“Good” points for multivariate polynomial interpolation and approximation

Marc Van Barel, Matthias Humet, Laurent Sorber

Dept. of Computer Science, KU Leuven, Belgium

“Cittadella dei Musei”, Cagliari, Sardinia, Italy
September 2-5, 2013

Univariate polynomial interpolation and approximation (1)

- Consider a "difficult" function f defined on a subset Ω of the real line $\mathbb{R}$, e.g., $\Omega = [-1, 1]$.
- "Difficult" means that it is too costly to work directly with the function f.
- Instead we can use a function g belonging to a vector space $\mathcal{V}$:
  - approximating f according to a certain criterion,
  - cheap to determine,
  - easier to handle,
  - numerically sound (conditioning, numerical stability).
- Possible choices for $\mathcal{V}$ are:
  - polynomials up to a certain degree,
  - rational functions,
  - trigonometric functions,
  - ...

Univariate polynomial interpolation and approximation (2)

- In this talk,
  - the approximating function g is a polynomial function;
  - the approximation criterion is the ∞-norm, i.e.,

$$\|f\|_\infty = \max_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})|.$$

- Problem: not cheap to determine the minmax-approximant.
- Solution: compute cheaply a nearly-optimal approximant.
- An example of such a nearly-optimal approximant is the polynomial interpolating the function f in the points $\{\boldsymbol{x}_k \in \Omega\}_1^N$ with a corresponding small Lebesgue constant.

Lebesgue constant (univariate case)

- subset $\Omega \subset \mathbb{R}$, e.g., $\Omega = [-1, 1]$
- monomials $p_k(\boldsymbol{x}) = \boldsymbol{x}^{k-1}$, $k = 1, 2, \ldots, N$
- vector space $\mathcal{V} = \mathcal{P}^N = \{\sum_{k=1}^{N} c_k p_k\}$
- The **Lebesgue function** Λ and the **Lebesgue constant** λ associated with a set of points $\{\boldsymbol{x}_k\}_1^N$ is defined by

$$\Lambda(\boldsymbol{x}) = \sum_{i=1}^{N} |l_i(\boldsymbol{x})| \qquad \lambda = \max_{\boldsymbol{x} \in \Omega} \Lambda(\boldsymbol{x})$$

  where $l_i(\boldsymbol{x})$ are the Lagrange polynomials

$$\begin{cases} l_i \in \mathcal{P}^N \\ l_i\left(\boldsymbol{x}_j\right) = \delta_{i,j} \end{cases}$$

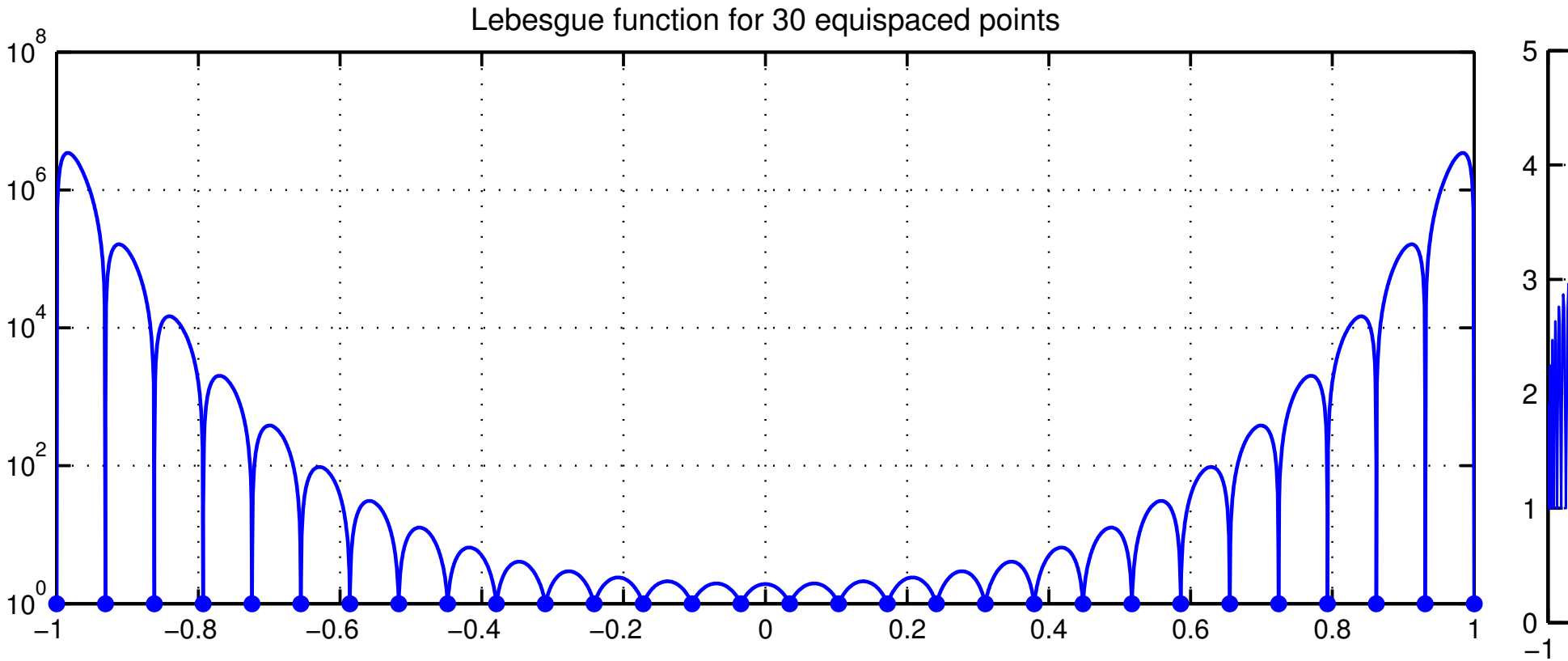
- For any function f:

$$\|f - p\|_\infty \leq (1 + \lambda) \, \|f - p^*\|_\infty$$

  Hence, nearly-optimal approximant when λ is small.

Example: Chebyshev points compared to equispaced points in $[-1, 1]$

For more details: see the book
Nick Trefethen, *Approximation Theory and Approximation Practice*, SIAM 2013.



15.2. Lebesgue constants for polynomial interpolation. . . .

For Chebyshev points, they satisfy

Lebesgue constant (multivariate case)

- subset $\Omega \subset \mathbb{R}^n$, e.g., $\Omega = [-1, 1]^2$, a triangle, a sphere, ...
- monomials $p_k(x_1, \ldots, x_n) = x_1^{\alpha_1(k)} \cdots x_n^{\alpha_n(k)} = \boldsymbol{x}^{\boldsymbol{\alpha}(k)}$ in a certain order ($\boldsymbol{x}, \boldsymbol{\alpha} \in \mathbb{R}^n$)
- vector space $\mathcal{P}^N = \{\sum_{k=1}^N c_k p_k\}$
- The **Lebesgue function** Λ and the **Lebesgue constant** λ associated with a set of points $\{\boldsymbol{x}_k\}_1^N$ is defined by

$$\Lambda(x) = \sum_{i=1}^{N} |l_i(\boldsymbol{x})| \qquad \lambda = \max_{\boldsymbol{x} \in \Omega} \Lambda(x)$$

  where $l_i(\boldsymbol{x})$ are the Lagrange polynomials

$$\begin{cases} l_i \in \mathcal{P}^N \\ l_i\left(\boldsymbol{x}_j\right) = \delta_{i,j} \end{cases}$$

- For any function f:

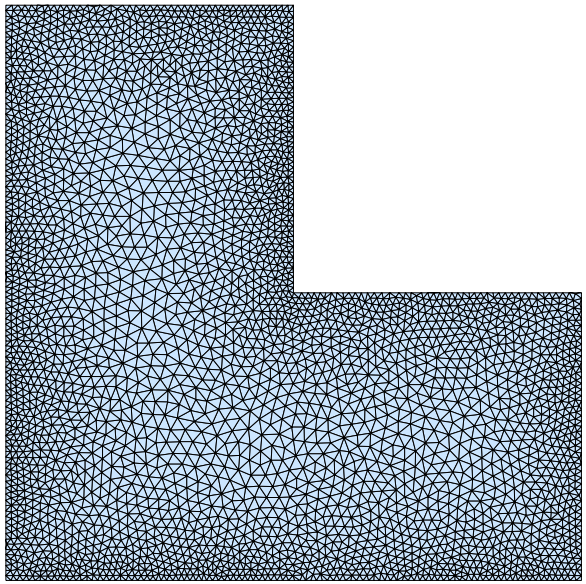
$$\|f - p\|_\infty \leq (1 + \lambda) \, \|f - p^*\|_\infty$$

  Hence, nearly-optimal approximant when λ is small.

Computing the Lebesgue constant

- Discretize subset $\Omega \in \mathbb{R}^n$ with "grid" $G \subset \Omega$ containing K points

$$\lambda = \max_{\boldsymbol{x} \in \Omega} \sum_{i=1}^{N} |l_i(\boldsymbol{x})| \quad \longrightarrow \quad \lambda \approx \max_{\boldsymbol{x} \in G} \sum_{i=1}^{N} |l_i(\boldsymbol{x})|$$



Example of a mesh G generated by DistMesh for the L-shape consisting of 3475 points.
P.-O. Persson, G. Strang, *A Simple Mesh Generator in MATLAB*. SIAM Review, Volume 46 (2), pp. 329-345, June 2004

Minimize Lebesgue constant

$$\min_{\{\boldsymbol{x}_k\}_1^N} \lambda \approx \left\| L_N\left(\{\boldsymbol{y}_k\}_1^K\right) \right\|_\infty$$
$$\text{s.t.} \quad \Phi_N\left(\{\boldsymbol{y}_k\}_1^K\right) = L_N\left(\{\boldsymbol{y}_k\}_1^K\right) \Phi_N\left(\{\boldsymbol{x}_k\}_1^N\right)$$

- Objective function is not differentiable
  $\rightarrow$ need derivative free optimization
- Many local minima
- Large problem size, e.g.: $n = 2$, total degree of 20 $\rightarrow$ 231 points $\sim$ 462 variables
- [Briani, Sommariva, Vianello; 2011] use state of the art Matlab optimization algorithms to compute minima for the simplex, square and disk.
  $\rightarrow$ very slow for larger point sets

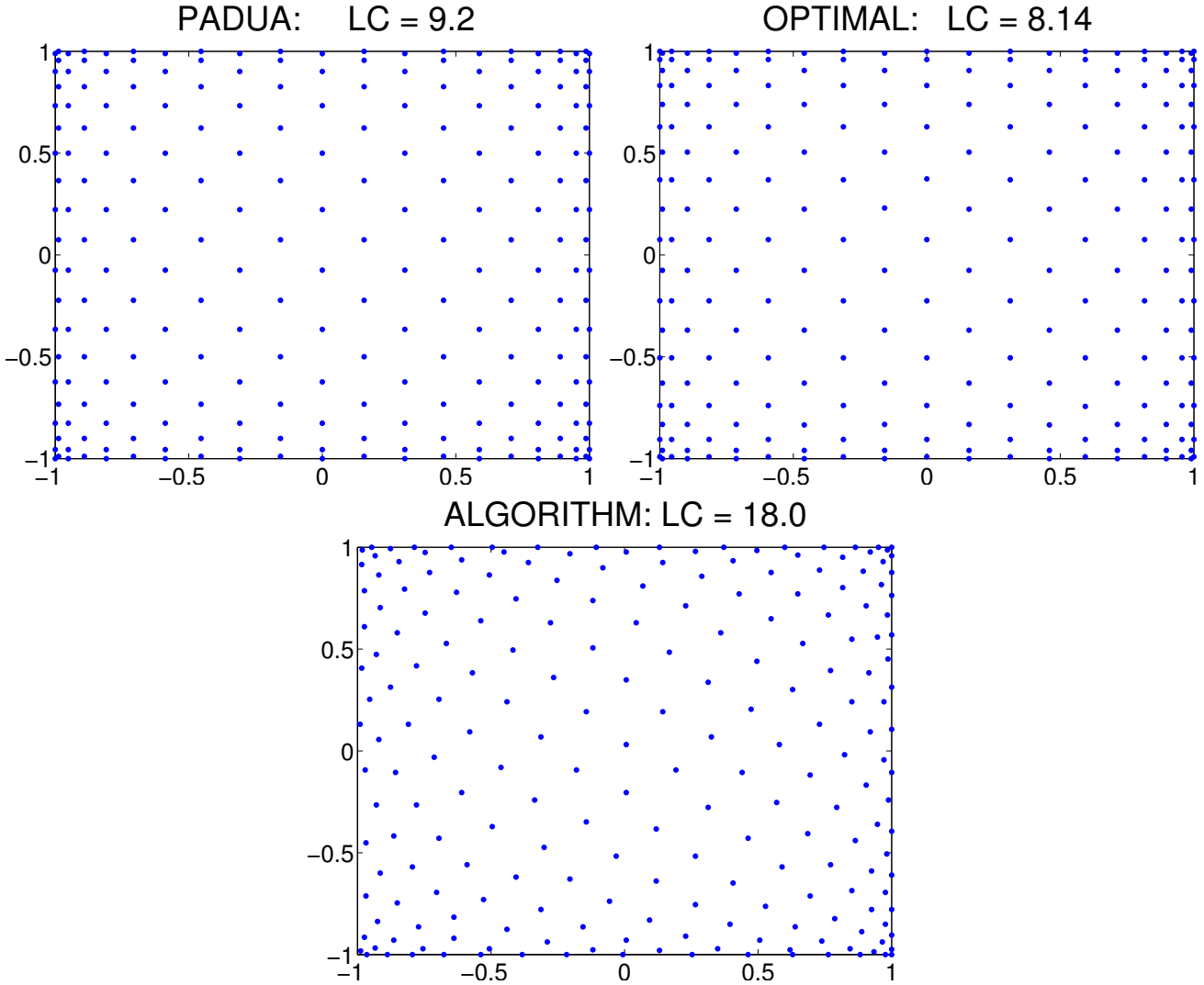
Alternatives to obtain low Lebesgue constant

- Compromise between a small λ and computational effort
- Special subsets: small λ with little computational cost

  E.g., direct formula for Padua points on the square $[-1,1]^2$
- Alternative algorithm ("greedy" algorithm)
  1. **First phase**: add $(k+1)$th point where $\sum_{i=1}^{k} \left| l_{(k),i}(\boldsymbol{x}) \right|$ is maximal on the subset Ω, i.e., on the "grid" G
     - ⋆ Remember: $\lambda = \max_{\boldsymbol{x}\in\Omega} \sum_{i=1}^{k} \left| l_{(k),i}(\boldsymbol{x}) \right|_{k+1}$
     - ⋆ By adding the point $\boldsymbol{x}^*$, we assure that $\sum_{i=1}^{k} \left| l_{(k+1),i}(\boldsymbol{x}^*) \right| = 1$
  2. **Second phase**: update points one by one
     - ⋆ remove point and add it again where $\sum_{i=1}^{N-1} \left| l_{(N-1),i}(\boldsymbol{x}) \right|$ is maximal

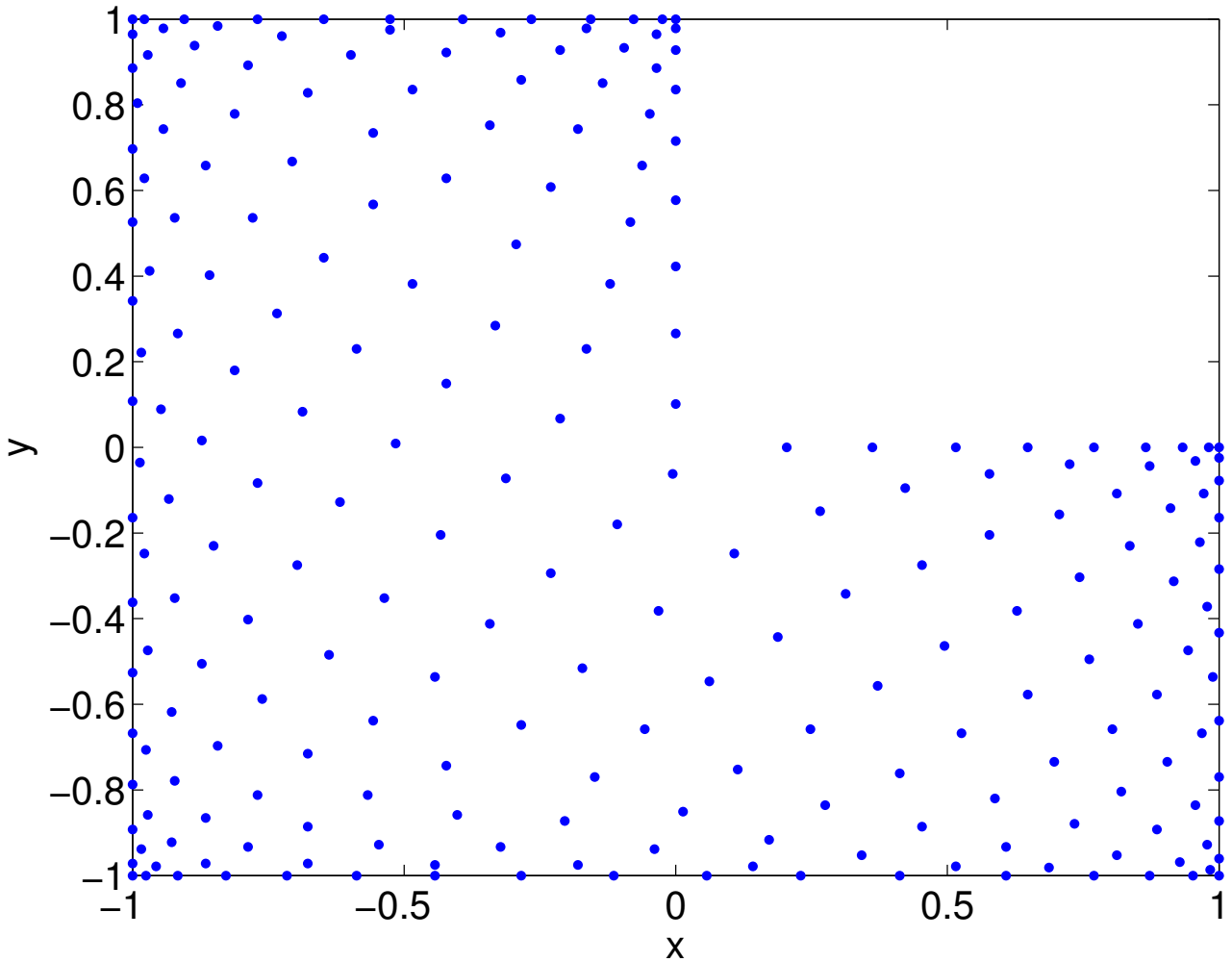
Compare points on the square

- 231 points (total degree 20), result after 20 iterations.
- Basis of product Chebyshev polynomials: $\phi_k(x, y) = T_i(x)T_j(y)$



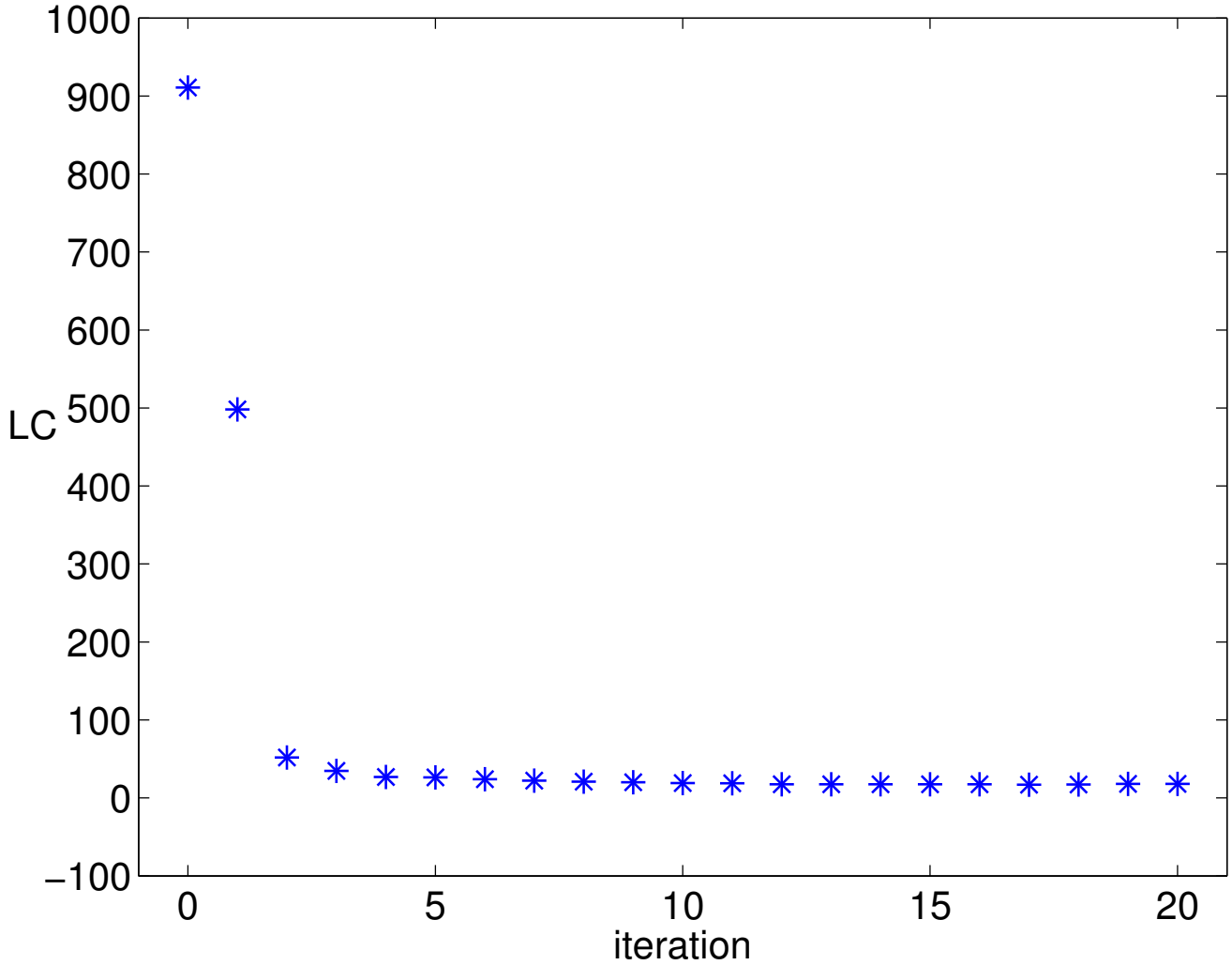
Compute points for the L-shape

- 231 points (total degree 20), result after 20 iterations.
- Basis of product Chebyshev polynomials: $\phi_k(x, y) = T_i(x)T_j(y)$
- Lebesgue constant: 31.5



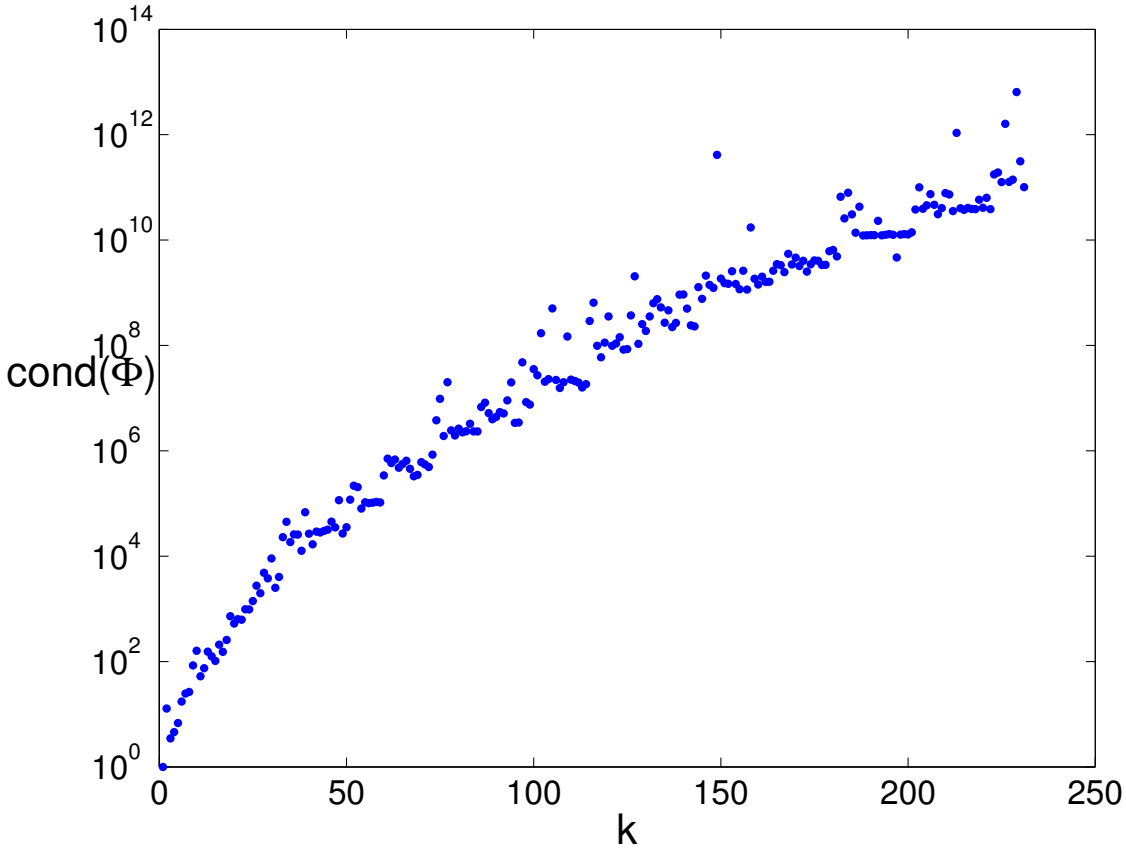
Compute points for the L-shape

- 231 points (total degree 20), 20 iterations
- Show Lebesgue constant after each iteration



Some problems with L-shape

- Product Chebyshev polynomials are not a good basis for the L-shape, see figure
- Alternative basis: orthogonal polynomials with respect to discrete inner product
  $\rightarrow$ need to recompute basis when points change too much



Alternative optimization algorithm

- Instead of minimizing the ∞-norm

$$\min_{\{\boldsymbol{x}_k\}_1^N} \lambda \approx \left\| L_N\left(\{\boldsymbol{y}_k\}_1^K\right)\right\|_\infty$$
$$\text{s.t.} \quad \Phi_N\left(\{\boldsymbol{y}_k\}_1^K\right) = L_N\left(\{\boldsymbol{y}_k\}_1^K\right)\Phi_N\left(\{\boldsymbol{x}_k\}_1^N\right)$$

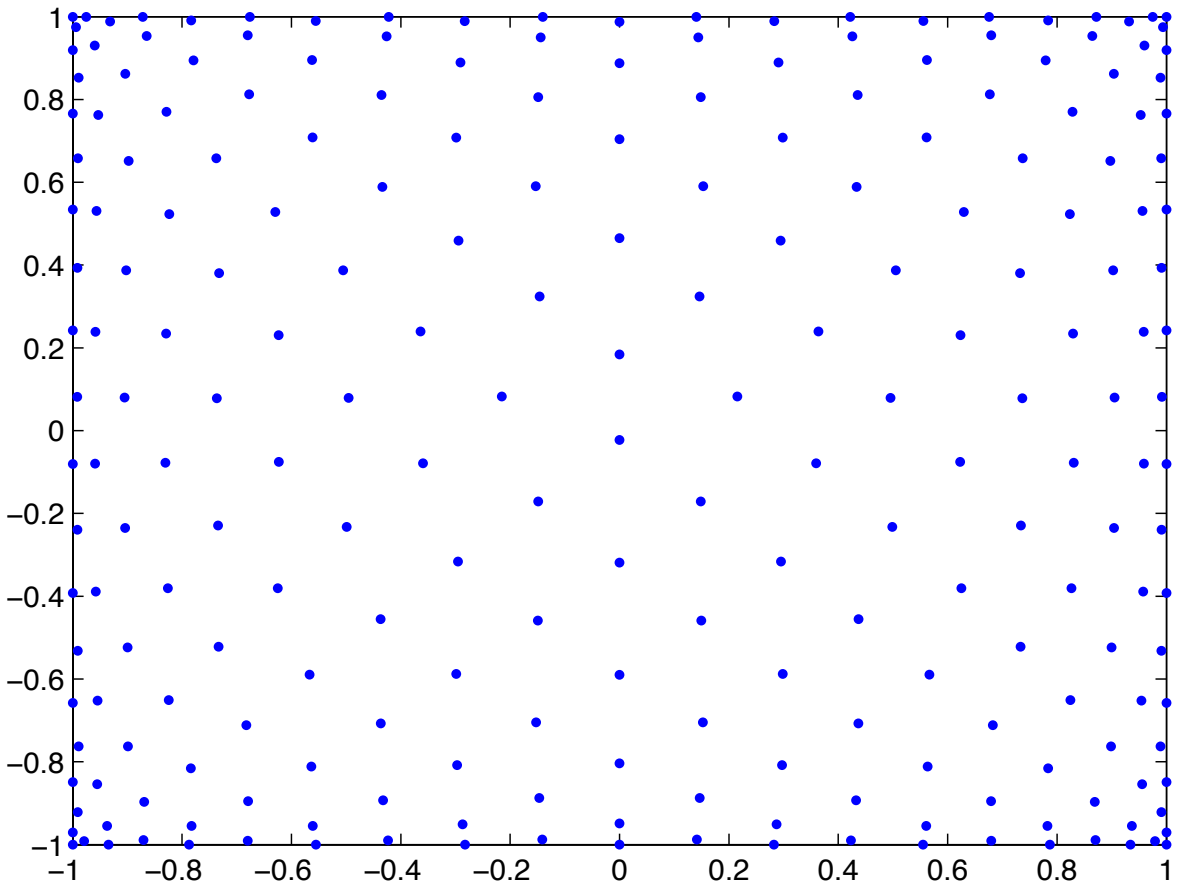
  we minimize the Frobenius norm

$$\min_{\{\boldsymbol{x}_k\}_1^N} \lambda \approx \left\| L_N\left(\{\boldsymbol{y}_k\}_1^K\right)\right\|_{\text{frob}}$$
$$\text{s.t.} \quad \Phi_N\left(\{\boldsymbol{y}_k\}_1^K\right) = L_N\left(\{\boldsymbol{y}_k\}_1^K\right)\Phi_N\left(\{\boldsymbol{x}_k\}_1^N\right)$$

- Efficient Matlab implementation of a bound-constrained nonlinear least-squares optimization problem by projected Gauss-Newton dogleg trust-region method

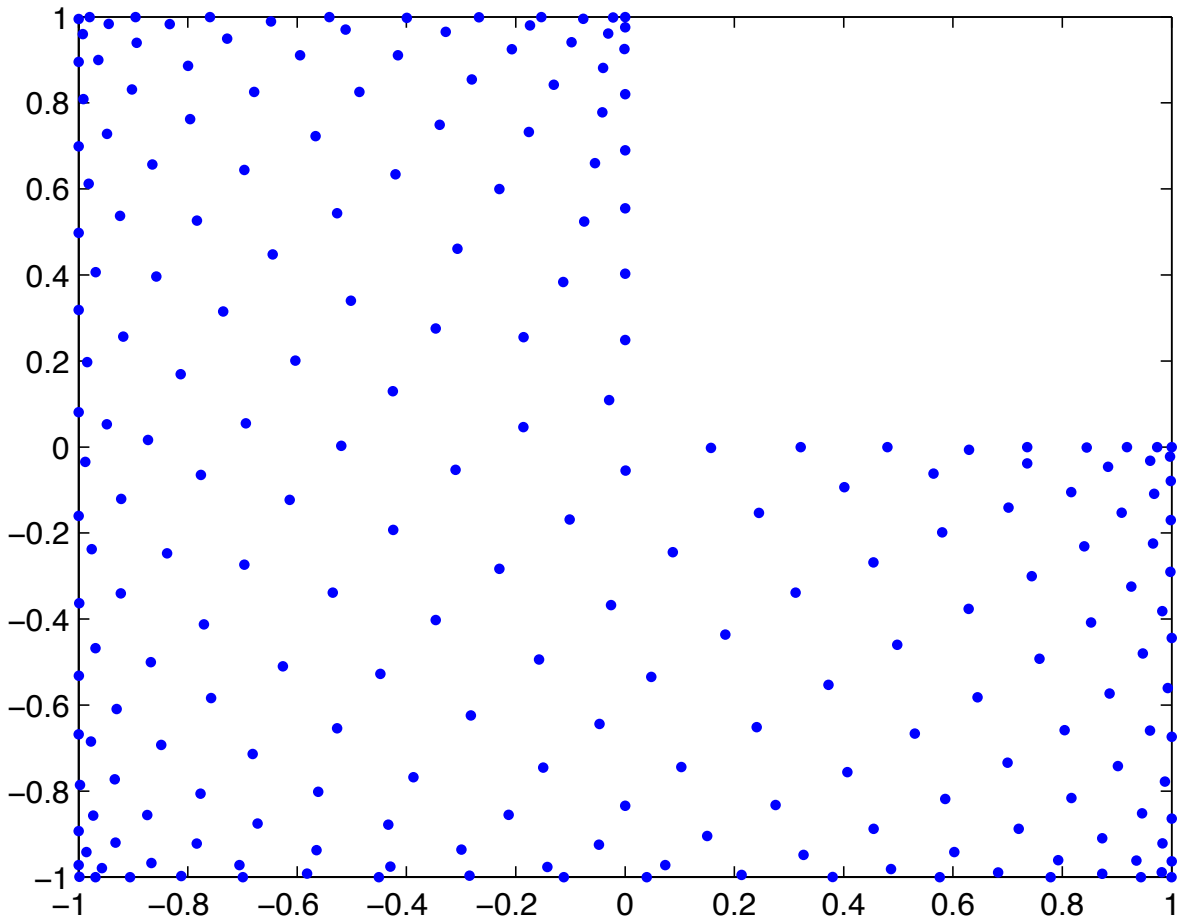
Alternative optimization algorithm: square

- 231 points (total degree 20)
- Lebesgue constant: 7.74 (Padua-points: 9.20; “best”: 8.14 !!)



Alternative optimization algorithm: L-shape

- 231 points (total degree 20)
- Lebesgue constant: 15.34



Extension of results of 1D to nD

- zeros of orthogonal polynomials are "good" points
  zeros of Chebyshev/Legendre/. . . polynomials in the interval $[-1, 1]$
  $\Rightarrow$ are the eigenvalues of corresponding Jacobi (tridiagonal) matrices
  no immediate generalization to the nD case
- when degree goes to infinity, the distribution of these zeros is $\frac{1}{\pi\sqrt{1-x^2}} \Rightarrow$ "good" points are points (almost) equidistant using as distance $d(a, b)$

$$d(a, b) = \left| \frac{1}{\pi} \int_a^b \frac{1}{\sqrt{1 - x^2}} dx \right|$$

leading to the so-called Dubiner metric in $[-1, 1]$:

$$d(a, b) = c\,|acos(a) - acos(b)|$$

Distribution of the zeros

- How can we approximate the distribution of these zeros?
  Look at the distribution of the zeros of random polynomials, i.e., polynomials whose coefficients with respect to orthogonal polynomials on the subset Ω are i.i.d. random numbers, normally distributed.
- Extension to the nD case: look at the distribution of the zero-hypersurfaces of the random polynomials written in an orthogonal basis on the subset Ω.
  $\Rightarrow$ 2D case: 0-level curves of random polynomials
- Illustration: 2 movies: square and L-shape
- Conjecture: the (real) solutions of each polynomial system of two of such random polynomials are distributed following the distribution of "good" points, i.e., leading to a small Lebesgue constant. (Recently proved in [Bloom, Levenberg, 2013])

Kernel density estimation

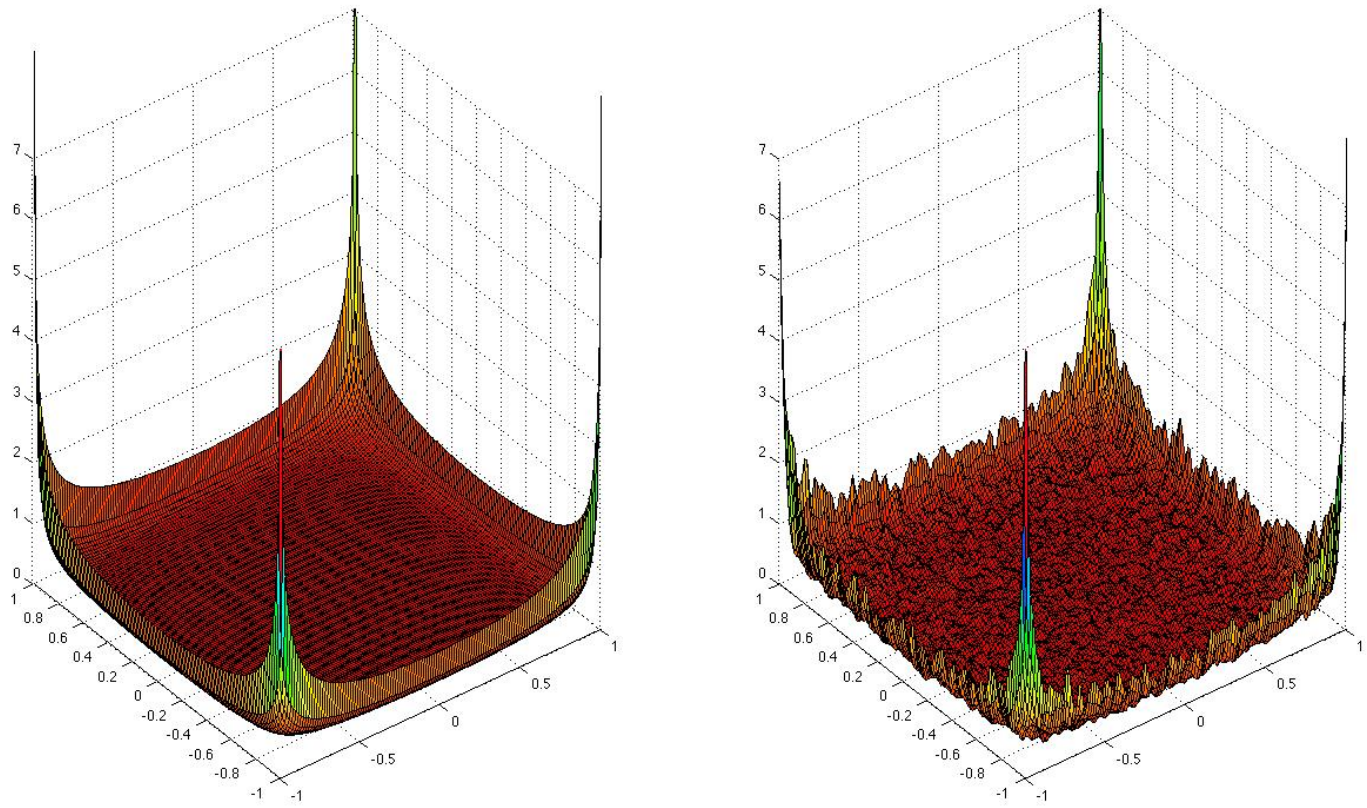


Figure : Kernel density estimation: the exact density and the estimated density on the unit square, total degree of 20, 340 samples of a system of two random polynomials, resulting in 41549 real solutions.

Solving a system of polynomial equations

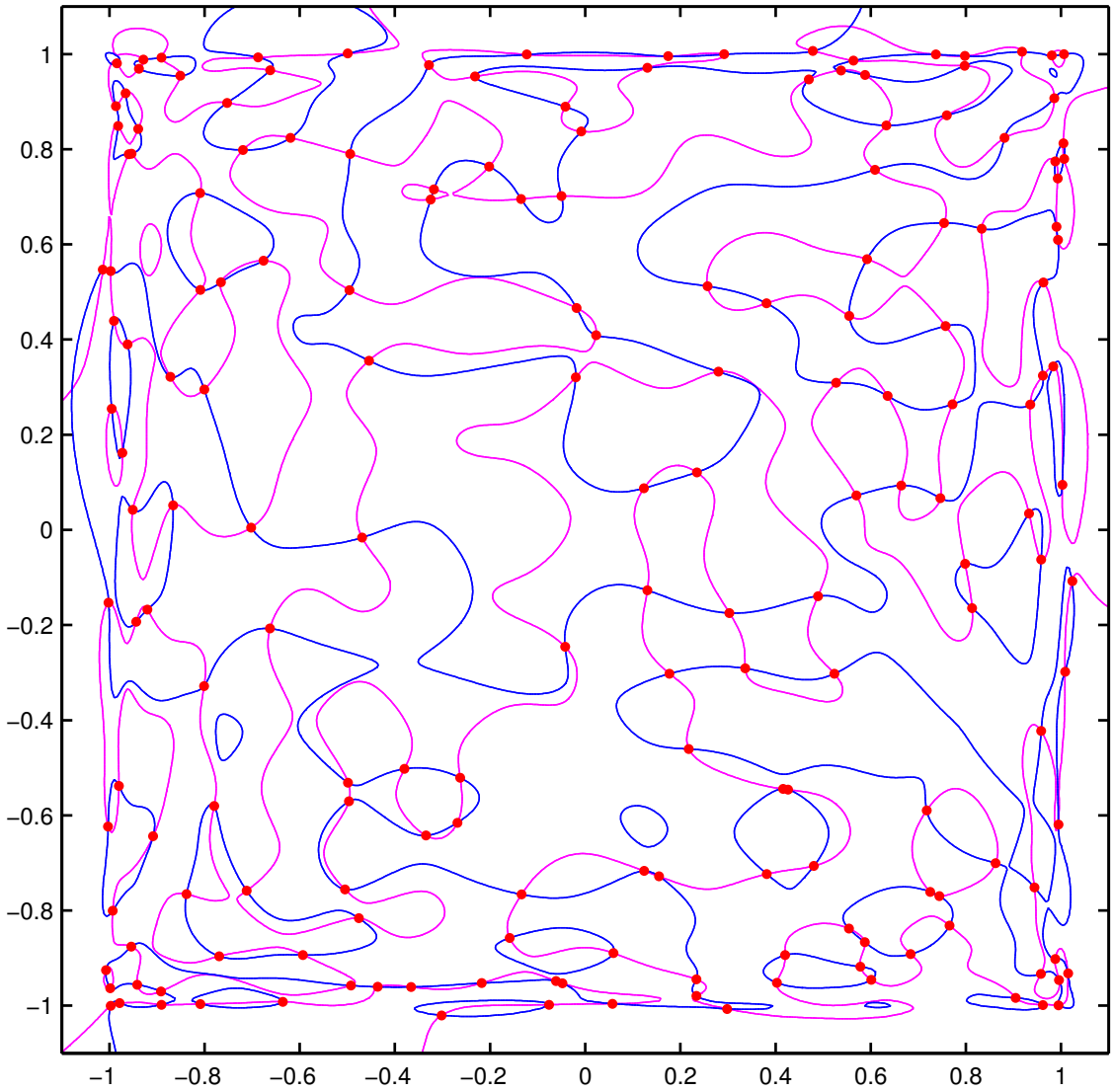


Figure : Common solutions of two random polynomials on the unit square, total degree of 25.

References I


S. Elhay, G. H. Golub, and J. Kautsky.
Updating and downdating of orthogonal polynomials with data fitting applications.
SIAM J. Matrix Anal. Appl., 12(2):327–353, 1991.


M. Van Barel and A. A. Chesnokov.
A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations.
Numer. Algorithms, 55:383–402, 2010.


M. Briani, A. Sommariva, and M. Vianello.
Computing Fekete and Lebesgue points: Simplex, square, disk.
J. Comput. Appl. Math., 236:2477–2486, 2012.
Not yet available on JCAM website...

References II


J.-P. Calvi and N. Levenberg.
Uniform approximation by discrete least squares polynomials.
J. Approx. Theory, 152:82–100, 2008.


L. Bos and M. Vianello.
Low cardinality admissible meshes on quadrangles, triangles and disks.
Math. Inequal. Appl., 15(1):229–235, 2012.


P.-O. Persson and G. Strang.
A simple mesh generator in MATLAB.
SIAM Rev., 46(2):329–345, 2004.


T. Bloom and N. Levenberg.
Random polynomials and pluripotential-theoretic extremal functions.
ArXiv e-prints, April 2013.

Happy 60th Birthday, Cornelis