Scalable Spline Algorithms for the Approximation of Large and Noisy Scattered Data

Oleg Davydov
Justus–Liebig University Giessen
Germany

Foundations of Computational Mathematics

Minneapolis, 5–14 August 2002

JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

Scattered Data Problem

$\Omega \subset \mathbb{R}^d$	bounded domain ($d > 1$)
$\Xi = \{\xi_i\}_{i=1}^N \subset \Omega$	arbitrarily distributed points in Ω
$\{z_i\}_{i=1}^N \subset \mathbb{R}$	known values of $f : \Omega \to \mathbb{R}$

Find $s : \Omega \to \mathbb{R}$ that approximates f on Ω

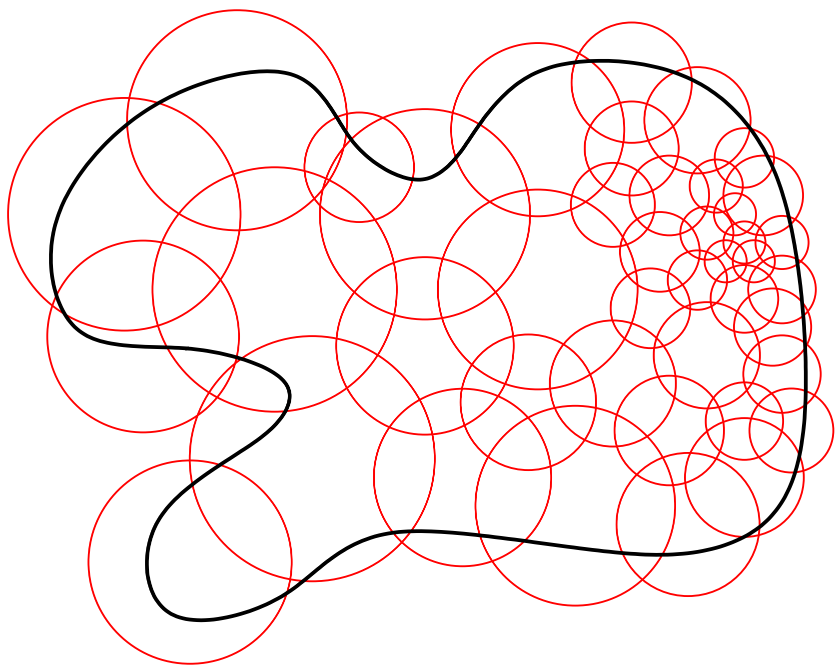
Desirable features of a scattered data algorithm:

- Good approximation quality
- No artefacts in s such as oscillations or ridges (not present in f)
- Usable s: fast evaluation, etc.
- Robustness w.r.t. noise in $\{z_i\}_{i=1}^N$
- Scalability: Linear time and cost w.r.t. N

Two-Stage Methods

Let $\omega \subset \Omega$ be a “small subdomain”.
The values $f(x)$, $x \in \omega$, do not have much to do with $z_i \approx f(\xi_i)$ for ξ_i situated far away from ω.

Stage 1: Cover Ω with a number of overlapping subdomains ω_μ, $\mu \in \mathcal{M}$, and compute suitable local approximations $p_\mu : \omega_\mu \to \mathbb{R}$ for all $\mu \in \mathcal{M}$.



Stage 2: “Blend” the local approximations p_μ, $\mu \in \mathcal{M}$, together to a smooth (say, C^1 or C^2) function $s : \Omega \to \mathbb{R}$.

Two-Stage Methods Based on Splines

Several versions of the two-stage methods have been developed since 1970th.

Surveys on scattered data fitting:

Schumaker 1976
Barnhill 1977
Lawson 1977
Franke 1982
Alfeld 1989
Franke & Nielson 1991
Fasshauer & Schumaker 1998

A locally supported polynomial spline basis $\{B_j\}_{j=1}^D$ can be used in Stage 2 (**Schumaker 1976**):

$$s = \sum_{j=1}^{D} \lambda_j(p_\mu) B_j,$$

where $\{\lambda_j\}_{j=1}^D$ are **dual functionals**, i.e.

$$\lambda_i(s_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

What Spline Bases?

Possible candidates are **any smooth, locally supported, piecewise polynomial bases**:

- tensor products of univariate B-splines
- box splines
- simplex splines
- C^1 finite element bases
- stable local spline bases on triangulations (nodal or Bernstein-Bézier versions)

(All of these have been actually studied in the context of scattered data fitting.)

Two points to be taken into account:

1. Approximation power of the spline space, which essentially amounts to **(Riesz) stability** and **degree of polynomial reproduction**.

2. Properties of the dual functionals λ_j: It is e.g. easier to work with Lagrange functionals (point evaluations) than with functionals involving first or second order derivatives.

Advantages of Polynomial Splines

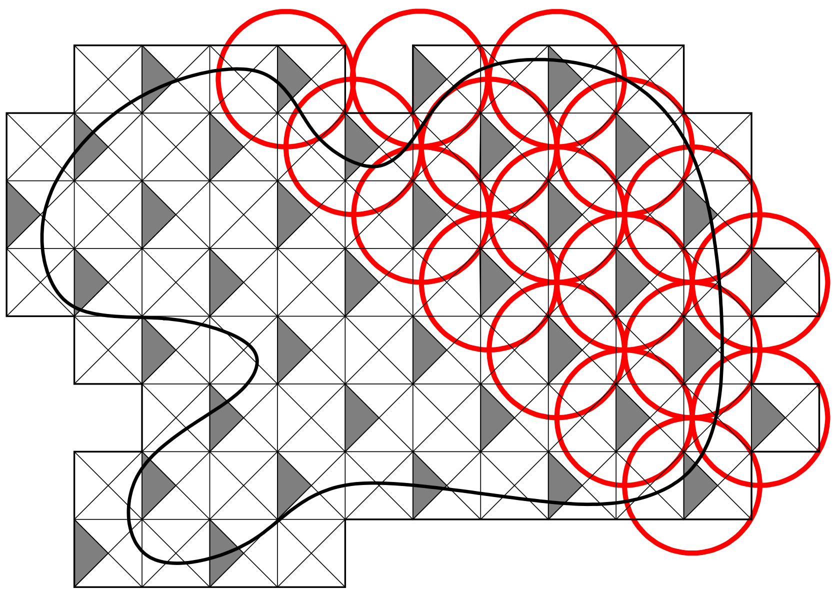
- **Scalable** algorithms for scattered data fitting thanks to the **locality** of the basis functions
- **Usability:** Fast and stable evaluation by using **recurrent relations** for basis splines (box splines, simplex splines), or **Bernstein-Bézier techniques**
- **Approximation power** is essentially determined by the approximation properties of the **local approximations** p_μ
- **Splines do not produce artefacts unless the local approximations are bad**
- **If the local approximations are resistant to noise, than so are the resulting spline surfaces**

Quality of local approximations is decisive!

Our Bases

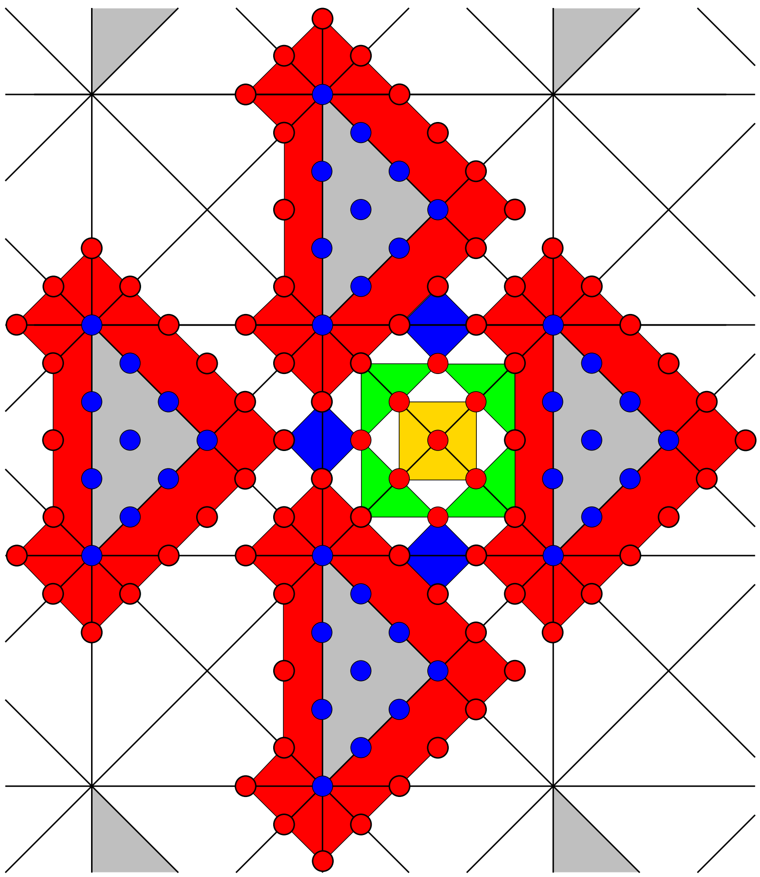
D. & Zeilfelder, preprint

- C^1 **piecewise cubics** or C^2 **piecewise sextics** on a four-directional mesh in $\mathbb{R}^2$
- In the C^1 case a modification of the FVS finite element bases; related to piecewise cubic local Lagrange bases by **Nürnberger, Schumaker & Zeilfelder, 2000**
- **Dual functionals:** Bernstein-Bézier coefficients corresponding to domain points completely filling certain triangles



Computation of Bernstein-Bézier coefficients by extension of local polynomials using smoothness conditions

C^1 cubics:



Approximation Error

$$q = \begin{cases} 3 & \text{for } C^1 \\ 6 & \text{for } C^2 \end{cases}$$

h the gridsize

T_μ the triangle corresponding to ω_μ

p_μ the local polynomials approximation on ω_μ

Assume:
$z_i = f(\xi_i)$, $i = 1, \dots, N$,
$f \in W_p^{q+1}(\Omega)$ for some $1 \le p \le \infty$

Then:

$$\|f - s\|_{L_p(\Omega)} \le C_1 h^{q+1} |f|_{W_p^{q+1}(\Omega)} + C_2 \Big(\sum_{\mu \in \mathcal{M}} \|f - p_\mu\|^p_{L_p(T_\mu)} \Big)^{1/p}, \quad 1 \le p < \infty,$$

$$\|f - s\|_{L_\infty(\Omega)} \le C_1 h^{q+1} |f|_{W_\infty^{q+1}(\Omega)} + C_2 \max_{\mu \in \mathcal{M}} \|f - p_\mu\|_{L_\infty(T_\mu)}.$$

Local Approximations

$\Xi_\mu := \Xi \cap \omega_\mu$ (local portion of data)

p_μ: **least squares** polynomial of total degree $\leq q_\mu$

$$\sum_{\xi_i \in \Xi_\mu} |z_i - p_\mu(\xi_i)|^2 = \min_{\deg(p) \leq q_\mu} \sum_{\xi_i \in \Xi_\mu} |z_i - p(\xi_i)|^2,$$

where q_μ is the **greatest acceptable degree**.

We start by examining the possibility to choose $q_\mu = 3$, resp. $q_\mu = 6$. If it does not work, we successively **drop the degree** by one, until we find an acceptable value for q_μ ($q_\mu = 0$ is always acceptable, but poor).

Philosophy: Ξ_μ may contain too little information for a higher degree polynomial approximation. (Example: a lot of points close to a circle are not good for the approximation with quadratic polynomials.)

The strategy of dropping the degree has shown (in our tests) a much better performance than that of looking for additional points

Acceptable Degree

$\{b_1, \dots, b_m\}$: **Bernstein polynomial basis** w.r.t. the triangle T_μ for the polynomials of degree q ($m = \binom{q+2}{2}$)

$M := [b_j(\xi_i)]_{j=1,\dots,m,\ \xi_i \in \Xi_\mu}$: **collocation matrix**

$\sigma_{\min}(M)$: the **minimal singular value** of M;

Degree q is **acceptable** if

$$1/\sigma_{\min}(M) \leq \kappa,$$

where κ is a tolerance value.

Best values of κ are very low (between 1 and 5) for the real world data in our tests.
If κ is high, we get surfaces with artefacts, especially for "complicated" types of data: track data, noisy data, data with high variations of density in xy-plane.

Why is κ Significant?

κ has a direct influence on the approximation power of the least squares polynomial p_μ

Let $L(f)$ be the least squares polynomial of degree q_μ computed from the values $z_i = f(\xi_i)$, $\xi_i \in \Xi_i$. Then $L : C(\omega_\mu) \to C(\omega_\mu)$ is a linear operator.

We have

$$K_1/\sigma_{\min}(M) \le \|L\|_{C\to C} \le K_2\sqrt{\#\Xi_\mu}/\sigma_{\min}(M),$$

in particular,

$$\|L\|_{C\to C} \le K_2\sqrt{\#\Xi_\mu}\,\kappa,$$

where K_1, K_2 are the stability constants of the Bernstein basis in the following sense:

$$K_1\|c\|_2 \le \|\sum_{j=1}^{m} c_j b_j\|_{C(\omega_\mu)} \le K_2\|c\|_2.$$

Since the operator L exactly reproduces polynomials of degree q_μ, we have

$$\begin{aligned}\|f - L(f)\|_{L_\infty(\omega_\mu)} &\leq (1 + \|L\|_{C\to C}) E_{q_\mu}(f, \omega_\mu) \\ &\leq (1 + K_2 \sqrt{\#\Xi_\mu}\,\kappa) E_{q_\mu}(f, \omega_\mu)\end{aligned}$$

where

$$E_{q_\mu}(f, \omega_\mu) := \inf_{\deg(p) \leq q_\mu} \|f - p\|_{L_\infty(\omega_\mu)}$$

is the best approximation of $f|_{\omega_\mu}$ by polynomials of degree q_μ.

It is well known that

$$E_{q_\mu}(f, \omega_\mu) \leq C\, \mathrm{diam}(\omega_\mu)^{q_\mu + 1} |f|_{W_\infty^{q_\mu+1}(\omega_\mu)}$$

C is an absolute constant: Recall that ω_μ is a circle, and q_μ does not exceed 3, resp. 6.

Thus, we get the following **estimate for the overall approximation quality**:

$$\|f-s\|_{L_\infty(\Omega)} \le C_1 h^{q+1} |f|_{W_\infty^{q+1}(\Omega)} + C_2 \max_{\mu\in\mathcal{M}} \|f-p_\mu\|_{L_\infty(T_\mu)}.$$

$$\le C_1 h^{q+1} |f|_{W_\infty^{q+1}(\Omega)}$$

$$+ C_3 \max_{\mu\in\mathcal{M}} h^{q_\mu+1} d_\mu^{q_\mu+1} (1+K_2 \sqrt{\#\Xi_\mu}\,\kappa)\, |f|_{W_\infty^{q_\mu+1}(\omega\mu)}$$

where

$$d_\mu := \frac{\text{diam}(\omega_\mu)}{\text{diam}(T_\mu)}.$$

(In most cases d_μ is bounded.)

As before,

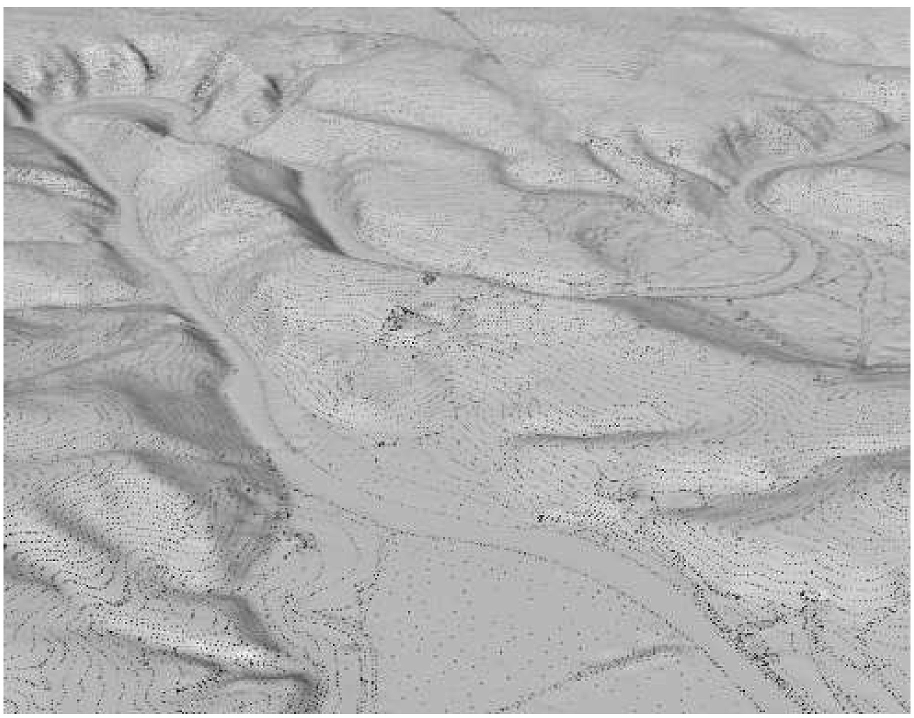
$$q = \begin{cases} 3 & \text{for } C^1, \\ 6 & \text{for } C^2, \end{cases}$$

and

$$0 \le q_\mu \le q.$$

Visualization and Rendering of Large Digital Terrain Models

The usability and efficiency of the C^1 method in the context of interactive visualization and rendering of large terrain data has been demonstrated by **Haber, Zeilfelder, D. & Seidel, 2001**, where real-time frame rates for typical fly-through sequences are achieved.



The C^1 method was implemented within the scope of the visualization project. The implementation of the C^2 method in **D. & Zeilfelder** follows similar ideas.

Acknowledgement

The picture of the previous page was generated from data used by the permission of the Landesamt für Kataster-, Vermessungs- und Kartenwesen des Saarlands under license numbers G-07-00 (9/26/00) and D-90/2000 (10/17/2000).

Ongoing and Future Work

Adaptive local approximations

Multiresolution

Nonlinear approximation

Adaptive meshes

Real world applications

Literature

O. Davydov and F. Zeilfelder, Scattered data fitting by direct extension of local polynomials with bivariate splines, preprint (includes a number of **numerical examples** with well known benchmark data sets and with large “real world” data)

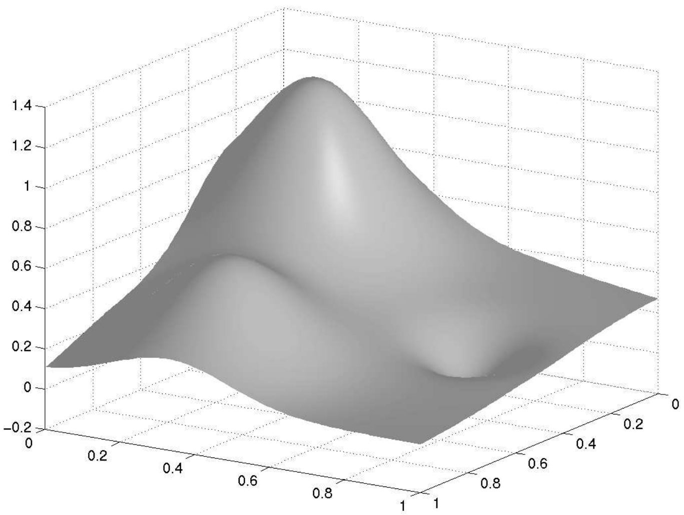
J. Haber, F. Zeilfelder, O. Davydov, and H.-P. Seidel, Smooth approximation and rendering of large scattered data sets, in Proceedings of IEEE Visualization 2001.

O. Davydov, On the approximation power of local least squares polynomials, to appear in Proc. ”Algorithms for Approximation IV.”

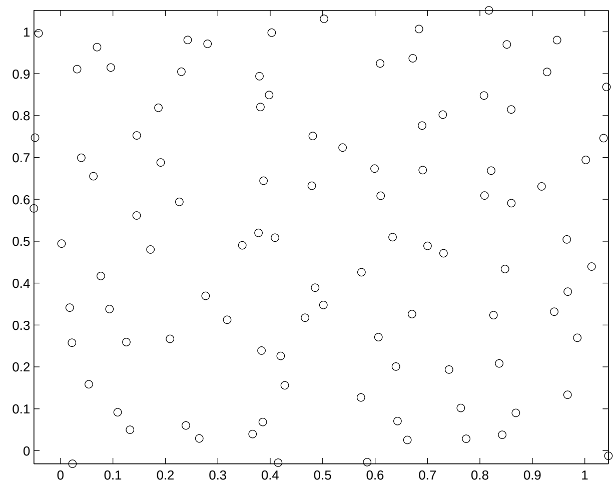
Preprint versions are available from
http://www.uni-giessen.de/~gcn5/davydov/

Numerical Examples

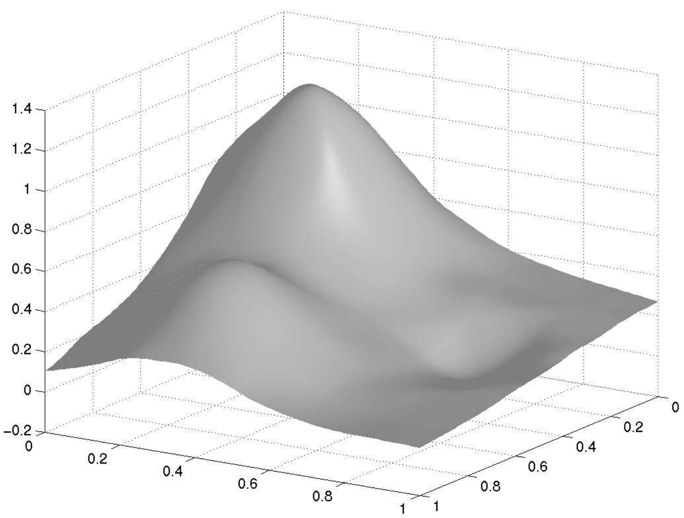
Franke Test Function



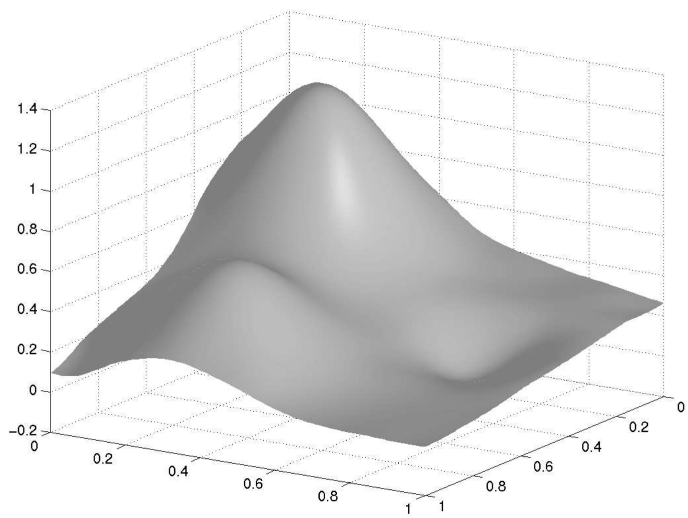
Original function



100 Points by R. Franke

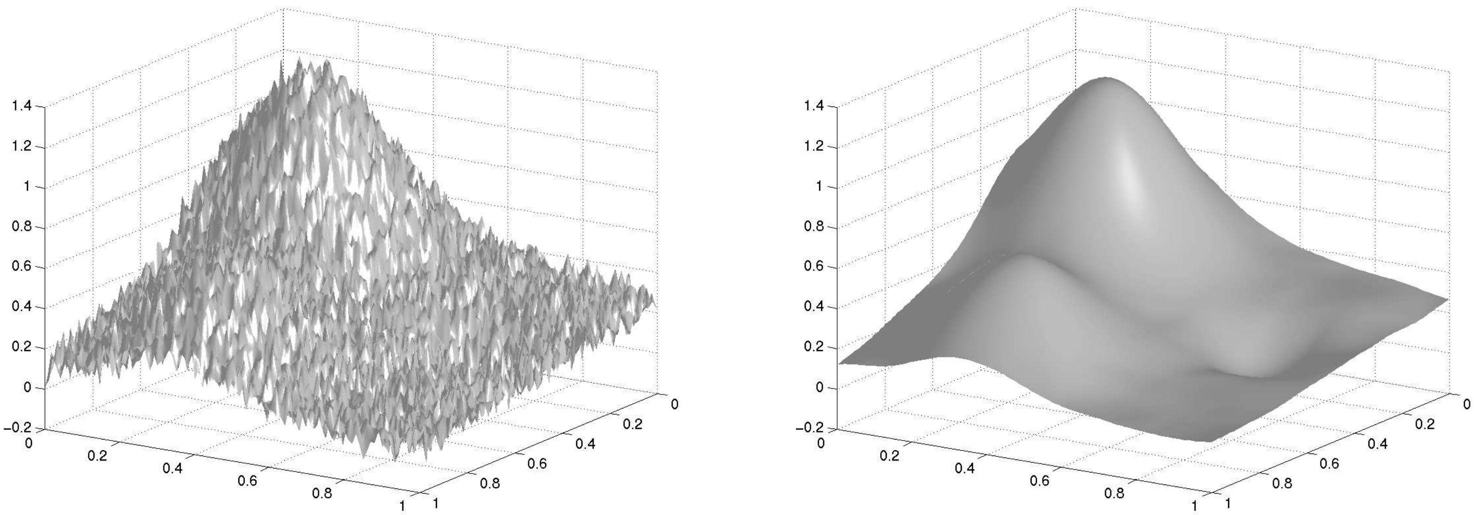


C^1 spline
231 degrees of freedom
max $= 0.0434$
mean $= 0.00704$
rms $= 0.0101$



C^2 spline
476 degrees of freedom
max $= 0.0355$
mean $= 0.00515$
rms $= 0.00761$

Denoising



(a) (b)

(a) Franke test function with normally distributed random errors on the 100x100 grid (standard deviation of the noise $\sigma = 0.05$)

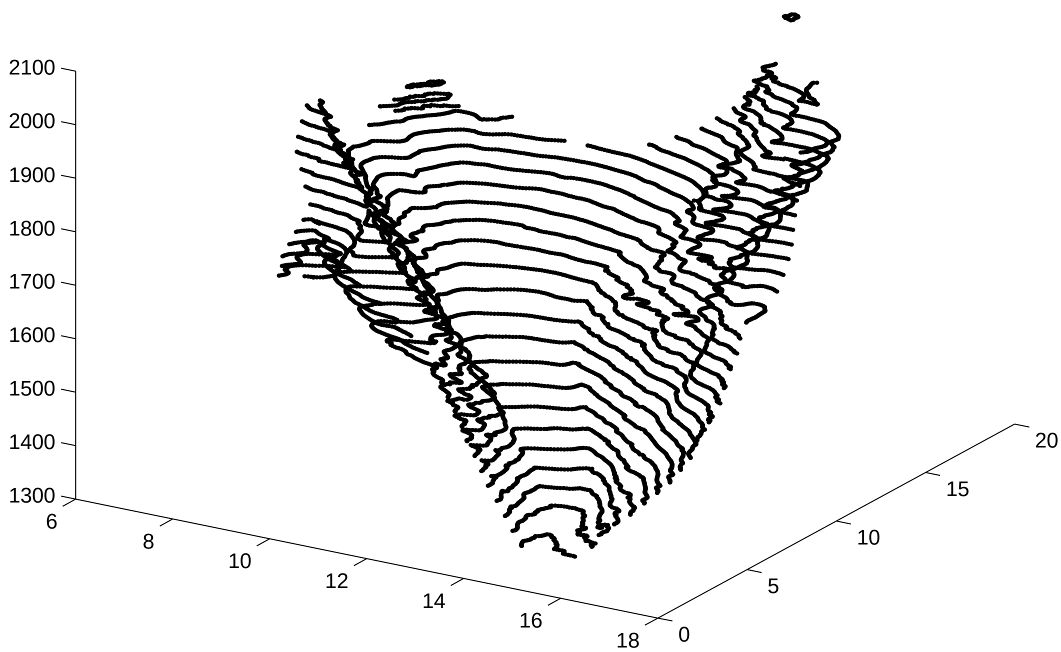
(b) C^1 Spline reconstruction (dim=304)

Error to the original function:
max=0.0274, mean=0.00415, rms=0.00552

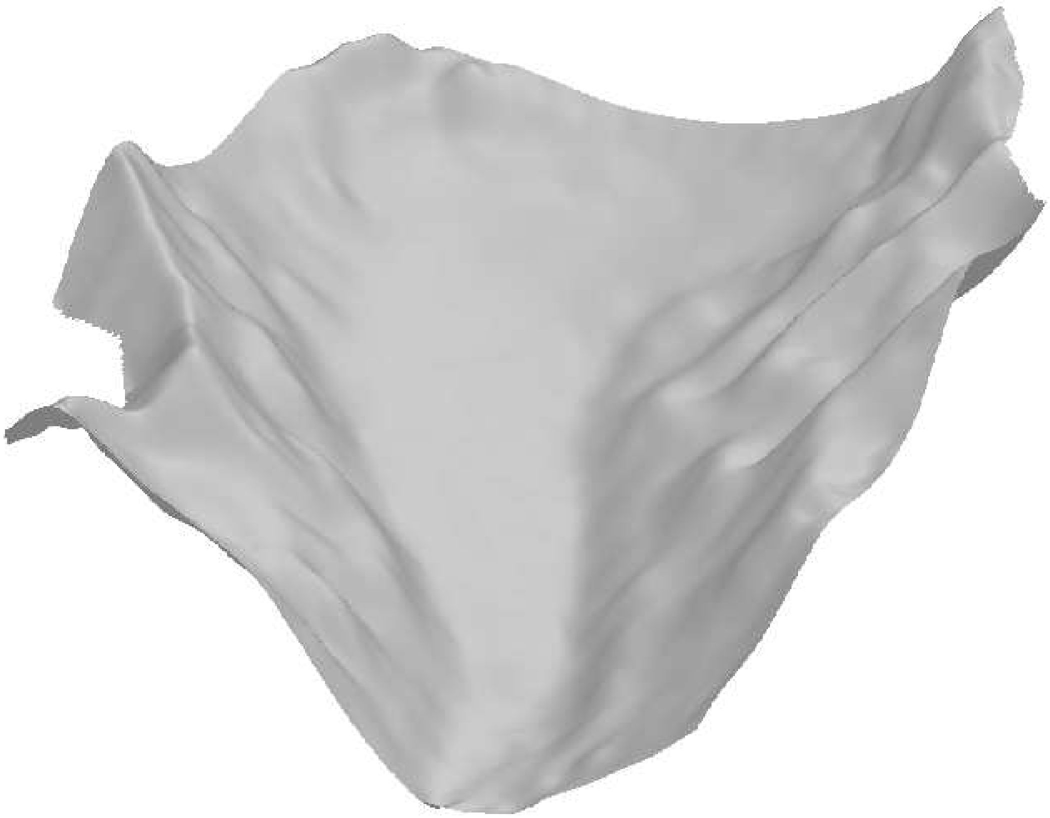
Cputime 3.04 sec

Glacier

8,345 points (available from the homepage of R. Franke): 44 digitized height contours of a glacier



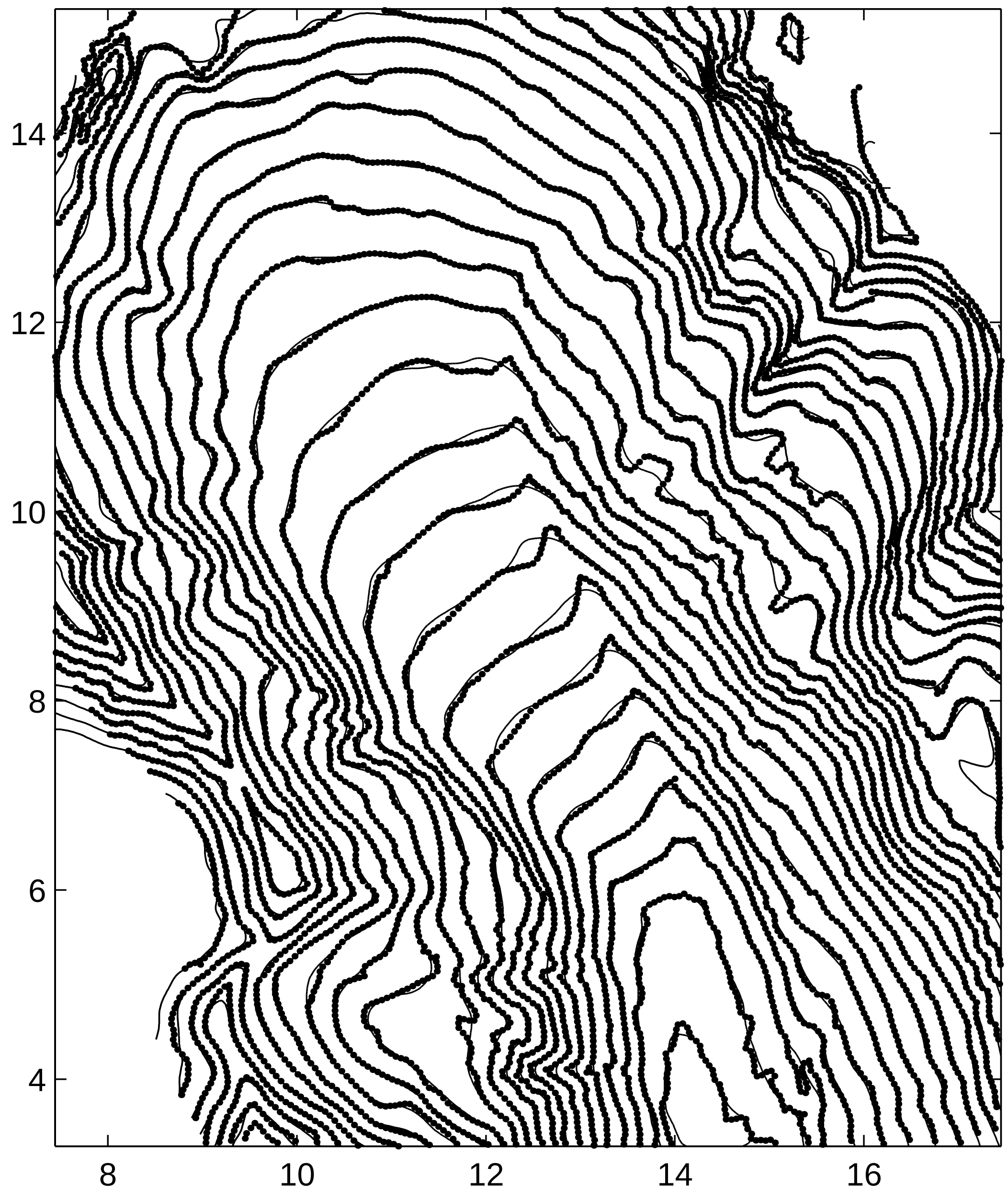
C^2 spline reconstruction



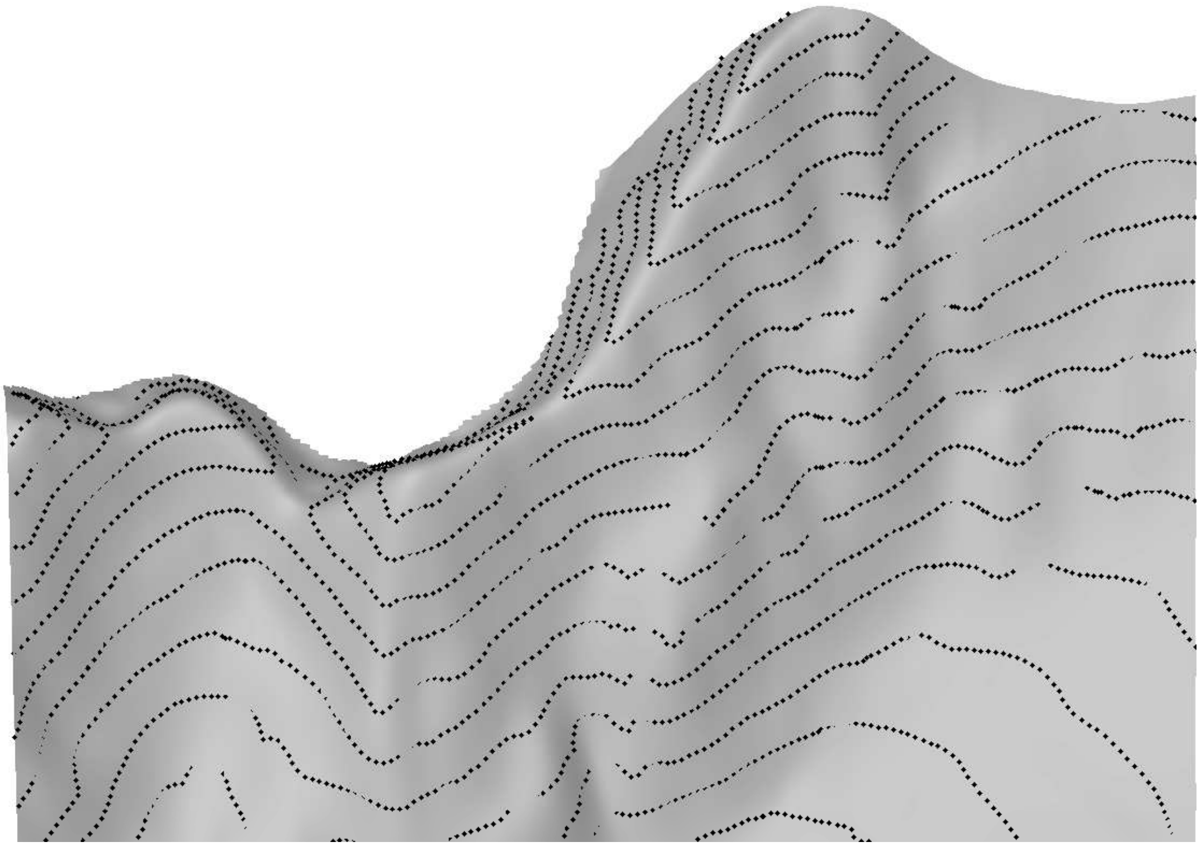
dim=7,254

Cputime 27.2 sec

Contour plot:

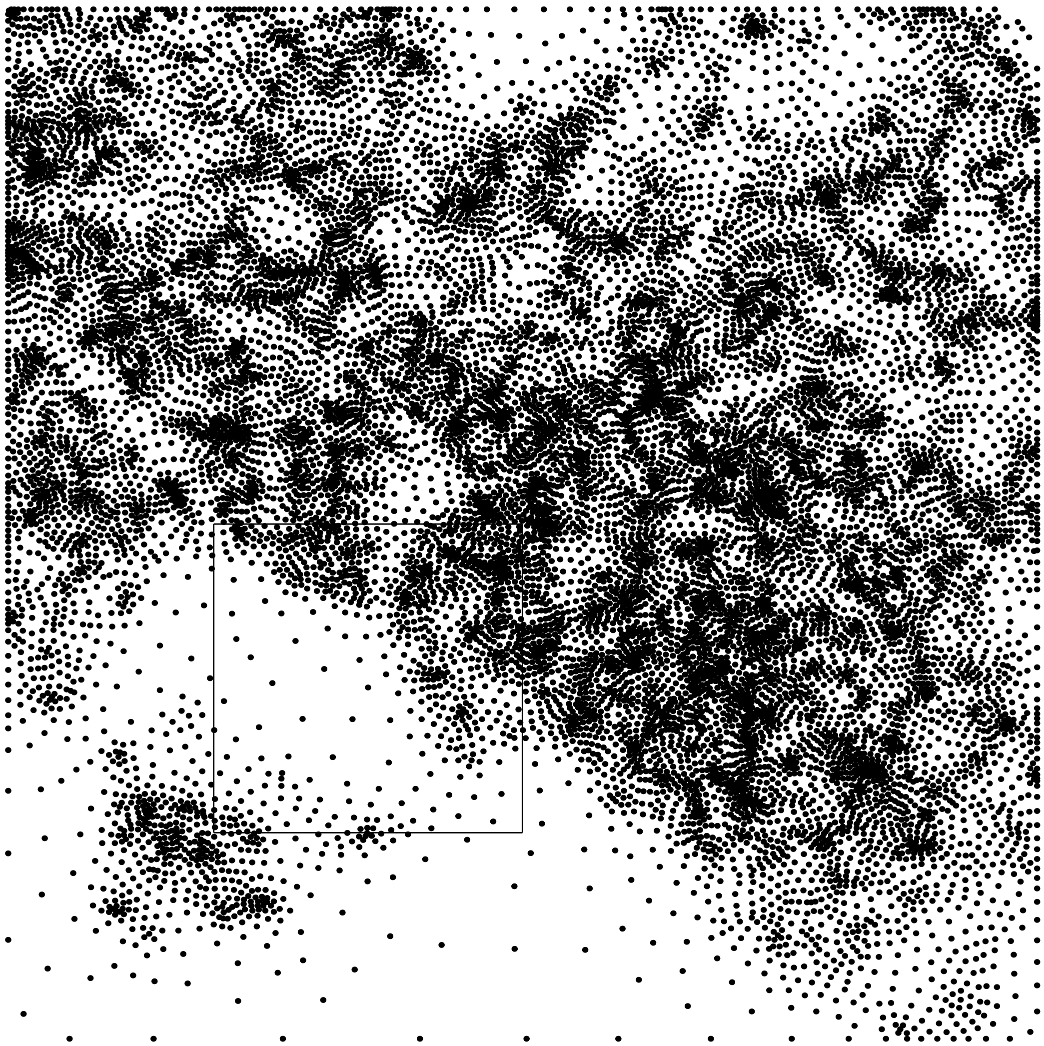


Screenshot with data points:

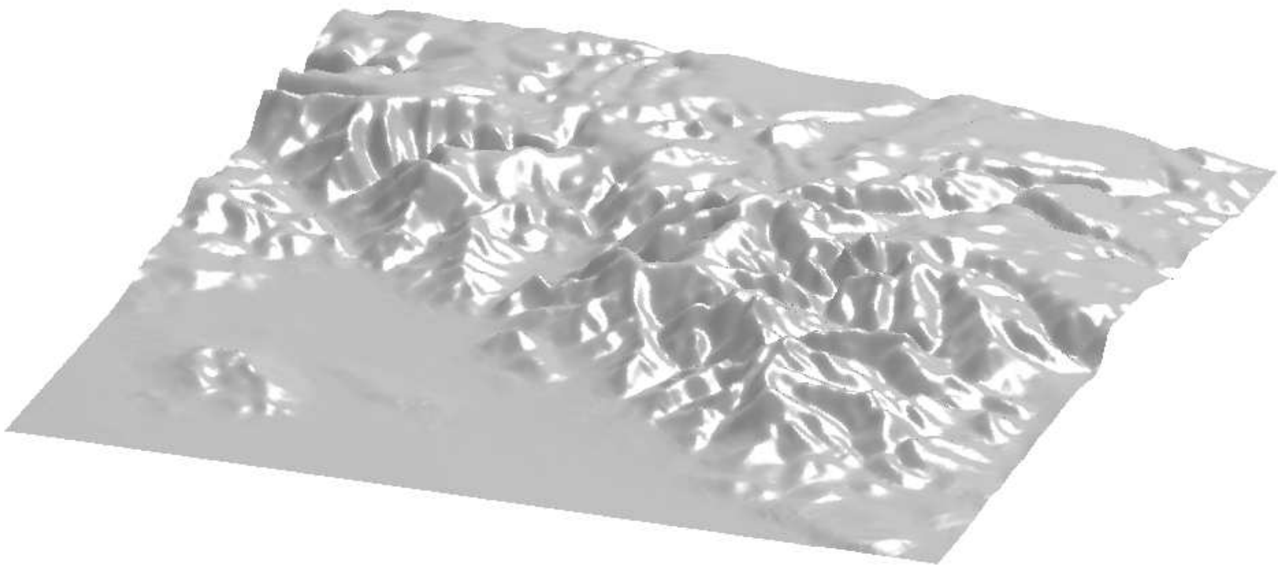


Black Forest

xy-locations (15,885 data points from a hilly area)



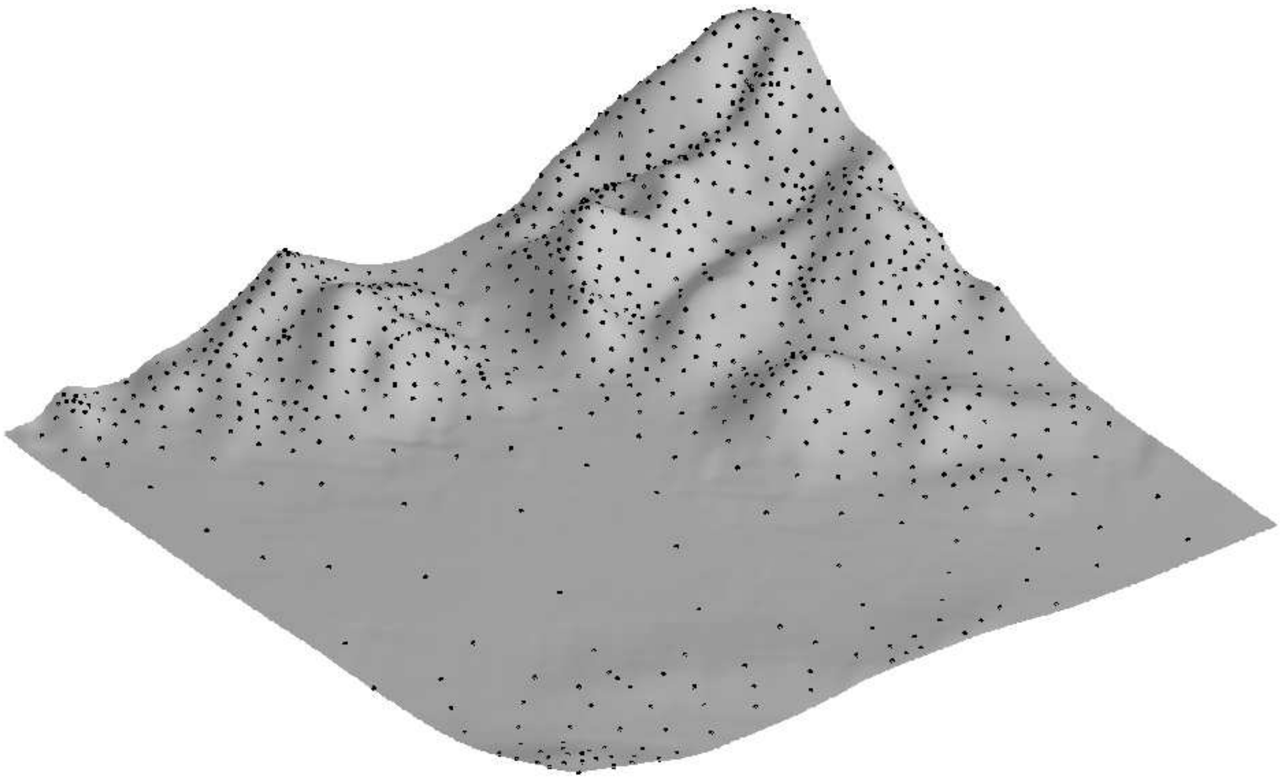
C^2 spline reconstruction (dim=91,526, cpu=12.6 sec)



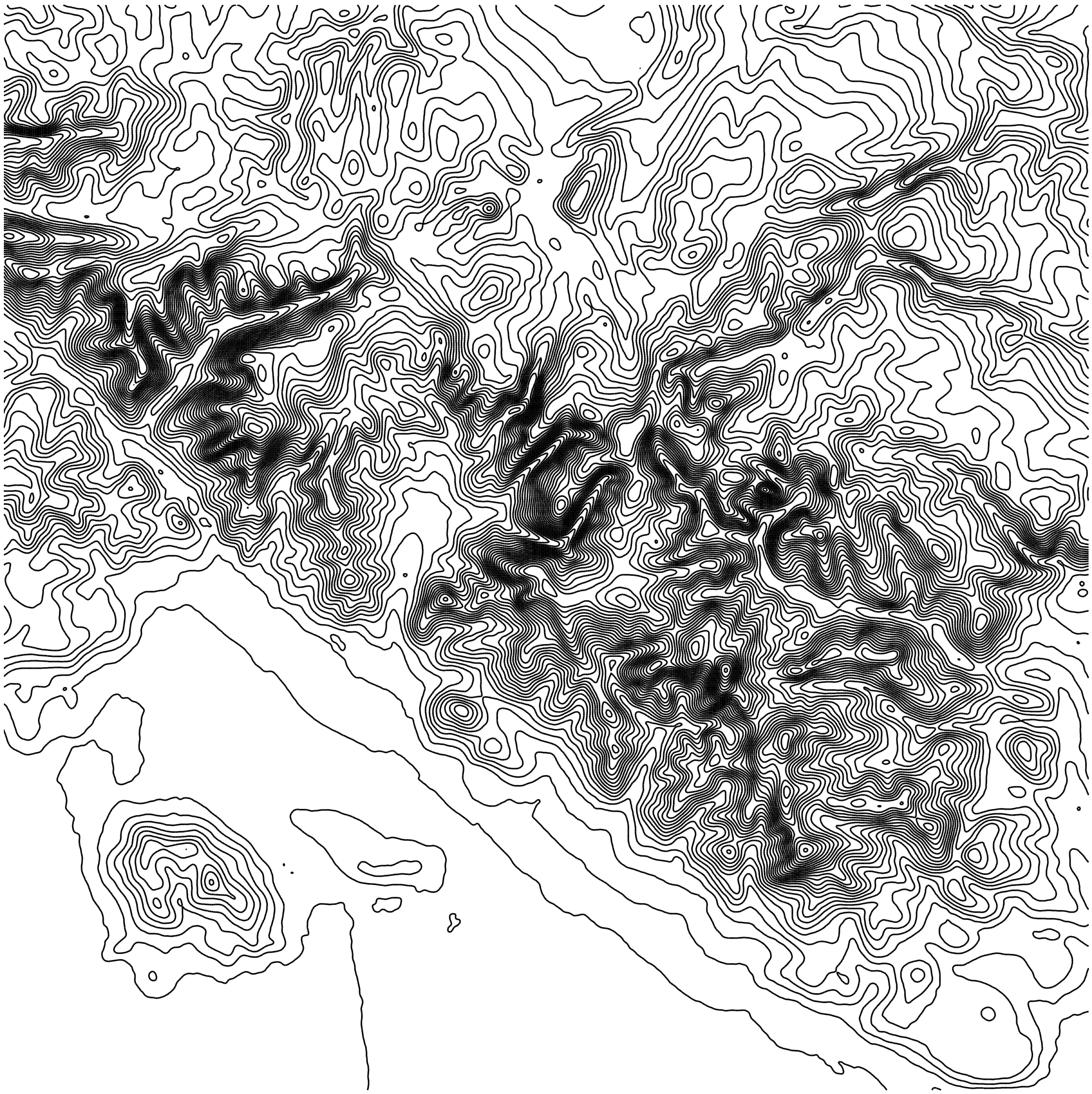
dim=91,526

Cputime 12.6 sec

Surface in the area indicated with a box on page 25:



Contour plot:

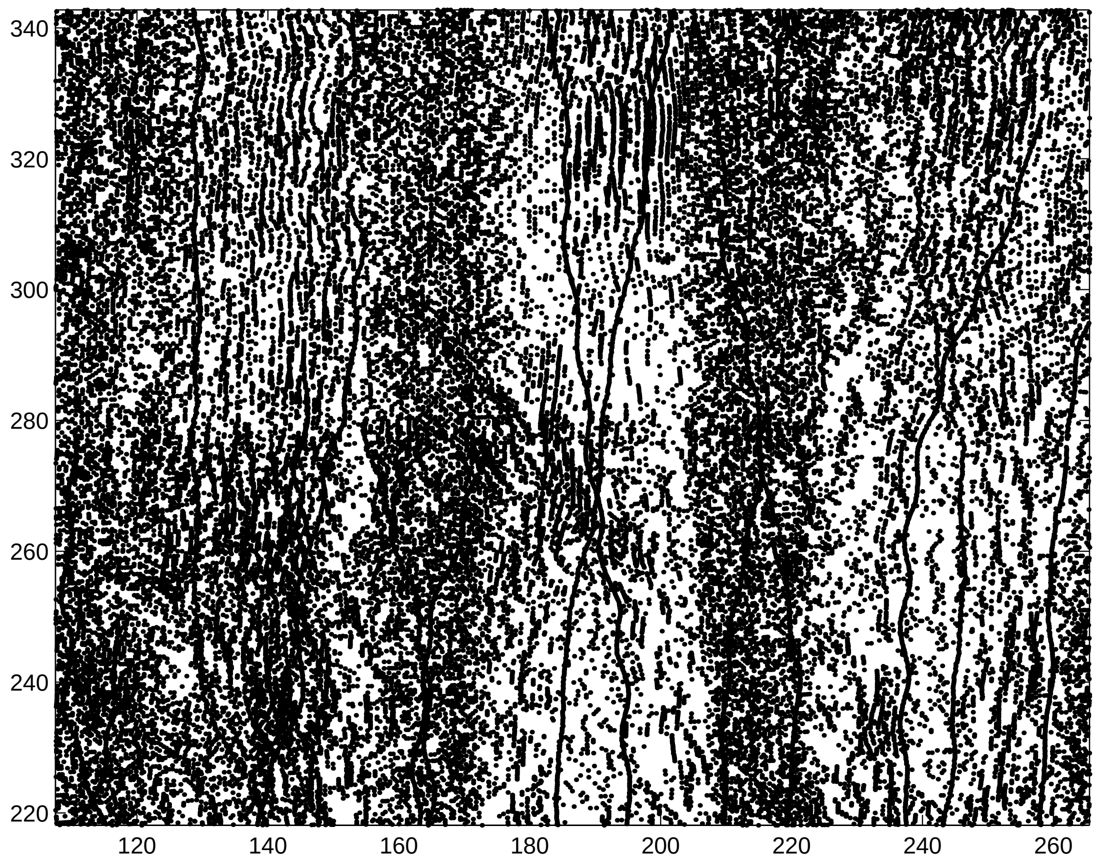


Rotterdam Port

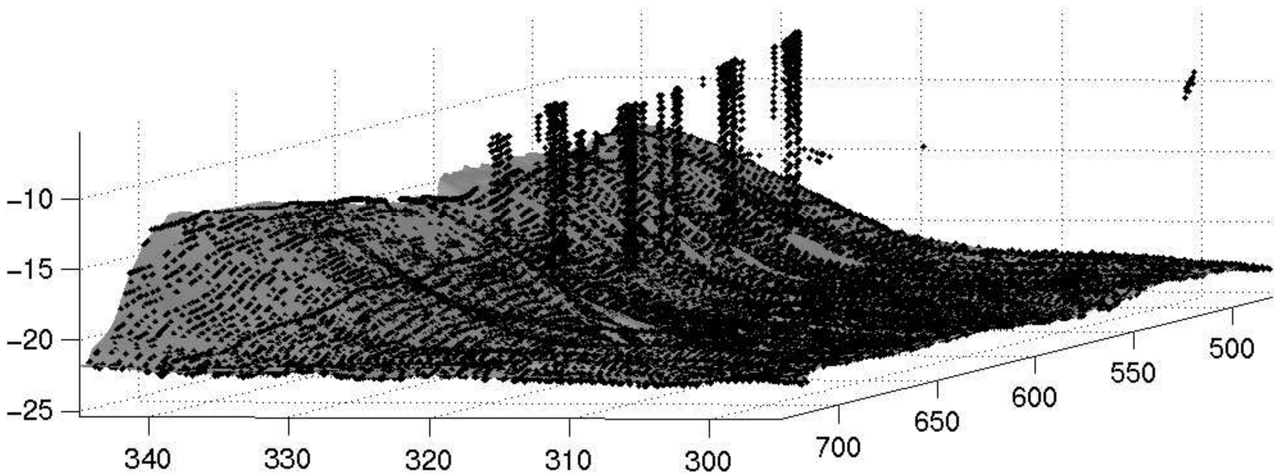
634,604 noisy raw data points (with outliers) from the measurements of the port of Rotterdam (using *high density multibeam echosounder*).

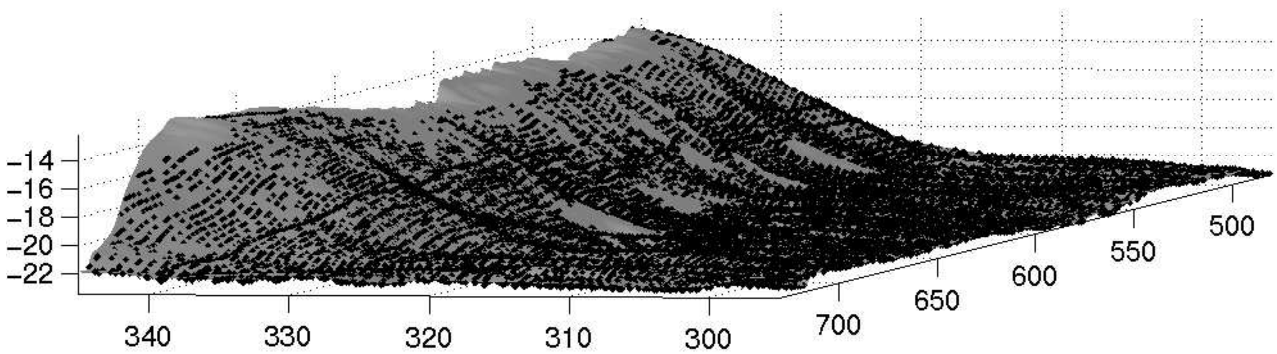
Quality Positioning Services BV, Zeist, Holland.

Typical distribution of the xy-points:



Outliers:

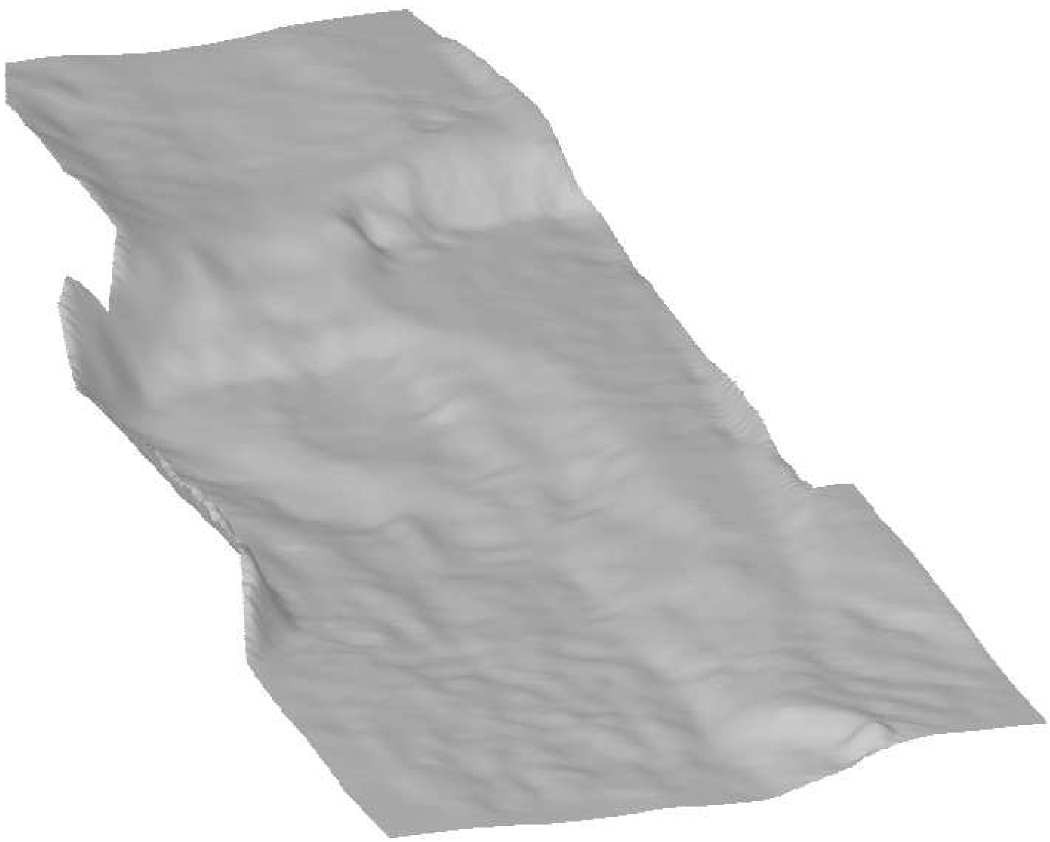

−10
−15
−20
−25
340
330
320
310
300
700
650
600
550
500


−14
−16
−18
−20
−22
340
330
320
310
300
700
650
600
550
500

Despiking / data cleaning:

We compute the C^1 spline of a relatively small dimension (22,399 parameters) und eliminate all data points, for which the z-values are at a greater distance from the spline than the rms error on the full data set.

C^1 spline with 22,399 parameters: (Cputime 42.5 sec)



Approximation of the cleaned data (619,205 points).

C^1 spline with 142,027 parameters: (Cputime 127 sec)