Meshfree Approximation with MATLAB

Lecture I: Introduction

Greg Fasshauer

Department of Applied Mathematics
Illinois Institute of Technology

Dolomites Research Week on Approximation
September 8–11, 2008

Outline

Rolland Hardy

- Professor of Civil and Construction Engineering at Iowa State University (retired 1989).
- Introduced multiquadrics (MQs) in the early 1970s (see, *e.g.*, [Hardy (1971)]).
- His work was primarily concerned with applications in geodesy and mapping.

Robert L. Harder and Robert N. Desmarais

- Aerospace engineers at MacNeal-Schwendler Corporation (MSC Software), and NASA's Langley Research Center.
- Introduced thin plate splines (TPSs) in 1972 (see, *e.g.*, [Harder and Desmarais (1972)]).
- Work was concerned mostly with aircraft design.

Jean Duchon

- Senior researcher in mathematics at the Université Joseph Fourier in Grenoble, France.
- Provided foundation for the variational approach minimizing the integral of $\nabla^2 f$ in $\mathbb{R}^2$ in the mid 1970s (see [Duchon (1976), Duchon (1977), Duchon (1978), Duchon (1980)]).
- This also leads to thin plate splines.

Jean Meinguet

- Mathematics professor at Université Catholique de Louvain in Louvain, Belgium (retired 1996).
- Introduced surface splines in the late 1970s (see, *e.g.*, [Meinguet (1979a), Meinguet (1979b), Meinguet (1979c), Meinguet (1984)]).
- Surface splines and thin plate splines are both considered as polyharmonic splines.

Richard Franke

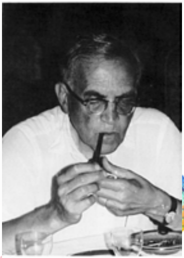
- Mathematician at the Naval Postgraduate School in Monterey, California (retired 2001).
- Compared various scattered data interpolation methods in [Franke (1982a)], and concluded MQs and TPSs were the best.
- Conjectured that the interpolation matrix for MQs is invertible.

Wolodymyr (Wally) Madych and Stuart Alan Nelson

- Both professors of mathematics. Madych at the University of Connecticut, and Nelson at Iowa State University (now retired).
- Proved Franke's conjecture (and much more) based on a variational approach in their 1983 manuscript [Madych and Nelson (1983)]. Manuscript was never published.

Charles Micchelli

- Used to be a mathematician at IBM Watson Research Center. Now a professor at the State University of New York.
- Published [Micchelli (1986)] in which he also proved Franke's conjecture. His proofs are rooted in the work of [Bochner (1932), Bochner (1933)] and [Schoenberg (1937), Schoenberg (1938a), Schoenberg (1938b)] on positive definite and completely monotone functions.
- We will follow his approach throughout much of these lectures.



Ed Kansa

- Physicist at Lawrence Livermore National Laboratory, California (retired).
- First suggested the use of radial basis functions for the solution of PDEs [Kansa (1986)].
- Later papers [Kansa (1990a), Kansa (1990b)] proposed “Kansa’s method” (or non-symmetric collocation).

Grace Wahba

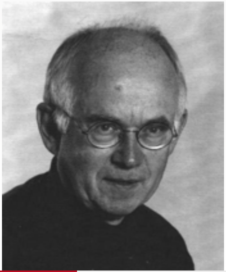
- Professor of statistics at the University of Wisconsin-Madison.
- Studied the use of thin plate splines for statistical purposes in the context of smoothing noisy data and data on spheres.
- Introduced ANOVA and cross validation approaches to the radial basis function setting (see, *e.g.*, [Wahba (1979), Wahba (1981), Wahba and Wendelberger (1980)]).
- One of the first monographs on the subject is [Wahba (1990)].

Nira Dyn

- Collaborated early with Grace Wahba on connections between numerical analysis and statistics via radial basis function methods (see [Dyn *et al.* (1979), Dyn and Wahba (1982)]).
- Professor of applied mathematics at Tel-Aviv University.
- Was one of the first proponents of radial basis function methods in the approximation theory community (see her surveys [Dyn (1987), Dyn (1989)]).
- Has since worked on many issues related to radial basis functions.

Robert Schaback

- Professor of mathematics at the University of Göttingen, Germany.
- Introduced compactly supported radial basis functions (CSRBFs) in [Schaback (1995a)].
- Another popular family of CSRBFs was presented by Holger Wendland (professor of mathematics at Sussex University, UK) in his Ph.D. thesis at Göttingen (see also [Wendland (1995)]).
- Both have contributed extensively to the field of radial basis functions. Especially the recent monograph [Wendland (2005)].



Meshfree local regression methods have been used independently in statistics for well over 100 years (see, *e.g.*, [Cleveland and Loader (1996)] and the references therein).



In fact, the basic moving least squares method (local regression) can be traced back at least to the work of [Gram (1883), Woolhouse (1870), De Forest (1873), De Forest (1874)].

Donald Shepard

- Professor at the Schneider Institutes for Health Policy at Brandeis University.
- As an undergraduate student at Harvard University he suggested the use of what are now called *Shepard functions* in the late 1960s.
- The publication [Shepard (1968)] discusses the basic inverse distance weighted Shepard method and some modifications thereof. The method was at the time incorporated into a computer program, SYMAP, for map making.

Peter Lancaster and Kes Šalkauskas

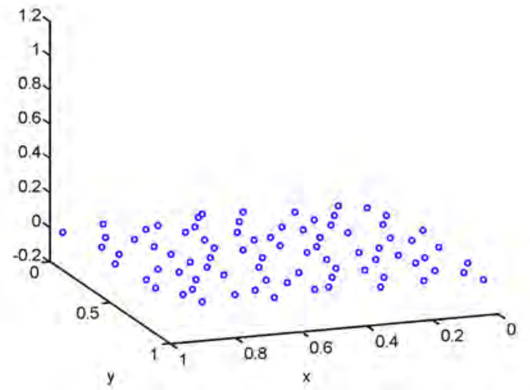
- Professors of mathematics at the University of Calgary, Canada (both retired).
- Published [Lancaster and Šalkauskas (1981)] introducing the moving least squares method (a generalization of Shepard functions).
- An interesting [Interview with Peter Lancaster].

Problem (Scattered Data Fitting)

Given data $(\boldsymbol{x}_j, y_j)$, $j = 1, \ldots, N$, *with* $\boldsymbol{x}_j \in \mathbb{R}^s$, $y_j \in \mathbb{R}$, *find a (continuous) function* $\mathcal{P}_f$ *such that* $\mathcal{P}_f(\boldsymbol{x}_j) = y_j$, $j = 1, \ldots, N$.

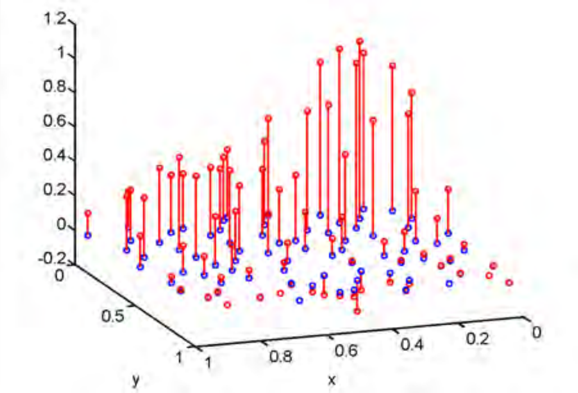
Problem (Scattered Data Fitting)

Given data $(\boldsymbol{x}_j, y_j)$, $j = 1, \ldots, N$, *with* $\boldsymbol{x}_j \in \mathbb{R}^s$, $y_j \in \mathbb{R}$, *find a (continuous) function* $\mathcal{P}_f$ *such that* $\mathcal{P}_f(\boldsymbol{x}_j) = y_j$, $j = 1, \ldots, N$.



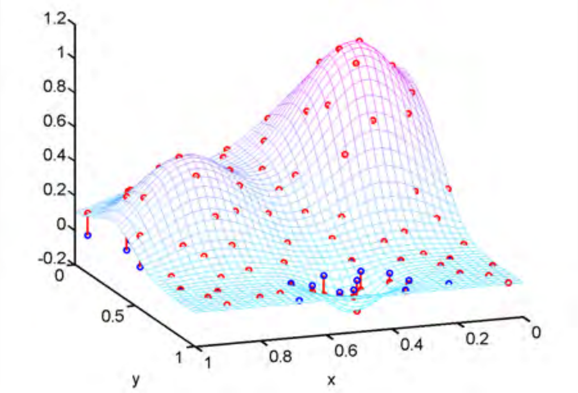
Problem (Scattered Data Fitting)

Given data $(\boldsymbol{x}_j, y_j)$, $j = 1, \ldots, N$, *with* $\boldsymbol{x}_j \in \mathbb{R}^s$, $y_j \in \mathbb{R}$, *find a (continuous) function* $\mathcal{P}_f$ *such that* $\mathcal{P}_f(\boldsymbol{x}_j) = y_j$, $j = 1, \ldots, N$.



Problem (Scattered Data Fitting)

Given data $(\boldsymbol{x}_j, y_j)$, $j = 1, \ldots, N$, *with* $\boldsymbol{x}_j \in \mathbb{R}^s$, $y_j \in \mathbb{R}$, *find a (continuous) function* $\mathcal{P}_f$ *such that* $\mathcal{P}_f(\boldsymbol{x}_j) = y_j$, $j = 1, \ldots, N$.



Standard setup

A convenient and common approach:
Assume $\mathcal{P}_f$ is a linear combination of certain basis functions B_k, *i.e.*,

$$\mathcal{P}_f(\boldsymbol{x}) = \sum_{k=1}^{N} c_k B_k(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^s. \tag{1}$$

Standard setup

A convenient and common approach:
Assume $\mathcal{P}_f$ is a linear combination of certain basis functions B_k, *i.e.*,

$$\mathcal{P}_f(\boldsymbol{x}) = \sum_{k=1}^{N} c_k B_k(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^s. \tag{1}$$

Solving the interpolation problem under this assumption leads to a system of linear equations of the form

$$A\boldsymbol{c} = \boldsymbol{y},$$

where the entries of the interpolation matrix A are given by
$A_{jk} = B_k(\boldsymbol{x}_j)$, $j, k = 1, \ldots, N$, $\boldsymbol{c} = [c_1, \ldots, c_N]^T$, and $\boldsymbol{y} = [y_1, \ldots, y_N]^T$.

Standard setup (cont.)

The scattered data fitting problem will be well-posed, *i.e.*, a solution to the problem will exist and be unique, if and only if the matrix A is non-singular.

Standard setup (cont.)

The scattered data fitting problem will be well-posed, *i.e.*, a solution to the problem will exist and be unique, if and only if the matrix A is non-singular.

In 1D it is well known that one can interpolate to arbitrary data at N distinct data sites using a polynomial of degree $N - 1$.

Standard setup (cont.)

The scattered data fitting problem will be well-posed, *i.e.*, a solution to the problem will exist and be unique, if and only if the matrix A is non-singular.

In 1D it is well known that one can interpolate to arbitrary data at N distinct data sites using a polynomial of degree $N-1$.

If the dimension is higher, there is the following negative result (see [Mairhuber (1956), Curtis (1959)]).

Theorem (Mairhuber-Curtis)

If $\Omega \subset \mathbb{R}^s$, $s \geq 2$, contains an interior point, then there exist no Haar spaces of continuous functions except for one-dimensional ones.

In order to understand this theorem we need

Definition

Let the finite-dimensional linear function space $\mathcal{B} \subseteq C(\Omega)$ have a basis $\{B_1, \ldots, B_N\}$. Then $\mathcal{B}$ is a *Haar space* on Ω if

$$\det A \neq 0$$

for any set of distinct $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ in Ω. Here A is the matrix with entries $A_{jk} = B_k(\boldsymbol{x}_j)$.

In order to understand this theorem we need

Definition

Let the finite-dimensional linear function space $\mathcal{B} \subseteq C(\Omega)$ have a basis $\{B_1, \ldots, B_N\}$. Then $\mathcal{B}$ is a *Haar space* on Ω if

$$\det A \neq 0$$

for any set of distinct $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ in Ω. Here A is the matrix with entries $A_{jk} = B_k(\boldsymbol{x}_j)$.

Existence of a Haar space guarantees invertibility of the interpolation matrix A, *i.e.*, existence and uniqueness of an interpolant of the form (1) to data specified at $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from the space $\mathcal{B}$.

In order to understand this theorem we need

Definition

Let the finite-dimensional linear function space $\mathcal{B} \subseteq C(\Omega)$ have a basis $\{B_1, \ldots, B_N\}$. Then $\mathcal{B}$ is a *Haar space* on Ω if

$$\det A \neq 0$$

for any set of distinct $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ in Ω. Here A is the matrix with entries $A_{jk} = B_k(\boldsymbol{x}_j)$.

Existence of a Haar space guarantees invertibility of the interpolation matrix A, *i.e.*, existence and uniqueness of an interpolant of the form (1) to data specified at $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from the space $\mathcal{B}$.

Example

Univariate polynomials of degree $N-1$ form an N-dimensional Haar space for data given at $x_1, \ldots, x_N$.

Interpretation of Mairhuber-Curtis

The Mairhuber-Curtis theorem tells us that if we want to have a well-posed multivariate scattered data interpolation problem we can no longer fix in advance the set of basis functions we plan to use for interpolation of arbitrary scattered data.

Interpretation of Mairhuber-Curtis

The Mairhuber-Curtis theorem tells us that if we want to have a well-posed multivariate scattered data interpolation problem we can no longer fix in advance the set of basis functions we plan to use for interpolation of arbitrary scattered data.

Instead, the basis should depend on the data locations.

Interpretation of Mairhuber-Curtis

The Mairhuber-Curtis theorem tells us that if we want to have a well-posed multivariate scattered data interpolation problem we can no longer fix in advance the set of basis functions we plan to use for interpolation of arbitrary scattered data.

Instead, the basis should depend on the data locations.

Example

It is not possible to perform unique interpolation with (multivariate) polynomials of degree N to data given at arbitrary locations in $\mathbb{R}^2$.

Proof of Mairhuber-Curtis

Proof of Theorem 2.

Let $s \geq 2$ and assume that $\mathcal{B}$ is a Haar space with basis $\{B_1, \ldots, B_N\}$ with $N \geq 2$.

Proof of Mairhuber-Curtis

Proof of Theorem 2.

Let $s \geq 2$ and assume that $\mathcal{B}$ is a Haar space with basis $\{B_1, \ldots, B_N\}$ with $N \geq 2$.

We need to show that this leads to a contradiction.

Proof of Mairhuber-Curtis

Proof of Theorem 2.

Let $s \geq 2$ and assume that $\mathcal{B}$ is a Haar space with basis $\{B_1, \ldots, B_N\}$ with $N \geq 2$.
We need to show that this leads to a contradiction.
By the definition of a Haar space

$$\det \left(B_k(\boldsymbol{x}_j) \right) \neq 0 \tag{2}$$

for any distinct $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.

Proof of Mairhuber-Curtis

Proof of Theorem 2.

Let $s \geq 2$ and assume that $\mathcal{B}$ is a Haar space with basis $\{B_1, \ldots, B_N\}$ with $N \geq 2$.
We need to show that this leads to a contradiction.
By the definition of a Haar space

$$\det \left(B_k(\boldsymbol{x}_j)\right) \neq 0 \tag{2}$$

for any distinct $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.

▸ Mairhuber-Curtis Maplet

Proof of Mairhuber-Curtis

Proof of Theorem 2.

Let $s \geq 2$ and assume that $\mathcal{B}$ is a Haar space with basis $\{B_1, \ldots, B_N\}$ with $N \geq 2$.
We need to show that this leads to a contradiction.
By the definition of a Haar space

$$\det \left(B_k(\boldsymbol{x}_j) \right) \neq 0 \tag{2}$$

for any distinct $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.

Mairhuber-Curtis Maplet

Since the determinant is a continuous function of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ we must have had $\det = 0$ at some point along path. This contradicts (2). □

We want to construct a (continuous) function $\mathcal{P}_f$ that interpolates samples obtained from a test function f_s data sites $\boldsymbol{x}_j \in [0,1]^s$, *i.e.*, want

$$\mathcal{P}_f(\boldsymbol{x}_j) = f_s(\boldsymbol{x}_j), \qquad \boldsymbol{x}_j \in [0,1]^s$$

We want to construct a (continuous) function $\mathcal{P}_f$ that interpolates samples obtained from a test function f_s data sites $\boldsymbol{x}_j \in [0, 1]^s$, *i.e.*, want

$$\mathcal{P}_f(\boldsymbol{x}_j) = f_s(\boldsymbol{x}_j), \qquad \boldsymbol{x}_j \in [0, 1]^s$$

Assume for now that $s = 1$.

- For small N one can use univariate polynomials
- If N is relatively large it's better to use splines
- Simplest approach: C^0 piecewise linear splines ("connect the dots")

We want to construct a (continuous) function $\mathcal{P}_f$ that interpolates samples obtained from a test function f_s data sites $\boldsymbol{x}_j \in [0,1]^s$, *i.e.*, want

$$\mathcal{P}_f(\boldsymbol{x}_j) = f_s(\boldsymbol{x}_j), \qquad \boldsymbol{x}_j \in [0,1]^s$$

Assume for now that $s = 1$.

- For small N one can use univariate polynomials
- If N is relatively large it's better to use splines
- Simplest approach: C^0 piecewise linear splines ("connect the dots")

Basis for space of piecewise linear interpolating splines:

$$\{B_k = |\cdot - x_k| : \; k = 1, \ldots, N\}$$

So

$$\mathcal{P}_f(x) = \sum_{k=1}^{N} c_k |x - x_k|, \qquad x \in [0,1]$$

and c_k determined by interpolation conditions

$$\mathcal{P}_f(x_j) = f_1(x_j), \qquad j = 1, \ldots, N$$

- Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites x_k as suggested by Mairhuber-Curtis

Norm Maplet

- Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites x_k as suggested by Mairhuber-Curtis

Norm Maplet

- $B(x) = |x|$ is called basic function

- Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites x_k as suggested by Mairhuber-Curtis

Norm Maplet

- $B(x) = |x|$ is called basic function
- The points x_k to which the basic function is shifted to form the basis functions are usually referred to as centers or *knots*.

- Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites x_k as suggested by Mairhuber-Curtis

Norm Maplet

- $B(x) = |x|$ is called basic function
- The points x_k to which the basic function is shifted to form the basis functions are usually referred to as centers or *knots*.
- Technically, one could choose these centers different from the data sites. However, usually centers coincide with the data sites.

- Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites x_k as suggested by Mairhuber-Curtis

Norm Maplet

- $B(x) = |x|$ is called basic function
- The points x_k to which the basic function is shifted to form the basis functions are usually referred to as centers or *knots*.
- Technically, one could choose these centers different from the data sites. However, usually centers coincide with the data sites.
- This simplifies the analysis of the method, and is sufficient for many applications.

- Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites x_k as suggested by Mairhuber-Curtis

▸ Norm Maplet

- $B(x) = |x|$ is called basic function
- The points x_k to which the basic function is shifted to form the basis functions are usually referred to as centers or *knots*.
- Technically, one could choose these centers different from the data sites. However, usually centers coincide with the data sites.
- This simplifies the analysis of the method, and is sufficient for many applications.
- In fact, relatively little is known about the case when centers and data sites differ.

- Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites x_k as suggested by Mairhuber-Curtis

▸ Norm Maplet

- $B(x) = |x|$ is called basic function
- The points x_k to which the basic function is shifted to form the basis functions are usually referred to as centers or *knots*.
- Technically, one could choose these centers different from the data sites. However, usually centers coincide with the data sites.
- This simplifies the analysis of the method, and is sufficient for many applications.
- In fact, relatively little is known about the case when centers and data sites differ.
- B_k are (radially) symmetric about their centers x_k
  $\longrightarrow$ radial basis function

Now the coefficients c_k in the scattered data interpolation problem are found by solving the linear system

$$\begin{bmatrix} |x_1 - x_1| & |x_1 - x_2| & \dots & |x_1 - x_N| \\ |x_2 - x_1| & |x_2 - x_2| & \dots & |x_2 - x_N| \\ \vdots & \vdots & \ddots & \vdots \\ |x_N - x_1| & |x_N - x_2| & \dots & |x_N - x_N| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix} \quad (3)$$

Now the coefficients c_k in the scattered data interpolation problem are found by solving the linear system

$$\begin{bmatrix} |x_1 - x_1| & |x_1 - x_2| & \dots & |x_1 - x_N| \\ |x_2 - x_1| & |x_2 - x_2| & \dots & |x_2 - x_N| \\ \vdots & \vdots & \ddots & \vdots \\ |x_N - x_1| & |x_N - x_2| & \dots & |x_N - x_N| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix} \quad (3)$$

- The matrix in (3) is a distance matrix

Now the coefficients c_k in the scattered data interpolation problem are found by solving the linear system

$$\begin{bmatrix} |x_1 - x_1| & |x_1 - x_2| & \dots & |x_1 - x_N| \\ |x_2 - x_1| & |x_2 - x_2| & \dots & |x_2 - x_N| \\ \vdots & \vdots & \ddots & \vdots \\ |x_N - x_1| & |x_N - x_2| & \dots & |x_N - x_N| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix} \tag{3}$$

- The matrix in (3) is a distance matrix
- Distance matrices have been studied in geometry and analysis in the context of isometric embeddings of metric spaces for a long time (see, *e.g.*, [Baxter (1991), Blumenthal (1938), Bochner (1941), Micchelli (1986), Schoenberg (1938a), Wells and Williams (1975)]).

Now the coefficients c_k in the scattered data interpolation problem are found by solving the linear system

$$\begin{bmatrix} |x_1 - x_1| & |x_1 - x_2| & \dots & |x_1 - x_N| \\ |x_2 - x_1| & |x_2 - x_2| & \dots & |x_2 - x_N| \\ \vdots & \vdots & \ddots & \vdots \\ |x_N - x_1| & |x_N - x_2| & \dots & |x_N - x_N| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix} \quad (3)$$

- The matrix in (3) is a distance matrix
- Distance matrices have been studied in geometry and analysis in the context of isometric embeddings of metric spaces for a long time (see, *e.g.*, [Baxter (1991), Blumenthal (1938), Bochner (1941), Micchelli (1986), Schoenberg (1938a), Wells and Williams (1975)]).
- It is known that the distance matrix based on the Euclidean distance between a set of distinct points in $\mathbb{R}^s$ is always non-singular (see below).

Now the coefficients c_k in the scattered data interpolation problem are found by solving the linear system

$$\begin{bmatrix} |x_1 - x_1| & |x_1 - x_2| & \dots & |x_1 - x_N| \\ |x_2 - x_1| & |x_2 - x_2| & \dots & |x_2 - x_N| \\ \vdots & \vdots & \ddots & \vdots \\ |x_N - x_1| & |x_N - x_2| & \dots & |x_N - x_N| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix} \quad (3)$$

- The matrix in (3) is a distance matrix
- Distance matrices have been studied in geometry and analysis in the context of isometric embeddings of metric spaces for a long time (see, *e.g.*, [Baxter (1991), Blumenthal (1938), Bochner (1941), Micchelli (1986), Schoenberg (1938a), Wells and Williams (1975)]).
- It is known that the distance matrix based on the Euclidean distance between a set of distinct points in $\mathbb{R}^s$ is always non-singular (see below).
- Therefore, our scattered data interpolation problem is well-posed.

Since distance matrices are non-singular for Euclidean distances in any space dimension s we have an immediate generalization:

Since distance matrices are non-singular for Euclidean distances in any space dimension s we have an immediate generalization:
For the scattered data interpolation problem on $[0,1]^s$ we can take

$$\mathcal{P}_f(\boldsymbol{x}) = \sum_{k=1}^{N} c_k \|\boldsymbol{x} - \boldsymbol{x}_k\|_2, \qquad \boldsymbol{x} \in [0,1]^s, \tag{4}$$

and find the c_k by solving

$$\begin{bmatrix} \|\boldsymbol{x}_1 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 & \cdots & \|\boldsymbol{x}_1 - \boldsymbol{x}_N\|_2 \\ \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_2 - \boldsymbol{x}_2\|_2 & \cdots & \|\boldsymbol{x}_2 - \boldsymbol{x}_N\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\boldsymbol{x}_N - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_N - \boldsymbol{x}_2\|_2 & \cdots & \|\boldsymbol{x}_N - \boldsymbol{x}_N\|_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_s(\boldsymbol{x}_1) \\ f_s(\boldsymbol{x}_2) \\ \vdots \\ f_s(\boldsymbol{x}_N) \end{bmatrix}.$$

Since distance matrices are non-singular for Euclidean distances in any space dimension s we have an immediate generalization:
For the scattered data interpolation problem on $[0,1]^s$ we can take

$$\mathcal{P}_f(\boldsymbol{x}) = \sum_{k=1}^{N} c_k \|\boldsymbol{x} - \boldsymbol{x}_k\|_2, \qquad \boldsymbol{x} \in [0,1]^s, \tag{4}$$

and find the c_k by solving

$$\begin{bmatrix} \|\boldsymbol{x}_1 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 & \dots & \|\boldsymbol{x}_1 - \boldsymbol{x}_N\|_2 \\ \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_2 - \boldsymbol{x}_2\|_2 & \dots & \|\boldsymbol{x}_2 - \boldsymbol{x}_N\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\boldsymbol{x}_N - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_N - \boldsymbol{x}_2\|_2 & \dots & \|\boldsymbol{x}_N - \boldsymbol{x}_N\|_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_s(\boldsymbol{x}_1) \\ f_s(\boldsymbol{x}_2) \\ \vdots \\ f_s(\boldsymbol{x}_N) \end{bmatrix}.$$

- Note that the basis is again data dependent

Since distance matrices are non-singular for Euclidean distances in any space dimension s we have an immediate generalization:
For the scattered data interpolation problem on $[0,1]^s$ we can take

$$\mathcal{P}_f(\boldsymbol{x}) = \sum_{k=1}^{N} c_k \|\boldsymbol{x} - \boldsymbol{x}_k\|_2, \qquad \boldsymbol{x} \in [0,1]^s, \tag{4}$$

and find the c_k by solving

$$\begin{bmatrix} \|\boldsymbol{x}_1 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 & \cdots & \|\boldsymbol{x}_1 - \boldsymbol{x}_N\|_2 \\ \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_2 - \boldsymbol{x}_2\|_2 & \cdots & \|\boldsymbol{x}_2 - \boldsymbol{x}_N\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\boldsymbol{x}_N - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_N - \boldsymbol{x}_2\|_2 & \cdots & \|\boldsymbol{x}_N - \boldsymbol{x}_N\|_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_s(\boldsymbol{x}_1) \\ f_s(\boldsymbol{x}_2) \\ \vdots \\ f_s(\boldsymbol{x}_N) \end{bmatrix}.$$

- Note that the basis is again data dependent
- Piecewise linear splines in higher space dimensions are usually constructed differently (via a cardinal basis on an underlying computational mesh)

Since distance matrices are non-singular for Euclidean distances in any space dimension s we have an immediate generalization:
For the scattered data interpolation problem on $[0,1]^s$ we can take

$$\mathcal{P}_f(\boldsymbol{x}) = \sum_{k=1}^{N} c_k \|\boldsymbol{x} - \boldsymbol{x}_k\|_2, \qquad \boldsymbol{x} \in [0,1]^s, \tag{4}$$

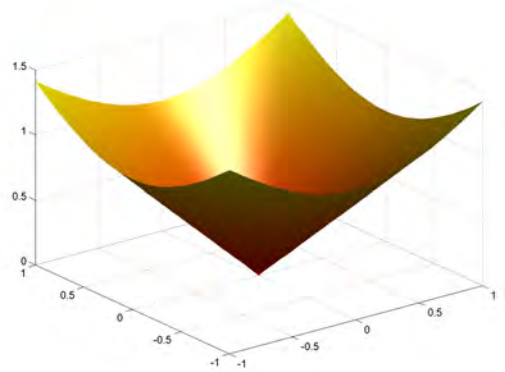
and find the c_k by solving

$$\begin{bmatrix} \|\boldsymbol{x}_1 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 & \dots & \|\boldsymbol{x}_1 - \boldsymbol{x}_N\|_2 \\ \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_2 - \boldsymbol{x}_2\|_2 & \dots & \|\boldsymbol{x}_2 - \boldsymbol{x}_N\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\boldsymbol{x}_N - \boldsymbol{x}_1\|_2 & \|\boldsymbol{x}_N - \boldsymbol{x}_2\|_2 & \dots & \|\boldsymbol{x}_N - \boldsymbol{x}_N\|_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_s(\boldsymbol{x}_1) \\ f_s(\boldsymbol{x}_2) \\ \vdots \\ f_s(\boldsymbol{x}_N) \end{bmatrix}.$$

- Note that the basis is again data dependent
- Piecewise linear splines in higher space dimensions are usually constructed differently (via a cardinal basis on an underlying computational mesh)
- For $s > 1$ the space $\text{span}\{\|\cdot - \boldsymbol{x}_k\|_2,\ k = 1, \dots, N\}$ is not the same as piecewise linear splines

Norm RBF

A typical basis function for the Euclidean distance matrix fit, $B_k(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_k\|_2$ with $\boldsymbol{x}_k = \boldsymbol{0}$ and $s = 2$.



In order to show the non-singularity of our distance matrices we use the Courant-Fischer theorem (see *e.g.*, [Meyer (2000)]):

Theorem

Let A be a real symmetric $N \times N$ matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$, then

$$\lambda_k = \max_{\dim \mathcal{V}=k} \min_{\substack{\boldsymbol{x}\in\mathcal{V} \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T A \boldsymbol{x} \quad \text{and} \quad \lambda_k = \min_{\dim \mathcal{V}=N-k+1} \max_{\substack{\boldsymbol{x}\in\mathcal{V} \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T A \boldsymbol{x}.$$

In order to show the non-singularity of our distance matrices we use the Courant-Fischer theorem (see *e.g.*, [Meyer (2000)]):

Theorem

Let A be a real symmetric $N \times N$ matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$, then

$$\lambda_k = \max_{\dim \mathcal{V}=k} \min_{\substack{\boldsymbol{x}\in\mathcal{V} \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T A \boldsymbol{x} \quad \text{and} \quad \lambda_k = \min_{\dim \mathcal{V}=N-k+1} \max_{\substack{\boldsymbol{x}\in\mathcal{V} \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T A \boldsymbol{x}.$$

Definition

A real symmetric matrix A is called *conditionally negative definite of order one* (or *almost negative definite*) if its associated quadratic form is negative, *i.e.*

$$\sum_{j=1}^{N} \sum_{k=1}^{N} c_j c_k A_{jk} < 0 \qquad (5)$$

for all $\boldsymbol{c} = [c_1, \ldots, c_N]^T \neq \boldsymbol{0} \in \mathbb{R}^N$ that satisfy $\sum_{j=1}^{N} c_j = 0$.

Now we have

Theorem

An $N \times N$ matrix A which is almost negative definite and has a non-negative trace possesses one positive and $N-1$ negative eigenvalues.

Now we have

Theorem

An $N \times N$ matrix A which is almost negative definite and has a non-negative trace possesses one positive and $N-1$ negative eigenvalues.

Proof.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ denote the eigenvalues of A. From the Courant-Fischer theorem we get

$$\lambda_2 = \min_{\dim \mathcal{V}=N-1} \max_{\substack{\boldsymbol{x}\in\mathcal{V} \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T A \boldsymbol{x} \leq \max_{\substack{\boldsymbol{c}:\ \sum c_k=0 \\ \|\boldsymbol{c}\|=1}} \boldsymbol{c}^T A \boldsymbol{c} < 0,$$

so that A has at least $N-1$ negative eigenvalues.

Now we have

Theorem

An $N \times N$ matrix A which is almost negative definite and has a non-negative trace possesses one positive and $N-1$ negative eigenvalues.

Proof.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ denote the eigenvalues of A. From the Courant-Fischer theorem we get

$$\lambda_2 = \min_{\dim \mathcal{V}=N-1} \max_{\substack{\boldsymbol{x}\in\mathcal{V} \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T A \boldsymbol{x} \leq \max_{\substack{\boldsymbol{c}:\ \sum c_k=0 \\ \|\boldsymbol{c}\|=1}} \boldsymbol{c}^T A \boldsymbol{c} < 0,$$

so that A has at least $N-1$ negative eigenvalues.
But since $\operatorname{tr}(A) = \sum_{k=1}^{N} \lambda_k \geq 0$, A also must have at least one positive eigenvalue. □

Non-singularity of distance matrix

It is known that $\varphi(r) = r$ is a strictly conditionally negative definite function of order one, i.e., the matrix A with $A_{jk} = \|\boldsymbol{x}_j - \boldsymbol{x}_k\|$ is almost negative definite.

Non-singularity of distance matrix

It is known that $\varphi(r) = r$ is a strictly conditionally negative definite function of order one, i.e., the matrix A with $A_{jk} = \|\boldsymbol{x}_j - \boldsymbol{x}_k\|$ is almost negative definite.

Also, since $A_{jj} = \varphi(0) = 0$, $j = 1, \ldots, N$, implies $\mathrm{tr}(A) = 0$.

Non-singularity of distance matrix

It is known that $\varphi(r) = r$ is a strictly conditionally negative definite function of order one, i.e., the matrix A with $A_{jk} = \|\boldsymbol{x}_j - \boldsymbol{x}_k\|$ is almost negative definite.

Also, since $A_{jj} = \varphi(0) = 0$, $j = 1, \ldots, N$, implies $\mathrm{tr}(A) = 0$.

Therefore, our distance matrix is non-singular by the above theorem.

One of our main MATLAB subroutines

- Forms the matrix of pairwise Euclidean distances of two (possibly different) sets of points in $\mathbb{R}^s$ (`dsites` and `ctrs`).

```
function DM = DistanceMatrix(dsites,ctrs)
[M,s] = size(dsites); [N,s] = size(ctrs);
DM = zeros(M,N);
for d=1:s
    [dr,cc] = ndgrid(dsites(:,d),ctrs(:,d));
    DM = DM + (dr-cc).^2;
end
DM = sqrt(DM);
```

One of our main MATLAB subroutines

- Forms the matrix of pairwise Euclidean distances of two (possibly different) sets of points in $\mathbb{R}^s$ (`dsites` and `ctrs`).

```
function DM = DistanceMatrix(dsites,ctrs)
[M,s] = size(dsites); [N,s] = size(ctrs);
DM = zeros(M,N);
for d=1:s
    [dr,cc] = ndgrid(dsites(:,d),ctrs(:,d));
    DM = DM + (dr-cc).^2;
end
DM = sqrt(DM);
```

```
>> [dr,cc] = ndgrid([0 1 2 3],[4 5 6 7])
dr =
     0     0     0     0
     1     1     1     1
     2     2     2     2
     3     3     3     3
cc =
     4     5     6     7
     4     5     6     7
     4     5     6     7
     4     5     6     7
```

One of our main MATLAB subroutines

- Forms the matrix of pairwise Euclidean distances of two (possibly different) sets of points in $\mathbb{R}^s$ (`dsites` and `ctrs`).

```
function DM = DistanceMatrix(dsites,ctrs)
[M,s] = size(dsites); [N,s] = size(ctrs);
DM = zeros(M,N);
for d=1:s
    [dr,cc] = ndgrid(dsites(:,d),ctrs(:,d));
    DM = DM + (dr-cc).^2;
end
DM = sqrt(DM);
```

Works for any space dimension!

Alternate forms of `DistanceMatrix.m`

Program (`DistanceMatrixA.m`)

```
function DM = DistanceMatrixA(dsites,ctrs)
[M,s] = size(dsites); [N,s] = size(ctrs);
DM = zeros(M,N);
for d=1:s
    DM = DM + (repmat(dsites(:,d),1,N) - ...
                    repmat(ctrs(:,d)',M,1)).^2;
end
DM = sqrt(DM);
```

Note: uses less memory than the `ndgrid`-based version

Remark

Both of these subroutines can easily be modified to produce a p-norm distance matrix by making the obvious changes to the code.

Alternate forms of `DistanceMatrix.m` (cont.)

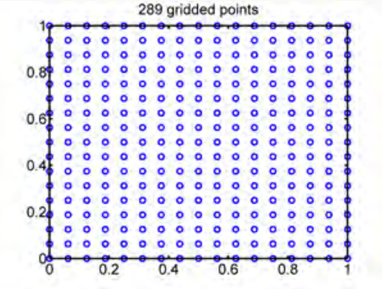
Program (`DistanceMatrixB.m`)

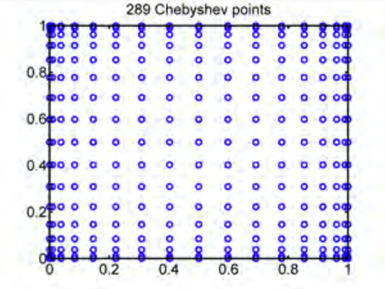
```
function DM = DistanceMatrixB(dsites,ctrs)
M = size(dsites,1); N = size(ctrs,1);
DM = repmat(sum(dsites.*dsites,2),1,N) - ...
        2*dsites*ctrs' + ...
        repmat((sum(ctrs.*ctrs,2))',M,1);
DM = sqrt(DM);
```

Note: For 2-norm distance only. Basic idea suggested by a former student – fast and memory efficient since no `for`-loop used

Depending on the type of approximation problem we are given, we may or may not be able to select where the data is collected, i.e., the location of the data sites or *design*.
Standard choices in low space dimensions include

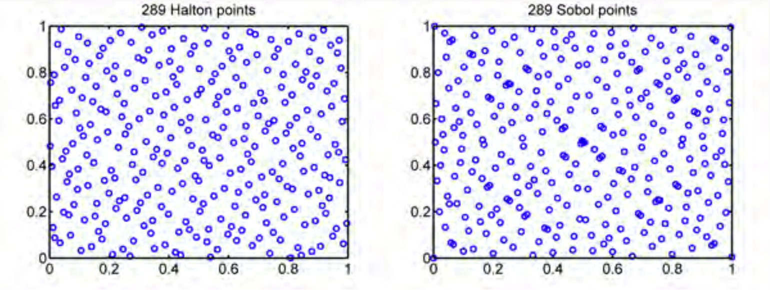
- tensor products of equally spaced points
- tensor products of Chebyshev points





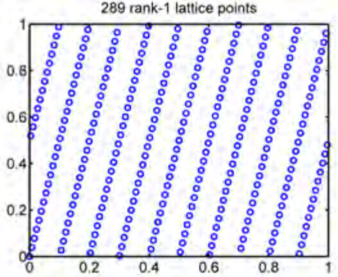
In higher space dimensions it is important to have space-filling (or low-discrepancy) quasi-random point sets. Examples include

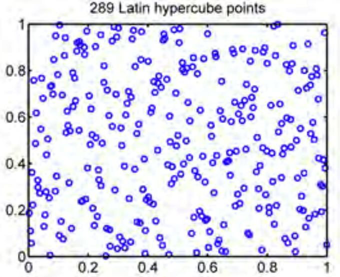
- Halton points ▸ more info
- Sobol' points
- lattice designs
- Latin hypercube designs
- and quite a few others (digital nets, Faure, Niederreiter, etc.)



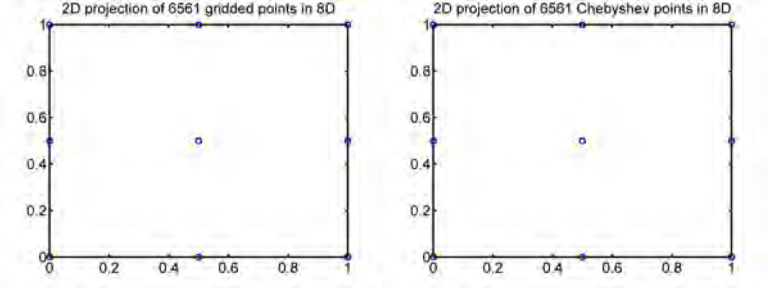
In higher space dimensions it is important to have space-filling (or low-discrepancy) quasi-random point sets. Examples include

- Halton points ▸ more info
- Sobol' points
- lattice designs
- Latin hypercube designs
- and quite a few others (digital nets, Faure, Niederreiter, etc.)

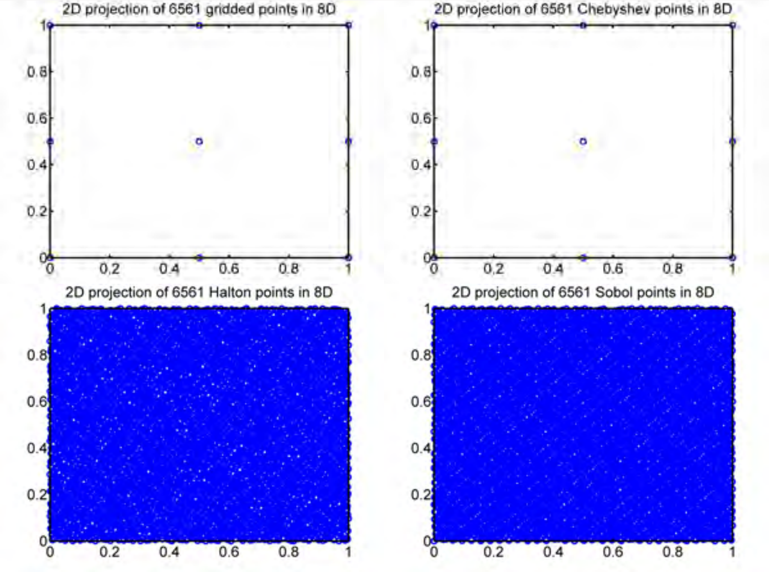




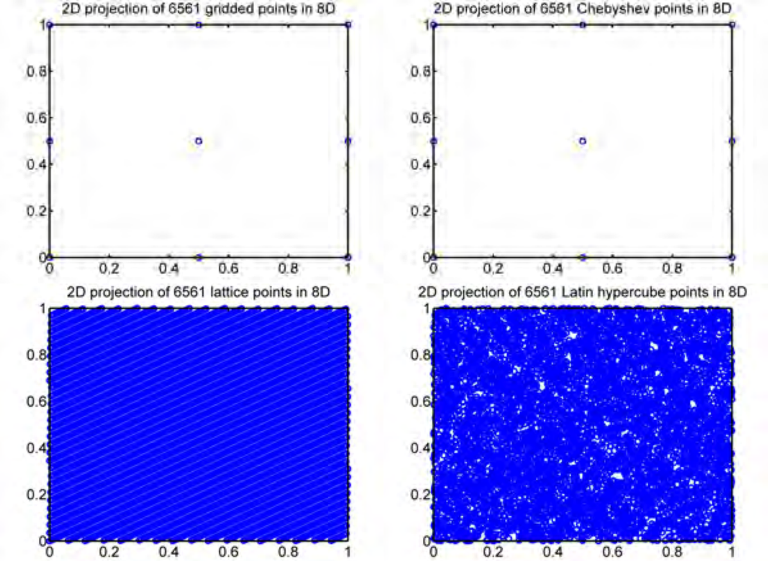
The difference between the standard (tensor product) designs and the quasi-random designs shows especially in higher space dimensions:



The difference between the standard (tensor product) designs and the quasi-random designs shows especially in higher space dimensions:



The difference between the standard (tensor product) designs and the quasi-random designs shows especially in higher space dimensions:



Program (DistanceMatrixFit.m)

```
1   s = 3;
2   k = 2; N = (2^k+1)^s;
3   neval = 10; M = neval^s;
4   dsites = CreatePoints(N,s,'h');
5   ctrs = dsites;
6   epoints = CreatePoints(M,s,'u');
7   rhs = testfunctionsD(dsites);
8   IM = DistanceMatrix(dsites,ctrs);
9   EM = DistanceMatrix(epoints,ctrs);
10  Pf = EM * (IM\rhs);
11  exact = testfunctionsD(epoints);
12  maxerr = norm(Pf-exact,inf)
13  rms_err = norm(Pf-exact)/sqrt(M)
```

Note the simultaneous evaluation of the interpolant at the entire set of evaluation points on line 10.

Root-mean-square error:

$$\text{RMS-error} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left[\mathcal{P}_f(\xi_j) - f(\xi_j)\right]^2} = \frac{1}{\sqrt{M}}\|\mathcal{P}_f - f\|_2, \qquad (6)$$

where the ξ_j, $j = 1, \ldots, M$ are the *evaluation points*.

Root-mean-square error:

$$\text{RMS-error} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left[\mathcal{P}_f(\xi_j) - f(\xi_j)\right]^2} = \frac{1}{\sqrt{M}}\|\mathcal{P}_f - f\|_2, \qquad (6)$$

where the ξ_j, $j = 1, \ldots, M$ are the *evaluation points*.

Remark

The basic MATLAB *code for the solution of any kind of RBF interpolation problem will be very similar to* `DistanceMatrixFit`*. Moreover, the data used — even for the distance matrix interpolation considered here — can also be "real" data. Just replace lines 4 and 7 by code that generates the data sites and data values for the right-hand side.*

Instead of reading points from files as in the book

```
function [points, N] = CreatePoints(N,s,gridtype)
% Computes a set of N points in [0,1]^s
% Note: could add variable interval later
% Inputs:
% N: number of interpolation points
% s: space dimension
% gridtype: 'c'=Chebyshev, 'f'=fence(rank-1 lattice),
%    'h'=Halton, 'l'=latin hypercube, 'r'=random uniform,
%    's'=Sobol, 'u'=uniform
% Outputs:
% points: an Nxs matrix (each row contains one s-D point)
% N: might be slightly less than original N for
%    Chebyshev and gridded uniform points
% Calls on: chebsamp, lattice, haltonseq, lhsamp, i4_sobol,
%           gridsamp
% Also needs: fdnodes,gaussj,i4_bit_hi1,i4_bit_lo0,i4_xor
```

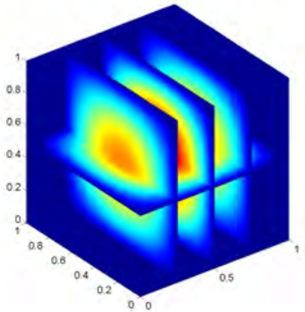
Credits: Hans Bruun Nielsen [DACE], Toby Driscoll, Fred Hickernell, Daniel Dougherty, John Burkardt

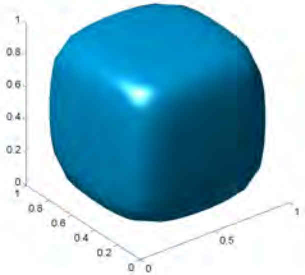
Test function

$$f_s(\boldsymbol{x}) = 4^s \prod_{d=1}^{s} x_d(1 - x_d), \qquad \boldsymbol{x} = (x_1, \ldots, x_s) \in [0,1]^s$$

Program

```
function tf = testfunctionsD(x)
[N,s] = size(x);
tf = 4^s*prod(x.*(1-x),2);
```





The tables and figures below show some examples computed with `DistanceMatrixFit`.

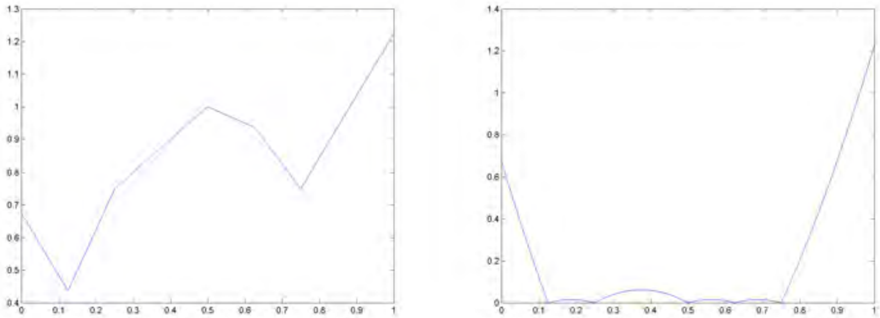
The number M of evaluation points for $s = 1, 2, \ldots, 6$, was 1000, 1600, 1000, 256, 1024, and 4096, respectively (*i.e.*, `neval` = 1000, 40, 10, 4, 4, and 4, respectively).

Note that, as the space dimension s increases, more and more of the (uniformly gridded) evaluation points lie on the boundary of the domain, while the data sites (which are given as Halton points) are located in the interior of the domain.

The value k listed in the tables is the same as the `k` in line 2 of `DistanceMatrixFit`.

	1D		2D		3D	
k	N	RMS-error	N	RMS-error	N	RMS-error
1	3	5.896957e-001	9	1.937341e-001	27	9.721476e-002
2	5	3.638027e-001	25	6.336315e-002	125	6.277141e-002
3	9	1.158328e-001	81	2.349093e-002	729	2.759452e-002
4	17	3.981270e-002	289	1.045010e-002		
5	33	1.406188e-002	1089	4.326940e-003		
6	65	5.068541e-003	4225	1.797430e-003		
7	129	1.877013e-003				
8	257	7.264159e-004				
9	513	3.016376e-004				
10	1025	1.381896e-004				
11	2049	6.907386e-005				
12	4097	3.453179e-005				

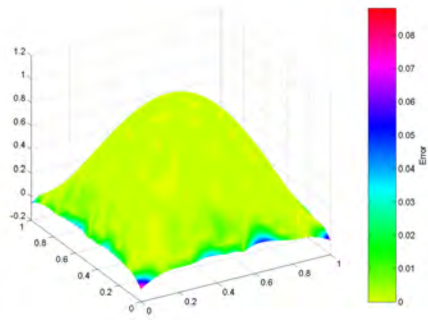
	4D		5D		6D	
k	N	RMS-error	N	RMS-error	N	RMS-error
1	81	1.339581e-001	243	9.558350e-002	729	5.097600e-002
2	625	6.817424e-002	3125	3.118905e-002		

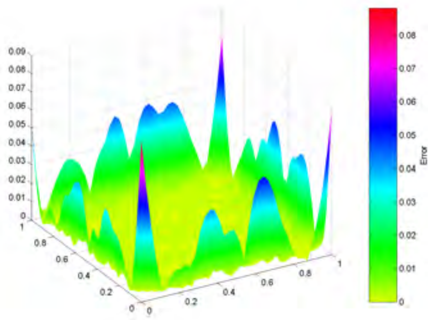


Left: distance matrix fit for $s = 1$ with 5 Halton points for f_1
Right: corresponding error

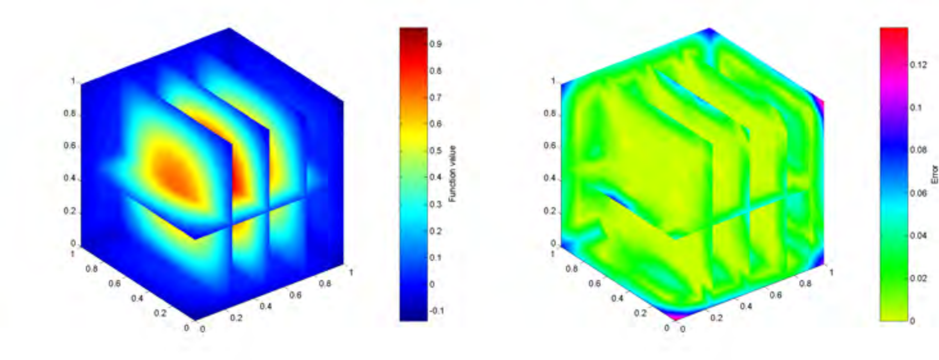
Remark

Note the piecewise linear nature of the interpolant. If we use more points then the fit becomes more accurate (see table) but then we can't recognize the piecewise linear nature of the interpolant.





Left: distance matrix fit for $s = 2$ with 289 Halton points for f_2
Right: corresponding error
Interpolant is false-colored according to absolute error



Left: distance matrix fit for $s = 3$ with 729 Halton points for f_3 (colors represent function values)
Right: corresponding error

Remark

We can see clearly that most of the error is concentrated near the boundary of the domain.

In fact, the absolute error is about one order of magnitude larger near the boundary than it is in the interior of the domain.

This is no surprise since the data sites are located in the interior.

Even for uniformly spaced data sites (including points on the boundary) the main error in radial basis function interpolation is usually located near the boundary.

Observations

From this first simple example we can observe a number of other features. Most of them are characteristic for the radial basis function interpolants.

- The basis functions $B_k = \|\cdot - \boldsymbol{x}_k\|_2$ are radially symmetric.
- As the MATLAB scripts show, the method is extremely simple to implement for any space dimension s.
  - No underlying computational mesh is required to compute the interpolant. The process of mesh generation is a major factor when working in higher space dimensions with polynomial-based methods such as splines or finite elements.
  - All that is required for our method is the pairwise distance between the data sites. Therefore, we have what is known as a meshfree (or *meshless*) method.

Observations (cont.)

- The accuracy of the method improves if we add more data sites.
  - It seems that the RMS-error in the tables above are reduced by a factor of about two from one row to the next.
  - Since we use $(2^k + 1)^s$ uniformly distributed random data points in row k this indicates a convergence rate of roughly $\mathcal{O}(h)$, where h can be viewed as something like the average distance or meshsize of the set $\mathcal{X}$ of data sites.
- The interpolant used here (as well as many other radial basis function interpolants used later) requires the solution of a system of linear equations with a dense $N \times N$ matrix. This makes it very costly to apply the method in its simple form to large data sets.
- Moreover, as we will see later, these matrices also tend to be rather ill-conditioned.

General Radial Basis Function Interpolation

Use data-dependent linear function space

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

Here $\varphi : [0, \infty) \to \mathbb{R}$ is strictly positive definite and radial

General Radial Basis Function Interpolation

Use data-dependent linear function space

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

Here $\varphi : [0, \infty) \to \mathbb{R}$ is strictly positive definite and radial

To find c_j solve interpolation equations

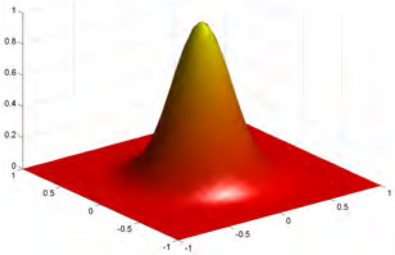
$$\mathcal{P}_f(\boldsymbol{x}_i) = f(\boldsymbol{x}_i), \quad i = 1, \ldots, N$$

Leads to linear system with matrix

$$A_{ij} = \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \quad i, j = 1, \ldots, N$$

Radial Basic Functions

$\varphi(r)$	Name
$e^{-(\varepsilon r)^2}$	Gaussian
$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$	inverse MQ
$\sqrt{1+(\varepsilon r)^2}$	multiquadric
$r^2 \log r$	thin plate spline
$(1-r)_+^4(4r+1)$	Wendland CSRBF

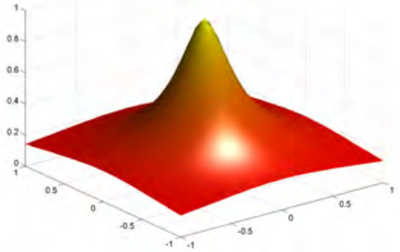


$r = \|\boldsymbol{x} - \boldsymbol{x}_k\|$ (radial distance)
ε (positive shape parameter)

Radial Basic Functions

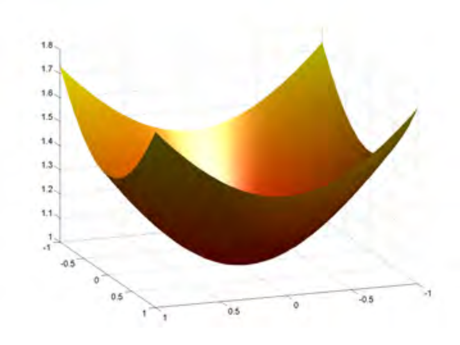
$\varphi(r)$	Name
$e^{-(\varepsilon r)^2}$	Gaussian
$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$	inverse MQ
$\sqrt{1+(\varepsilon r)^2}$	multiquadric
$r^2 \log r$	thin plate spline
$(1-r)_+^4(4r+1)$	Wendland CSRBF

$r = \|\boldsymbol{x} - \boldsymbol{x}_k\|$ (radial distance)
ε (positive shape parameter)



Radial Basic Functions

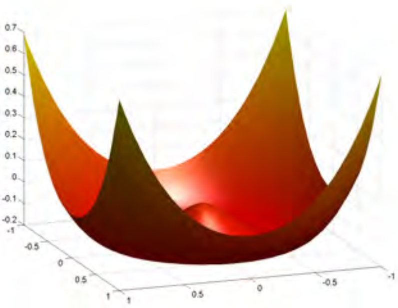
$\varphi(r)$	Name
$e^{-(\varepsilon r)^2}$	Gaussian
$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$	inverse MQ
$\sqrt{1+(\varepsilon r)^2}$	multiquadric
$r^2 \log r$	thin plate spline
$(1-r)_+^4(4r+1)$	Wendland CSRBF



$r = \|\boldsymbol{x} - \boldsymbol{x}_k\|$ (radial distance)
ε (positive shape parameter)

Radial Basic Functions

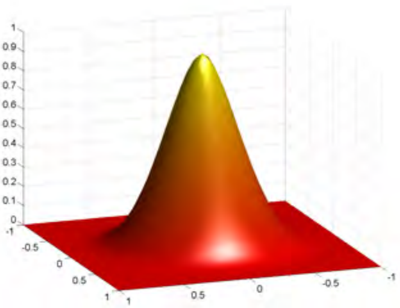
$\varphi(r)$	Name
$e^{-(\varepsilon r)^2}$	Gaussian
$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$	inverse MQ
$\sqrt{1+(\varepsilon r)^2}$	multiquadric
$r^2 \log r$	thin plate spline
$(1-r)_+^4(4r+1)$	Wendland CSRBF



$r = \|\boldsymbol{x} - \boldsymbol{x}_k\|$ (radial distance)
ε (positive shape parameter)

Radial Basic Functions

$\varphi(r)$	Name
$e^{-(\varepsilon r)^2}$	Gaussian
$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$	inverse MQ
$\sqrt{1+(\varepsilon r)^2}$	multiquadric
$r^2 \log r$	thin plate spline
$(1-r)_+^4(4r+1)$	Wendland CSRBF



$r = \|\boldsymbol{x} - \boldsymbol{x}_k\|$ (radial distance)
ε (positive shape parameter)

```
function rbf_definition
global rbf
%%% CPD0
  rbf = @(ep,r) exp(-(ep*r).^2);     % Gaussian RBF
% rbf = @(ep,r) 1./sqrt(1+(ep*r).^2);    % IMQ RBF
% rbf = @(ep,r) 1./(1+(ep*r).^2).^2;   % generalized IMQ
% rbf = @(ep,r) 1./(1+(ep*r).^2);    % IQ RBF
% rbf = @(ep,r) exp(-ep*r);    % basic Matern
% rbf = @(ep,r) exp(-ep*r).*(1+ep*r);    % Matern linear
%%% CPD1
% rbf = @(ep,r) ep*r;     % linear
% rbf = @(ep,r) sqrt(1+(ep*r).^2);   % MQ RBF
%%% CPD2
% rbf = @tps;   % TPS (defined in separate function tps.m)

function rbf = tps(ep,r)
rbf = zeros(size(r));
nz = find(r~=0);   % to deal with singularity at origin
rbf(nz) = (ep*r(nz)).^2.*log(ep*r(nz));
```

Program (RBFInterpolation_sD.m)

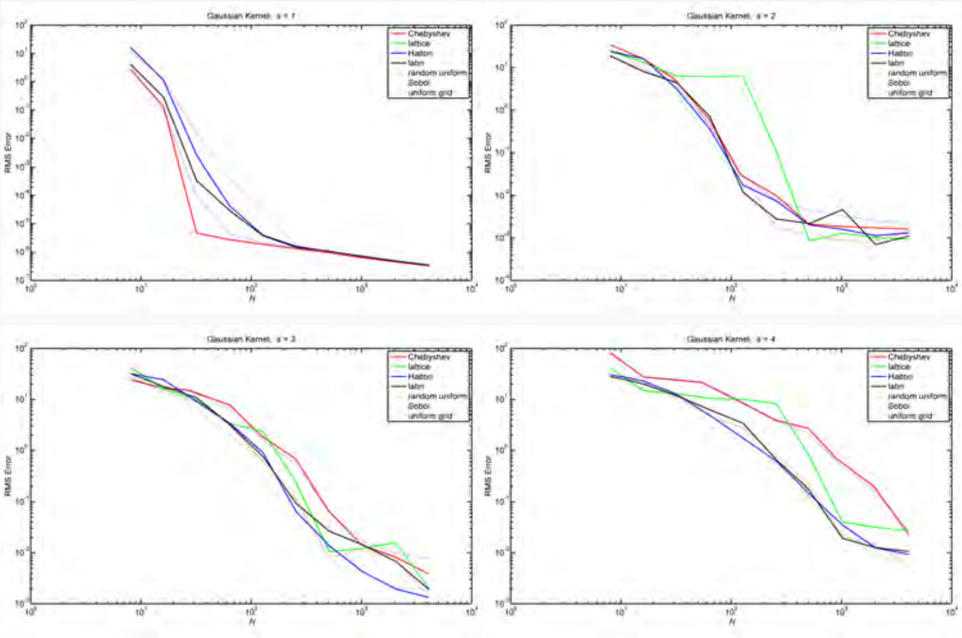
```
s = 2;  N = 289;  M = 500;
global rbf;  rbf_definition;  epsilon = 6/s;
[dsites, N] = CreatePoints(N,s,'h');
ctrs = dsites;
epoints = CreatePoints(M,s,'r');
rhs = testfunctionsD(dsites);
DM_data = DistanceMatrix(dsites,ctrs);
IM = rbf(epsilon,DM_data);
DM_eval = DistanceMatrix(epoints,ctrs);
EM = rbf(epsilon,DM_eval);
Pf = EM * (IM\rhs);
exact = testfunctionsD(epoints);
maxerr = norm(Pf-exact,inf)
rms_err = norm(Pf-exact)/sqrt(M)
```

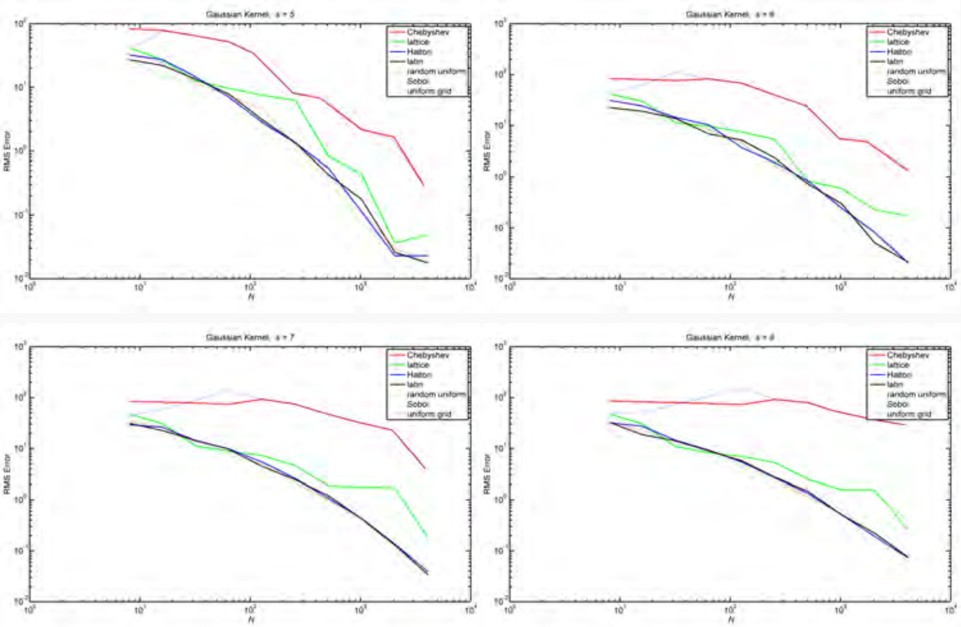
Bore-hole test function used in the computer experiments literature ([An and Owen (2001), Morris et al. (1993)]):

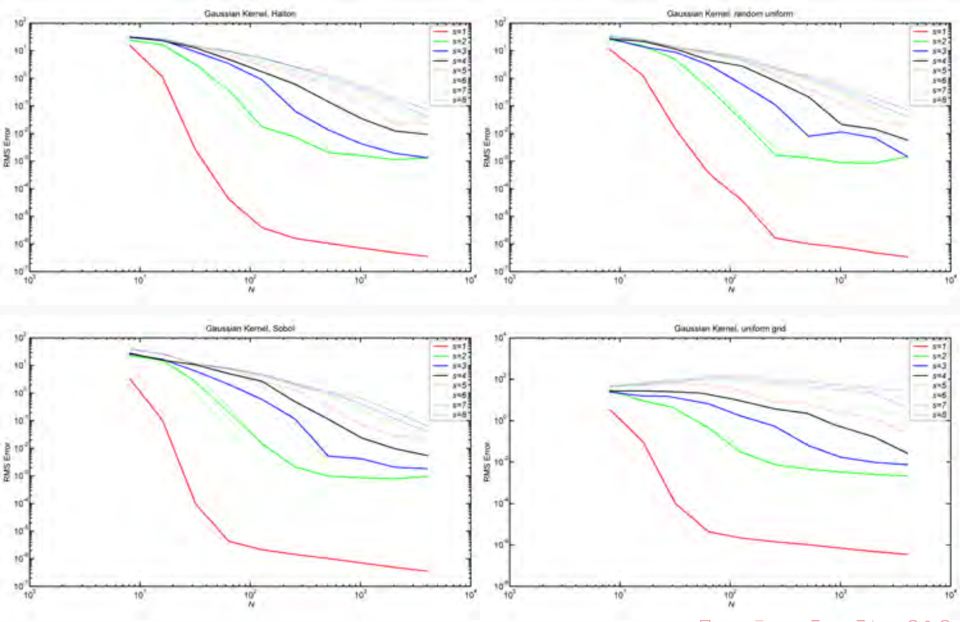
$$f(r_w, r, T_u, T_l, H_u, H_l, L, K_w) = \frac{2\pi T_u(H_u - H_l)}{\log\left(\frac{r}{r_w}\right)\left[1 + \frac{2LT_u}{\log\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{T_u}{T_l}\right]}$$

models flow rate of water from an upper to lower aquifer
Meaning and range of values:

- radius of borehole: $0.05 \le r_w \le 0.15$ (m)
- radius of surrounding basin: $100 \le r \le 50000$ (m)
- transmissivities of aquifers: $63070 \le T_u \le 115600$, $63.1 \le T_l \le 116$ (m^2/yr)
- potentiometric heads: $990 \le H_u \le 1110$, $700 \le H_l \le 820$ (m)
- length of borehole: $1120 \le L \le 1680$ (m)
- hydraulic conductivity of borehole: $9855 \le K_w \le 12045$ (m/yr)


Gaussian Kernel, s = 1
Gaussian Kernel, s = 2
Gaussian Kernel, s = 3
Gaussian Kernel, s = 4
Chebyshev
lattice
Halton
random uniform
Sobol
uniform grid
RMS Error
N


Gaussian Kernel, s = 5
Gaussian Kernel, s = 6
Gaussian Kernel, s = 7
Gaussian Kernel, s = 8
Chebyshev
lattice
Halton
latin
random uniform
Sobol
uniform grid
RMS Error
N

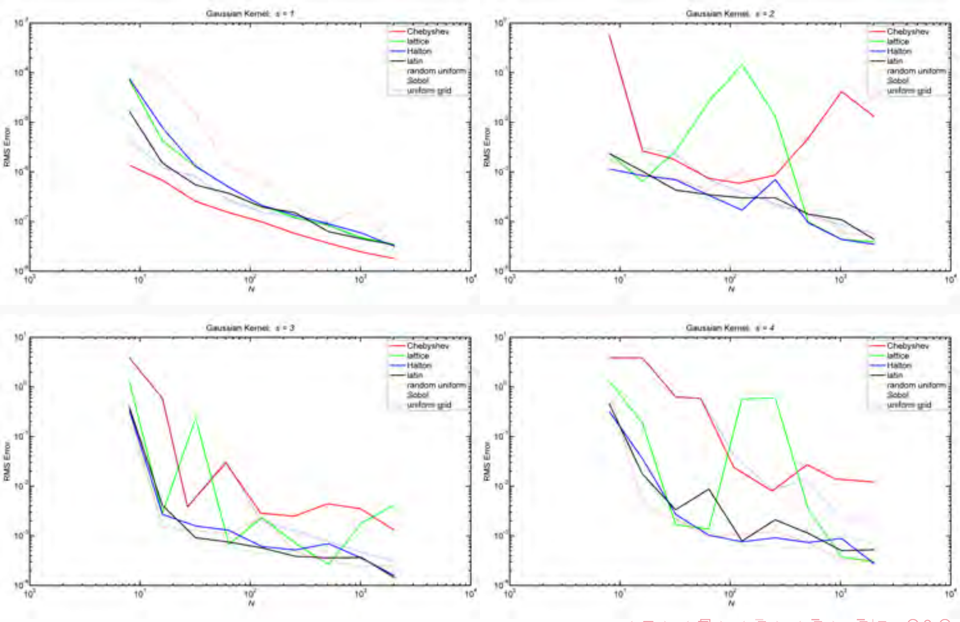

Gaussian Kernel, Halton
Gaussian Kernel, random uniform
Gaussian Kernel, Sobol
Gaussian Kernel, uniform grid
RMS Error
N
s=1
s=2
s=3
s=4
s=5
s=6
s=7
s=8

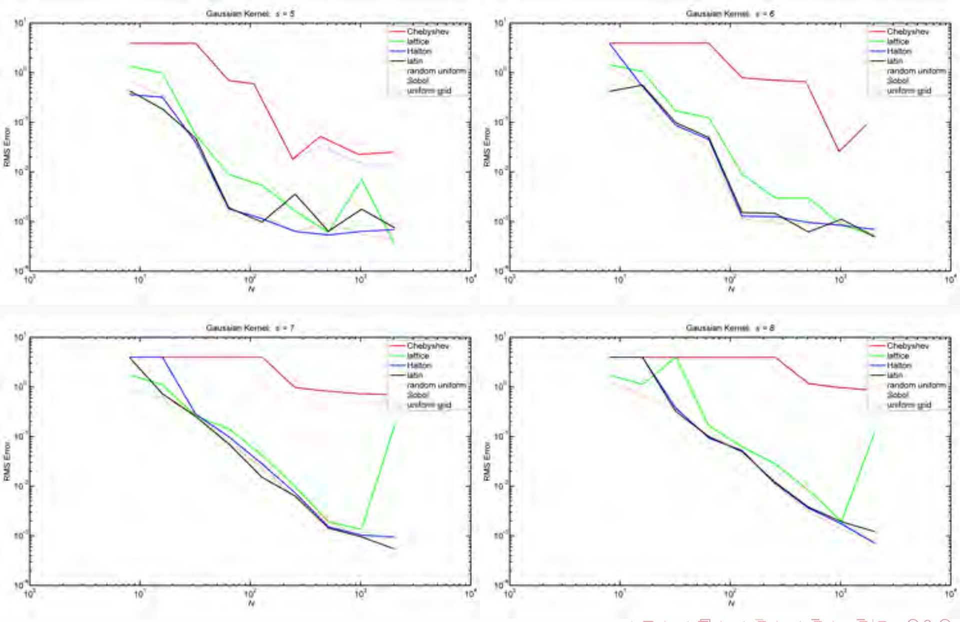
- In the non-stationary setting the convergence rate deteriorates with increasing dimension – regardless of the choice of design

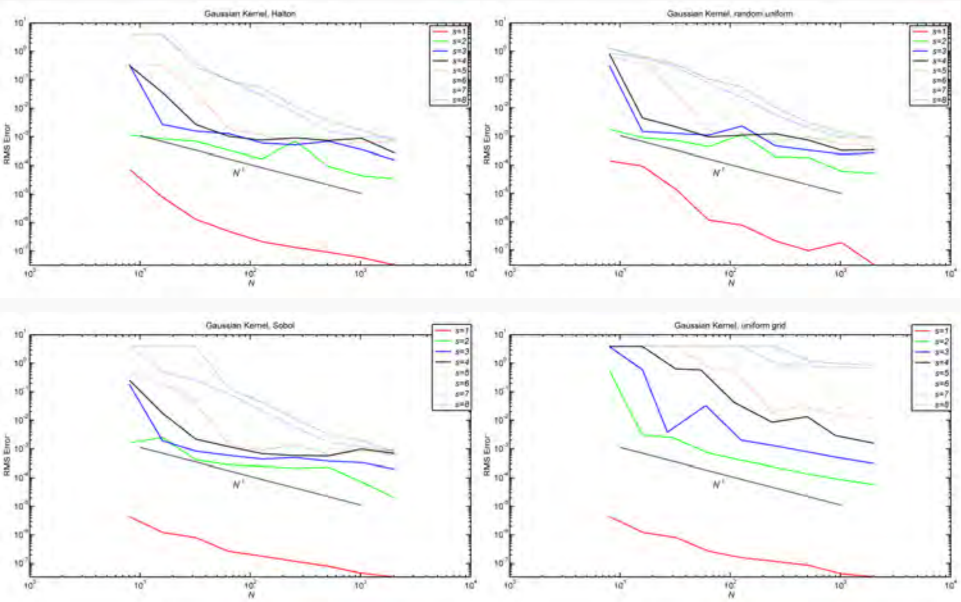
- In the non-stationary setting the convergence rate deteriorates with increasing dimension – regardless of the choice of design

- We know that the stationary setting is even worse since it's saturated

- In the non-stationary setting the convergence rate deteriorates with increasing dimension – regardless of the choice of design
- We know that the stationary setting is even worse since it's saturated
- Try an "optimal" non-stationary scheme...


Gaussian Kernel: ε = 1
Gaussian Kernel: ε = 2
Gaussian Kernel: ε = 3
Gaussian Kernel: ε = 4
Chebyshev
lattice
Halton
latin
random uniform
Sobol
uniform grid
RMS Error
N


Gaussian Kernel: s = 5
Gaussian Kernel: s = 6
Gaussian Kernel: s = 7
Gaussian Kernel: s = 8
Chebyshev
lattice
Halton
latin
random uniform
Sobol
uniform grid
RMS Error
N


Gaussian Kernel, Halton
Gaussian Kernel, random uniform
Gaussian Kernel, Sobol
Gaussian Kernel, uniform grid
RMS Error
N
N^{-1}
s=1
s=2
s=3
s=4
s=5
s=6
s=7
s=8

- Convergence rates seem to hold (dimension-independent?) – provided we use space-filling design

References I

Blumenthal, L. M. (1938).
Distance Geometries.
Univ. of Missouri Studies, **13**, 142pp.

Bochner, S. (1932).
Vorlesungen über Fouriersche Integrale.
Akademische Verlagsgesellschaft (Leipzig).

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.

References II

- Meyer, C. D. (2000).
  Matrix Analysis and Applied Linear Algebra.
  SIAM (Philadelphia).
- Wahba, G. (1990).
  Spline Models for Observational Data.
  CBMS-NSF Regional Conference Series in Applied Mathematics 59, SIAM (Philadelphia).
- Wells, J. H. and Williams, R. L. (1975).
  Embeddings and Extensions in Analysis.
  Springer (Berlin).
- Wendland, H. (2005).
  Scattered Data Approximation.
  Cambridge University Press.
- An, J. and Owen, A. (2001).
  Quasi-regression.
  J. Complexity **17**, pp. 588–607.

References III

Baxter, B. J. C. (1991).
Conditionally positive functions and p-norm distance matrices.
Constr. Approx. **7**, pp. 427–440.

Bochner, S. (1933).
Monotone Funktionen, Stieltjes Integrale und harmonische Analyse.
Math. Ann. **108**, pp. 378–410.

Bochner, S. (1941).
Hilbert distances and positive definite functions,
Ann. of Math. **42**, pp. 647–656.

Cleveland, W. S. and Loader, C. L. (1996).
Smoothing by local regression: Principles and methods.
in *Statistical Theory and Computational Aspects of Smoothing*, W. Haerdle and M. G. Schimek (eds.), Springer (New York), pp. 10–49.

Curtis, P. C., Jr. (1959)
n-parameter families and best approximation.
Pacific J. Math. **9**, pp. 1013–1027.

References IV

De Forest, E. L. (1873).
On some methods of interpolation applicable to the graduation of irregular series.
Annual Report of the Board of Regents of the Smithsonian Institution for 1871, pp. 275–339.

De Forest, E. L. (1874).
Additions to a memoir on methods of interpolation applicable to the graduation of irregular series.
Annual Report of the Board of Regents of the Smithsonian Institution for 1873, pp. 319–353.

Duchon, J. (1976).
Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces.
Rev. Francaise Automat. Informat. Rech. Opér., Anal. Numer. **10**, pp. 5–12.

References V

Duchon, J. (1977).
Splines minimizing rotation-invariant semi-norms in Sobolev spaces.
in *Constructive Theory of Functions of Several Variables, Oberwolfach 1976*, W. Schempp and K. Zeller (eds.), Springer Lecture Notes in Math. 571, Springer-Verlag (Berlin), pp. 85–100.

Duchon, J. (1978).
Sur l'erreur d'interpolation des fonctions de plusieurs variables par les D^m-splines.
Rev. Francaise Automat. Informat. Rech. Opér., Anal. Numer. **12**, pp. 325–334.

Duchon, J. (1980).
Fonctions splines homogènes à plusiers variables.
Université de Grenoble.

Dyn, N. (1987).
Interpolation of scattered data by radial functions.
in *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker, and F. Utreras (eds.), Academic Press (New York), pp. 47–61.

References VI

Dyn, N. (1989).
Interpolation and approximation by radial and related functions.
in *Approximation Theory VI*, C. Chui, L. Schumaker, and J. Ward (eds.), Academic Press (New York), pp. 211–234.

Dyn, N., Wahba, G. and Wong, W. H. (1979).
Smooth Pycnophylactic Interpolation for Geographical Regions: Comment.
J. Amer. Stat. Assoc. **74**, pp. 530–535.

Dyn, N. and Wahba, G. (1982).
On the estimation of functions of several variables from aggregated data.
SIAM J. Math. Anal. **13**, pp. 134–152.

Franke, R. (1982a).
Scattered data interpolation: tests of some methods.
Math. Comp. **48**, pp. 181–200.

References VII

Gram, J. P. (1883).
Über Entwicklung reeler Functionen in Reihen mittelst der Methode der kleinsten Quadrate.
J. Math. **94**, pp. 41–73.

Halton, J. H. (1960).
On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals.
Numer. Math. **2**, pp. 84–90.

Harder, R. L. and Desmarais, R. N. (1972).
Interpolation using surface splines.
J. Aircraft **9**, pp. 189–191.

Hardy, R. L. (1971).
Multiquadric equations of topography and other irregular surfaces.
J. Geophys. Res. **76**, pp. 1905–1915.

References VIII

Kansa, E. J. (1986).
Application of Hardy's multiquadric interpolation to hydrodynamics.
Proc. 1986 Simul. Conf. **4**, pp. 111–117.

Kansa, E. J. (1990a).
Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — I: Surface approximations and partial derivative estimates.
Comput. Math. Appl. **19**, pp. 127–145.

Kansa, E. J. (1990b).
Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — II: Solutions to parabolic, hyperbolic and elliptic partial differential equations.
Comput. Math. Appl. **19**, pp. 147–161.

Lancaster, P. and Šalkauskas, K. (1981).
Surfaces generated by moving least squares methods.
Math. Comp. **37**, pp. 141–158.

References IX

Madych, W. R. and Nelson, S. A. (1983).
Multivariate interpolation: a variational theory.
Manuscript.

Madych, W. R. and Nelson, S. A. (1988).
Multivariate interpolation and conditionally positive definite functions.
Approx. Theory Appl. **4**, pp. 77–89.

Madych, W. R. and Nelson, S. A. (1990a).
Multvariate interpolation and conditionally positive definite functions, II.
Math. Comp. **54**, pp. 211–230.

Madych, W. R. and Nelson, S. A. (1992).
Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation.
J. Approx. Theory **70**, pp. 94–114.

References X

Mairhuber, J. C. (1956).
On Haar's theorem concerning Chebyshev approximation problems having unique solutions.
Proc. Am. Math. Soc. **7**, pp. 609–615.

Meinguet, J. (1979a).
Multivariate interpolation at arbitrary points made simple.
Z. Angew. Math. Phys. **30**, pp. 292–304.

Meinguet, J. (1979b).
An intrinsic approach to multivariate spline interpolation at arbitrary points.
in *Polynomial and Spline Approximations*, N. B. Sahney (ed.), Reidel (Dordrecht), pp. 163–190.

Meinguet, J. (1979c).
Basic mathematical aspects of surface spline interpolation.
in *Numerische Integration*, G. Hämmerlin (ed.), Birkhäuser (Basel), pp. 211–220.

References XI

Meinguet, J. (1984).
Surface spline interpolation: basic theory and computational aspects.
in *Approximation Theory and Spline Functions*, S. P. Singh, J. H. W. Burry, and B. Watson (eds.), Reidel (Dordrecht), pp. 127–142.

Micchelli, C. A. (1986).
Interpolation of scattered data: distance matrices and conditionally positive definite functions.
Constr. Approx. **2**, pp. 11–22.

Morris, M. D., Mitchell, T. J. and Ylvisaker, D. (1993).
Bayesian design and analysis of computer experiments: use of derivatives in surface prediction.
Technometrics, **35**/3, pp. 243–255.

Schaback, R. (1995a).
Creating surfaces from scattered data using radial basis functions.
in *Mathematical Methods for Curves and Surfaces*, M. Dæhlen, T. Lyche, and L. Schumaker (eds.), Vanderbilt University Press (Nashville), pp. 477–496.

References XII

Schoenberg, I. J. (1937).
On certain metric spaces arising from Euclidean spaces by a change of metric and their imbedding in Hilbert space.
Ann. of Math. **38**, pp. 787–793.

Schoenberg, I. J. (1938a).
Metric spaces and completely monotone functions.
Ann. of Math. **39**, pp. 811–841.

Schoenberg, I. J. (1938b).
Metric spaces and positive definite functions.
Trans. Amer. Math. Soc. **44**, pp. 522–536.

Shepard, D. (1968).
A two dimensional interpolation function for irregularly spaced data.
Proc. 23rd Nat. Conf. ACM, pp. 517–524.

References XIII

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. (1981).
Spline Interpolation and smoothing on the sphere.
SIAM J. Sci. Statist. Comput. **2**, pp. 5–16.

Wahba, G. (1982).
Erratum: Spline interpolation and smoothing on the sphere.
SIAM J. Sci. Statist. Comput. **3**, pp. 385–386.

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.

References XIV

Wendland, H. (1995).
Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree.
Adv. in Comput. Math. **4**, pp. 389–396.

Wong, T.-T., Luk, W.-S. and Heng, P.-A. (1997).
Sampling with Hammersley and Halton points.
J. Graphics Tools **2**, pp. 9–24.

Woolhouse, W. S. B. (1870).
Explanation of a new method of adjusting mortality tables, with some observations upon Mr. Makeham's modification of Gompertz's theory.
J. Inst. Act. **15**, pp. 389–410.

DACE: A Matlab Kriging Toolbox.
available online at
http://www2.imm.dtu.dk/ hbn/dace/.

References XV

Highham, N. J.
An interview with Peter Lancaster.
http://www.siam.org/news/news.php?id=126.

MATLAB Central File Exchange.
available online at
http://www.mathworks.com/matlabcentral/fileexchange/.

Halton Points

Halton points (see [Halton (1960), Wong *et al.* (1997)]) are created from van der Corput sequences.

Halton Points

Halton points (see [Halton (1960), Wong *et al.* (1997)]) are created from van der Corput sequences.

Construction of a van der Corput sequence:

Start with unique decomposition of an arbitrary $n \in \mathbb{N}_0$ with respect to a prime base p, *i.e.*,

$$n = \sum_{i=0}^{k} a_i p^i,$$

where each coefficient a_i is an integer such that $0 \le a_i < p$.

Halton Points

Halton points (see [Halton (1960), Wong *et al.* (1997)]) are created from van der Corput sequences.

Construction of a van der Corput sequence:

Start with unique decomposition of an arbitrary $n \in \mathbb{N}_0$ with respect to a prime base p, *i.e.*,

$$n = \sum_{i=0}^{k} a_i p^i,$$

where each coefficient a_i is an integer such that $0 \le a_i < p$.

Example

Let $n = 10$ and $p = 3$. Then

$$10 = 1 \cdot 3^0 + 0 \cdot 3^1 + 1 \cdot 3^2,$$

so that $k = 2$ and $a_0 = a_2 = 1$ and $a_1 = 0$.

Next, define $h_p : \mathbb{N}_0 \to [0, 1)$ via

$$h_p(n) = \sum_{i=0}^{k} \frac{a_i}{p^{i+1}}$$

Next, define $h_p : \mathbb{N}_0 \to [0, 1)$ via

$$h_p(n) = \sum_{i=0}^{k} \frac{a_i}{p^{i+1}}$$

Example

$$h_3(10) = \frac{1}{3} + \frac{1}{3^3} = \frac{10}{27}$$

Next, define $h_p : \mathbb{N}_0 \to [0,1)$ via

$$h_p(n) = \sum_{i=0}^{k} \frac{a_i}{p^{i+1}}$$

Example

$$h_3(10) = \frac{1}{3} + \frac{1}{3^3} = \frac{10}{27}$$

$h_{p,N} = \{h_p(n) : \ n = 0, 1, 2, \ldots, N\}$ is called van der Corput sequence

Next, define $h_p : \mathbb{N}_0 \to [0, 1)$ via

$$h_p(n) = \sum_{i=0}^{k} \frac{a_i}{p^{i+1}}$$

Example

$$h_3(10) = \frac{1}{3} + \frac{1}{3^3} = \frac{10}{27}$$

$h_{p,N} = \{h_p(n) : \; n = 0, 1, 2, \ldots, N\}$ is called van der Corput sequence

Example

$$h_{3,10} = \{0, \tfrac{1}{3}, \tfrac{2}{3}, \tfrac{1}{9}, \tfrac{4}{9}, \tfrac{7}{9}, \tfrac{2}{9}, \tfrac{5}{9}, \tfrac{8}{9}, \tfrac{1}{27}, \tfrac{10}{27}\}$$

Generation of Halton point set in $[0, 1)^s$:

- take s (usually distinct) primes $p_1, \ldots, p_s$
- determine corresponding van der Corput sequences $h_{p_1,N}, \ldots, h_{p_s,N}$
- form s-dimensional Halton points by taking van der Corput sequences as coordinates:

$$H_{s,N} = \{(h_{p_1}(n), \ldots, h_{p_s}(n)) : \; n = 0, 1, \ldots, N\}$$

set of $N + 1$ Halton points

Some properties of Halton points

- Halton points are *nested* point sets, *i.e.*, $H_{s,M} \subset H_{s,N}$ for $M < N$
- Can even be constructed sequentially
- In low space dimensions, the multi-dimensional Halton sequence quickly “fills up” the unit cube in a well-distributed pattern
- For higher dimensions ($s \approx 40$) Halton points are well distributed only if N is large enough
- The origin is not part of the point set produced by `haltonseq.m`

Return