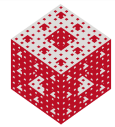
MATH 590: Meshfree Methods

Chapter 19: Least Squares RBF Approximation with MATLAB

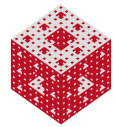
Greg Fasshauer

Department of Applied Mathematics
Illinois Institute of Technology

Fall 2010



Outline

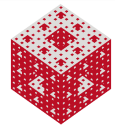


Up to now we have looked only at interpolation.

However, many times it makes more sense to approximate the given data by a least squares fit.

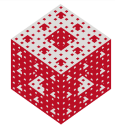
This is especially true

- if the data are contaminated with noise,
- or if there are so many data points that efficiency considerations force us to approximate from a space spanned by fewer basis functions than data points.



As we saw in Chapter 18 we can interpret kernel interpolation as a constrained optimization problem, i.e., the kernel interpolant automatically minimizes the native space norm among all interpolants in the native space.

We now take this point of view again, but start with a more general formulation.



Let us assume we are seeking a function $\mathcal{P}_f$ of the form

$$\mathcal{P}_f(\boldsymbol{x}) = \sum_{j=1}^{M} c_j K(\boldsymbol{x}, \boldsymbol{x}_j), \qquad \boldsymbol{x} \in \mathbb{R}^s,$$

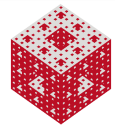
where the number M of basis functions is in general less than or equal the number N of data sites.
We then want to determine the coefficients $\boldsymbol{c} = [c_1, \ldots, c_M]^T$ so that we minimize the quadratic form

$$\frac{1}{2}\boldsymbol{c}^T \mathrm{Q} \boldsymbol{c} \tag{1}$$

with some symmetric positive definite matrix Q subject to the linear constraints

$$\mathrm{A}\boldsymbol{c} = \boldsymbol{f} \tag{2}$$

where A is an $N \times M$ matrix with full rank, and the right-hand side $\boldsymbol{f} = [f_1, \ldots, f_N]^T$ is given.



Such a constrained quadratic minimization problem can be converted to a system of linear equations by introducing Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$.

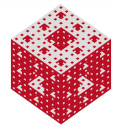
Thus, we consider finding the minimum of

$$\frac{1}{2}\boldsymbol{c}^T \mathrm{Q}\boldsymbol{c} - \boldsymbol{\lambda}^T [\mathrm{A}\boldsymbol{c} - \boldsymbol{f}] \tag{3}$$

with respect to $\boldsymbol{c}$ and $\boldsymbol{\lambda}$.

Since Q is assumed to be a positive definite matrix, it is well known that

- the functional to be minimized is convex,
- and thus the minimization problem has a unique minimum.



Therefore, the standard necessary condition for such a minimum:

- differentiate with respect to $\boldsymbol{c}$ and $\boldsymbol{\lambda}$, and
- find the zeros of those derivatives

is also sufficient.

This leads to

$$\begin{aligned} \mathrm{Q}\boldsymbol{c} - \mathrm{A}^T\boldsymbol{\lambda} &= \mathbf{0} \\ \mathrm{A}\boldsymbol{c} - \boldsymbol{f} &= \mathbf{0} \end{aligned}$$

or, in matrix form,

$$\begin{bmatrix} \mathrm{Q} & -\mathrm{A}^T \\ \mathrm{A} & \mathrm{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{f} \end{bmatrix}.$$

By applying (block) Gaussian elimination to this block matrix (Q is invertible since it is assumed to be positive definite) we get

$$\boldsymbol{\lambda} = \left(\mathrm{A}\mathrm{Q}^{-1}\mathrm{A}^T\right)^{-1} \boldsymbol{f} \tag{4}$$

$$\boldsymbol{c} = \mathrm{Q}^{-1}\mathrm{A}^T \left(\mathrm{A}\mathrm{Q}^{-1}\mathrm{A}^T\right)^{-1} \boldsymbol{f}. \tag{5}$$

Example

If the quadratic form represents the native space norm of $\mathcal{P}_f = \sum_{j=1}^{M} c_j K(\cdot, \boldsymbol{x}_j)$, i.e.,

$$\|\mathcal{P}_f\|^2_{\mathcal{N}_K(\Omega)} = \sum_{i=1}^{M} \sum_{j=1}^{M} c_i c_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{c}^T \mathrm{Q} \boldsymbol{c}$$

with $\mathrm{Q}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{c} = [c_1, \ldots, c_M]^T$, and the linear side conditions are the interpolation conditions

$$\mathrm{A}\boldsymbol{c} = \boldsymbol{f} \qquad \Longleftrightarrow \qquad \mathcal{P}_f(\boldsymbol{x}_i) = f_i, \quad i = 1, \ldots, M,$$

with $\mathrm{A} = \mathrm{A}^T = \mathrm{Q}$ (symmetric), the same $\boldsymbol{c}$ as above and data vector $\boldsymbol{f} = [f_1, \ldots, f_M]^T$, then we see that the Lagrange multipliers (4) become

$$\boldsymbol{\lambda} = \mathrm{A}^{-1} \boldsymbol{f}$$

and (5) yields the coefficients

$$\boldsymbol{c} = \boldsymbol{\lambda}.$$

Therefore, as we saw earlier, the minimum norm interpolant is obtained by solving the interpolation equations alone.

Since we took the more general point of view that

- $\mathcal{P}_f$ is generated by M basis functions,
- and N linear constraints are specified,

the above formulation also covers both over- and under-determined least squares fitting where the quadratic form $\boldsymbol{c}^T Q \boldsymbol{c}$ represents an added smoothing (or regularization) term.

Remark

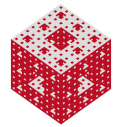
The smoothing term is not required to obtain a unique solution of the system $A\boldsymbol{c} = \boldsymbol{f}$ in the over-determined case ($N \geq M$), but in the under-determined case such a constraint is needed (c.f. the solution of under-determined linear systems via singular value decomposition in the numerical linear algebra literature (e.g., [Trefethen and Bau (1997)])).

Usually the regularized least squares approximation problem is formulated as minimization of

$$\frac{1}{2}\boldsymbol{c}^T Q\boldsymbol{c} + \omega \sum_{j=1}^{N} \left(\mathcal{P}_f(\boldsymbol{x}_j) - f_j\right)^2$$

$$\iff \tfrac{1}{2}\boldsymbol{c}^T Q\boldsymbol{c} + \omega(A\boldsymbol{c} - \boldsymbol{f})^T(A\boldsymbol{c} - \boldsymbol{f}). \tag{6}$$

- The quadratic form $\boldsymbol{c}^T Q\boldsymbol{c}$ controls the smoothness of the fitting function
- and the least squares term measures the closeness to the data.
- The parameter ω controls the tradeoff between these two terms with
  - large value of ω: more pointwise accuracy,
  - small value of ω: more smoothness.



Remark

- *The formulation (6) is used in*
  - *regularization theory ([Evgeniou et al. (2000), Girosi (1998)]),*
  - *penalized least squares fitting ([von Golitschek and Schumaker (1990)]),*
  - *smoothing splines [Reinsch (1967), Schoenberg (1964)],*
  - *and in papers by Wahba on thin plate splines (e.g., [Kimeldorf and Wahba (1971), Wahba (1979), Wahba (1990), Wahba and Luo (1997), Wahba and Wendelberger (1980)]).*
- *The idea of smoothing a data fitting process by this kind of formulation seems to go back to at least [Whittaker (1923)].*
- *In practice a penalized least squares formulation is especially useful if the data f_i cannot be completely trusted, i.e., they are contaminated by noise.*
- *The problem of minimizing (6) is also known as ridge regression in the statistics literature.*
- *ω is usually chosen using GCV or MLE.*

Example

If we restrict ourselves to working with square symmetric systems, i.e., $A = A^T$, and assume the smoothness functional is given by the native space norm, i.e., $Q = A$, then we obtain the minimizer of the unconstrained quadratic functional (6) by solving the linear system

$$\left(A + \frac{1}{2\omega} I\right) \boldsymbol{c} = \boldsymbol{f} \tag{7}$$

which is the result of setting the derivative of (6) with respect to $\boldsymbol{c}$ equal to zero.

Thus, ridge regression corresponds to a diagonal stabilization/regularization of the usual interpolation system $A\boldsymbol{c} = \boldsymbol{f}$.

This approach is especially useful for smoothing of noisy data.

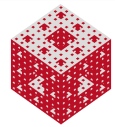
We implement this method and give some numerical examples below.

Now more general setting:

- still sample f on the set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of data sites,
- but now introduce a second set $\Xi = \{\boldsymbol{\xi}_i\}_{i=1}^{M}$ at which we center the basis functions.

Remark

- *Usually we will have $M \leq N$.*
- *The case $M = N$ with $\Xi = \mathcal{X}$ recovers the traditional interpolation setting discussed in earlier chapters.*



We let the kernel approximant be of the form

$$\mathcal{Q}_f(\boldsymbol{x}) = \sum_{j=1}^{M} c_j K(\boldsymbol{x}, \boldsymbol{\xi}_j), \quad \boldsymbol{x} \in \mathbb{R}^s. \tag{8}$$

The c_j can be found as the least squares solution of $A\boldsymbol{c} = \boldsymbol{f}$, i.e., by minimizing

$$\|\mathcal{Q}_f - f\|_2^2,$$

where the ℓ_2-norm

$$\|f\|_2^2 = \sum_{i=1}^{N} [f(\boldsymbol{x}_i)]^2, \qquad \boldsymbol{x}_i \in \mathcal{X},$$

is induced by the discrete inner product

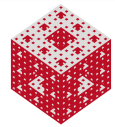
$$\langle f, g \rangle = \sum_{i=1}^{N} f(\boldsymbol{x}_i) g(\boldsymbol{x}_i), \qquad \boldsymbol{x}_i \in \mathcal{X}. \tag{9}$$

This approximation problem has a unique solution if the (rectangular) collocation matrix A with entries

$$A_{jk} = K(\boldsymbol{x}_j, \xi_k), \quad j = 1, \ldots, N, \ k = 1, \ldots, M,$$

has full rank.

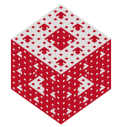
If $M \leq N$ and $\Xi \subseteq \mathcal{X}$, then A does have full rank provided we use "standard" kernels.
This is true, since in this case A will have an $M \times M$ square submatrix which is non-singular (by virtue of being an interpolation matrix).



The over-determined ($M > N$) linear system $A\boldsymbol{c} = \boldsymbol{f}$ can be solved using standard algorithms from numerical linear algebra such as QR or SVD.

Remark

Therefore in MATLAB *the solution of the least squares problem is computed on line 10 using backslash which automatically produces a least squares solution.*



Program (RBFApproximation2D.m)

```
 1  rbf = @(e,r) exp(-(e*r).^2); ep = 1;
 2  M = 81;  ctrs = CreatePoints(M,2,'u');
 3  N = 1089;  dsites = CreatePoints(N,2,'h');
 4  neval = 40;  epoints = CreatePoints(neval^2,2,'u');
 5  DM_data = DistanceMatrix(dsites,ctrs);
 6  CM = rbf(ep,DM_data);
 7  rhs = testfunctionsD(dsites);
 8  DM_eval = DistanceMatrix(epoints,ctrs);
 9  EM = rbf(ep,DM_eval);
10  Pf = EM * (CM\rhs);
11  exact = testfunctionsD(epoints);
12  maxerr = norm(Pf-exact,inf);
13  figure; fview = [100,30];
14a plot(dsites(:,1),dsites(:,2),'bo',...
14b      ctrs(:,1),ctrs(:,2),'r+');
15  xe = reshape(epoints(:,1),neval,neval);
16  ye = reshape(epoints(:,2),neval,neval);
17  PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
18  PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

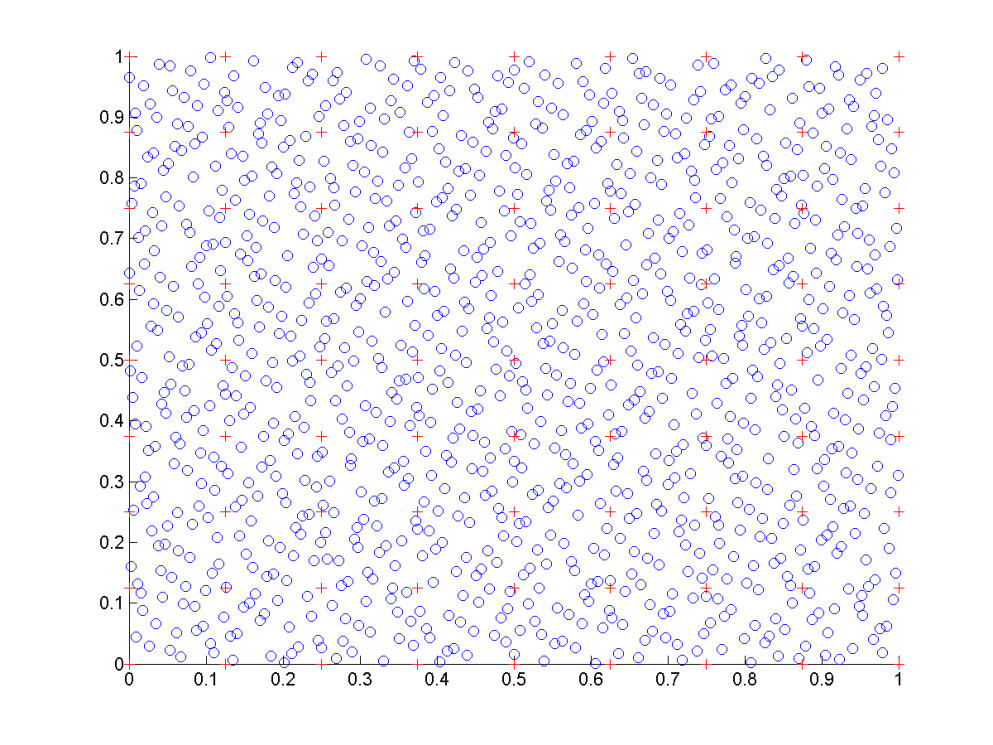
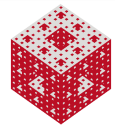
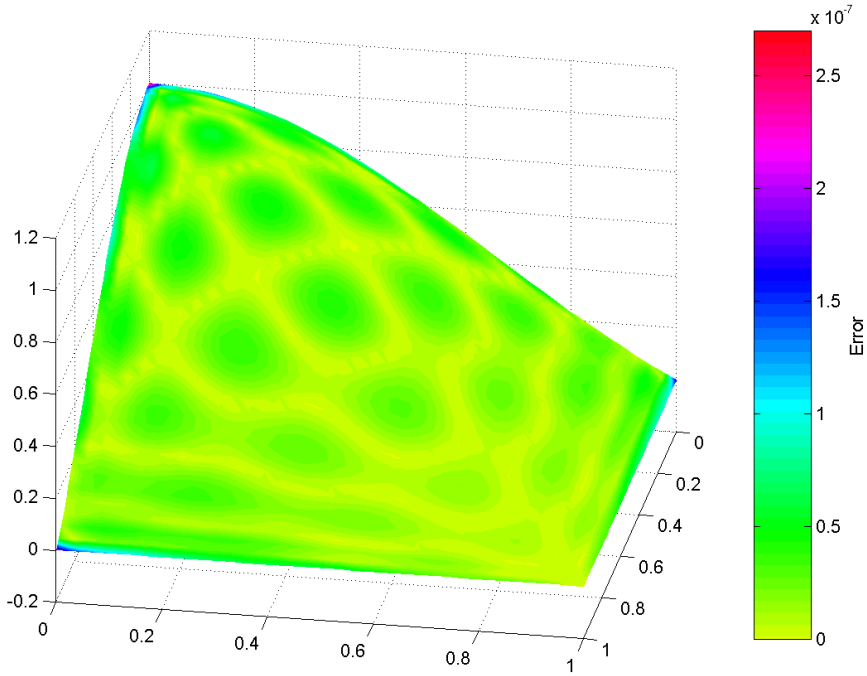


Figure: 1089 Halton data sites ($\circ$) and 81 uniformly gridded centers ($+$).





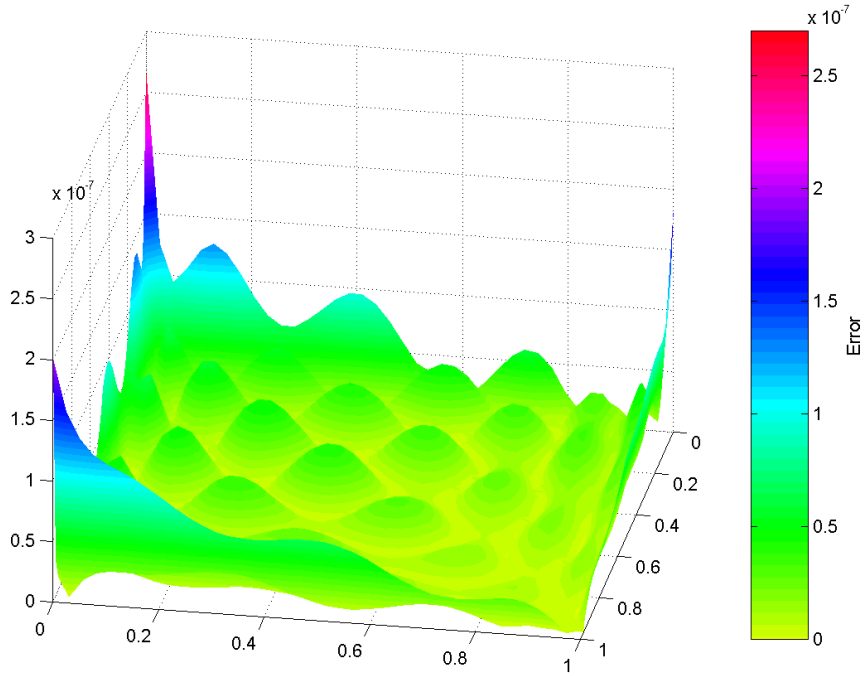
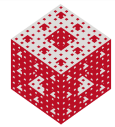
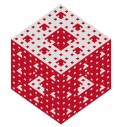


Figure: Least squares approximation (left) to 1089 data points sampled from 2D sinc function with 81 Gaussian basis functions with $\varepsilon = 1$ and maximum error (right) false-colored by magnitude of error.



Remark

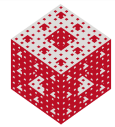
- *If* $\varepsilon = 1$*, then the collocation matrix is rank deficient with* MATLAB *reporting a numerical rank of 58.*
- *In order to have a full numerical rank for this problem* ε *needs to be at least 2.2 (in which case the maximum error deteriorates to* 5.255591e-004 *instead of* 2.173460e-007 *for* $\varepsilon = 1$*).*
- *There is not much theory available for the case of differing centers and data sites (see Chapter 20).*
- *Some care needs to be taken when computing least squares solutions based on sets of differing centers and data sites.*



We present two strategies for dealing with noisy data, i.e., data that we consider to be not reliable due to, e.g., measurement or transmission errors.

This situation arises frequently in practice.

We simulate a set of noisy data by sampling Franke's test function at a set $\mathcal{X}$ of data sites, and then adding uniformly distributed random noise of various strengths.

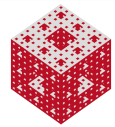


For this experiment we use thin plate splines since their native space norm corresponds to the bending energy of a thin plate and thus they have a tendency to produce "visually pleasing" smooth and tight surfaces.

Since the thin plate splines have a singularity at the origin a little extra care needs to be taken with their implementation:

Program (`tps.m`)

```
1  function rbf = tps(e,r)
2  nz = find(r~=0);    % to find singularity at origin
3  rbf = zeros(size(r));    % limit at origin
4  rbf(nz) = (e*r(nz)).^2.*log(e*r(nz));
```



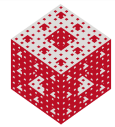
Strategy I

Compute a straightforward least squares approximation to the (large) set of data using a (small) set of basis functions as we did in the previous section.

In the statistics literature this approach is known as regression splines.

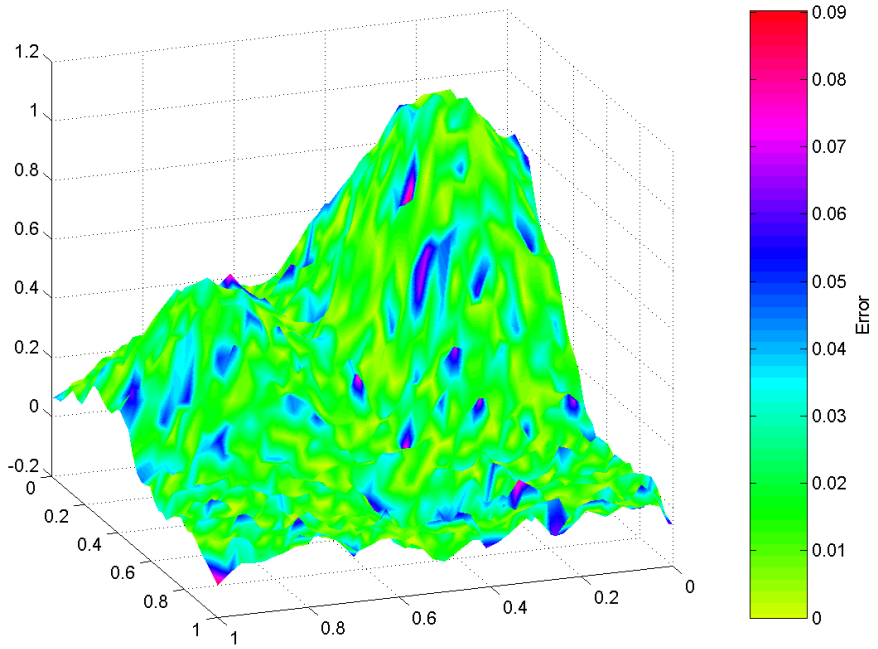
Remark

We will not address the question of how to choose the centers for the basis functions at this point.



Program (RBFApproximation2Dlinear.m)

```
 1  rbf = @tps; ep = 1;
 2  M = 81;  ctrs = CreatePoints(M,2,'u');
 3  N = 1089;  dsites = CreatePoints(N,2,'h');
 4  neval = 40;  epoints = CreatePoints(neval^2,2,'u');
 5  DM_data = DistanceMatrix(dsites,ctrs);
 6  CM = rbf(ep,DM_data);
 7  PM = [ones(N,1) dsites]; PtM = [ones(M,1) ctrs]';
 8  CM = [CM PM; [PtM zeros(3,3)]];
 9  rhs = testfunctionsD(dsites);
10  rhs = rhs + 0.03*randn(size(rhs));
11  rhs = [rhs; zeros(3,1)];
12  DM_eval = DistanceMatrix(epoints,ctrs);
13  EM = rbf(ep,DM_eval);
14  PM = [ones(neval^2,1) epoints]; EM = [EM PM];
15  Pf = EM * (CM\rhs);
16  exact = testfunctionsD(epoints);
17  maxerr = norm(Pf-exact,inf);
18  figure; fview = [160,20];
19  xe = reshape(epoints(:,1),neval,neval);
20  ye = reshape(epoints(:,2),neval,neval);
21a plot(dsites(:,1),dsites(:,2),'bo',...
21b      ctrs(:,1),ctrs(:,2),'r+');
22  PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
23  PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```



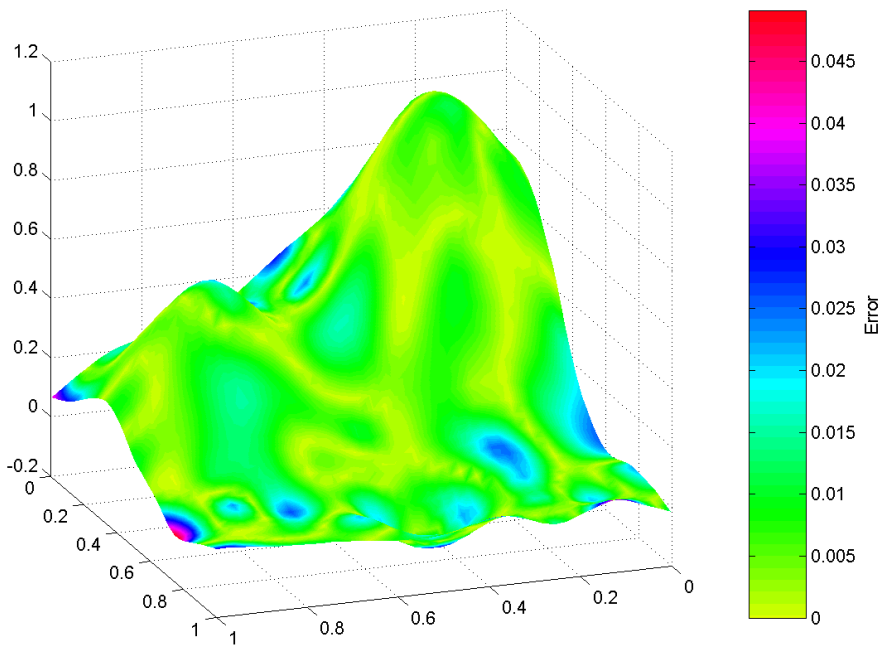
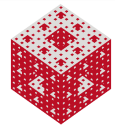


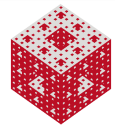
Figure: Thin plate spline interpolant (left) to 1089 noisy data points sampled from Franke's function, and least squares approximation with 81 uniformly spaced thin plate spline basis functions (right) false-colored by magnitude of error.



Method	RMS-error	max-error
Interpolation	2.482624e-002	9.914837e-002
LSQ approximation	9.665743e-003	5.490050e-002

Table: Errors and condition numbers for interpolation and least squares approximation of noisy data.

- Clearly, this simple least squares approach performs much better than straightforward interpolation to the noisy data (see the left plot of the figure and line 1 of the table).
- Least squares approximation with $M < N$ is much cheaper to compute.
- It is not clear how to choose the smaller set of RBF centers.
- There is not much mathematical theory to guarantee if (or when) this approach is well-posed, i.e., the collocation matrix has full rank.



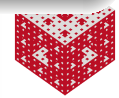
Strategy II

We can smooth out noisy data by using the ridge regression method (see (7)).

This method is popular in the statistics and neural network community.

Remark

- *In the* MATLAB *program we do ridge regression with thin plate splines (including the linear term in the basis expansion) for smoothing of noisy data.*
- *The smoothing parameter* ω *of (7) is defined on line 7.*
- *The diagonal stabilization of A is performed on line 17. Note that the stabilization only affects the A part of the matrix, and not the extra rows and columns added for polynomial precision.*



Program (TPS_RidgeRegression2D.m)

```
 1  rbf = @tps;  ep = 1;  omega = 100;
 2  N = 1089;  dsites = CreatePoints(N,2,'h');
 3  ctrs = dsites;
 4  neval = 40;  epoints = CreatePoints(neval^2,2,'u');
 5  DM_data = DistanceMatrix(dsites,ctrs);
 6  rhs = testfunctionsD(dsites);
 7  randn('state',3);  rhs = rhs + 0.03*randn(size(rhs));
 8  rhs = [rhs; zeros(3,1)];
 9  IM = rbf(ep,DM_data);
10  IM = IM + eye(size(IM))/(2*omega);
11  PM=[ones(N,1) dsites];  IM=[IM PM; [PM' zeros(3,3)]];
12  fprintf('Condition number estimate: %e\n',condest(IM))
13  DM_eval = DistanceMatrix(epoints,ctrs);
14  EM = rbf(ep,DM_eval);
15  PM = [ones(neval^2,1) epoints]; EM = [EM PM];
16  Pf = EM * (IM\rhs);
17  exact = testfunctionsD(epoints);
18  maxerr = norm(Pf-exact,inf);
19  PlotSurf(xe,ye,Pf,neval,exact,maxerr,[160,20]);
20  PlotError2D(xe,ye,Pf,exact,maxerr,neval,[160,20]);
```

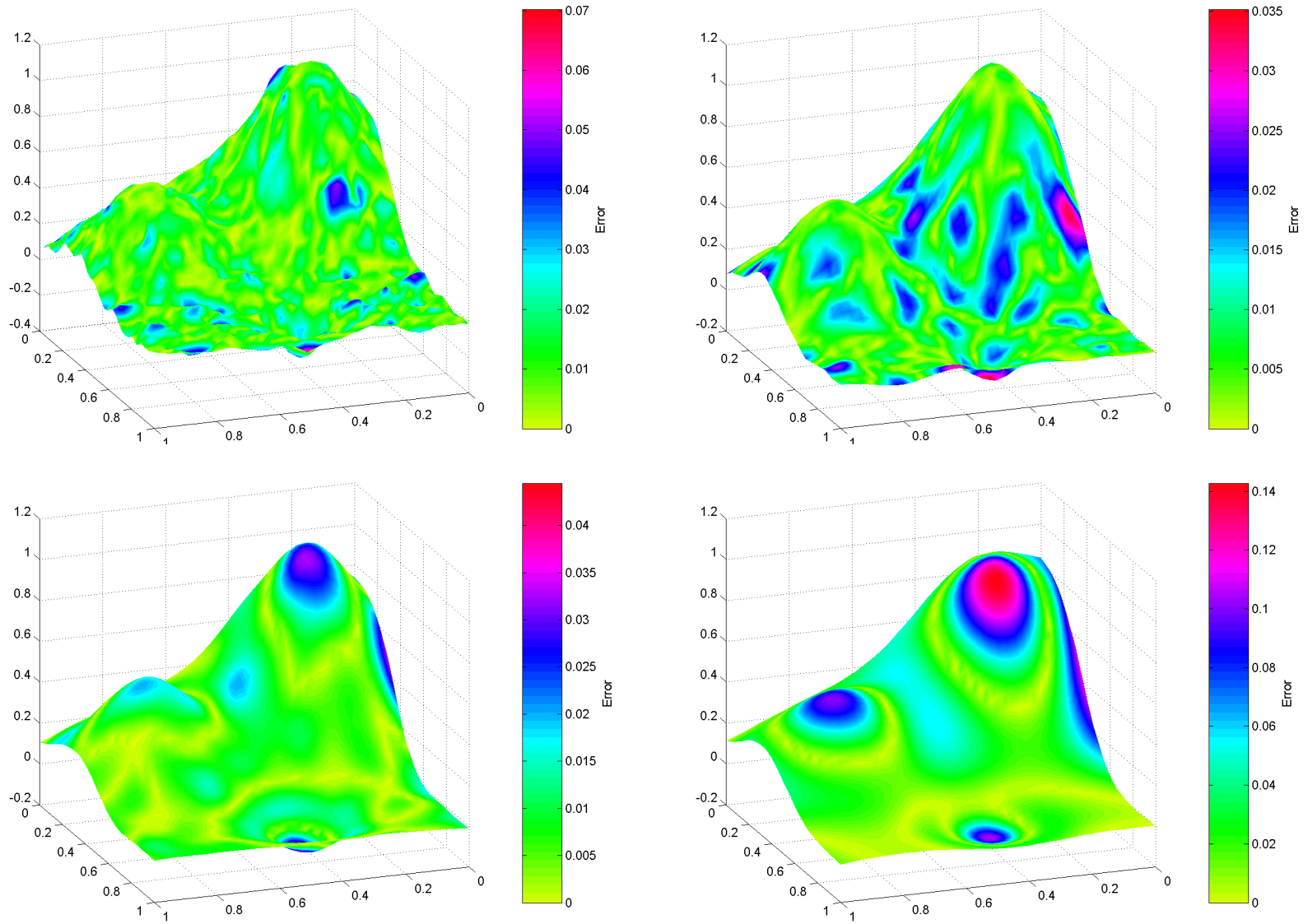
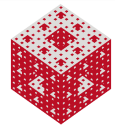


Figure: Thin plate spline ridge regression to 1089 noisy data points sampled from Franke's function with $\omega = 1000$ (top left), $\omega = 100$ (top right), $\omega = 10$ (bottom left), and $\omega = 1$ (bottom right).

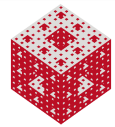
Method	ω	RMS-error	max-error	cond(A)
Interpolation	∞	2.482624e-002	9.914837e-002	1.502900e+007
LSQ approximation	NA	9.665743e-003	5.490050e-002	NA
Ridge regression	1000	1.713843e-002	7.580288e-002	2.537652e+006
Ridge regression	100	1.078358e-002	4.215865e-002	3.839384e+005
Ridge regression	10	9.173961e-003	3.349371e-002	4.571167e+004
Ridge regression	1	2.764272e-002	1.041350e-001	9.317936e+003

Table: Errors and condition numbers for various least squares approximants to noisy data.



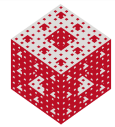
Remark

- *The results illustrate very nicely the smoothing effect obtained by varying ω.*
  - *A very large value of ω emphasizes the fitting component of the functional to be minimized in (6) resulting in a rather rough surface,*
  - *A small value of ω gives preference to the smoothing term.*
  - *The "optimal" value of ω lies somewhere in the middle.*
- *In practice one would usually use cross validation to obtain the optimal value of ω.*
- *Besides the visual smoothing of the approximating surface, a small value of ω also has a stabilizing effect on the collocation matrix.*
  - *The diagonal of the matrix becomes more and more dominant.*
  - *The condition estimates also verify this behavior.*



Some other strategies that might be used to fit noisy data using kernel-based methods are

- MLS approximation (see Chapters 22-25)
- approximate MLS approximation (see Chapters 26-27)
- Riley's algorithm, also known as iterated Tikhonov regularization (see [Fasshauer (2008)])



References I

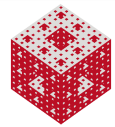

Buhmann, M. D. (2003).
Radial Basis Functions: Theory and Implementations.
Cambridge University Press.


Fasshauer, G. E. (2007).
Meshfree Approximation Methods with MATLAB.
World Scientific Publishers.


Higham, D. J. and Higham, N. J. (2005).
MATLAB *Guide*.
SIAM (2nd ed.), Philadelphia.


Iske, A. (2004).
Multiresolution Methods in Scattered Data Modelling.
Lecture Notes in Computational Science and Engineering 37, Springer Verlag (Berlin).


Trefethen, L. N. and Bau, D. (1997).
Numerical Linear Algebra.
SIAM (Philadelphia, PA).



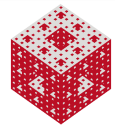
References II


G. Wahba (1990).
Spline Models for Observational Data.
CBMS-NSF Regional Conference Series in Applied Mathematics 59, SIAM (Philadelphia).


Wendland, H. (2005a).
Scattered Data Approximation.
Cambridge University Press (Cambridge).


Evgeniou, T., Pontil, M. and Poggio, T. (2000).
Regularization networks and support vector machines.
Adv. Comput. Math. **13** 1, pp. 1–50.


Girosi, F. (1998).
An equivalence between sparse approximation and support vector machines.
Neural Computation **10**, pp. 1455–1480.



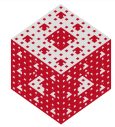
References III


von Golitschek, M. and Schumaker, L. L. (1990).
Data fitting by penalized least squares.
in *Algorithms for Approximation II*, M. G. Cox and J. C. Mason (eds.), Chapman & Hall (London), pp. 210–227.


Kimeldorf, G. and Wahba, G. (1971).
Some results on Tchebycheffian spline functions.
J. Math. Anal. Applic. **33**, pp. 82–95.


Reinsch, C. H. (1967).
Smoothing by spline functions.
Numer. Math. **10**, pp. 177–183.


Schoenberg, I. J. (1964).
Spline functions and the problem of graduation.
Proc. Nat. Acad. Sci. **52**, pp. 947–950.



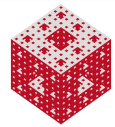
References IV

Wahba, G. (1979).
Convergence rate of “thin plate” smoothing splines when the data are noisy (preliminary report).
Springer Lecture Notes in Math. **757**, pp. 233–245.

Wahba, G. and Luo, Z. (1997).
Smoothing spline ANOVA fits for very large, nearly regular data sets, with application to historical global climate data.
in *The Heritage of P. L. Chebyshev: a Festschrift in honor of the 70th birthday of T. J. Rivlin*, *Ann. Numer. Math.* **4** 1–4, pp. 579–597.

Wahba, G. and Wendelberger, J. (1980).
Some new mathematical methods for variational objective analysis using splines and cross validation.
Monthly Weather Review **108**, pp. 1122–1143.

Whittaker, E. T. (1923).
On a new method of graduation.
Proc. Edinburgh Math. Soc. **41**, pp. 63–75.



References V

Fasshauer, G. E. (2008).
Tutorial on Meshfree Approximation Methods with MATLAB.
Dolomites Research Notes On Approximation **1**, ISSN: 2035-6803.
http://drna.di.univr.it/.