Two Stage Scattered Data Fitting

Oleg Davydov

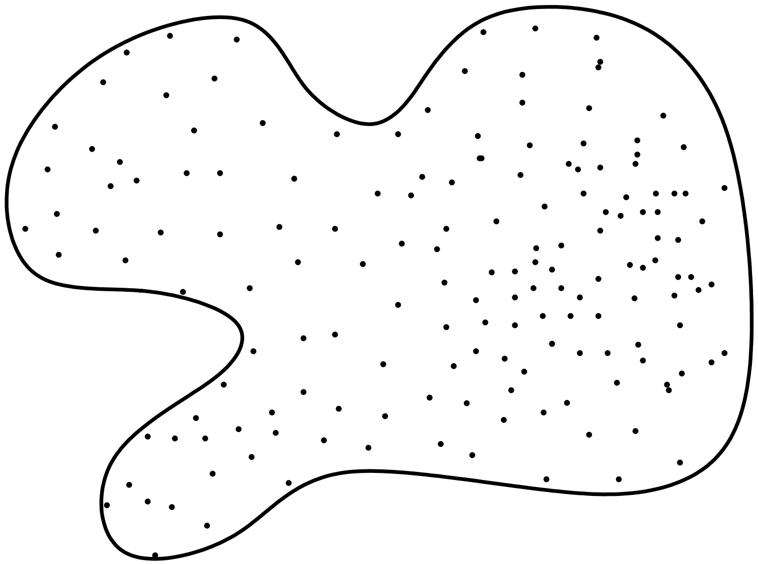
University of Strathclyde

Department of Mathematics, University of Sussex

30 November 2006

Scattered Data Problem

$\Omega \subset \mathbb{R}^d$	a bounded domain ($d > 1$)
$\Xi = \{\xi_i\}_{i=1}^M \subset \Omega$	arbitrarily distributed points in Ω
$\{z_i\}_{i=1}^M \subset \mathbb{R}$	known values of $f : \Omega \to \mathbb{R}$
Find:	$s : \Omega \to \mathbb{R}$, an **approximation of f**

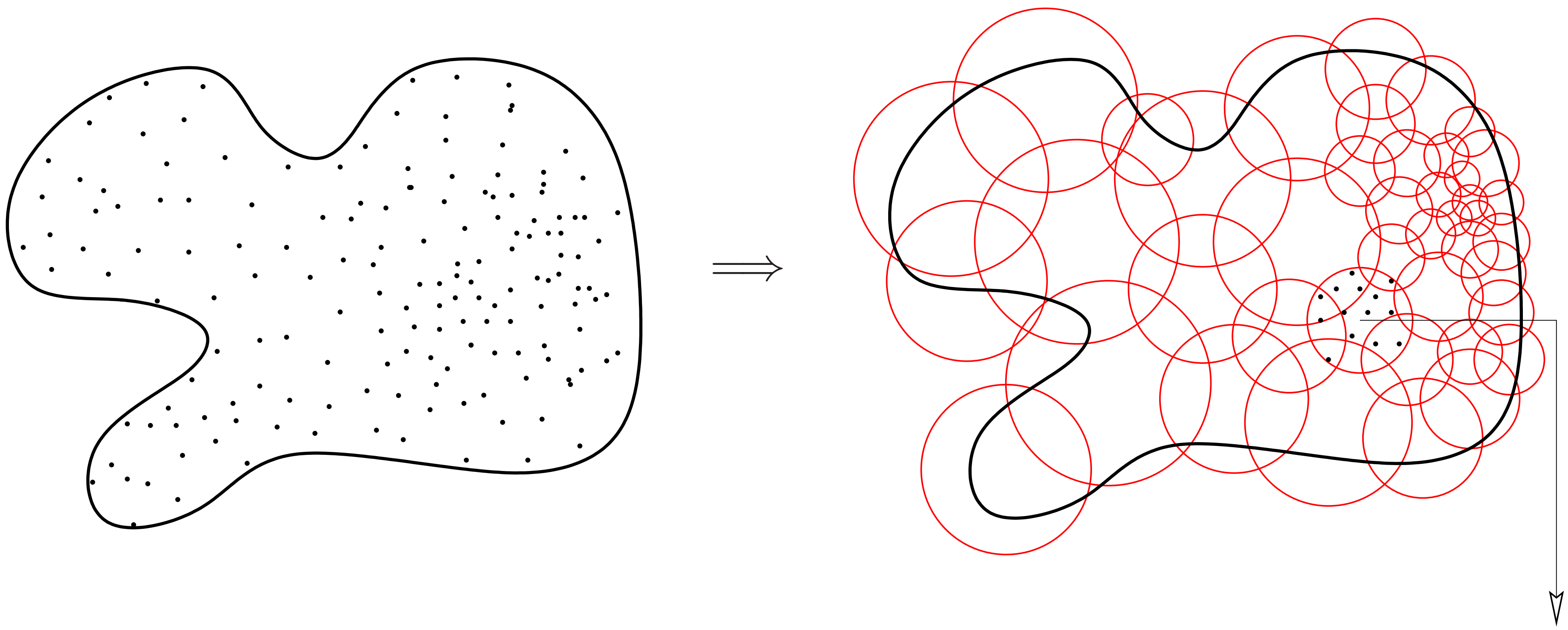


Two-Stage Scattered Data Fitting

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

Stage 1

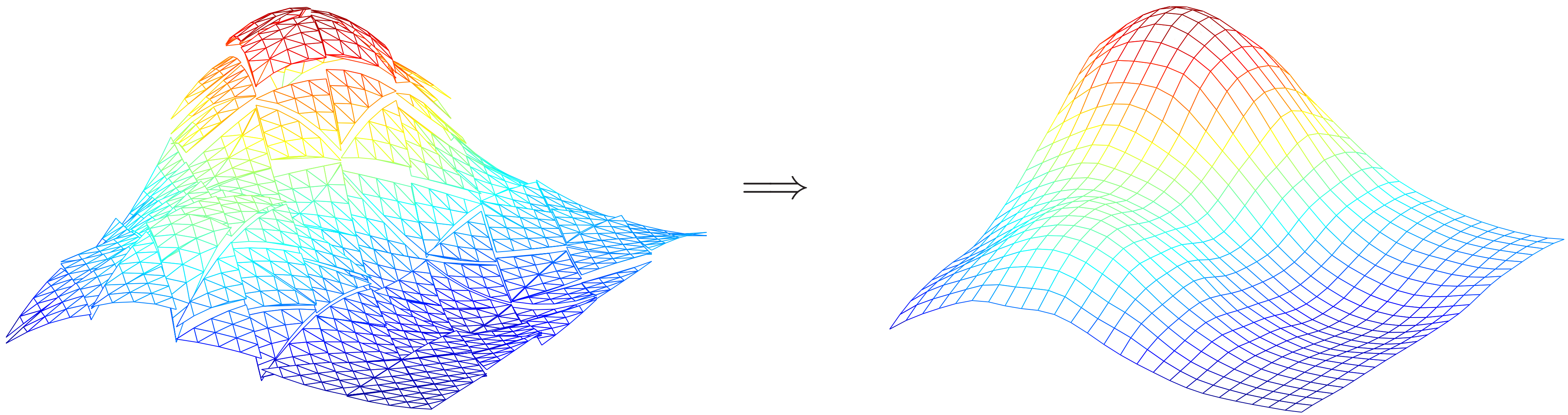
(a) Cover Ω with **overlapping subdomains** ω_μ, $\mu \in \mathcal{M}$.



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

Stage 2

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



Advantages

Motivation

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

- **“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 ω_μ (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 Ξ_μ
- reliability of local approximations

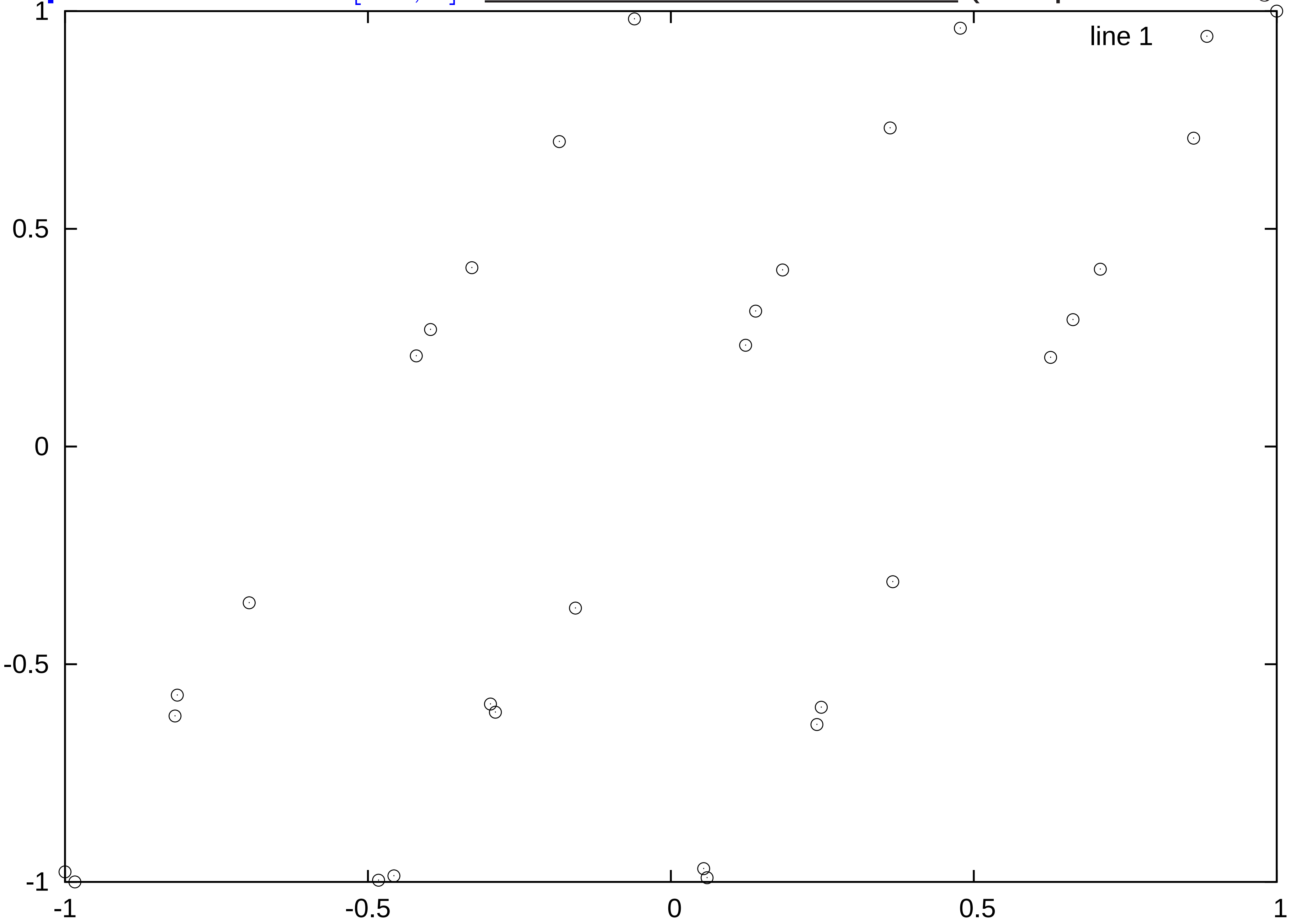
Ideally, all this should be done adaptively, depending on particular data.

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_μ – locally supported basis splines
  (local polynomial exactness; NURBS; Bézier surfaces; adaptive meshes)

- **Partition of unity method:** $s = \sum_{\mu \in \mathcal{M}} p_\mu \Omega_\mu,$ where $\Omega_\mu \geq 0, \quad \sum_{\mu \in \mathcal{M}} \Omega_\mu \equiv 1$

  (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)

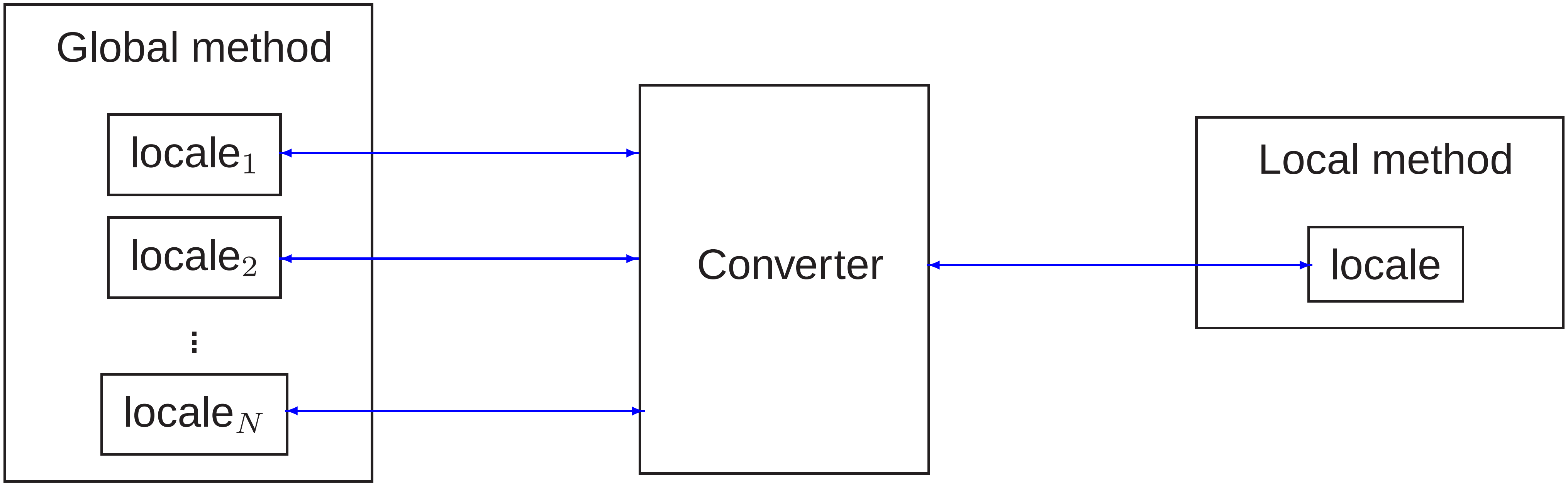
TSFIT (Two-Stage FITting)

- Authors: D. & Zeilfelder
- C library of functions
- Available under GNU General Public License
- Homepage `http://www.maths.strath.ac.uk/~aas04108/tsfit/`
- 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)

Design



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]

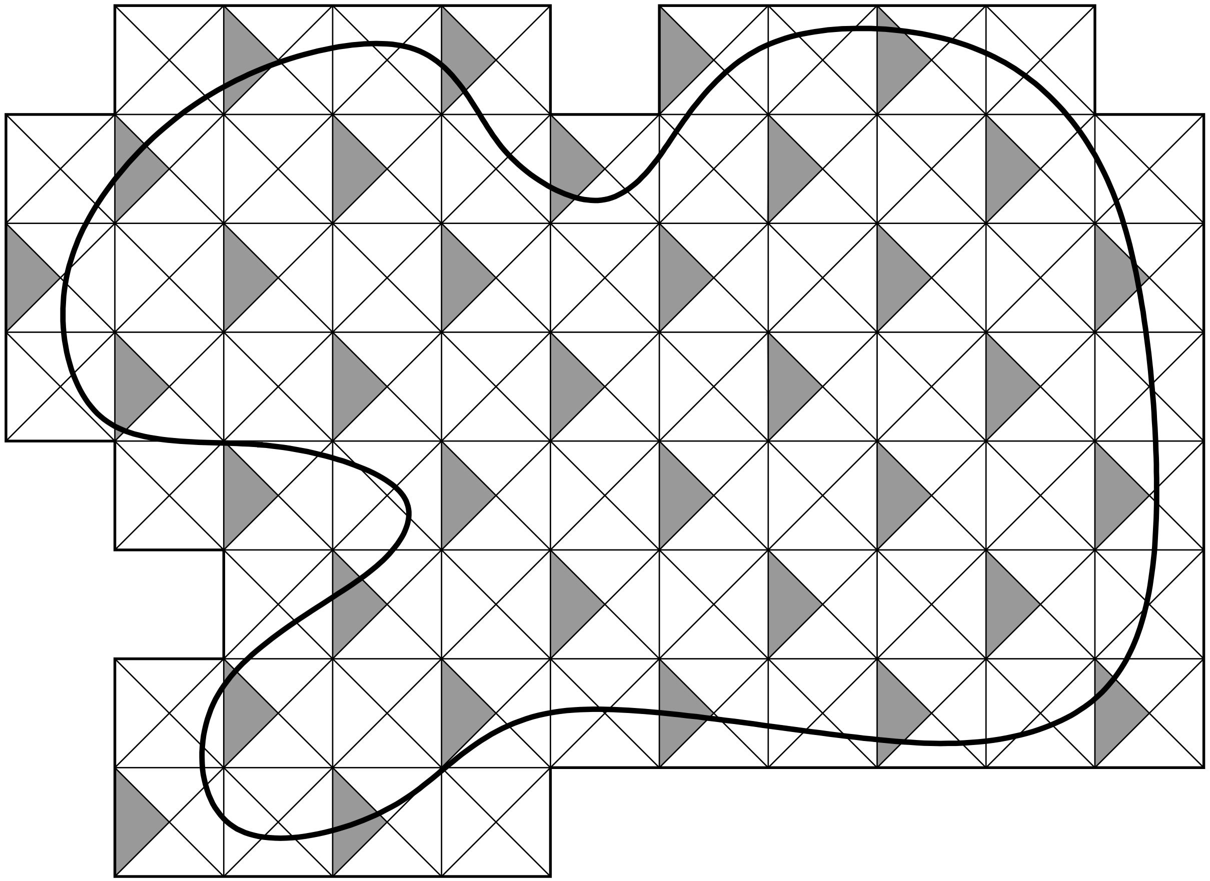
(b) **Second Stage ("global methods"):**

- C^1 cubic and C^2 sextic "direct extension" splines on the four-directional mesh. [D. & Zeilfelder]

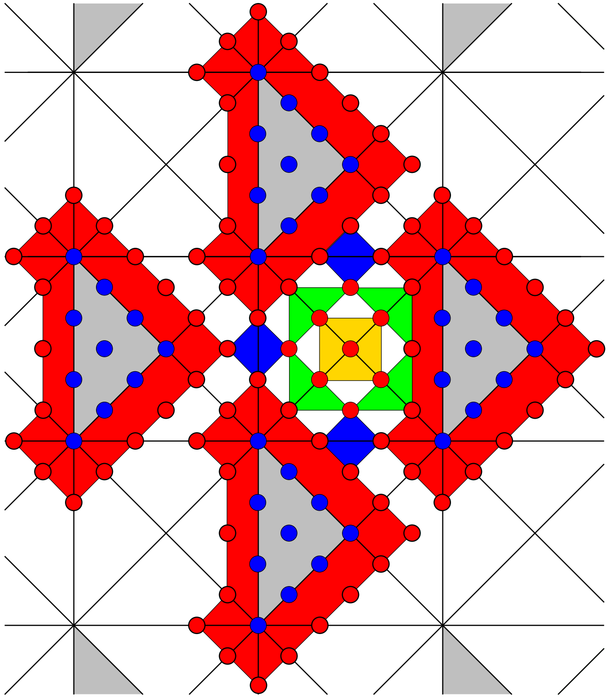
Second Stage: Direct Extension Splines

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

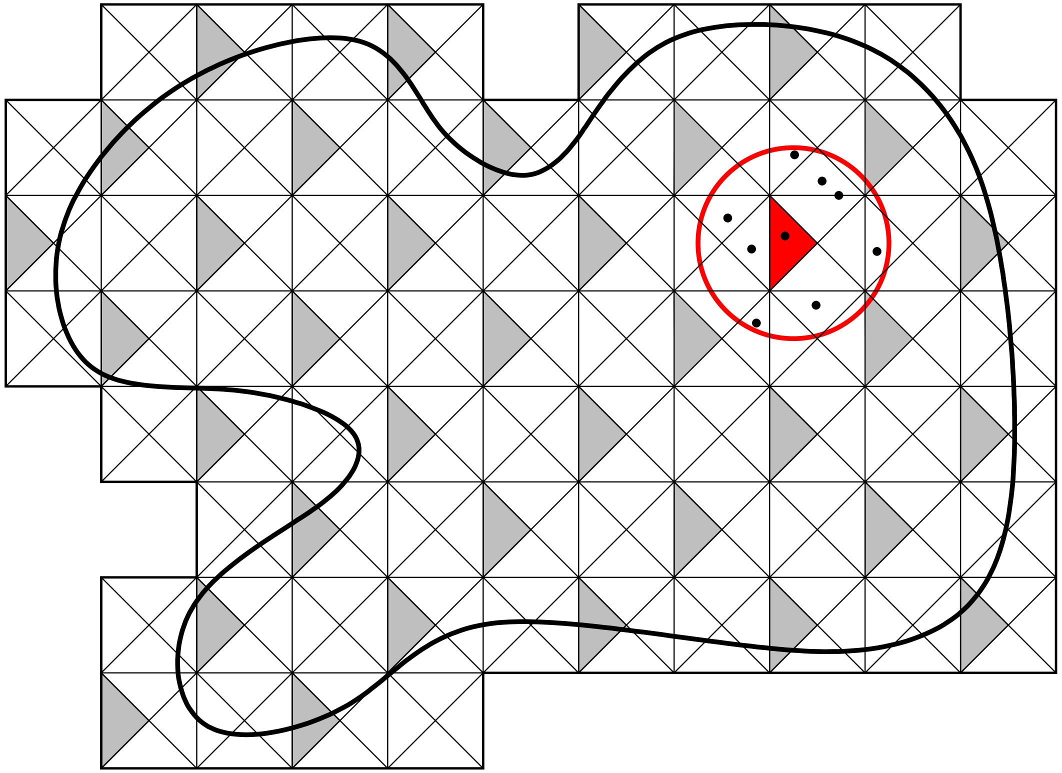
C^1 cubic or C^2 sextic splines on the four-directional mesh



Extension algorithm
for the C^1 case:



Choice of Local Domain and Points



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

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

(to achieve this, we extend ω by scaling step by step).

Thinning of Ξ_μ , if $\#\Xi_\mu > M_{\max}$.

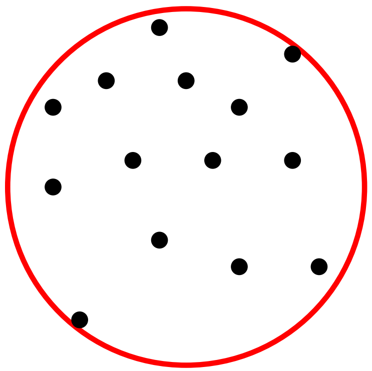
Local Approximation

"Local" Problem:

Ξ_μ	local points ξ_i in ω_μ
$f\|_{\Xi_\mu}$	corresponding z_i-values of $f : \Omega \to \mathbb{R}$
$\mathcal{P}_\mu$	local approximation space

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

$\Xi_\mu \subset \omega_\mu$:



Fundamental differences to the "global" problem:

- The set Ξ_μ is small, i.e. $\#\Xi_\mu$ is bounded by a constant
- $\mathrm{diam}(\omega_\mu) \to 0$
- 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:

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

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_Ξ 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 Ξ

Computable estimate of $\|L_\Xi\|$

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

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

We have

$$K_1\, \sigma_{min}^{-1}(C) \le \|L_\Xi\| \le K_2 \sqrt{\#\Xi}\; \sigma_{min}^{-1}(C), \qquad (*)$$

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

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

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

We accept a local approximation only if $\sigma_{min}^{-1}(C) \le \kappa$, where κ 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

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

which is obviously possible. Then we have

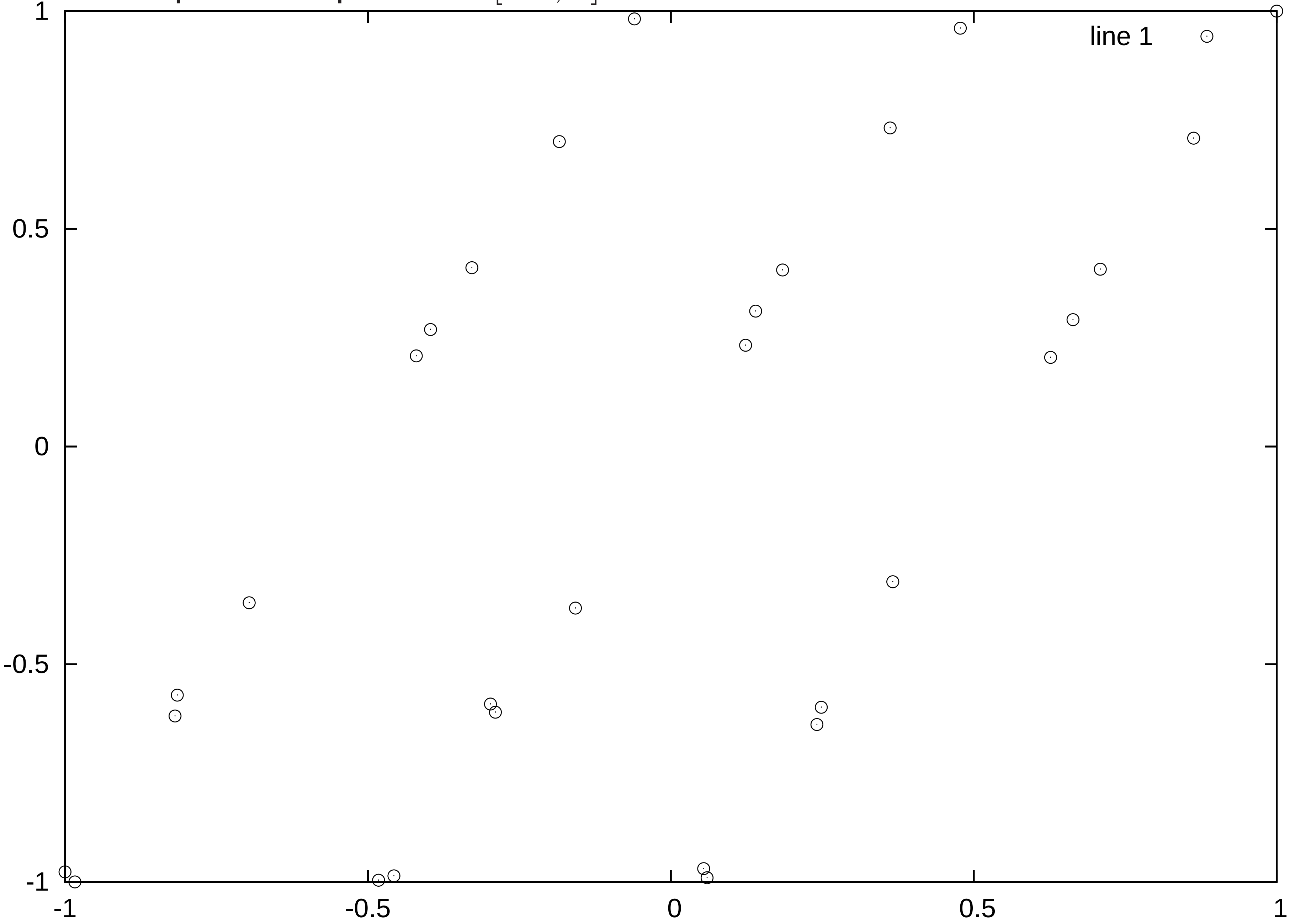
$$\|L_\Xi\tilde{f}\|_{C(\omega)} \geq K_1\|C^+\tilde{f}|_\Xi\|_2 = K_1\sigma_{min}^{-1}(C)\|\tilde{f}|_\Xi\|_2.$$

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

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

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 $f(x,y) = x + 3\sin(xy)$

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

degree	error on $[-0.8, 0.8]^2$	$\sigma_{min}^{-1}(C)$
1	2.54 (48%)	0.49
2	0.23 (4.4%)	2.04
3	6.20 (120%)	83.3
5	12.6 (240%)	2127

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)

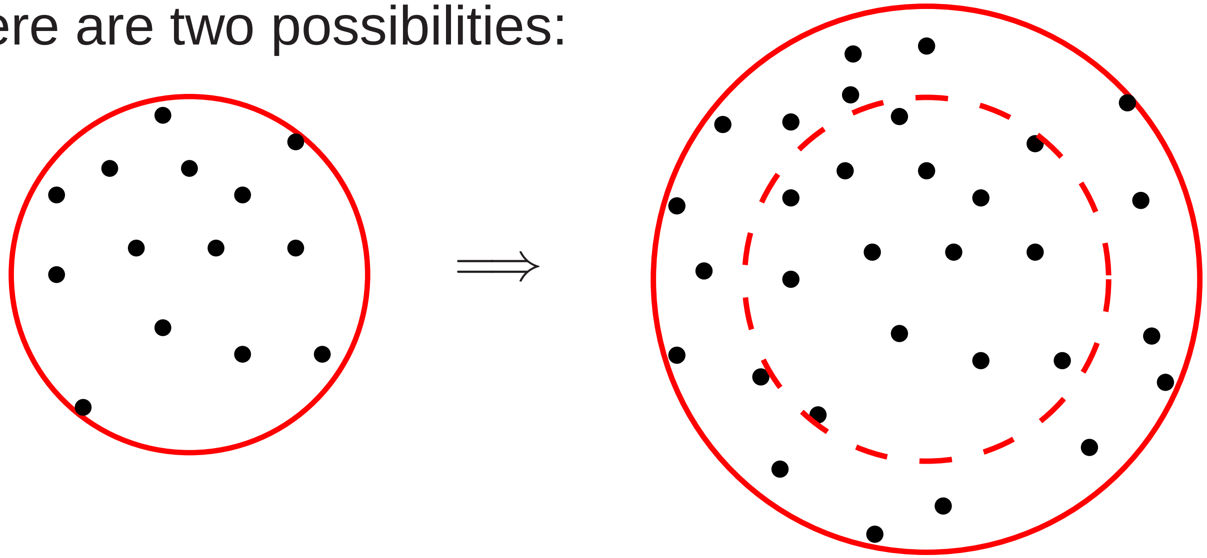
Degree 3: useless (120% error); cubics can approximate f better than quadratics, but $\sigma_{min}^{-1}(C) \approx 83$ magnifies the error

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. cond$(C) \approx 523$ for cubics
cond$(C) \approx 17000$ for degree 5

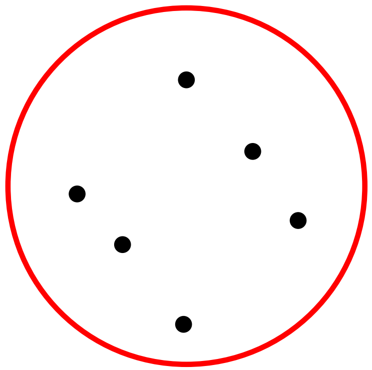
If $\sigma_{min}^{-1}(C)$ **is too large**, there are two possibilities:

1. Extend Ξ:



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 κ.

Given: Local points Ξ_μ and an appropriately scaled polynomial basis

Find: Approximation of f on a cell T_μ 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 $\sigma_{\min}(C)^{-1} \leq \kappa$, 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

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

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

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

d_ω – diameter of ω, δ – scaling parameter.

Approximation error:

$$\|f - H(f)\|_{C(\omega)} \le \left(1 + \|L_{\Xi}^{H}\|\right) \inf_{H \in \mathcal{H}_\omega} \|f - H\|_{C(\omega)},$$

where $\|L_{\Xi}^{H}\|$ is the norm of the least squares operator for the hybrid space.

Similar to the pure polynomial case, $\|L_{\Xi}^{H}\|$ 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 ϕ, scaling coefficient δ, and tolerances κ_P, κ_H.

Given: Local points Ξ_μ 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 $\sigma_{\min}(C)^{-1} > \kappa_H$, use the polynomial method. STOP

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

5. Let ξ 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 $\sigma_{\min}(C)^{-1} > \kappa_H$. Otherwise, go to **4**.

2. Standard RBF approximation

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

where ϕ is p.d. or c.p.d. of minimal order $\leq q+1$.

Interpolation:

$$\begin{aligned} R(f)(\theta) &= f(\theta), \quad \text{all } \theta \in \Theta, \\ \sum_{\theta \in \Theta} a_\theta p(\theta) &= 0, \qquad \text{all } p \in \Pi_q, \end{aligned}$$

(Constrained) least squares:

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

$$\sum_{\theta \in \Theta} a_\theta p(\theta) = 0, \qquad \text{all} \quad p \in \Pi_q.$$

Approximation error of RBF interpolation

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

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

where

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

$B_{1/\delta}(\mathbf{0})$ – ball in $\mathbb{R}^d$ with center at origin and radius $1/\delta$,

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

$|f|_{\phi_\omega}$ is in general different for different local subdomains ω

Approximation error in the case of thin plate splines.

Thin plate spline:

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

(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,

$$|f|^2_{\phi_\omega^{\mathrm{TP},\beta}} = (\delta d_\omega)^\beta |f|^2_{\phi^{\mathrm{TP},\beta}}, \qquad \inf_{p\in\Pi_q} \Big\| \phi(\|\cdot\|_2) - p \Big\|_{C(B_{1/\delta}(\mathbf{0}))} \leq C_q (1/\delta)^\beta,$$

Estimate:

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

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

2. Adjust the polynomial degree q to the knot set Y_μ
(as with the polynomial method).

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

Parameters: Polynomial degree q, RBF ϕ (p.d. or c.p.d. of order 1),
scaling coefficient δ, und tolerances $M_{\min}$, $M_{\max}$, κ_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 obtaned from data fitting

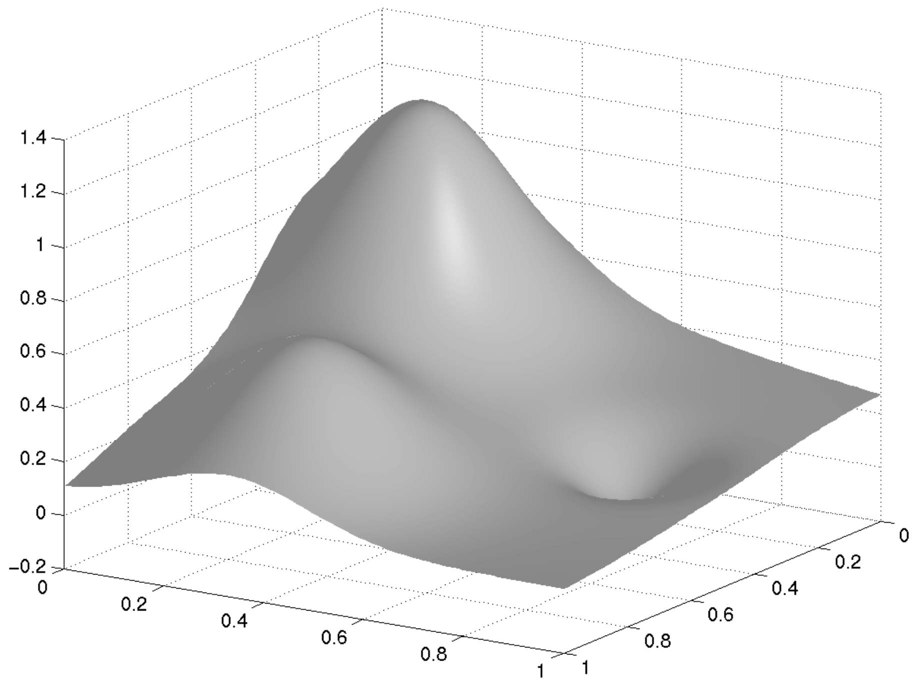
- Adaptive irregular meshes (multivariate counterpart of univarite free knot splines; use C^1 and C^2 polynomial splines on irregular triangulations and local refinement algorithms from FEM)

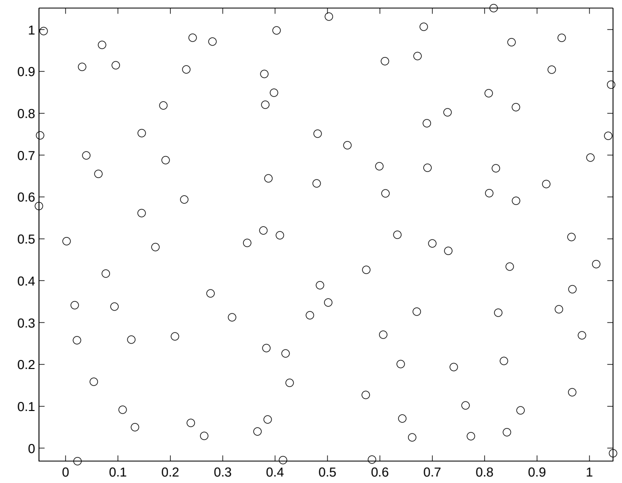
- Fitting functional data on manifolds (joint work with Larry Schumaker) and in higher dimensions

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

Numerical Examples

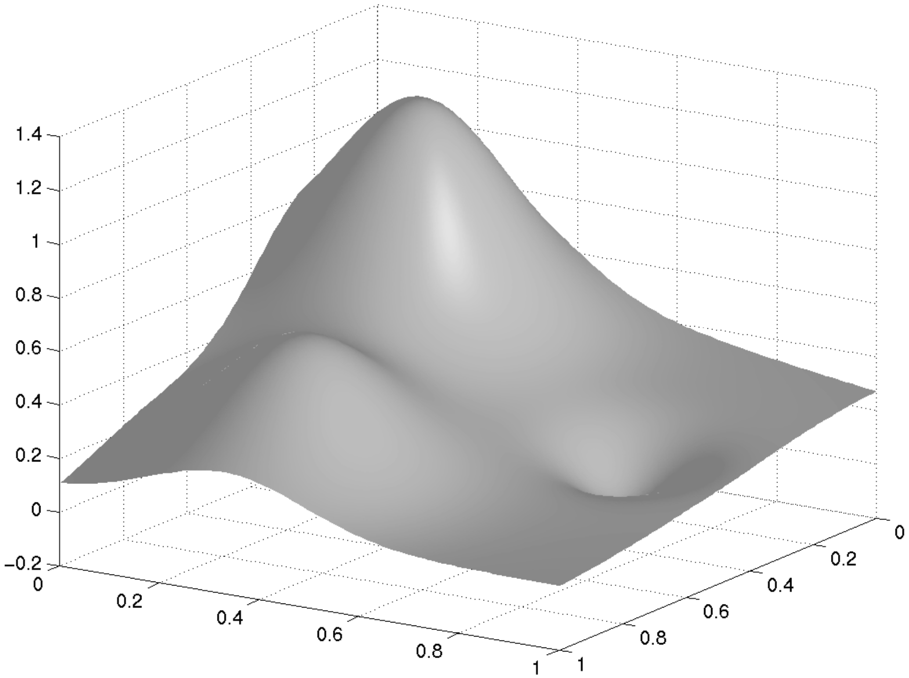
1. Recovery of Franke test function from its values at 100 points.
(The data `ds3` is available from `http://www.math.nps.navy.mil/~rfranke/`)



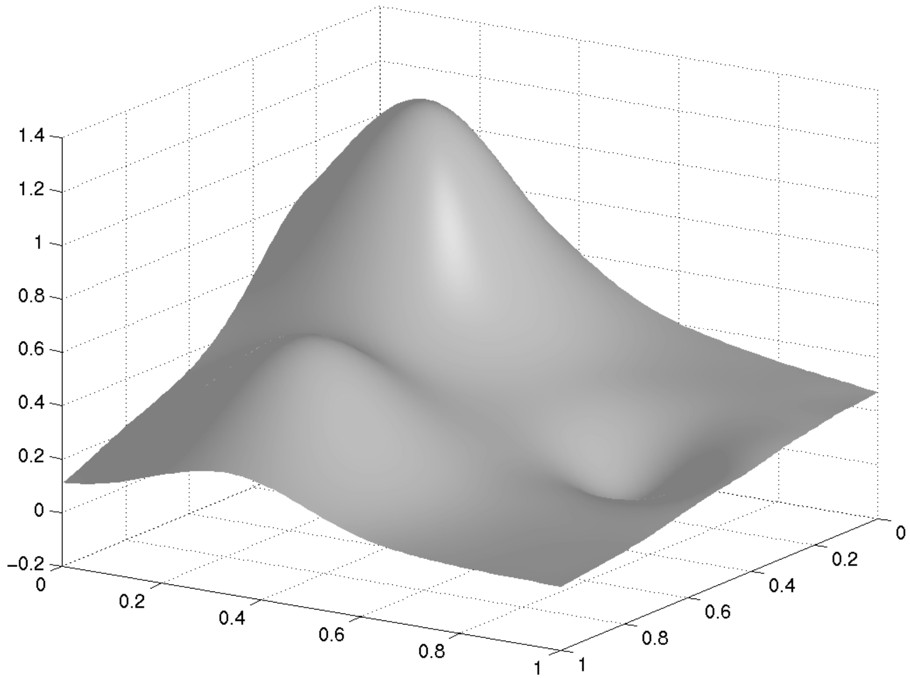


$$f(x,y) = \frac{3}{4}\exp\left[-\frac{(9x-2)^2+(9y-2)^2}{4}\right] + \frac{3}{4}\exp\left[-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}\right]$$
$$+\frac{1}{2}\exp\left[-\frac{(9x-7)^2+(9y-3)^2}{4}\right] - \frac{1}{5}\exp\left[-(9x-4)^2-(9y-7)^2\right].$$

Exact



Approximation (H_{MQ})



(H_{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 (5×5 square grid with both diagonals of the squares drawn in, $M_{min} = 16$, $M_{max} = 100$).

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

Global methods from Franke's 1979 report are denoted (G_{MQ}), ($\mathrm{G}_{\mathrm{IMQ}}$), etc.

Pure polynomial method is denoted by (P).

method	q	κ_H	δ	*max*	*mean*	*rms*	n_T^{aver}
$(\mathrm{H}_{\mathrm{MQ}})$	0	10^5	0.4	$1.6\ 10^{-2}$	$1.9\ 10^{-3}$	$3.0\ 10^{-3}$	21.1
$(\mathrm{H}_{\mathrm{IMQ}})$	0	10^4	0.5	$1.5\ 10^{-2}$	$2.0\ 10^{-3}$	$3.1\ 10^{-3}$	21.3
$(\mathrm{H}_{\mathrm{G}})$	0	10^4	0.4	$1.9\ 10^{-2}$	$2.2\ 10^{-3}$	$3.5\ 10^{-3}$	19.3
$(\mathrm{H}_{\mathrm{TP}})$	1	10^5	2.0	$5.7\ 10^{-2}$	$7.8\ 10^{-3}$	$1.3\ 10^{-2}$	20.7
$(\mathrm{H}_{\mathrm{TP3}})$	1	10^5	2.0	$4.7\ 10^{-2}$	$4.5\ 10^{-3}$	$7.5\ 10^{-3}$	20.3
$(\mathrm{H}_{\mathrm{TP4}})$	2	10^5	2.0	$3.0\ 10^{-2}$	$3.4\ 10^{-3}$	$5.5\ 10^{-3}$	14.9
$(\mathrm{H}_{\mathrm{TP5}})$	2	10^6	2.0	$2.8\ 10^{-2}$	$3.4\ 10^{-3}$	$5.2\ 10^{-3}$	13.4
$(\mathrm{H}_{\mathrm{W2}})$	0	10^4	2.0	$3.9\ 10^{-2}$	$4.1\ 10^{-3}$	$7.1\ 10^{-3}$	22.7
$(\mathrm{H}_{\mathrm{B3}})$	0	10^5	2.0	$3.3\ 10^{-2}$	$3.6\ 10^{-3}$	$6.0\ 10^{-3}$	22.4
$(\mathrm{H}_{\mathrm{W4}})$	0	10^4	2.0	$2.1\ 10^{-2}$	$2.1\ 10^{-3}$	$3.6\ 10^{-3}$	21.2
$(\mathrm{H}_{\mathrm{W6}})$	0	10^5	2.0	$1.6\ 10^{-2}$	$1.9\ 10^{-3}$	$3.0\ 10^{-3}$	20.9
(P)	6			$3.8\ 10^{-2}$	$5.2\ 10^{-3}$	$7.6\ 10^{-3}$	
$(\mathrm{G}_{\mathrm{MQ}})$				$2.3\ 10^{-2}$	$1.8\ 10^{-3}$	$3.6\ 10^{-3}$	
$(\mathrm{G}_{\mathrm{IMQ}})$				$2.5\ 10^{-2}$	$2.8\ 10^{-3}$	$5.2\ 10^{-3}$	
$(\mathrm{G}_{\mathrm{G}})$				$6.2\ 10^{-2}$	$6.0\ 10^{-3}$	$1.1\ 10^{-2}$	
$(\mathrm{G}_{\mathrm{TP}})$	1			$5.2\ 10^{-2}$	$5.3\ 10^{-3}$	$9.5\ 10^{-3}$	
$(\mathrm{G}_{\mathrm{TP3}})$	1			$2.5\ 10^{-2}$	$3.1\ 10^{-3}$	$5.8\ 10^{-3}$	

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

N	n_x	M_{min}	κ_H	δ	*max*	*mean*	*rms*	n_T^{aver}
10^2	5	16	10^5	0.4	$4.60\ 10^{-2}$	$3.98\ 10^{-3}$	$7.46\ 10^{-3}$	19
10^3	16	40	10^{12}	1.0	$1.69\ 10^{-4}$	$1.53\ 10^{-6}$	$6.47\ 10^{-6}$	47
10^4	50	40	10^{15}	1.6	$4.64\ 10^{-7}$	$5.62\ 10^{-9}$	$1.51\ 10^{-8}$	48

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

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

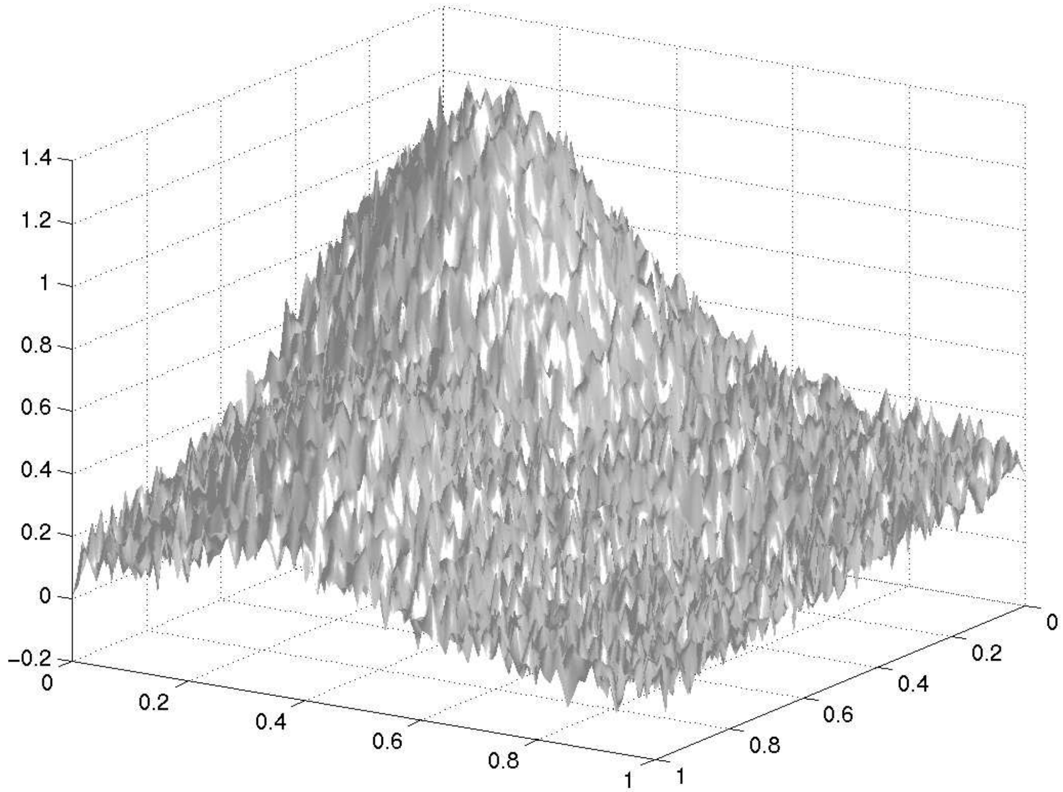
Table 3: Maximum error using the RBF interpolation method with multiquadric. Other parameters: $M_{min} = 20$ if $N = 10^2$ and $M_{min} = 100$ otherwise.

Thin plate splines with RBF interpolation method.

N	spline grid	max ($\beta = 3/2$)	max ($\beta = 7/4$)
10^2	5×5	$8.55 \cdot 10^{-2}$	$6.92 \cdot 10^{-2}$
10^3	16×16	$5.22 \cdot 10^{-3}$	$3.37 \cdot 10^{-3}$
10^4	50×50	$2.60 \cdot 10^{-4}$	$1.17 \cdot 10^{-4}$
10^5	158×158	$2.37 \cdot 10^{-5}$	$7.09 \cdot 10^{-6}$

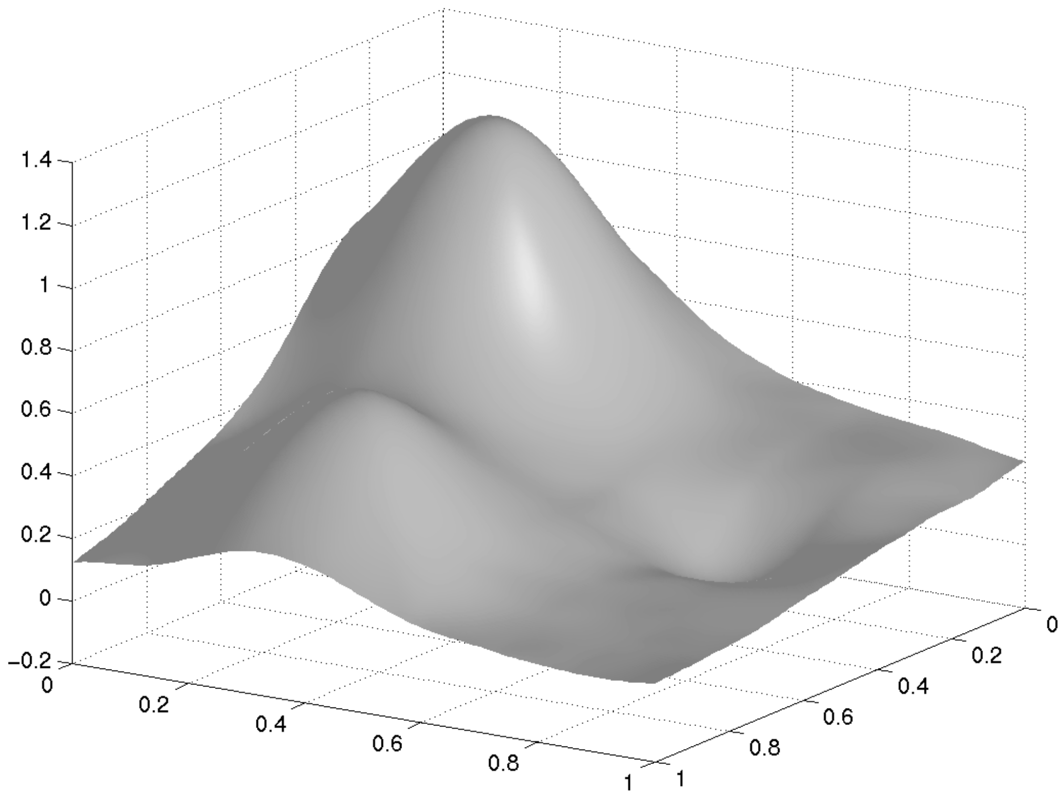
Table 4: Maximum error using the local RBF interpolation scheme based on $\phi(r) = r^\beta$, $\beta = 3/2$ and $7/4$. Parameter values: $\delta = 1$, $\tilde{q} = 0$, $M_{min} = S = 100$. Approximation order about $h^{\beta+1}$

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



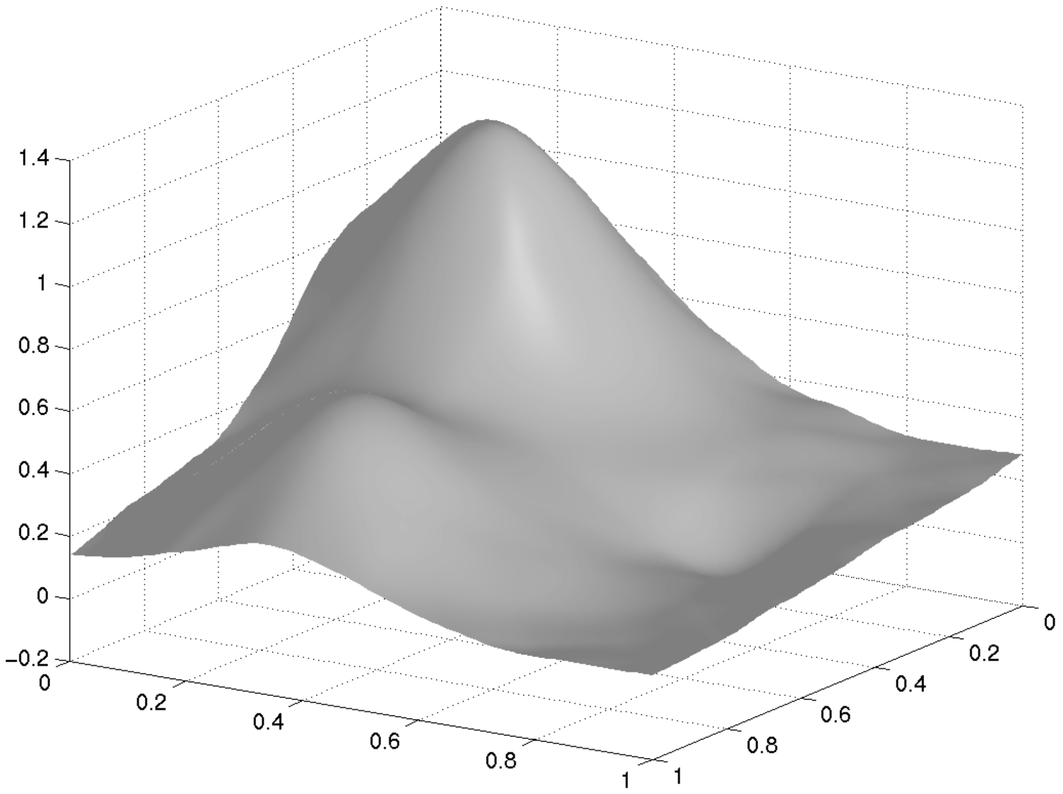
dim=304

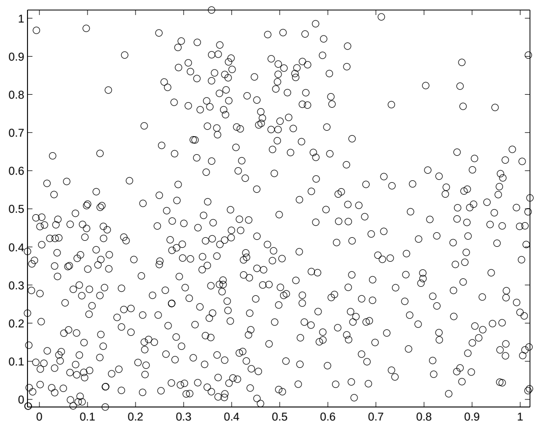
Error w.r.t. the original function: max=0.0274, mean=0.00415, rms=0.00552

Parameters: $\kappa = 1$, $M_{\max}$=300

Noise reduction: $\sigma/rms = 9.058$

Reconstruction of the Franke test function from 500 contaminated values by [McMahon & Franke, 1992] (standard deviation $\sigma = 0.05$)



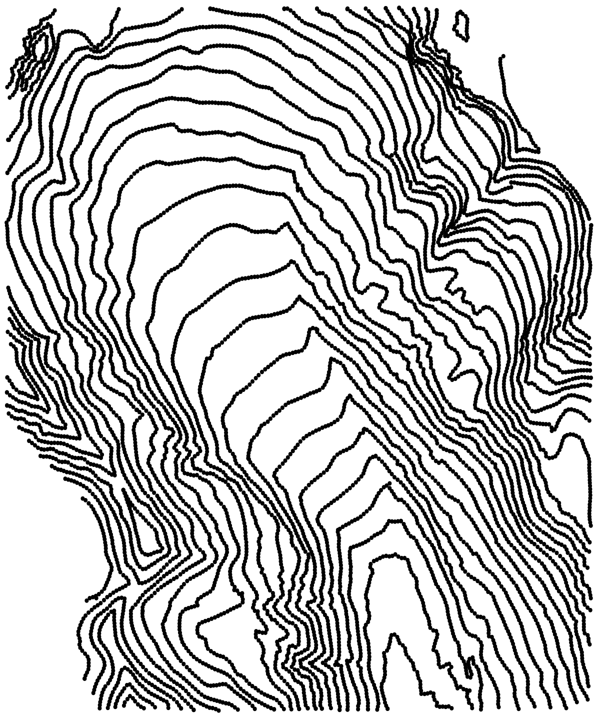


231 degrees of freedom, rms error: 0.0178

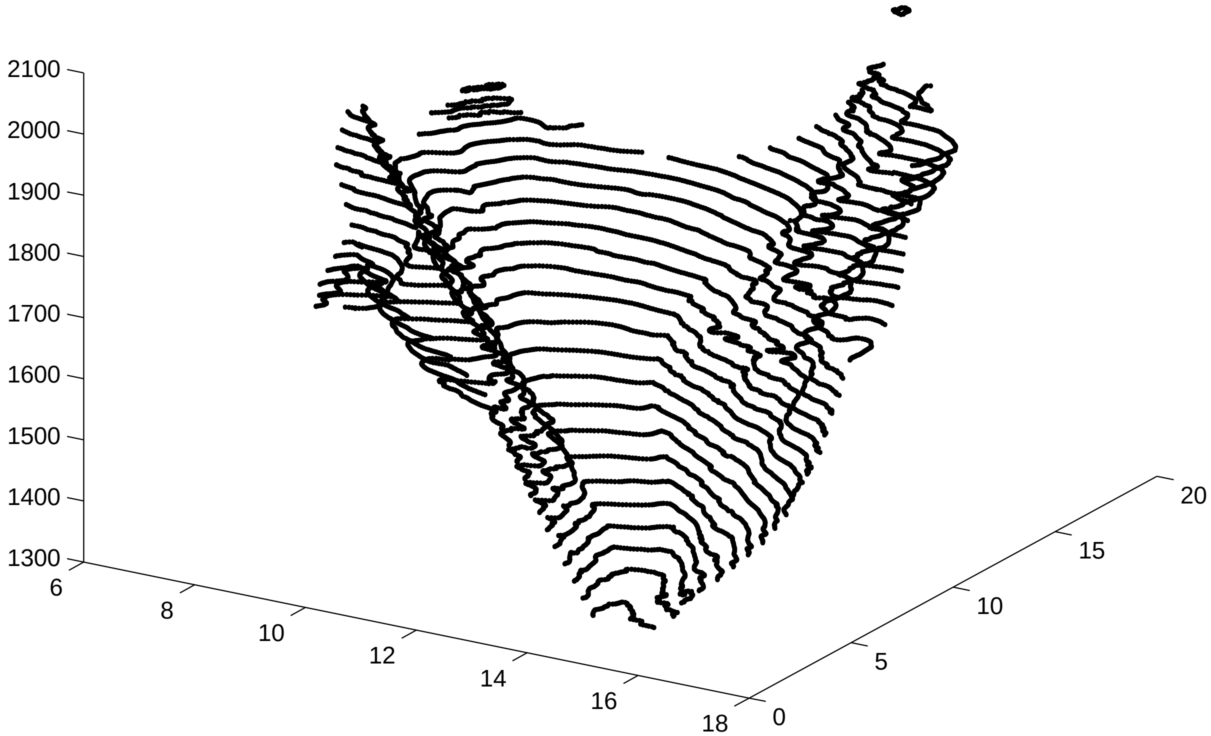
Noise reduction: $\sigma / rms = 2.81$

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).

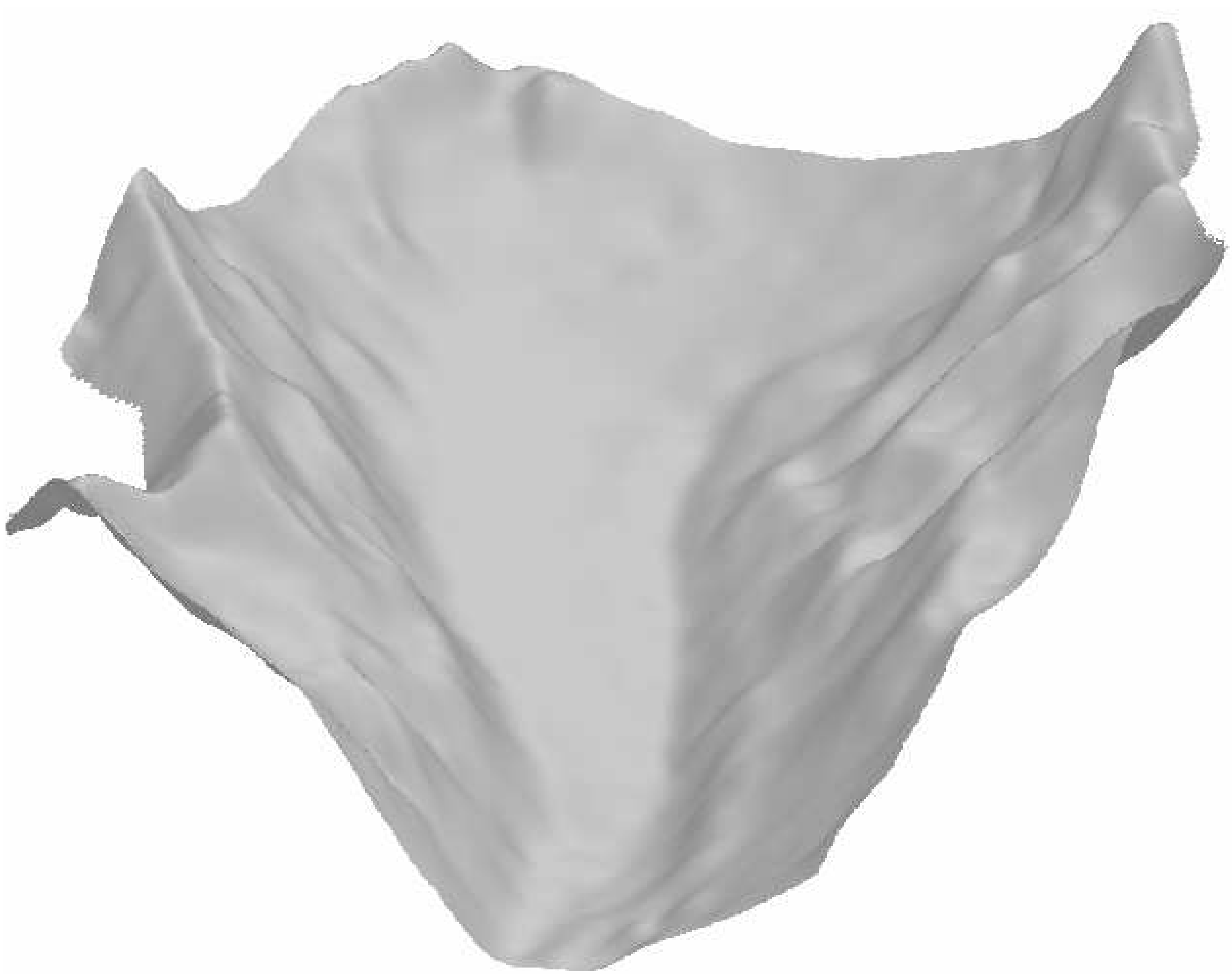
Location of data points



3D perspective view

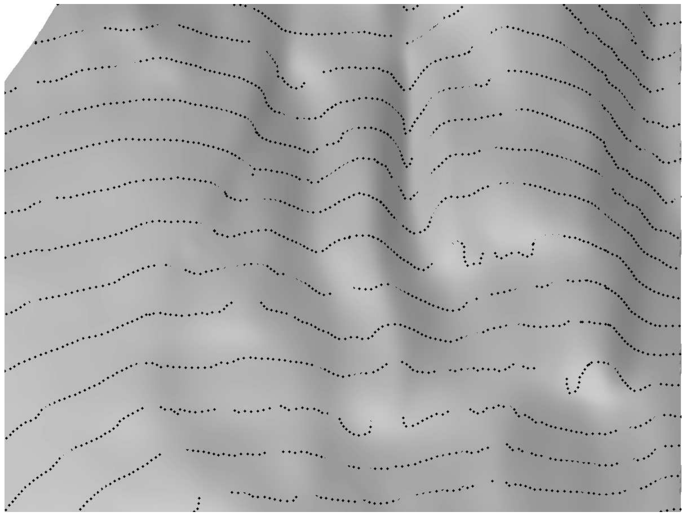


Glacier test: The C^2 spline approximation ($n_x = 20$, $n_y = 24$) with (H_{MQ}).

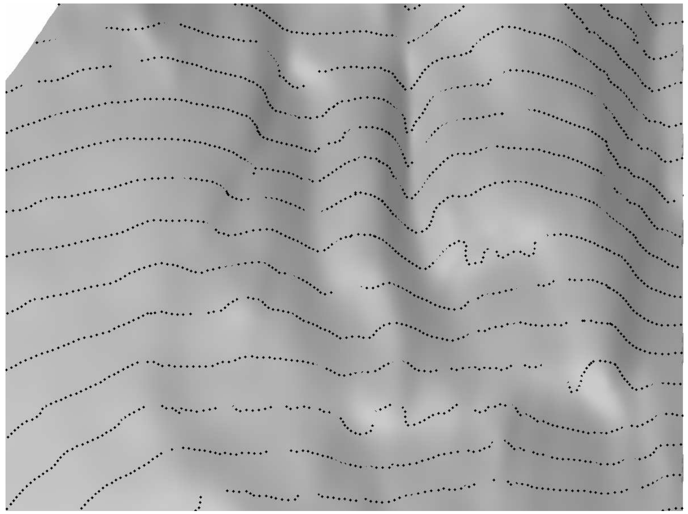


Zoom

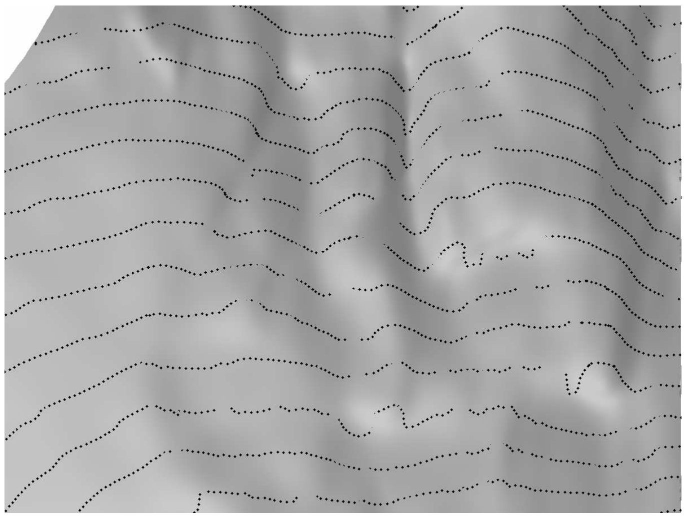
P (20×24)



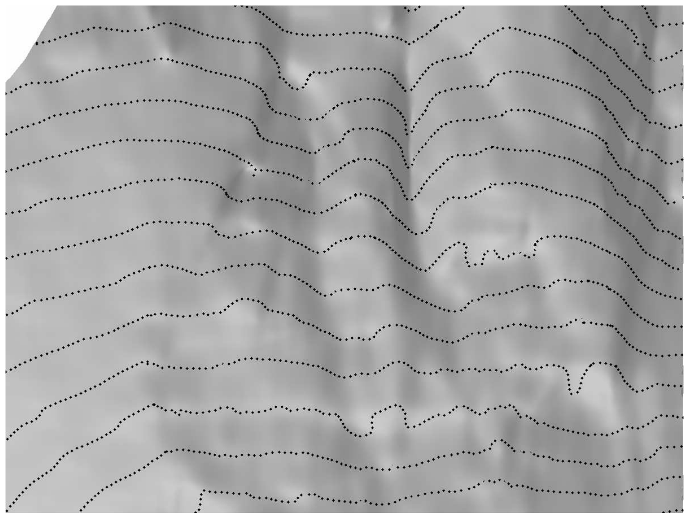
H_{MQ} (20×24)



R_{MQ} (20×24)

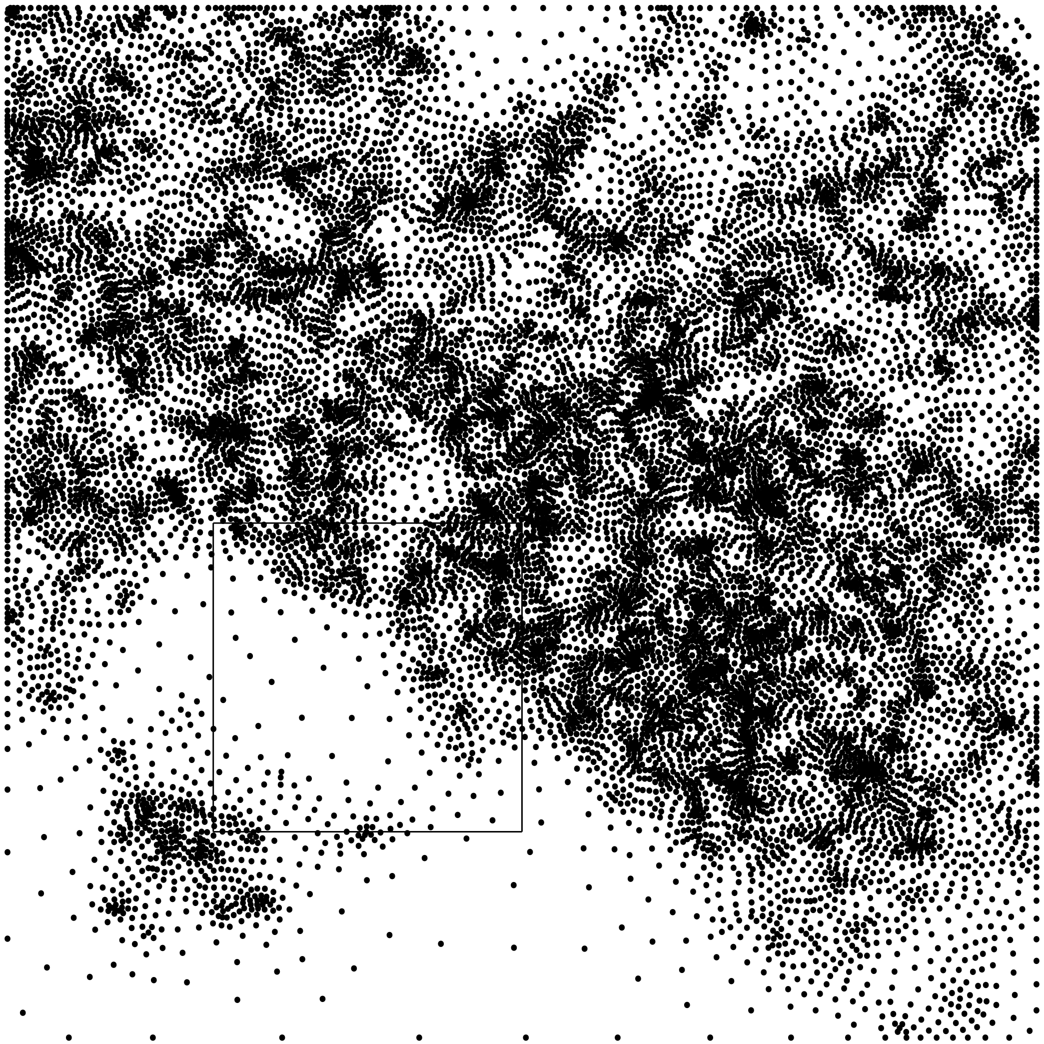


H_{MQ} (40×48)

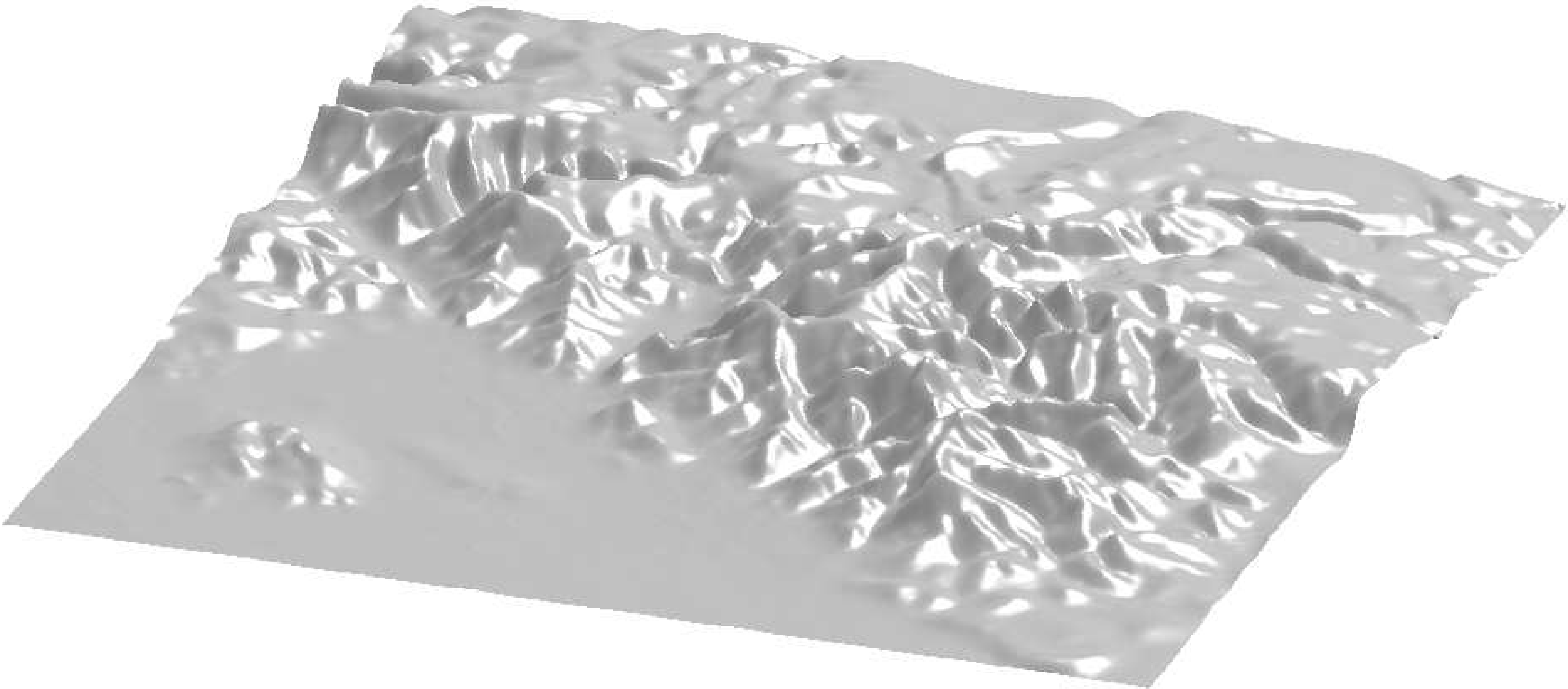


5. Black Forest data

15,885 data points from an area of 144 km^2
(elevation range 1200 m)

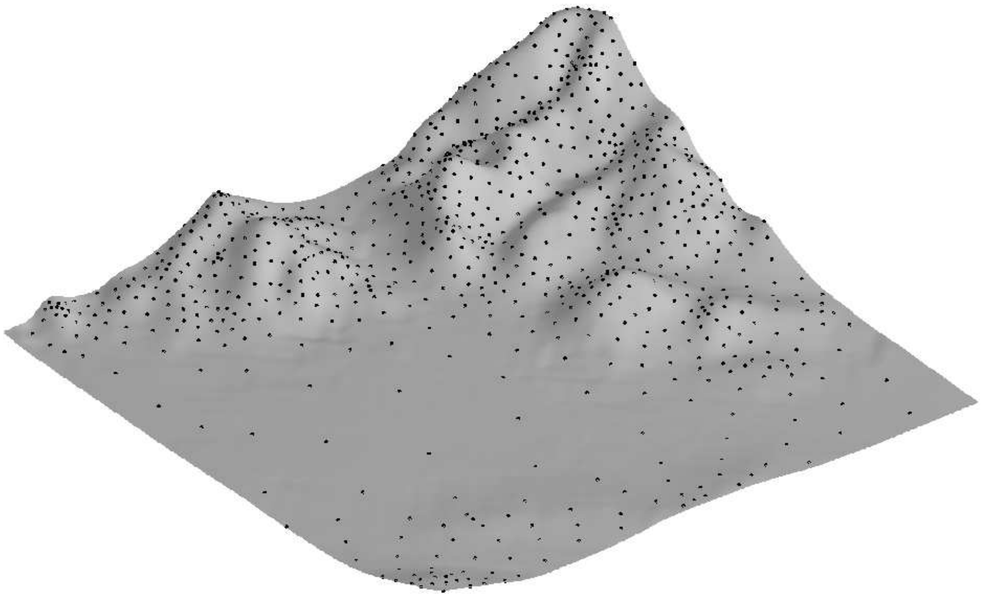


The C^2 spline approximation ($n_x = n_y = 80$) with (H_{MQ})

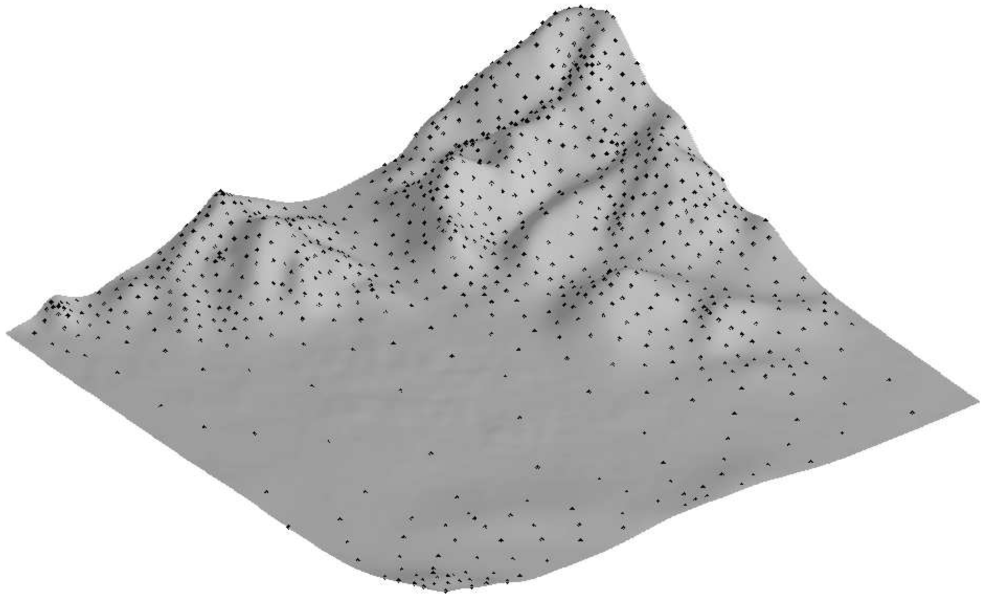


Zoom

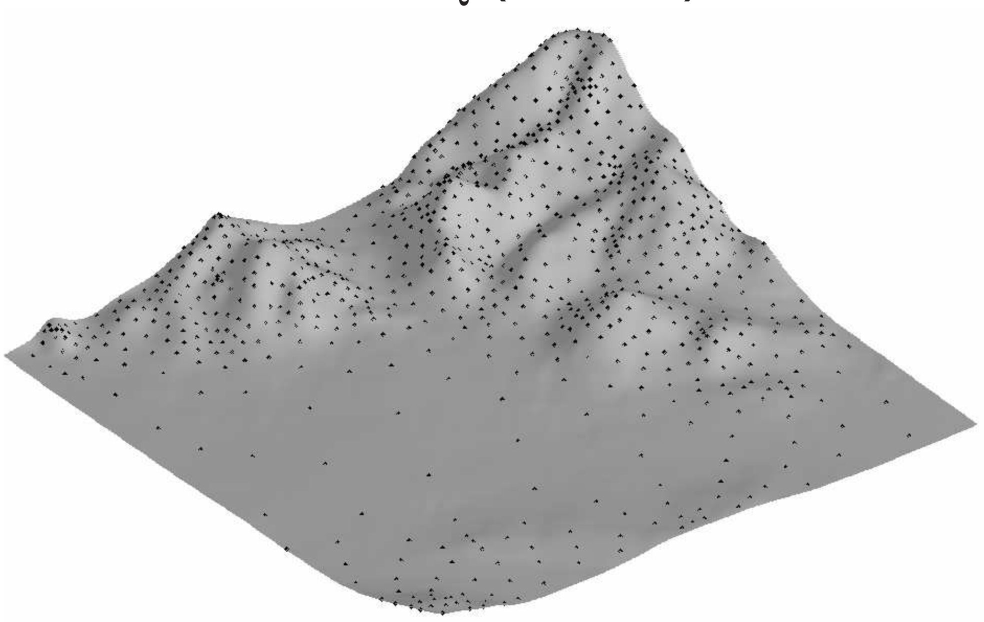
P (80×80)



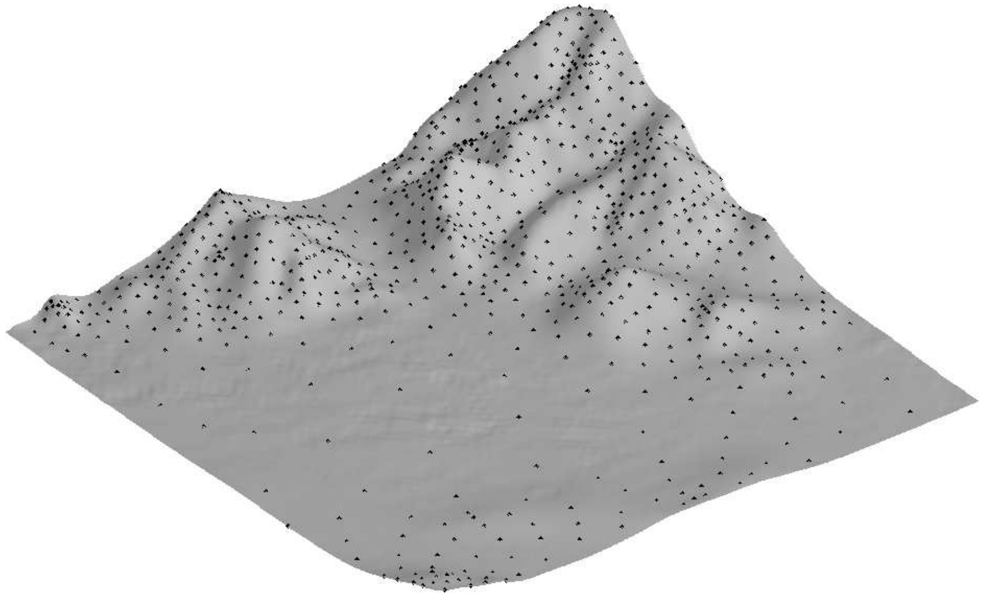
H_{MQ} (80×80)



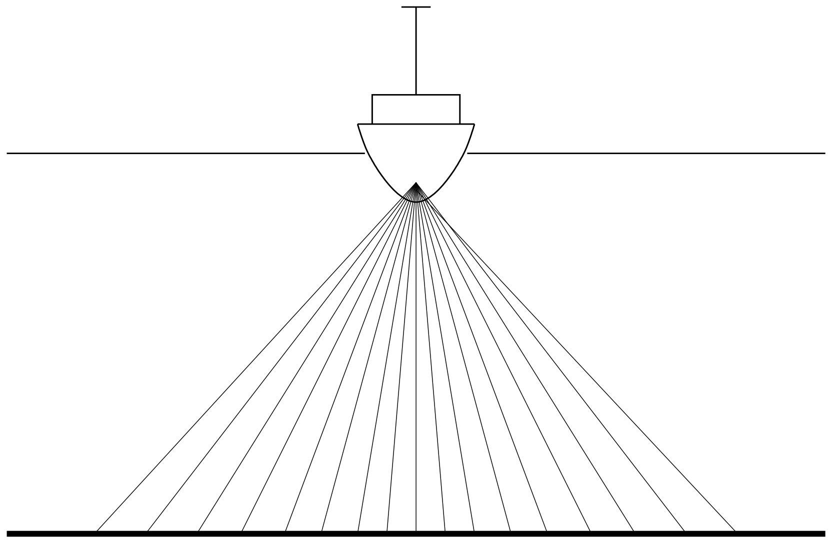
R_{MQ} (80×80)

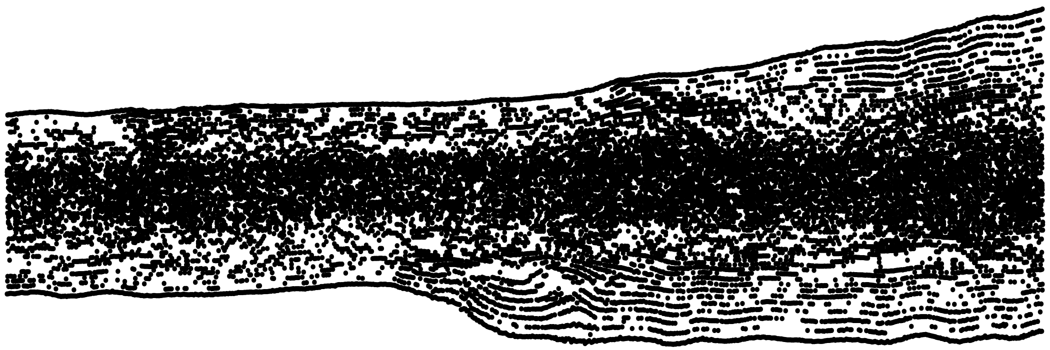


H_{MQ} (160×160)

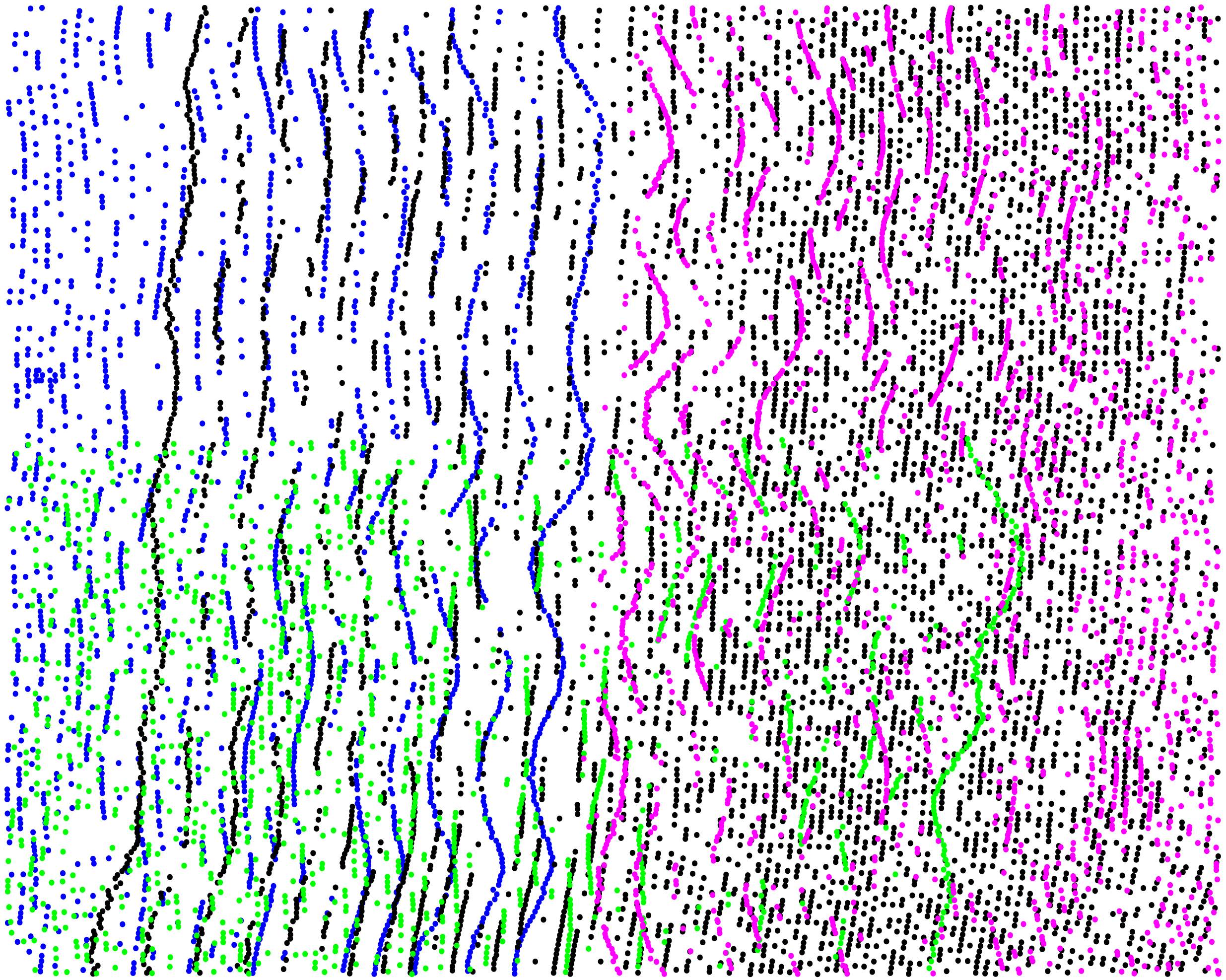


6. Rotterdam Port multibeam echosounder data 634,604 raw data points (courtesy Quality Positioning Services, Zeist, Holland).





Zoom to the local distribution of the xy-points

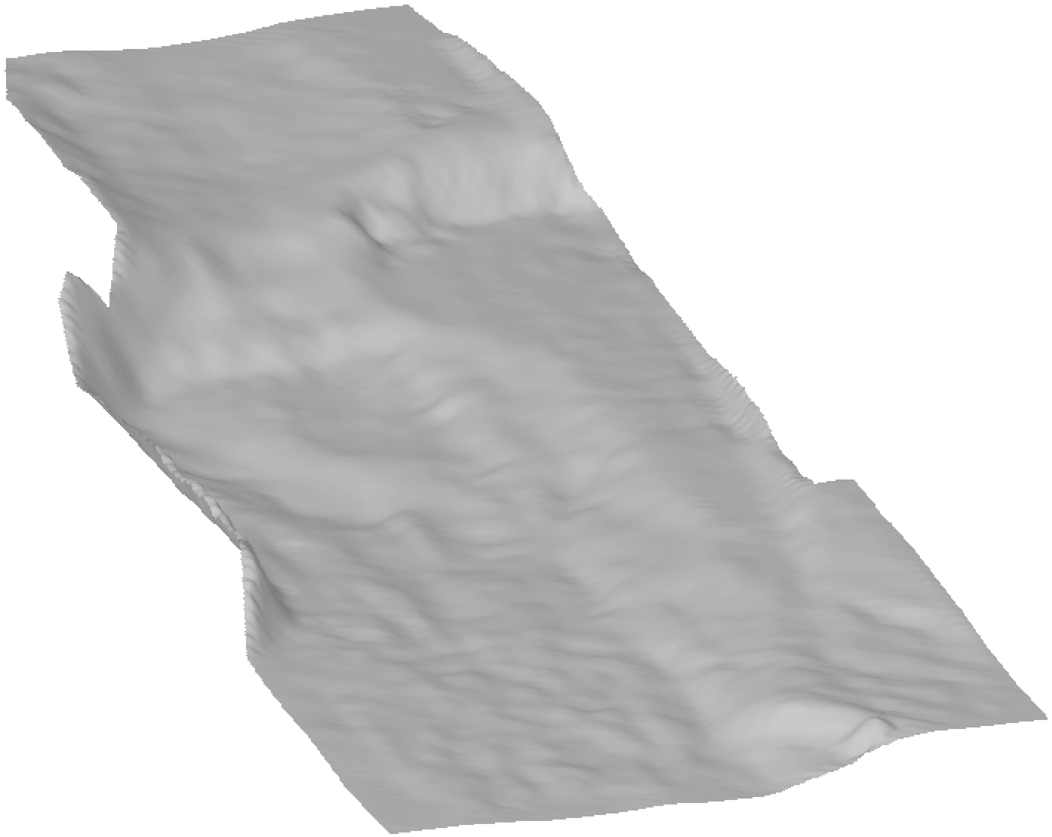


Special features of multibeam data:

- **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.

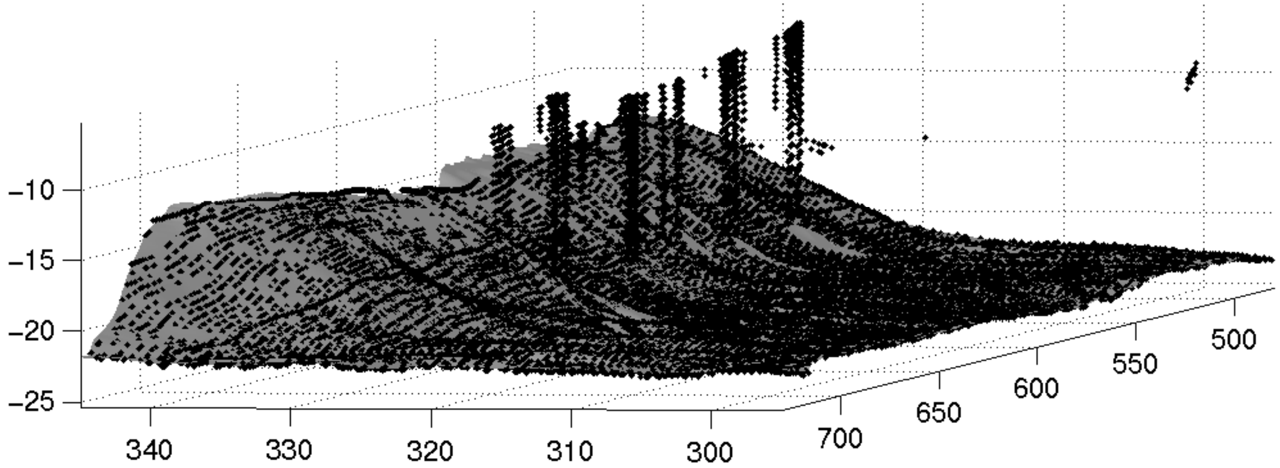
Processing of the Rotterdam Port data

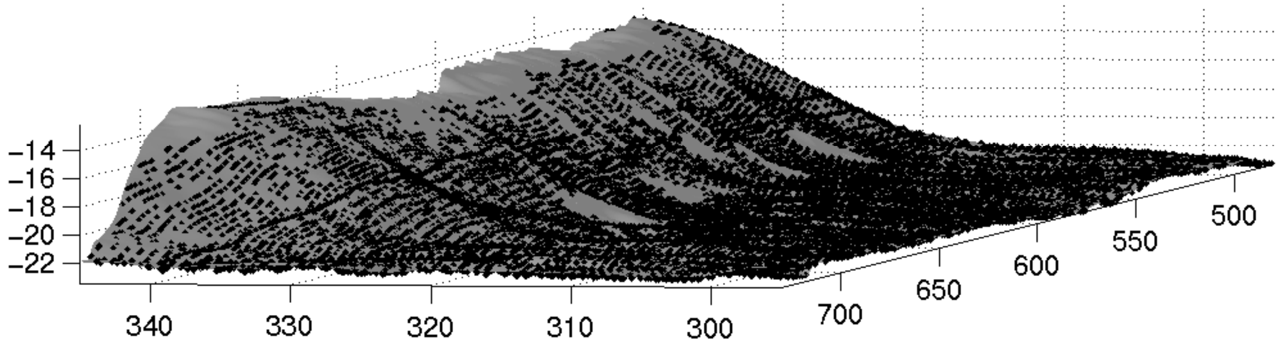
1) A coarse approximation (22,399 degrees of freedom, 23.8 s computational time, rms error 0.61 m):



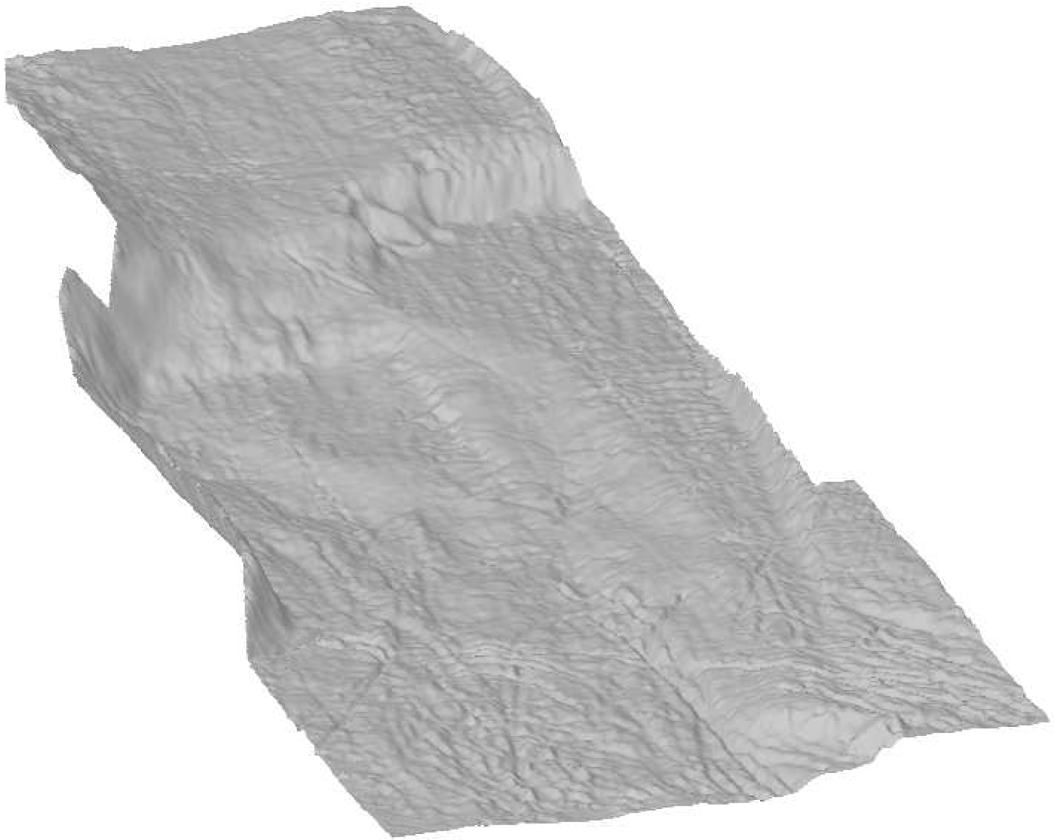
Vertically exaggerated (4×). The z-values of the data: -27.6 m to -5.3 m.

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):



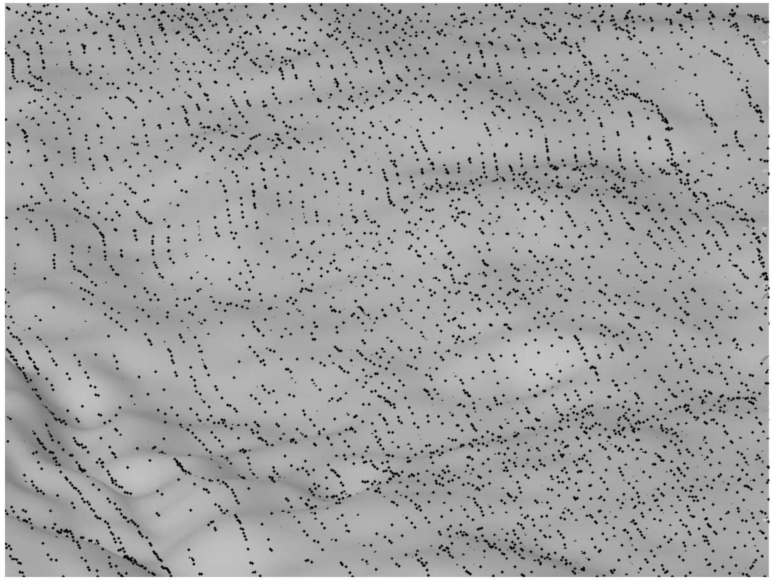
View from the above

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

Zoom

(P)

$(\mathrm{H_{MQ}})$



$(\mathrm{R_{MQ}})$