Two Stage Scattered Data Fitting

Oleg Davydov

University of Strathclyde

Department of Mathematics, University of Sussex

30 November 2006
Scattered Data Problem

\[ \Omega \subset \mathbb{R}^d \]  
\[ \Xi = \{ \xi_i \}_{i=1}^{M} \subset \Omega \]  
\[ \{ z_i \}_{i=1}^{M} \subset \mathbb{R} \]  

Find:  
\[ s : \Omega \rightarrow \mathbb{R} \text{, an approximation of } f \]
Two-Stage Scattered Data Fitting

Known from the 1970th (Schumaker; Barnhill; Lawson; Foley)

Stage 1

(a) Cover $\Omega$ with **overlapping subdomains** $\omega_\mu$, $\mu \in \mathcal{M}$.

(b) Compute **local approximations** $p_\mu$ to the data $(\xi_i, z_i)$, $\xi_i \in \Xi_\mu \subset \Xi \cap \omega_\mu$. 
Stage 2

Create a smooth global function $s : \Omega \rightarrow \mathbb{R}$ using the information provided by the local approximations $p_\mu, \mu \in \mathcal{M}$.
Advantages

Motivation

- **Efficiency**: Linear computational complexity if $\#\Xi_\mu = O(1)$ and $\#\mathcal{M} = O(N)$

- **“Local approximability”**: Distant data samples do not contain essential new information needed for local approximation if the smoothness of the underlying function is not too high (a function in a Sobolev space, say).

Features achievable with some effort

- **Approximation quality**: Resulting approximation error (after Stage 2) should be comparable with the approximation error of local approximations.

- **Convenient structure of the surface**: E.g. NURBS or Bézier surfaces well known in CAGD; adaptive meshes; multilevel compression algorithms (spline wavelets, hierarchical bases)

- **Artefact-free surface**: $C^1$, $C^2$ or even higher smoothness surfaces without artificial discontinuities or ridges. Local polynomial exactness. No spurious oscillations.
Complexity

A two-stage algorithm must take care of

- choice of subdomains $\omega_\mu$ (usually controlled by the parameters of the method used in the second stage, such as density and shape of spline mesh)
- choice of appropriate local data sites $\Xi_\mu$
- reliability of local approximations

Ideally, all this should be done adaptively, depending on particular data.

Instead, traditional two-stage algorithms choose data sites and spline mesh heuristically and ignore the question of the error of the local approximations.

Typical approach to local approximations: to estimate the value of the unknown function at a point, take, say, the 15 closest data sites, and build least squares cubic polynomial approximation using this data.

This is problematic: no error bound if the local data sites are poorly located (near to a zero set of a cubic polynomial)
Example: 30 Points in $[-1, 1]^2$. Cubics should not be used! (compare: $\dim \Pi_3 = 10$)
Local methods (Stage 1)

- Least squares polynomials
- **Interpolation with positive definite functions** (radial basis functions, kriging)
- **Weighted averages** (Shepard’s interpolation, etc.)
- . . .
Global methods (Stage 2)

- **Spline quasi-interpolation:**  
  \[ s = \sum_{\mu \in \mathcal{M}} \lambda_{\mu}(p_{\mu})B_{\mu} \]  
  \( B_{\mu} \) – locally supported basis splines  
  (local polynomial exactness; NURBS; Bézier surfaces; adaptive meshes)

- **Partition of unity method:**  
  \[ s = \sum_{\mu \in \mathcal{M}} p_{\mu}\Omega_{\mu}, \quad \text{where} \quad \Omega_{\mu} \geq 0, \quad \sum_{\mu \in \mathcal{M}} \Omega_{\mu} \equiv 1 \]  
  (preserves exact interpolation if the local method interpolates exactly)

- **Subdivision surfaces:**  
  Computer Graphics applications

- **Gridding:**  
  no parametric surface, just evaluations on a fixed uniform grid  
  (image processing; FFT; wavelets)
TSFIT (Two-Stage FITting)

- Authors: D. & Zeilfelder
- C library of functions
- Available under GNU General Public License
- Homepage [http://www.maths.strath.ac.uk/~aas04108/tsfit/](http://www.maths.strath.ac.uk/~aas04108/tsfit/)
- LAPACK and BLAS required
- Tested on LINUX machines (x86 and x86_64)
- Test data sets available with the package
Features

- **Goal:** Combine any local method with any global method
- Comparison of various two-stage methods
- Object-oriented style programming with standard C
- Extendibility
- Convincing performance on examples of large, difficult, truly scattered and noisy real world data (contour data, multibeam echosounder data)
Example locale for a global method: 1) point in the domain, and 2) function and derivative values at this point.

Example locale for a local method: 1) triangle to define the Bernstein-Bèzier basis, and 2) coefficients of a polynomial w.r.t. this basis.
Methods Already Implemented

Version 0.91 (November 2005): 2D only

(a) First Stage ("local methods"):  
- Least squares polynomials with degree adapted to the local constellation of data sites by estimating the norm of the least squares operator using the singular value decomposition of the local collocation matrices. [D. & Zeilfelder]
- Hybrid method: Polynomials + radial basis functions (RBF). Local RBF knots are selected by a greedy procedure, the norm of the least squares operator is again controlled with the help of the singular value decomposition. [D., Morandi & Sestini]
- RBF interpolation or least squares: Constants + radial basis functions. Knots are selected by thinning to achieve good separation. [D., Sestini & Morandi]

(b) Second Stage ("global methods"):  
- $C^1$ cubic and $C^2$ sextic “direct extension” splines on the four-directional mesh. [D. & Zeilfelder]
Second Stage: Direct Extension Splines

(D. & Zeilfelder; Applications to computer graphics: Haber et al, IEEE Vis 2001)

$C^1$ cubic or $C^2$ sextic splines on the four-directional mesh

Extension algorithm for the $C^1$ case:
Choice of Local Domain and Points

Parameters: tolerances $M_{\text{min}}$, $M_{\text{max}}$

For each cell $T_\mu$ of a spline partition, find $\omega \supset T_\mu$ s.t. for $\Xi_\mu = \Xi \cap \omega$ it holds: $\#\Xi_\mu \geq M_{\text{min}}$

(to achieve this, we extend $\omega$ by scaling step by step).

Thinning of $\Xi_\mu$, if $\#\Xi_\mu > M_{\text{max}}$. 
Local Approximation

“Local” Problem:

\[ \Xi_\mu \quad \text{local points } \xi_i \text{ in } \omega_\mu \]
\[ f|_{\Xi_\mu} \quad \text{corresponding } z_i \text{-values of } f : \Omega \to \mathbb{R} \]
\[ P_\mu \quad \text{local approximation space} \]

Find: \( p_\mu \in P_\mu \) with small local error \( \| f - p_\mu \|_{C(\omega_\mu)} \)

Fundamental differences to the “global” problem:
- The set \( \Xi_\mu \) is small, i.e. \( \#\Xi_\mu \) is bounded by a constant
- \( \text{diam}(\omega_\mu) \to 0 \)
- Any smooth function is “polynomial-like” locally
Details on Local Polynomial Approximation

\( \mathcal{P}_\mu = \Pi_q \), the space of polynomials of total degree \( \leq q \)

Let \( p_\mu = L_\Xi(f) \) be local least squares polynomial:

\[
\sum_{\xi \in \Xi} |f(\xi) - p_\mu(\xi)|^2 = \min_{p \in \Pi_q} \sum_{\xi \in \Xi} |f(\xi) - p(\xi)|^2, \quad \Xi = \Xi_\mu.
\]

Approximation error:

\[
\|f - p_\mu\|_{C(\omega)} \leq \left(1 + \|L_\Xi\|\right) \inf_{p \in \Pi_q} \|f - p\|_{C(\omega)},
\]

\[
\omega = \omega_\mu, \quad \|L_\Xi\| = \|L_\Xi\|_{C(\omega) \to C(\omega)}.
\]

\( L_\Xi \) is a well-defined bounded operator if the least squares problem is non-degenerate. (It is exact for polynomials of degree \( q \).

We need to make sure that \( \|L_\Xi\| \) does not blow up for any \( \Xi \).
**Computable estimate of** $\| L_\Xi \|$

Let $P_1, \ldots, P_m$ span the space $\mathcal{P}_\mu$ on $\omega$. Consider the local collocation matrix

$$C = [P_j(\xi_i)]_{i,j}.$$

We have

$$K_1 \sigma_{\text{min}}^{-1}(C) \leq \| L_\Xi \| \leq K_2 \sqrt{\# \Xi} \sigma_{\text{min}}^{-1}(C),$$

where $\sigma_{\text{min}}(C)$ is the minimal singular value of $C$, and

$$K_1 \leq \frac{\| \sum_{j=1}^m a_j P_j \|_{C(\omega)}}{\left( \sum_{j=1}^m |a_j|^2 \right)^{1/2}} \leq K_2.$$

(If the basis $\{P_1, \ldots, P_m\}$ is properly scaled, then $K_1, K_2 > 0$ are independent of $\omega$.)

---

We accept a local approximation only if $\sigma_{\text{min}}^{-1}(C) \leq \kappa$, where $\kappa$ is a user specified tolerance.
Proof of (*)

Let

\[
L_\Xi(f) = \sum_{j=1}^{m} a_j P_j.
\]

It follows by a well-known result in numerical linear algebra that the vector \(a = (a_1, \ldots, a_m)^T\) can be computed as the product of the pseudoinverse \(C^+\) of \(C\) with the vector \(f|_\Xi\). Therefore,

\[
\|a\|_2 = \|C^+ f|_\Xi\|_2 \leq \|C^+\|_2 \|f|_\Xi\|_2 = \sigma_{\min}^{-1}(C) \|f|_\Xi\|_2.
\]

Since

\[
\|L_\Xi(f)\|_{C(\omega)} \leq K_2 \|a\|_2
\]

and

\[
\|f|_\Xi\|_2 \leq \sqrt{\#\Xi} \|f|_\Xi\|_\infty \leq \sqrt{\#\Xi} \|f\|_{C(\omega)};
\]

the upper bound in (*) follows:

\[
\|L_\Xi(f)\|_{C(\omega)} \leq K_2 \|a\|_2 \leq K_2 \sigma_{\min}^{-1}(C) \|f|_\Xi\|_2 \leq K_2 \sqrt{\#\Xi} \sigma_{\min}^{-1}(C) \|f\|_{C(\omega)}.
\]
To prove the lower bound, we choose a function $\tilde{f} \in C(\omega)$ such that

$$
\|C^+ \tilde{f}\|_2 = \|C^+\|_2 \|\tilde{f}\|_2, \quad \|\tilde{f}\|_\infty = \|\tilde{f}\|_{C(\omega)},
$$

which is obviously possible. Then we have

$$
\|L\tilde{f}\|_{C(\omega)} \geq K_1 \|C^+ \tilde{f}\|_2 = K_1 \sigma_{min}^{-1}(C) \|\tilde{f}\|_2.
$$

Since $\|\tilde{f}\|_2 \geq \|\tilde{f}\|_\infty = \|\tilde{f}\|_{C(\omega)}$, we arrive at the inequality

$$
\|L\tilde{f}\|_{C(\omega)} \geq K_1 \sigma_{min}^{-1}(C) \|\tilde{f}\|_{C(\omega)}
$$

which completes the proof.
Recall: Example of 30 points in $[-1, 1]^2$ where cubics should not be used.
Error of least squares polynomial for $f(x, y) = x + 3 \sin(xy)$

$$-2.6 < f(x, y) < 2.6, \quad (x, y) \in [-0.8, 0.8]^2$$

<table>
<thead>
<tr>
<th>degree</th>
<th>error on $[-0.8, 0.8]^2$</th>
<th>$\sigma_{\min}^{-1}(C)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.54 (48%)</td>
<td>0.49</td>
</tr>
<tr>
<td>2</td>
<td>0.23 (4.4%)</td>
<td>2.04</td>
</tr>
<tr>
<td>3</td>
<td>6.20 (120%)</td>
<td>83.3</td>
</tr>
<tr>
<td>5</td>
<td>12.6 (240%)</td>
<td>2127</td>
</tr>
</tbody>
</table>

Degree 1: poor approximation ($\sigma_{\min}^{-1}(C) \approx 0.5$ reasonable, but degree too low)

Degree 2: the best choice ($\sigma_{\min}^{-1}(C) \approx 2$ reasonable; quadratics approximate better than linear polynomials)

Degree 3: useless (120% error); cubics can approximate $f$ better than quadratics, but $\sigma_{\min}^{-1}(C) \approx 83$ magnifies the error

Degree 5: useless; the high $\sigma_{\min}^{-1}(C)$ magnifies the error and compensates the positive effect of a higher order approximation

No numerical instability: e.g. $\text{cond}(C) \approx 523$ for cubics

condition $(C) \approx 17000$ for degree 5
If $\sigma_{min}^{-1}(C)$ is too large, there are two possibilities:

1. Extend $\Xi$:

2. Use a less demanding $\mathcal{P}$

Reduction of the polynomial degree $q$:

(bad for $q = 2$, good for $q = 1$)

Both 1. and 2. worsen the best approximation $\inf_{p \in \Pi_q} \|f - p\|_{C(\omega)}$
Algorithm

Parameters: Polynomial degree $q$ and tolerance $\kappa$.

Given: Local points $\Xi_\mu$ and an appropriately scaled polynomial basis

Find: Approximation of $f$ on a cell $T_\mu$ of a spline partition.

1. If $q = 0$, compute $L_{\Xi}(f)$. STOP

2. Compute the singular value decomposition of the local collocation matrix $C$.
   If $\sigma_{\text{min}}(C)^{-1} \leq \kappa$, compute $L_{\Xi}(f)$. STOP
   Otherwise, set $q = q - 1$ and go to 1.
Local Approximation with Radial Basis Functions

D., Morandi & Sestini: two algorithms tested recently

1. “Hybrid” polynomial/RBF local least squares approximation $H(f)$, where

$$\sum_{\xi \in \Xi} |f(\xi) - H(f)(\xi)|^2 = \min_{H \in \mathcal{H}} \sum_{\xi \in \Xi} |f(\xi) - H(\xi)|^2,$$

$$\mathcal{H}_\omega = \Pi_q + \text{span}\left\{ \phi\left(\frac{\| \cdot - \theta \|_2}{\delta d_\omega}\right) : \theta \in \Theta \right\}, \quad \Theta \subset \Xi,$$

$\phi$ is a radial basis function, p.d. or c.p.d. of minimal order $\leq q + 1$

$$d_\omega - \text{diameter of } \omega, \quad \delta - \text{scaling parameter}.$$
Approximation error:

$$\| f - H(f) \|_{C(\omega)} \leq \left( 1 + \| L^H_\Xi \| \right) \inf_{H \in H_\omega} \| f - H \|_{C(\omega)},$$

where $\| L^H_\Xi \|$ is the norm of the least squares operator for the hybrid space.

Similar to the pure polynomial case, $\| L^H_\Xi \|$ can be estimated by minimum singular value of the corresponding collocation matrix.

Conversion to a polynomial needed by the “extension” spline:
Interpolation or least squares w.r.t. the evaluations of $H(f)$ on a local grid.
**Algorithm: Hybrid Method**

**Parameters:** Polynomial degree $q$, RBF $\phi$, scaling coefficient $\delta$, and tolerances $\kappa_P, \kappa_H$.

**Given:** Local points $\Xi_\mu$ and an appropriately scaled polynomial basis

1. If $\#\Xi_\mu < \dim \Pi_q + 3$, use the polynomial method. STOP

2. Initialize the knot set $Y_\mu \subset \Xi_\mu$ with 3 Points in good location, and compute the singular value decomposition of the hybrid collocation matrix $C$.

3. If $\sigma_{\text{min}}(C)^{-1} > \kappa_H$, use the polynomial method. STOP

4. Compute the hybrid approximation $H_\mu(f)$ with knots $Y_\mu$. STOP, if $\dim \Pi_q + \#Y_\mu = \#\Xi_\mu$.

5. Let $\xi$ be a point in $\Xi \setminus Y_\mu$ of the highest error $|f(\xi) - H_\mu(f, \xi)|$. Set $Y_\mu = Y_\mu \cup \{\xi\}$, and compute the singular value decomposition of the hybrid collocation matrix $C$. STOP, if $\sigma_{\text{min}}(C)^{-1} > \kappa_H$. Otherwise, go to 4.
2. Standard RBF approximation

\[ R(f) = \tilde{p} + \sum_{\theta \in \Theta} a_\theta \phi \left( \frac{\| \cdot - \theta \|_2}{\delta d_\omega} \right), \quad \tilde{p} \in \Pi_q, \quad \Theta \subset \Xi, \]

where \( \phi \) is p.d. or c.p.d. of minimal order \( \leq q + 1 \).

**Interpolation:**

\[ R(f)(\theta) = f(\theta), \quad \text{all } \theta \in \Theta, \]
\[ \sum_{\theta \in \Theta} a_\theta p(\theta) = 0, \quad \text{all } p \in \Pi_q, \]

**(Constrained) least squares:**

\[ \sum_{\xi \in \Xi} |f(\xi) - R(f)(\xi)|^2 = \min_{R \in \mathcal{H}_\omega} \sum_{\xi \in \Xi} |f(\xi) - R(\xi)|^2, \]
\[ \sum_{\theta \in \Theta} a_\theta p(\theta) = 0, \quad \text{all } p \in \Pi_q. \]
Approximation error of RBF interpolation

Adjusting some results from the literature (in particular, Madych & Nelson; Schaback; Jetter, Stöckler & Ward), we get

\[ \|f - R(f)\|_{C(\omega)} \leq \left(1 + \frac{1}{\nu(\Pi_q, \Theta)}\right) \sqrt{\inf_{p \in \Pi_q} \left\| \phi(\| \cdot \|_2) - p\right\|_{C(B_{1/\delta}(0))}} \|f\|_{\phi_\omega}, \]

where

\[ |f|_{\phi_\omega} := \left(\int_{\mathbb{R}^d} \frac{\hat{f}(x)^2}{\hat{\Phi}_\omega(x)} \, dx\right)^{1/2}, \quad \phi_\omega := \phi\left(\frac{\| \cdot - \theta\|_2}{\delta d_\omega}\right), \quad \Phi_\omega(\cdot) := \phi_\omega(\| \cdot \|_2), \]

\[ B_{1/\delta}(0) = \text{ ball in } \mathbb{R}^d \text{ with center at origin and radius } 1/\delta, \]

\[ \nu(\Pi_q, \Theta) = \min_{p \in \Pi_q} \frac{\|p\|_\infty}{\|p\|_{C(\omega)}} = \text{ polynomial norming constant related to } \|L_\Theta\| \]

\[ |f|_{\phi_\omega} \text{ is in general different for different local subdomains } \omega \]
Approximation error in the case of thin plate splines.

Thin plate spline:

\[
\phi_{TP,\beta}(r) = \begin{cases} 
(-1)^{\lceil \beta/2 \rceil} r^\beta, & \beta \in \mathbb{R}_{>0} \setminus 2\mathbb{N}, \\
(-1)^{\beta/2+1} r^\beta \log r, & \beta \in 2\mathbb{N},
\end{cases}
\]

(c.p.d. of order \( \lceil \beta/2 \rceil \) if \( \beta \in \mathbb{R}_{>0} \setminus 2\mathbb{N} \), and \( \beta/2 + 1 \) if \( \beta \in 2\mathbb{N} \)).

Therefore,

\[
|f|_{\phi_{TP,\beta}}^2 = (\delta d_\omega)^\beta |f|_{\phi_{TP,\beta}}^2,
\]

\[
\inf_{p \in \Pi_q} \left\| \phi(\| \cdot \|_2) - p \right\|_{C(B_{1/\delta}(0))} \leq C_q (1/\delta)^\beta,
\]

Estimate:

\[
\|f - R(f)\|_{C(\omega)} \leq C_q \left(1 + \frac{1}{\nu(\Pi_q, \Theta)}\right) d_\omega^{\beta/2} |f|_{\phi}
\]

More elaborate estimates allow to replace \( d_\omega^{\beta/2} \) with \( d_\omega^{\beta+1} \) (even \( d_\omega^{\beta+2} \) for grid data: Buhmann) at the expense of using a stronger norm of \( f \).
Algorithm: RBF Method

1. Choose $\Xi_\mu$ and set $Y_\mu = \Xi_\mu$. Thin $Y_\mu$ if needed, s.t. $d_{\omega_\mu}/sd(Y_\mu) \leq S$. ($d_{\omega_\mu}$ — diameter of $\omega_\mu$, $sd(Y_\mu)$ — separation distance of $Y_\mu$.)

2. Adjust the polynomial degree $q$ to the knot set $Y_\mu$ (as with the polynomial method).

3. Compute either RBF interpolation $R^I_\mu(f)$ with the knots $Y_\mu$, or RBF least squares $R^{LS}_\mu(f)$ with the knots $Y_\mu$ and data $\Xi_\mu$.

Parameters: Polynomial degree $q$, RBF $\phi$ (p.d. or c.p.d. of order 1), scaling coefficient $\delta$, und tolerances $M_{\text{min}}, M_{\text{max}}, \kappa_P, S$. 
Work in Progress / Future Work

- **Improved local methods** (greater adaptivity in the local data selection, taking into account the information about level and type of noise, etc.)

- **Additional global methods**, with emphasis on splines (tensor product splines, NURBS, box splines, Powell-Sabin splines, etc.)

- Using spline wavelets and hierarchical splines for compression of surfaces obtained from data fitting

- **Adaptive irregular meshes** (multivariate counterpart of univariate free knot splines; use $C^1$ and $C^2$ polynomial splines on irregular triangulations and local refinement algorithms from FEM)

- Fitting functional data on **manifolds** (joint work with Larry Schumaker) and in higher dimensions

- **Methods tuned for particular type of data** (e.g. contour data)
Numerical Examples

1. Recovery of Franke test function from its values at 100 points.
(The data ds3 is available from http://www.math.nps.navy.mil/~rfranke/)

\[
f(x, y) = \frac{3}{4} \exp \left[-\frac{(9x - 2)^2 + (9y - 2)^2}{4}\right] + \frac{3}{4} \exp \left[-\frac{(9x + 1)^2}{49} - \frac{(9y + 1)}{10}\right] \\
+ \frac{1}{2} \exp \left[-\frac{(9x - 7)^2 + (9y - 3)^2}{4}\right] - \frac{1}{5} \exp \left[-(9x - 4)^2 - (9y - 7)^2\right].
\]
(H\textsubscript{MQ}): Hybrid method based on the multiquadric function \( \phi(r) = \sqrt{1 + r^2} \). Polynomial degree: \( q = 0 \). Spline space: \( C^2 \) sextic piecewise polynomials on the four-directional mesh (5 \times 5 square grid with both diagonals of the squares drawn in, \( M_{\text{min}} = 16 \), \( M_{\text{max}} = 100 \)).

We also consider methods based on other RBFs: (H\textsubscript{IMQ}), (H\textsubscript{G}), (H\textsubscript{TP}), (H\textsubscript{W2}), etc.

Global methods from Franke’s 1979 report are denoted (G\textsubscript{MQ}), (G\textsubscript{IMQ}), etc.

Pure polynomial method is denoted by (P).
<table>
<thead>
<tr>
<th>method</th>
<th>$q$</th>
<th>$\kappa_H$</th>
<th>$\delta$</th>
<th>max</th>
<th>mean</th>
<th>rms</th>
<th>$n_T^{\text{aver}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{MQ}$</td>
<td>0</td>
<td>$10^5$</td>
<td>0.4</td>
<td>$1.610^{-2}$</td>
<td>$1.910^{-3}$</td>
<td>$3.010^{-3}$</td>
<td>21.1</td>
</tr>
<tr>
<td>$H_{IMQ}$</td>
<td>0</td>
<td>$10^4$</td>
<td>0.5</td>
<td>$1.510^{-2}$</td>
<td>$2.010^{-3}$</td>
<td>$3.110^{-3}$</td>
<td>21.3</td>
</tr>
<tr>
<td>$H_G$</td>
<td>0</td>
<td>$10^4$</td>
<td>0.4</td>
<td>$1.910^{-2}$</td>
<td>$2.210^{-3}$</td>
<td>$3.510^{-3}$</td>
<td>19.3</td>
</tr>
<tr>
<td>$H_{TP}$</td>
<td>1</td>
<td>$10^5$</td>
<td>2.0</td>
<td>$5.710^{-2}$</td>
<td>$7.810^{-3}$</td>
<td>$1.310^{-2}$</td>
<td>20.7</td>
</tr>
<tr>
<td>$H_{TP3}$</td>
<td>1</td>
<td>$10^5$</td>
<td>2.0</td>
<td>$4.710^{-2}$</td>
<td>$4.510^{-3}$</td>
<td>$7.510^{-3}$</td>
<td>20.3</td>
</tr>
<tr>
<td>$H_{TP4}$</td>
<td>2</td>
<td>$10^5$</td>
<td>2.0</td>
<td>$3.010^{-2}$</td>
<td>$3.410^{-3}$</td>
<td>$5.510^{-3}$</td>
<td>14.9</td>
</tr>
<tr>
<td>$H_{TP5}$</td>
<td>2</td>
<td>$10^6$</td>
<td>2.0</td>
<td>$2.810^{-2}$</td>
<td>$3.410^{-3}$</td>
<td>$5.210^{-3}$</td>
<td>13.4</td>
</tr>
<tr>
<td>$H_{W2}$</td>
<td>0</td>
<td>$10^4$</td>
<td>2.0</td>
<td>$3.910^{-2}$</td>
<td>$4.110^{-3}$</td>
<td>$7.110^{-3}$</td>
<td>22.7</td>
</tr>
<tr>
<td>$H_{B3}$</td>
<td>0</td>
<td>$10^5$</td>
<td>2.0</td>
<td>$3.310^{-2}$</td>
<td>$3.610^{-3}$</td>
<td>$6.010^{-3}$</td>
<td>22.4</td>
</tr>
<tr>
<td>$H_{W4}$</td>
<td>0</td>
<td>$10^4$</td>
<td>2.0</td>
<td>$2.110^{-2}$</td>
<td>$2.110^{-3}$</td>
<td>$3.610^{-3}$</td>
<td>21.2</td>
</tr>
<tr>
<td>$H_{W6}$</td>
<td>0</td>
<td>$10^5$</td>
<td>2.0</td>
<td>$1.610^{-2}$</td>
<td>$1.910^{-3}$</td>
<td>$3.010^{-3}$</td>
<td>20.9</td>
</tr>
<tr>
<td>(P)</td>
<td></td>
<td>6</td>
<td></td>
<td>$3.810^{-2}$</td>
<td>$5.210^{-3}$</td>
<td>$7.610^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$G_{MQ}$</td>
<td></td>
<td></td>
<td></td>
<td>$2.310^{-2}$</td>
<td>$1.810^{-3}$</td>
<td>$3.610^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$G_{IMQ}$</td>
<td></td>
<td></td>
<td></td>
<td>$2.510^{-2}$</td>
<td>$2.810^{-3}$</td>
<td>$5.210^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$G_G$</td>
<td></td>
<td></td>
<td></td>
<td>$6.210^{-2}$</td>
<td>$6.010^{-3}$</td>
<td>$1.110^{-2}$</td>
<td></td>
</tr>
<tr>
<td>$G_{TP}$</td>
<td>1</td>
<td></td>
<td></td>
<td>$5.210^{-2}$</td>
<td>$5.310^{-3}$</td>
<td>$9.510^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$G_{TP3}$</td>
<td>1</td>
<td></td>
<td></td>
<td>$2.510^{-2}$</td>
<td>$3.110^{-3}$</td>
<td>$5.810^{-3}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Franke function test (ds3 data set): errors on a dense grid.
2. Approximation order tests with Franke function.

\( N \) is the number of points

<table>
<thead>
<tr>
<th>( N )</th>
<th>( n_x )</th>
<th>( M_{\text{min}} )</th>
<th>( \kappa_H )</th>
<th>( \delta )</th>
<th>( \text{max} )</th>
<th>( \text{mean} )</th>
<th>( \text{rms} )</th>
<th>( n_T^{\text{aver}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^2 )</td>
<td>5</td>
<td>16</td>
<td>( 10^5 )</td>
<td>0.4</td>
<td>( 4.60 \cdot 10^{-2} )</td>
<td>( 3.98 \cdot 10^{-3} )</td>
<td>( 7.46 \cdot 10^{-3} )</td>
<td>19</td>
</tr>
<tr>
<td>( 10^3 )</td>
<td>16</td>
<td>40</td>
<td>( 10^{12} )</td>
<td>1.0</td>
<td>( 1.69 \cdot 10^{-4} )</td>
<td>( 1.53 \cdot 10^{-6} )</td>
<td>( 6.47 \cdot 10^{-6} )</td>
<td>47</td>
</tr>
<tr>
<td>( 10^4 )</td>
<td>50</td>
<td>40</td>
<td>( 10^{15} )</td>
<td>1.6</td>
<td>( 4.64 \cdot 10^{-7} )</td>
<td>( 5.62 \cdot 10^{-9} )</td>
<td>( 1.51 \cdot 10^{-8} )</td>
<td>48</td>
</tr>
</tbody>
</table>

Table 2: Franke function test (random data): hybrid method with multiquadric.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( \delta = 0.4 )</th>
<th>( \delta = 0.8 )</th>
<th>( \delta = 1.2 )</th>
<th>( \delta = 1.6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S = 40 )</td>
<td>( S = 20 )</td>
<td>( S = 40/3 )</td>
<td>( S = 10 )</td>
<td></td>
</tr>
<tr>
<td>( 10^2 )</td>
<td>( 2.27 \cdot 10^{-2} )</td>
<td>( 2.81 \cdot 10^{-2} )</td>
<td>( 3.66 \cdot 10^{-2} )</td>
<td>( 4.48 \cdot 10^{-2} )</td>
</tr>
<tr>
<td>( 10^3 )</td>
<td>( 1.26 \cdot 10^{-5} )</td>
<td>( 4.44 \cdot 10^{-6} )</td>
<td>( 6.13 \cdot 10^{-5} )</td>
<td>( 5.42 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( 10^4 )</td>
<td>( 4.20 \cdot 10^{-6} )</td>
<td>( 1.98 \cdot 10^{-7} )</td>
<td>( 1.00 \cdot 10^{-7} )</td>
<td>( 2.36 \cdot 10^{-7} )</td>
</tr>
<tr>
<td>( 10^5 )</td>
<td>( 2.03 \cdot 10^{-6} )</td>
<td>( 9.28 \cdot 10^{-8} )</td>
<td>( 3.54 \cdot 10^{-8} )</td>
<td>( 5.60 \cdot 10^{-8} )</td>
</tr>
<tr>
<td>#knots</td>
<td>122.2</td>
<td>87.2</td>
<td>60.4</td>
<td>42.8</td>
</tr>
</tbody>
</table>

Table 3: Maximum error using the RBF interpolation method with multiquadric. Other parameters: \( M_{\text{min}} = 20 \) if \( N = 10^2 \) and \( M_{\text{min}} = 100 \) otherwise.
Thin plate splines with RBF interpolation method.

<table>
<thead>
<tr>
<th>N</th>
<th>spline grid</th>
<th>$\text{max } (\beta = 3/2)$</th>
<th>$\text{max } (\beta = 7/4)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^2$</td>
<td>5 × 5</td>
<td>$8.55 \cdot 10^{-2}$</td>
<td>$6.92 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>$10^3$</td>
<td>16 × 16</td>
<td>$5.22 \cdot 10^{-3}$</td>
<td>$3.37 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>$10^4$</td>
<td>50 × 50</td>
<td>$2.60 \cdot 10^{-4}$</td>
<td>$1.17 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>$10^5$</td>
<td>158 × 158</td>
<td>$2.37 \cdot 10^{-5}$</td>
<td>$7.09 \cdot 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 4: Maximum error using the local RBF interpolation scheme based on $\phi(r) = r^\beta$, $\beta = 3/2$ and $7/4$. Parameter values: $\delta = 1$, $\tilde{q} = 0$, $M_{\text{min}} = S = 100$. Approximation order about $h^{\beta+1}$.
3. Denoising

Franke test function with normally distributed random errors on the 100x100 grid (standard deviation $\sigma = 0.05$)
Reconstruction ($C_1$ spline, polynomial method) of the contaminated data

\[ \text{dim}=304 \]

\textbf{Error w.r.t. the original function:} \quad \text{max}=0.0274, \text{ mean}=0.00415, \text{ rms}=0.00552

\textbf{Parameters:} \quad \kappa = 1, \quad M_{\text{max}}=300

\textbf{Noise reduction:} \quad \sigma/rms = 9.058
Reconstruction of the Franke test function from 500 contaminated values by [McMahon & Franke, 1992] (standard deviation $\sigma = 0.05$)

231 degrees of freedom, \hspace{1em} rms error: 0.0178

Noise reduction: $\sigma/rms = 2.81$
4. Glacier data set: 8,345 points (44 digitized height contours of a glacier with $25\,m$ vertical spacing; also available from the homepage of Franke).

Location of data points

3D perspective view
Glacier test: The $C^2$ spline approximation ($n_x = 20$, $n_y = 24$) with (H$_{MQ}$).
5. Black Forest data

15,885 data points from an area of 144 km$^2$ (elevation range 1200 m)
The $C^2$ spline approximation ($n_x = n_y = 80$) with ($H_{MQ}$)
6. Rotterdam Port multibeam echosounder data: 634,604 raw data points (courtesy Quality Positioning Services, Zeist, Holland).
Zoom to the local distribution of the xy-points
Special features of multibeam data:

- **Huge data sets are produced very fast.** (Tens of millions of points per hour.)
- They should be **visualised in real time.**
- The measurement **errors** (noise, outliers) should be **removed in real time.**
- **Compression** is highly desirable.
Processing of the Rotterdam Port data

1) A coarse approximation (22,399 degrees of freedom, 23.8 s computational time, rms error 0.61 m):

Vertically exaggerated (4×). The z-values of the data: -27.6 m to -5.3 m.
2) Despiking / data cleaning: all points (12.980) with the error > 0.61 removed:
3) The final spline surface (142,027 degrees of freedom, 114.6 s computational time, rms error < 0.08 m w.r.t. the cleaned data):
View from the above

The surface faithfully represents a fine structure on the harbour floor such as remains from the dredge process.