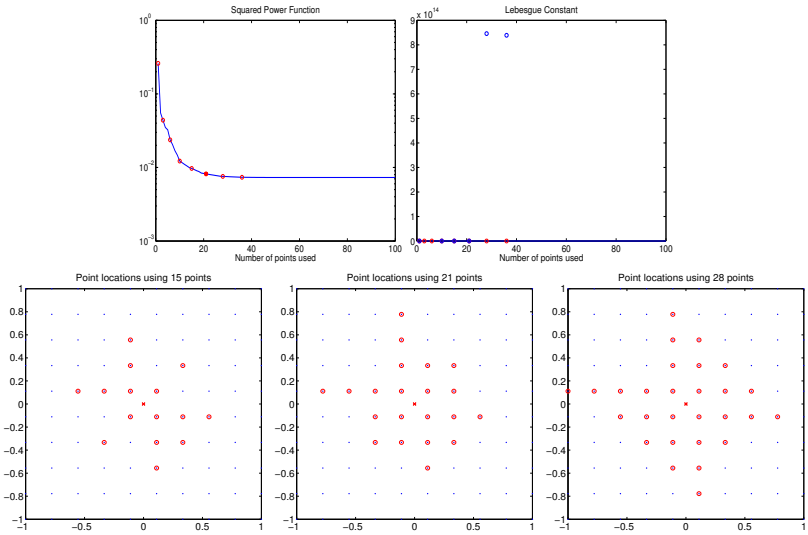

Figure 3: Local recovery in $W_2^6(\mathbb{R}^2)$ using the greedy point selection strategy at scale $c = 0.1$ on 100 regular points. One should use 21 points.

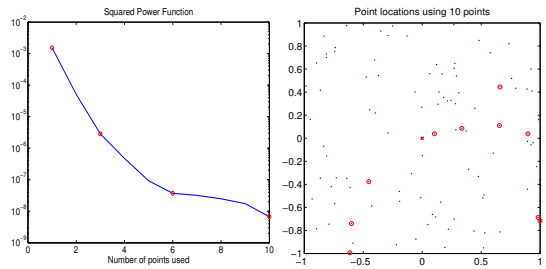

Figure 4: Local recovery in $W_2^6(\mathbb{R}^2)$ at the origin using the greedy point selection strategy at scale $c = 1$ up to $P^2 < 1.e-8$ on irregular points. One should use 10 points only.

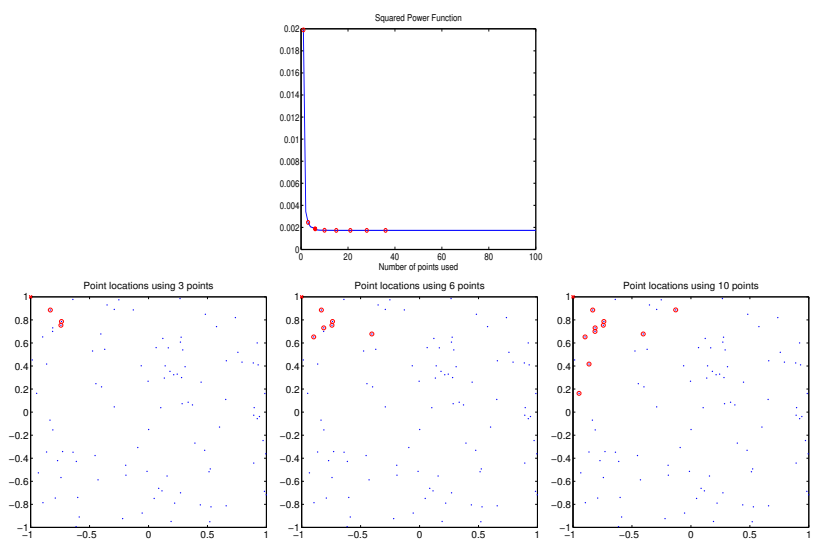

Figure 5: Local recovery in $W_2^3(\mathbb{R}^2)$ at the top left corner using the greedy point selection strategy.

6 Global Examples

We use the results of the previous section and the same scattered point set of $n = 100$ points offered for selection. But now we run the local greedy algorithm on each point of a $51 \times 51 = 2601$ evaluation grid to simulate an upsampling from rather irregular input data locations to a regular grid. Readers should see these examples as zoom-ins for much larger cases. We stick to a 51×51 grid, because we want reasonable plots and a comparison with the global interpolation on n points. The sequence of examples matches the sequence of the previous section. In all cases, we take $5 \cdot Q$ points for $q = \lceil m - d/2 \rceil$ and $Q = \binom{q-1+d}{d}$ when we work with the Matérn kernel generating $W_2^m(\mathbb{R}^d)$. This aims at optimal convergence rates $h^{m-d/2}$ in Sobolev spaces.

The case $m = 3$ requires $q = 2$ and $Q = 6$ for getting the expected rate of h^2. We offer the $5 \cdot Q = 30$ nearest points for each evaluation point z, and let the algorithm select the 6 best ones to form the set $X(z)$ of Section 2, following Figure 1. The results are in Figure 6, plotted over all 2601 evaluation points. The top left plot shows the squared Power Function when the full interpolation problem is solved on all $n = 100$ points. The local case is top right, working on 6 selected points, and the values must be larger than the first plot everywhere. The lower left plot shows the difference. It is only large near the boundary, where the global case has 0.007 and the local case has 0.009. Also, the Lebesgue constants only blow up near the corners. Note that we allowed 30 points for a selection of 6. Offering more points does not help.

The case $m = 1.5$ is selecting $Q = 3$ points out of $5Q = 15$ and produces Figure 7. Working locally does not lose more than about 10%. The case $m = 6$ at scale 0.1 on 100 regular data points is in Figure 8. This runs as expected.

Then we present results for scale 1.0 that leads to a condition estimate of $7.9 \cdot 10^{18}$ for the full problem. But we stop the greedy algorithm at a tolerance of 10^{-8} for the squared Power Function. Here, the global Power Function values may be polluted due to the bad condition of the full kernel matrix. The corresponding plots are omitted. But the right-hand plot shows that the algorithm often gets away with less than 21 points to reach the tolerance. We suggested 10 points for Figure 4, but there we worked at a single point in the interior.

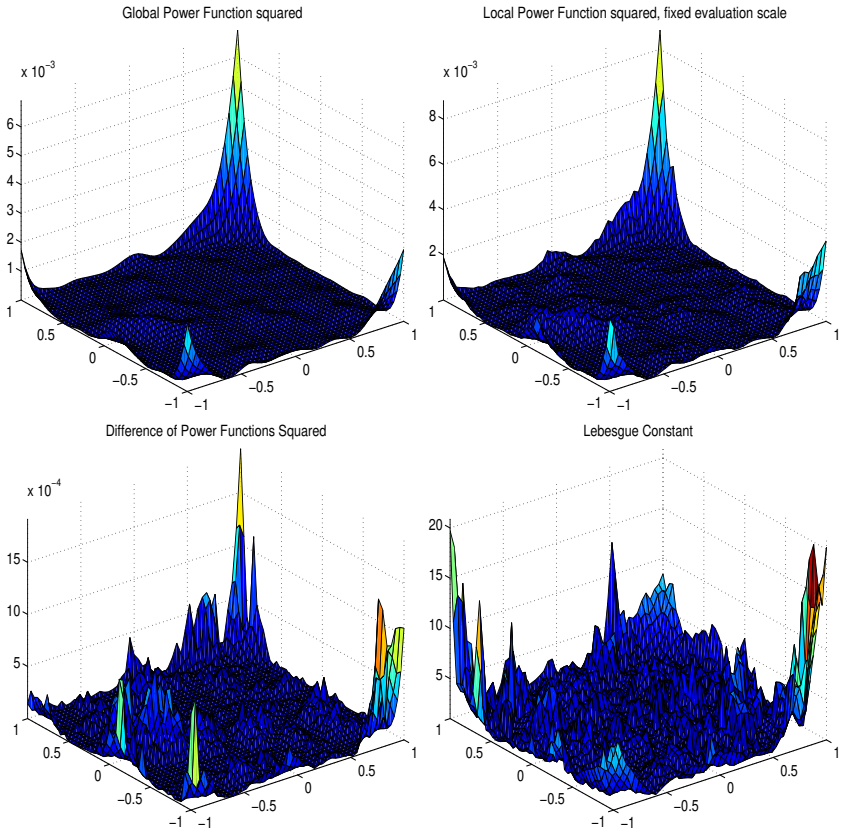

Figure 6: Local recovery in $W_2^3(\mathbb{R}^2)$ on 2601 points in $[-1,+1]^2$ using the greedy point selection strategy, with selection of 6 points out of 30.

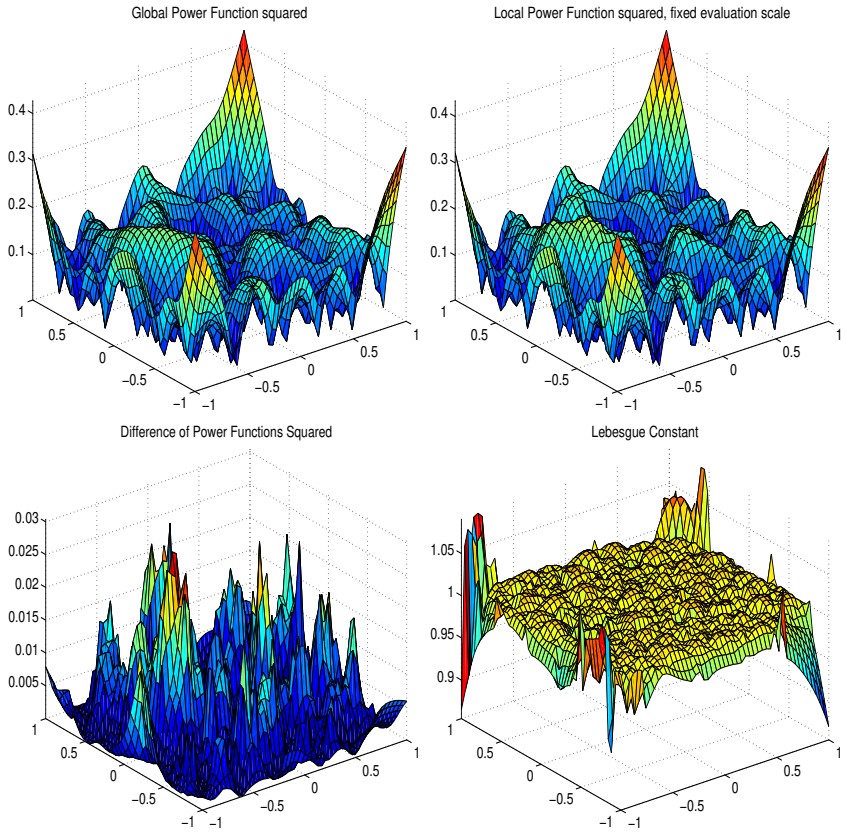

Figure 7: Local recovery in $W_2^{3/2}(\mathbb{R}^2)$ on 2601 points in $[-1,+1]^2$ using the greedy point selection strategy, with selection of 3 points out of 15.

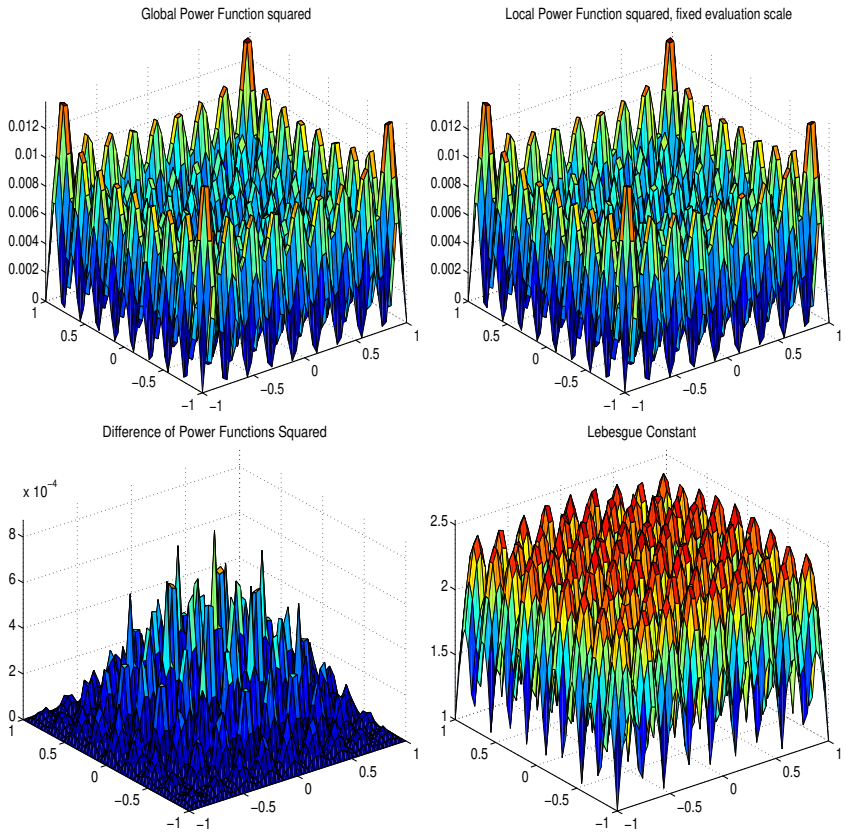

Figure 8: Local recovery in $W_2^6(\mathbb{R}^2)$ on 2601 regular points in $[-1,+1]^2$ at scale 0.1 using the greedy point selection strategy on 100 regular data points, with selection of 21 points out of 100.

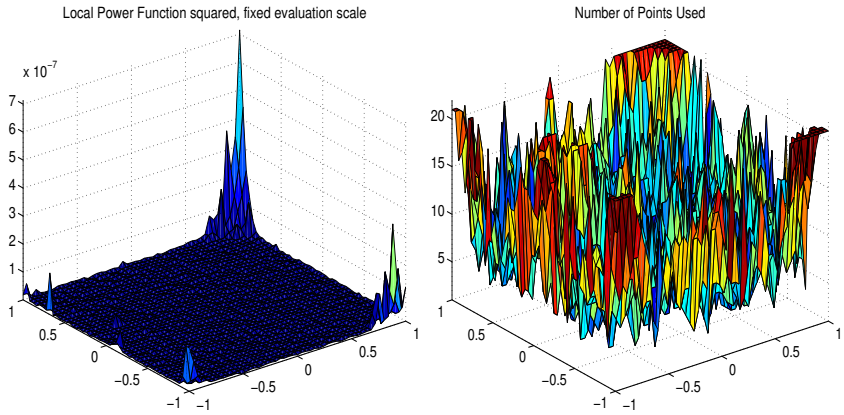

Figure 9: Local recovery in $W_2^6(\mathbb{R}^2)$ on 2601 points in $[-1,+1]^2$ using the greedy point selection strategy, with selection of 21 points out of 100, and with a lower bound of 10^{-8} for the squared Power Function. .

7 Function Reproduction

We now repeat the above examples, but focus on recovery of a smooth function, here the `peaks` function of MATLAB. The goal is to show the discontinuities induced by working locally.

Again, we start with $m = 3$, and let Figure 10 show results for the situation of Figure 6. Note that the local version selects only 6 points out of the 30 nearest neighbours. Similarly, Figure 10 presents the case of $m = 3/2$, selecting 3 out of 15 points.

To see the discontinuities in a close-up, we keep the 100 irregular data points in $[-1,+1]^2$, but now we place 2601 regular evaluation points into the interval $[0,0.4]^2$ containing only 13 irregular data points. Note that we can re-evaluate the global interpolant on the 2601 local points to have a zoom effect, but the local interpolant has to be recalculated, and it gets different and more exact. Figure 14 shows the case $m = 3$ again, to be compared to Figures 6 and 10. The top left plot shows the data points (red circles) and the evaluation points (blue dots), filling $[0,0.4]$ completely. The squared Power Functions differ by a factor of about 2, but are now around 10^{-5} instead of 10^{-2} in Figure 6. The second row and the last plot show that the error of the function recovery is about 0.1, while it is around 0.5 in Figure 10. The Lebesgue constants get better, see the lower left plot.

Figure 15 is for comparison to Figures 9 and 13. Since the iteration stops at the 10^{-8} threshold for the squared Power Function, the top right plot is chaotic, while the centre plot shows that using more points would go down to 10^{-11}. The actual ploints used are in the final plot, staying well below 21 proposed for $m = 6$. Due to the threshold,

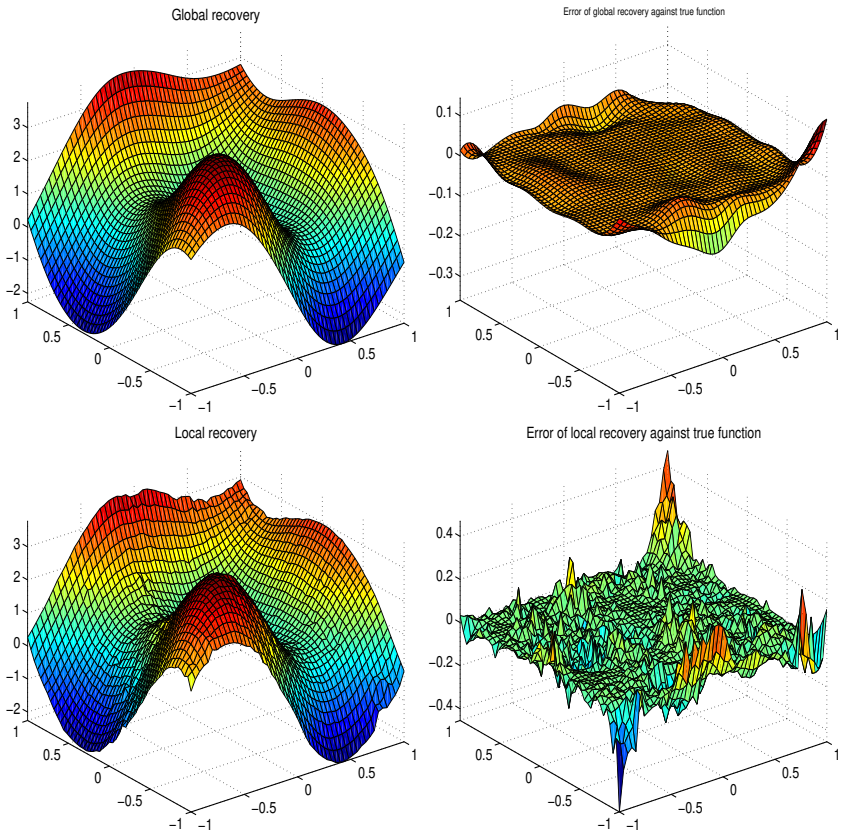

Figure 10: Local recovery of the `peaks` function in $W_2^3(\mathbb{R}^2)$, on 2601 points in $[-1,+1]^2$ using the greedy point selection strategy, with selection of 6 points out of 30.

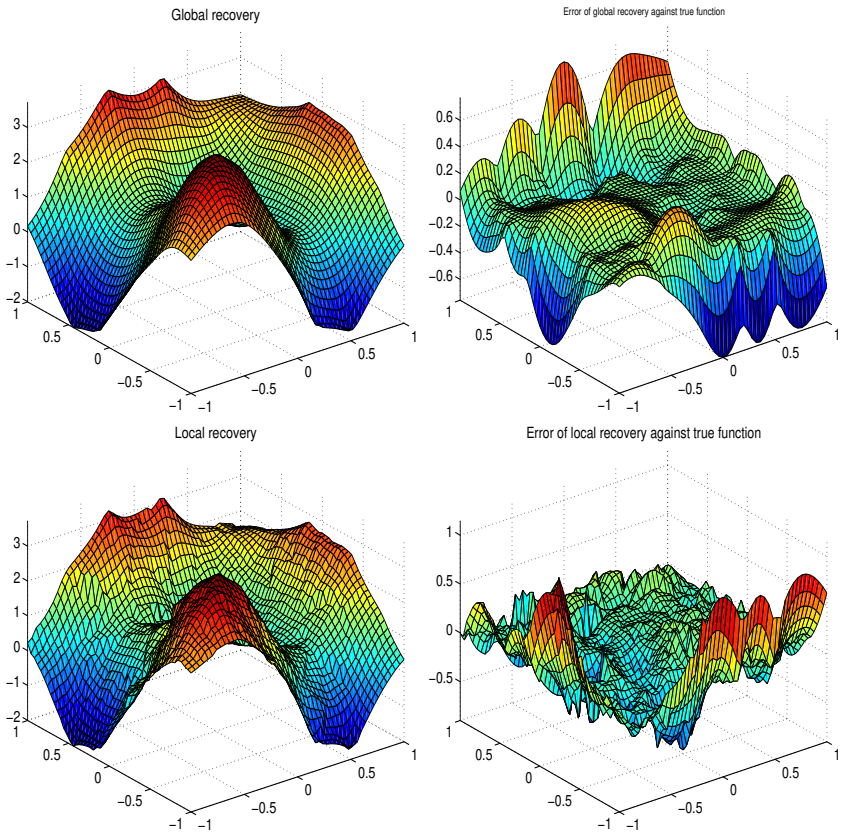

Figure 11: Local recovery of the `peaks` function in $W_2^{3/2}(\mathbb{R}^2)$, on 2601 points in $[-1,+1]^2$ using the greedy point selection strategy, with selection of 3 points out of 15.

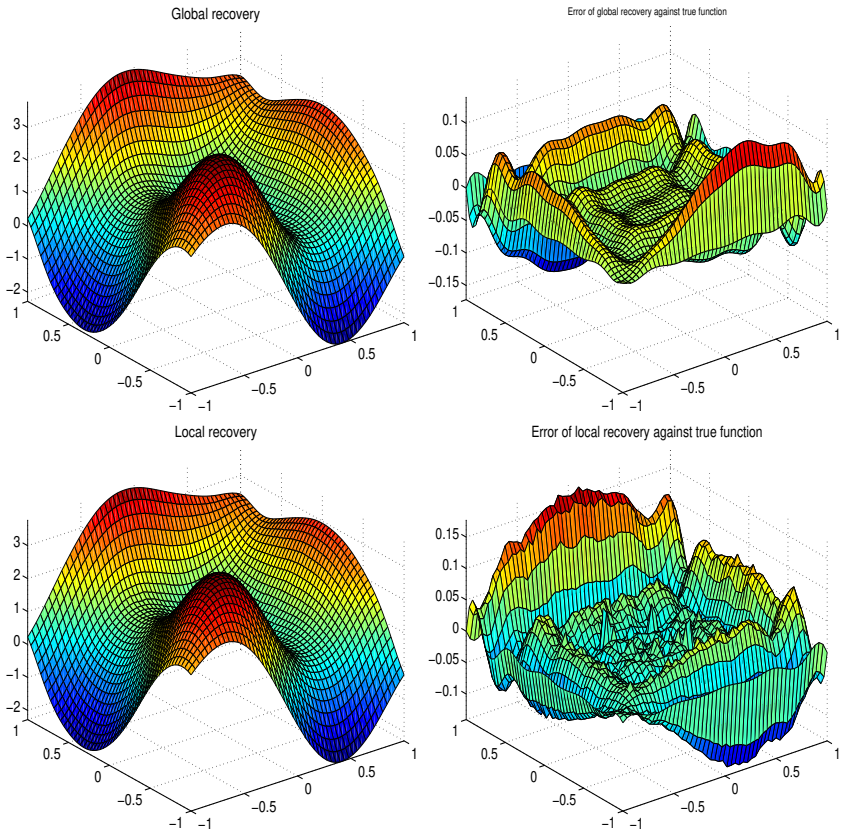

Figure 12: Local recovery of the `peaks` function in $W_2^6(\mathbb{R}^2)$ on 2601 regular points in $[-1,+1]^2$ at scale 0.1 using the greedy point selection strategy on 100 regular data points, with selection of 21 points out of 100.

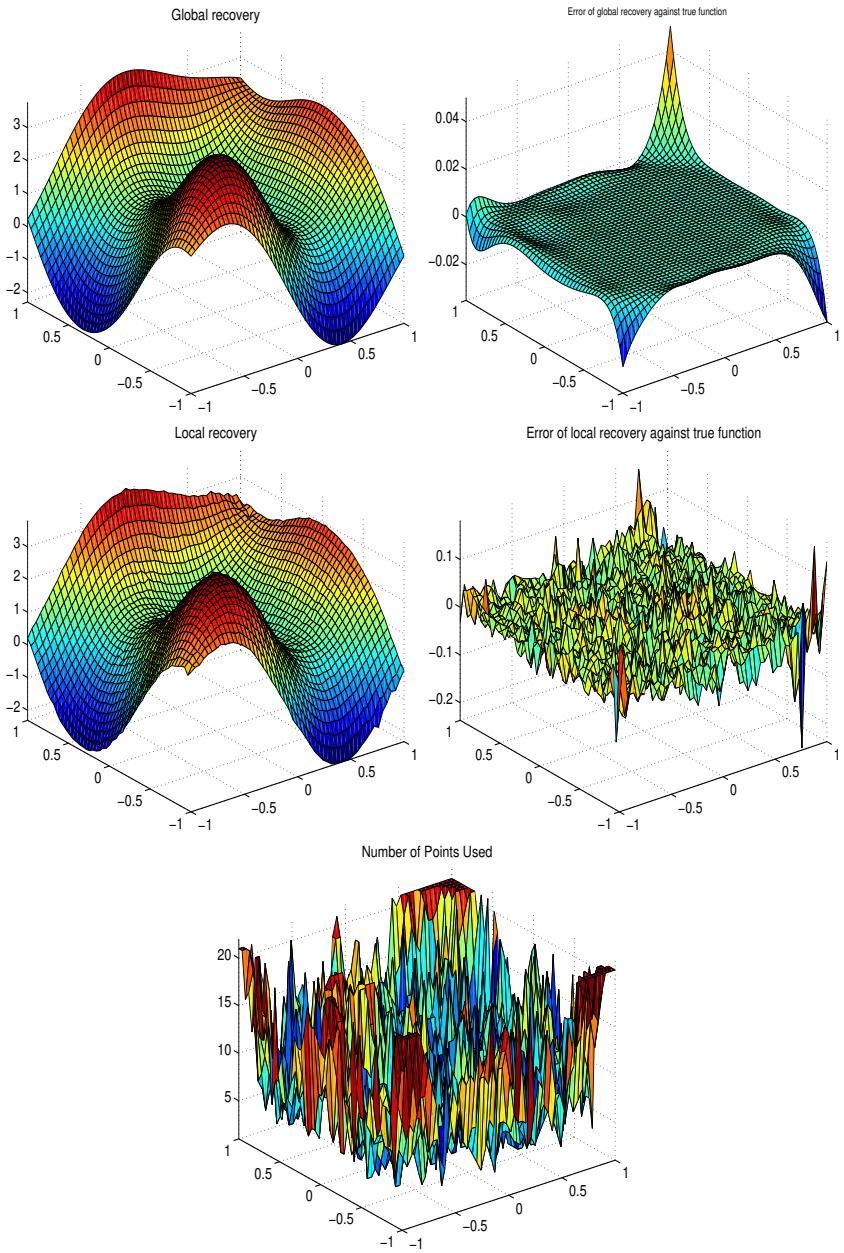

Figure 13: Local recovery of the `peaks` function in $W_2^6(\mathbb{R}^2)$ on 2601 regular points in $[-1,+1]^2$ at scale 1.0 using the greedy point selection strategy on 100 irregular data points, with a threshold of 10^{-8} on the squared Power Function. The final plot shows the number of points used locally.

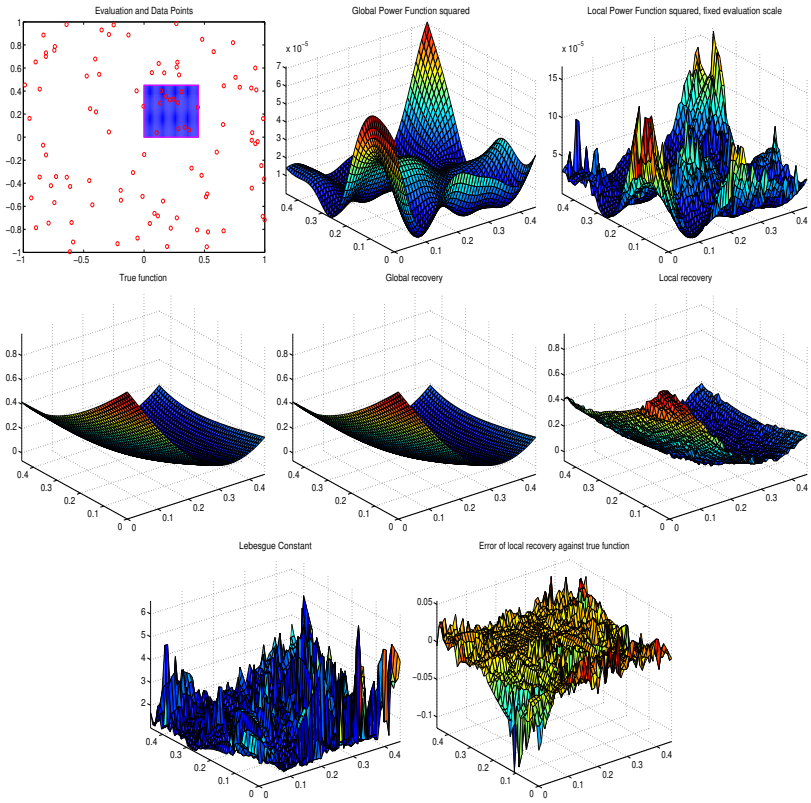

Figure 14: Local recovery of the `peaks` function in $W_2^3(\mathbb{R}^2)$ on 2601 regular points in $[0,0.4]^2$ at scale 1.0 using the greedy point selection strategy on 100 irregular data points in $[-1,+1]^2$.

the noise in the right centre plot is roughly constant everywhere, but still better than in Figure 13.

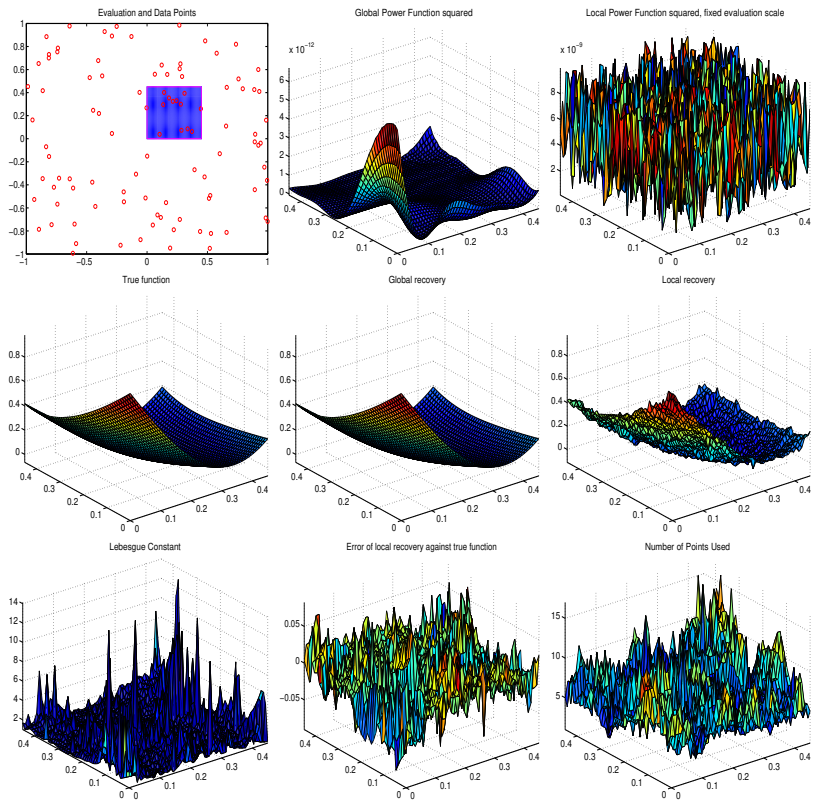

Figure 15: Local recovery of the `peaks` function in $W_2^6(\mathbb{R}^2)$ on 2601 regular points in $[0,0.4]^2$ at scale 1.0 using the greedy point selection strategy on 100 irregular data points in $[-1,+1]^2$. The squared Power function tolerance is 10^{-8}.

8 Convergence Rates

The previous sections tried to achieve $\mathcal{O}(h^{m-d/2})$ convergence in L_∞ in Sobolev spaces $W_2^m(\mathbb{R}^d)$ using well-selected sets of $Q = \binom{q-1+d}{d}$ points for $q = \lceil m - d/2 \rceil$. This number of points is necessary to let the set be in general position for polynomials in $\P_q^d$, and this is necessary for convergence like h^q. These rates hold pointwise by construction, and Figure 16 shows that they hold in the large.

A fixed evaluation grid of $441 = 21 \times 21$ was used in all cases. Since the worst situation should occur in corners or near boundaries, there are no serious changes when using

a finer evaluation grid. The greedy point selection method was run on each z of the evaluation grid, offering $5 \cdot Q$ data points of random sets X_N with N up to 10.000. To avoid additional randomness, the sets X_N were nested for increasing N. The plots show the maximum of the Power Function on the evaluation set. The expected rates are attained well, see the dotted lines marking the expected convergence rate. In all cases, the Power Function gets small enough for letting any function approximation be exact within plot precision.

9 Stability Issues

The case $m = 6$ is unexpectedly unstable already for reasonable h, needing a closer look. Inspecting the test runs, instabilities come up when the squared Power Function reaches machine precision, making the decision (4) unsafe. This occurs even when the local triangular $Q \times Q$ matrix of (5) is still within standard condition limits. The squared Power Function behaves like h^{2m-d} in L_∞ by standard convergence theory, and then fill distances of order

$$h < 10^{-\frac{15}{2m-d}} =: \overline{h}_{2m-d}$$

can be expected to fail in double precision. Of course, scaling and multipliers plays a major part here, but the ratios

$$\overline{h}_2 \,:\, \overline{h}_5 \,:\, \overline{h}_{10} \quad \text{like} \quad 0.0000000316 \,:\, 0.001 \,:\, 0.0316\,;$$

matching $m = 1.5, 3$, and 6 in $\mathbb{R}^2$ indicate that large m will fail unexpectedly early. Figure 16 goes down to 10.000 points with $h = 0.02$, illustrating the above argument. It can be expected that the case $m = 3/2$ runs up to using approximately 60 million points.

Note that this applies also to global interpolation, making small h for large m hazardous, if not impossible. But the condition of the global kernel matrix does not enter here, in contrast to the standard stability theory of kernel-based interpolation. The well-known phenomena like rank loss or bad condition are local, not global. Like in [31], they are connected to how well a kernel can be approximated by polynomials.

To circumvent the stability problems, the additional green line shows the results when the threshold for the Power Function is set to 10^{-6}.

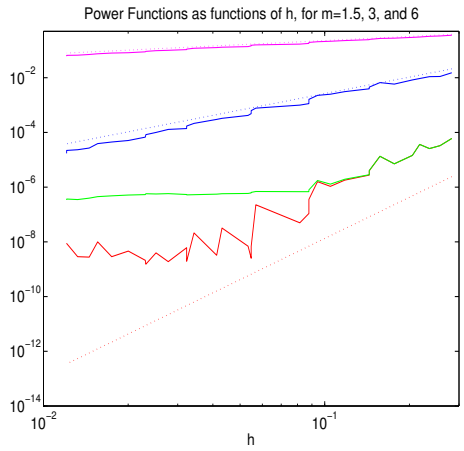

Figure 16: Power Functions as functions of h for $m = 1.5$ (magenta), 3 (blue), and 6 (red) based on minimal point sets. The dotted lines indicate the expected theoretical rates $h^{m-d/2}$.

10 Conclusions and Open Problems

The suggested selection algorithm for local point sets works as expected, leading to optimal convergence rates in Sobolev spaces using the minimal possible number of points for that purpose. Resulting functions are discontinuous, but since they converge in L_∞ to smooth functions, they can be called *asymptotically smooth*.

However, the method can run into instabilities for large data sets roughly at the limits of stability of the global problem. Since the method does not produce *scalable stencils* in the sense of [9], the instability problems may be overcome by going to scale-invariant techniques. This needs further work, along with comparisons to Moving Least Squares or Shepard-type techniques.

Furthermore, it is open whether the technique always produces sets that are in general position with respect to polynomials, and how the Lebesgue constants behave. In contrast to global techniques, the method can vary the kernel scale locally without sacrificing local convergence rates. This needs further work as well.

The method can easily be extended to finding good point sets for local approximations of derivatives [43, 8, 9], and this will have some influence on meshless RBF-FD methods [42, 17, 22, 3, 8, 16].

Also, the implications for the "flat limit" situation, see e.g. [12, 18, 23, 33, 34, 36, 14] are worth investigating.

There are no conflicts of interest.

References

[1] M.G. Armentano. Error estimates in Sobolev spaces for moving least square approximations. *SIAM J. Numer. Anal.*, 39(1):38–51, 2001.

[2] I. Babuska and J.M. Melenk. The Partition of Unity Method. *Int. J. Numer. Meths. Eng.*, 40:727–758, 1997.

[3] G.A. Barnett. *A Robust RBF-FD Formulation based on Polyharmonic Splines and Polynomials*. PhD thesis, Univ. of Colorado, Boulder, 2015.

[4] L.P. Bos and K. Salkauskas. Moving least squares are Backus-Gilbert optimal. *Journal of Approximation Theory*, 59:267–275, 1989.

[5] M.D. Buhmann. *Radial Basis Functions, Theory and Implementations*. Cambridge University Press, Cambridge,UK, 2003.

[6] R. Cavoretto. Adaptive radial basis function partition of unity interpolation, a bivariate algorithm for unstructured data. *J. Sci. Comput.*, 87, 2021.

[7] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc.*, 22(1):211–231, 2009.

[8] O. Davydov and R. Schaback. Minimal numerical differentiation formulas. *Numerische Mathematik*, 140:555–592, 2018.

[9] O. Davydov and R. Schaback. Optimal stencils in Sobolev spaces. *IMA Journal of Numerical Analysis*, 39:398–422, 2019.

[10] St. De Marchi, R. Schaback, and H. Wendland. Near-optimal data-independent point locations for radial basis function interpolation. *Adv. Comput. Math.*, 23(3):317–330, 2005.

[11] Di Tommaso F. Dell'Accio, F. Scattered data interpolation by Shepard's like methods: Classical results and recent advances. 9:32–44, 2016.

[12] T.A. Driscoll and B. Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Comput. Math. Appl.*, 43:413–422, 2002.

[13] R. Farwig. Multivariate interpolation of arbitrarily spaced data by moving least squares methods. *J. Comp. Appl. Math.*, 16:79–93, 1986.

[14] G. Fasshauer and M. McCourt. *Kernel-based Approximation Methods using MATLAB*, volume 19 of *Interdisciplinary Mathematical Sciences*. World Scientific, Singapore, 2015.

[15] G.E. Fasshauer. Approximate moving least-squares approximation with compactly supported radial weights. In *Meshfree methods for partial differential equations (Bonn, 2001)*, volume 26 of *Lect. Notes Comput. Sci. Eng.*, pages 105–116. Springer, Berlin, 2003.

[16] N. Flyer, B. Fornberg, V. Bayona, and G.A. Barnett. On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy. *Journal of Computational Physics*, 321:21–38, 2016.

[17] B. Fornberg, E. Lehto, and C. Powell. Stable calculation of Gaussian-based RBF-FD stencils. *Computers and Mathematics with Applications*, 65:627–637, 2013.

[18] B. Fornberg, G. Wright, and E. Larsson. Some observations regarding interpolants in the limit of flat radial basis functions. *Computers & Mathematics with Applications*, 47:37–55, 2004. doi:10.1016/S0898-1221(04)90004-1.

[19] R. Franke and G.M. Nielson. Scattered data interpolation and applications: A tutorial and survey. In H. Hagen and D. Roller, editors, *Geometric Modeling, Methods and Applications*, pages 131–160, 1991.

[20] A. Heryudono, E. Larsson, and A. Ramage. Preconditioning for radial basis function partition of unity methods. *J. Sci. Comput.*, 67:1089–1109, 2016.

[21] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37:141–158, 1981.

[22] E. Larsson, E. Lehto, A. Heryodono, and B. Fornberg. Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions. *SIAM J. Sci. Comput.*, 35:A2096–A2119, 2013.

[23] Elisabeth Larsson and Bengt Fornberg. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Comput. Math. Appl.*, 49:103–130, 2005.

[24] Elisabeth Larsson, Victor Shcherbakov, and Alfa Heryudono. A least squares radial basis function partition of unity method for solving PDEs. *SIAM Journal on Scientific Computing*, 39(6):A2538–A2563, 2017.

[25] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67:1517–1531, 1998.

[26] D. Mirzaei, R. Schaback, and M. Dehghan. On generalized moving least squares and diffuse derivatives. *IMA J. Numer. Anal.*, 32, No. 3:983–1000, 2012.

[27] M. Mouattamid and R. Schaback. Recursive kernels. *Analysis in Theory and Applications*, 25:301–316, 2009.

[28] St. Müller and R. Schaback. A Newton basis for kernel spaces. *Journal of Approximation Theory*, 161:645–655, 2009.

[29] A. Noorizadeghan and R. Schaback. Introducing the evaluation condition number: A novel assessment of conditioning in radial basis function methods. *Engineering Analysis with Boundary Elements*, 166, 2024.

[30] G. Santin and B. Haasdonk. Convergence rate of the data-independent P-greedy algorithm in kernel-based approximation. *Dolomites Res. Notes Approx.*, 10(Special Issue):68–78, 2017.

[31] R. Schaback. Lower bounds for norms of inverses of interpolation matrices for radial basis functions. *Journal of Approximation Theory*, 79(2):287–306, 1994.

[32] R. Schaback. Reconstruction of multivariate functions from scattered data. Manuscript, available via http://webvm.num.math.uni-goettingen.de/schaback/teaching/rbfbook_2.pdf, 1997.

[33] R. Schaback. Multivariate interpolation by polynomials and radial basis functions. *Constructive Approximation*, 21:293–317, 2005.

[34] R. Schaback. Limit problems for interpolation by analytic radial basis functions. *J. Comp. Appl. Math.*, 212:127–149, 2008.

[35] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 23th National Conference ACM*, pages 517–523. ACM New York, 1968.

[36] G. Song, J. Riddle, G.E. Fasshauer, and F.J. Hickernell. Multivariate interpolation with increasingly flat radial basis functions of finite smoothness. *Adv. Comp. Math.*, 36:485–501, 2012.

[37] V.N. Temlyakov. The best m–term approximation and greedy algorithms. *Advances in Computational Mathematics*, 8:249–265, 1998.

[38] V.N. Temlyakov. Greedy algorithms and m–term approximation with regard to redundant dictionaries. *Journal of Approximation Theory*, 98:117–145, 1999.

[39] H. Wendland. Local polynomial reproduction and moving least squares approximation. *IMA Journal of Numerical Analysis*, 21:285–300, 2001.

[40] H. Wendland. Fast evaluation of radial basis functions: Methods based on partition of unity. In C. K. Chui, L. L. Schumaker, and J. Stöckler, editors, *Approximation Theory X: Wavelets, Splines, and Applications*, pages 473–483. Vanderbilt University Press, 2002.

[41] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge,UK, 2005.

[42] G.B. Wright and B. Fornberg. Scattered node compact finite difference-type formulas generated from radial basis functions. *J. Comput. Phys.*, 212(1):99–123, 2006.

[43] Z. Wu. Hermite–Birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications*, 8/2:1–10, 1992.