A fast algorithm for computing the mock-Chebyshev interpolation

B. Ali Ibrahimoglu

Yıldız Technical University
Department of Mathematical Engineering

(NASCA 2018)

Problem statement

Problem statement

Linear interpolation:

- nodes:

$$x_0, \dots, x_n, \quad x_i \in [-1, 1]$$

- data:

$$f_0, \dots, f_n, \quad f_i \in \mathbb{C}$$

- basis functions:

$$\phi_0(x), \dots, \phi_n(x)$$

Interpolation conditions:

$$\sum_{i=0}^{n} a_i \phi_i(x_j) = f_j, \quad j = 0, \dots, n$$

Polynomial interpolation

Problem statement

Given:

$$(x_0, f_0), \dots, (x_n, f_n), \quad x_i \in [-1, 1]$$

Compute:

$$p_n(x) = \sum_{i=0}^{n} a_i x^i, \quad p_n(x_i) = f_i$$

Unisolvence:

$$x_i \neq x_j, \qquad i \neq j$$

Interpolation error:

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i), \quad \xi \in \,]-1, 1[$$

Polynomial interpolation

Lebesgue constant

Lagrange form:

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\ell_i(x)$$

$$\ell_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, \dots, n$$

$$\ell_i(x_j) = \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Lebesgue function:

$$L_n(x) = \sum_{i=0}^{n} |\ell_i(x)|$$

Lebesgue constant:

$$\Lambda_n(x_0, \dots, x_n) = \max_{x\in[-1,1]} L_n(x)$$

Polynomial interpolation

Error estimates

- Error formula 1:

$$\|f - p_n\|_\infty \leq \max_{x\in[-1,1]} \left(\frac{|f^{(n+1)}(x)|}{(n+1)!} \right) \max_{x\in[-1,1]} \prod_{j=0}^{n} |x - x_j|$$

- Error formula 2:

$$\|f - p_n\|_\infty \leq (1 + \Lambda_n) \max_{x\in[-1,1]} |f(x) - p_n^*(x)|,$$

$$\Lambda_n = \max_{x\in[-1,1]} \sum_{i=0}^{n} |\ell_i(x)|, \quad \ell_i(x) = \frac{\prod_{j=0,\ j\neq i}^{n}(x - x_j)}{\prod_{j=0,\ j\neq i}^{n}(x_i - x_j)}$$

$p_n^* \in \Pi_n$ is the best uniform polynomial approximation to f

$\Rightarrow$ Small Lebesgue constant implies $\|f - p_n\|_\infty \approx \|f - p_n^*\|_\infty$

Polynomial interpolation

Optimize:

$$\min_{\{x_0,\dots,x_n\}\subset[-1,1]} \Lambda_n(x_0,\dots,x_n)$$

Solution:

$$\Lambda_n^{\text{Optimal}} = \frac{2}{\pi}\ln(n+1) + 0.5212\ldots + o(1) \qquad \text{[Vértesi'90]}$$

- set T: Chebyshev points in $(-1,1)$

$$x_i = -\cos\left(\frac{\pi(2i+1)}{2(n+1)}\right), \quad i = 0,\dots,n,$$

$$\Lambda_n(T) = \frac{2}{\pi}\ln(n+1) + 0.9625\ldots + o(1)$$

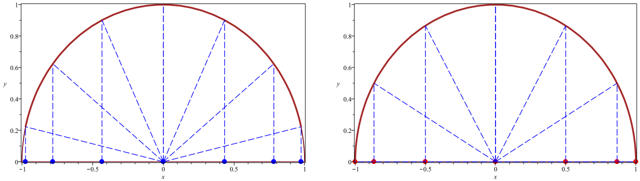
- set C: Chebyshev-Lobatto points on $[-1,1]$

$$x_i = -\cos\left(\frac{i\pi}{n}\right), \quad i = 0,\dots,n,$$

$$\Lambda_n(C) = \begin{cases} \Lambda_{n-1}(T), & n \text{ odd} \\ \Lambda_{n-1}(T) - \alpha_n, & 0 < \alpha_n < \frac{1}{n^2}, \quad n \text{ even} \end{cases}$$

Polynomial interpolation

Interpolation points

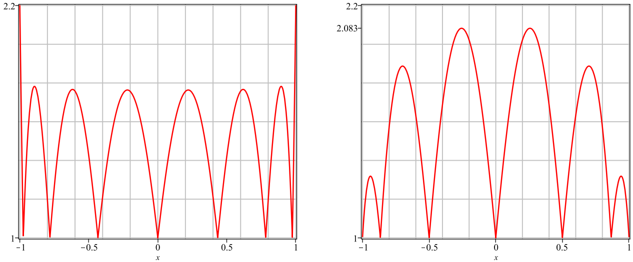


7 Chebyshev (•) and Chebyshev-Lobatto (•) points, $n = 6$

Polynomial interpolation

Lebesgue function

$$L_n(x) = \sum_{i=0}^{n} |\ell_i(x)|, \quad -1 \le x \le 1$$



Lebesgue function for Chebyshev and for Chebyshev-Lobatto points, $n + 1 = 7$

Mock-Chebyshev interpolation

When the data $f_0, \dots, f_n$ are provided at equidistant points only

Polynomial interpolation with equidistant points is unreliable:

- it is severely ill-conditioned ($\Lambda_n \to \infty$ exponentially fast)
  set E: equidistant points

$$x_i = -1 + \frac{2i}{n}, \quad i = 0, \dots, n,$$

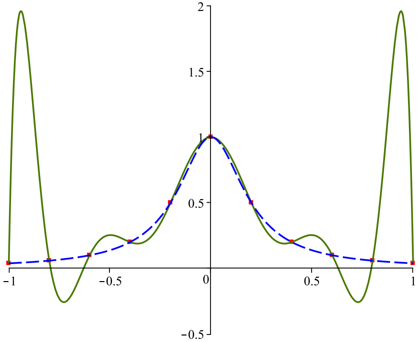
$$\Lambda_n(E) = \frac{2^{n+1}}{e\,n(\ln(n) + \gamma)} + o(1)$$

- it leads to Runge's phenomenon

  The Runge phenomenon: polynomial interpolation of a function f, using equidistant interpolation points on $[-1, 1]$, could diverge on parts of this interval even if f is analytic everywhere on the interval.

Mock-Chebyshev interpolation

Runge's phenomenon



Graphs of $f(x) = \frac{1}{1+25x^2}$ (dashed line) and its polynomial interpolant (solid line) at 11 equidistant points (solid boxes), $-1 \le x \le 1$

Mock-Chebyshev interpolation

When the data $f_0, \dots, f_n$ are provided at equidistant points only

Among all methods that have been proposed to defeat Runge's phenomenon, the mock-Chebyshev interpolation algorithm is the exception

- being blessed with the same $\mathcal{O}(1)$ condition method as standard Chebyshev interpolation with n points (w.r.t. numerical ill-conditioning)
- as it falls under the umbrella of standard Chebyshev interpolation (w.r.t. lack of a rigorous theory)

The mock-Chebyshev grid: a subset of $(n+1)$ points from an equispaced grid with $\mathcal{O}(n^2)$ points chosen to mimic the non-uniform $n+1$-point Chebyshev-Lobatto grid [Boyd, Xu, 2009].

Mock-Chebyshev interpolation

The Mock-Chebyshev Grid [Boyd, Xu, 2009]

$$\tilde{x}_j = \bar{x}_l, \quad \text{such that } |x_j - \bar{x}_l| = \min_k |x_j - \bar{x}_k|,$$
$$\forall k = 0, 1, \ldots, m, \quad j = 0, 1, \ldots, n, \quad l \in \{0, 1, \ldots, m\},$$

- x_j: points of Chebyshev-Lobatto grid with $(n+1)$ points

$$x_j = -\cos\left(\frac{j\pi}{n}\right), \quad j = 0, 1, \ldots, n$$

- $\bar{x}_k$: points of equispaced grid with $m = \mathcal{O}(n^2)$ points

$$\bar{x}_k = -1 + \frac{2k}{m}, \quad k = 0, 1, \ldots, m$$

- $\tilde{x}_j$: subset of $(n+1)$ points, imitating Chebyshev-Lobatto grid, selected from equispaced grid with $m = \mathcal{O}(n^2)$ points

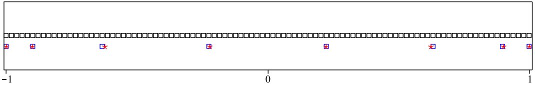
Mock-Chebyshev interpolation

The Mock-Chebyshev Grid [Boyd, Xu, 2009]

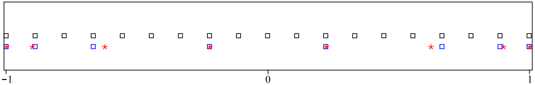
- To avoid the fact that the endpoints −1 and 1 can be selected more than once, m is greater than $\frac{2n^2}{\pi^2}$
- The computational cost is $\mathcal{O}(nm)$

Example: 8 mock-Chebyshev points for $x_j = -\cos(j\pi/7), \quad j = 0, 1, \ldots, 7$

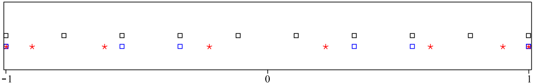
- $m = 99$



- $m = 18$



- for $m = 9$, the endpoints −1 and 1 repeat!



Mock-Chebyshev interpolation

Mock-X points $\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_n$ on $[a, b]$

Given:

a set of non-uniform $(n+1)$ points

$$X_n = \{x_j : j = 0, 1, \ldots, n\}, a = x_0 < x_1 < \cdots < x_n = b$$

Find:

a subset of uniformly distributed $(n+1)$ points

$$\tilde{X}_n = \{\tilde{x}_j : j = 0, 1, \ldots, n\}, a = \tilde{x}_0 < \tilde{x}_1 < \cdots < \tilde{x}_n = b,$$

in the sense that the subset $\tilde{X}_n$ of $(n+1)$ points mimic the behavior of the given non-uniform $(n+1)$ points

Mock-Chebyshev interpolation

A new approach for computing the mock-X points $\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_n$

Here we give a new method for generating the mock-X points $\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_n$:

$$\tilde{x}_j = \tilde{x}_0 + S_j \tilde{h}, \quad \tilde{h} = \frac{2}{S_n}, \quad \tilde{x}_0 = x_0, \quad S_0 = 0, \quad j = 1, 2, \ldots, n,$$

$$S_j = \left\lceil \frac{h_j(X_n)}{\min h_j(X_n)} \right\rceil + S_{j-1}, \quad h_j(X_n) = |x_j - x_{j-1}|, \quad j = 1, 2, \ldots, n$$

- where $\lceil \cdot \rceil$ denotes the ceil function. For computing S_j, we can also use the floor function $\lfloor \cdot \rfloor$ or the round function $\mathtt{round}(\cdot)$
- the points $\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_n$ are a subset of $(S_n + 1)$ equispaced points with gap $\tilde{h} = 2/S_n$

Mock-Chebyshev interpolation

The mock-Chebyshev grid with our method

When we take a set of $(n+1)$ Chebyshev-Lobatto points as X_n in our method, then we obtain the set of mock-Chebyshev points.

For this case, let $\bar{X}_m = \{\bar{x}_k = -1 + 2k/m, \quad k = 0, 1, \ldots, m\}$ be the superset of $\tilde{X}_n$. Then we have the following properties:

- $\min h_j(X_n) = h_1(X_n) = h_n(X_n), \quad j = 1, 2, \ldots, n$
- $m > \frac{4n^2}{\pi^2}$, therefore the distance between the points $\bar{x}_0, \bar{x}_1, \cdots, \ \bar{x}_m$ ensures that the endpoints -1 and 1 never repeat
- $\tilde{x}_j$ are symmetric with respect to the origin

Mock-Chebyshev interpolation

The mock-Chebyshev grid with our method

- For $n \leq 2i+1, \quad i = 0, 1, \ldots, \lfloor \frac{n-1}{2} \rfloor$,

$$\frac{x_i + x_{i+1}}{2} < -1 + \frac{4(x_{i+1}+1)}{(n-2)x_1 + n + 2} \leq \tilde{x}_{i+1},$$

$$\tilde{x}_{i+1} \leq x_{i+1} + \frac{i}{2}(x_1 + 1) < \frac{x_{i+1} + x_{i+2}}{2}$$

By using the symmetry, we conclude:

$$\frac{x_i + x_{i+1}}{2} < \tilde{x}_{i+1} < \frac{x_{i+1} + x_{i+2}}{2}, \quad i = 0, 1, \ldots, n-2$$

Mock-Chebyshev interpolation

The mock-Chebyshev grid with our method

- The distance between the points $\bar{x}_0, \bar{x}_1, \cdots, \bar{x}_m$ ensures that a grid point $\bar{x}_k$ nearest to a Chebyshev-Lobatto point x_j is never repeated. Moreover, numerical results suggest that

$$\max_{0 \le j \le n} |x_j - \tilde{x}_j| = O((n \ln n)^{-1})$$

- The computational cost is $\mathcal{O}(n)$ (Compare this with $\mathcal{O}(nm)$)

Mock-Chebyshev interpolation

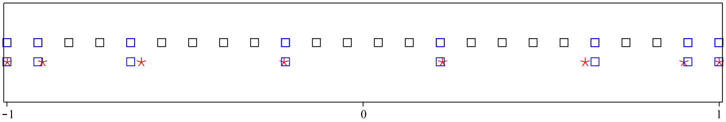
Example: 8 mock-Chebyshev points with our method

$$X_7 = \left\{x_j = -\cos\left(\frac{j\pi}{7}\right), \quad j = 0, 1, \ldots, 7\right\}$$

$$\tilde{X}_7 = \left\{-1, -\frac{21}{23}, -\frac{15}{23}, -\frac{5}{23}, \frac{5}{23}, \frac{15}{23}, \frac{21}{23}, 1\right\}$$

$$\bar{X}_{23} = \left\{\bar{x}_k = -1 + \frac{2k}{23}, \quad k = 0, 1, \ldots, 23\right\}$$

$$\tilde{X}_7 \subset \bar{X}_{23}$$

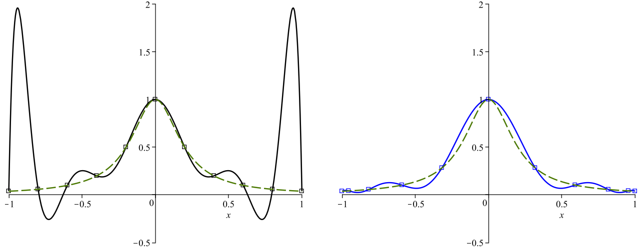


Graph of 8 Chebyshev-Lobatto points ($*$) with their mock-Chebyshev points ($\square$) with gap 2/23 for degree $n = 7$

Mock-Chebyshev interpolation

Mock-Chebyshev interpolation for Runge function, $n = 10$

$$f(x) = \frac{1}{1+25x^2}, \quad -1 \le x \le 1$$

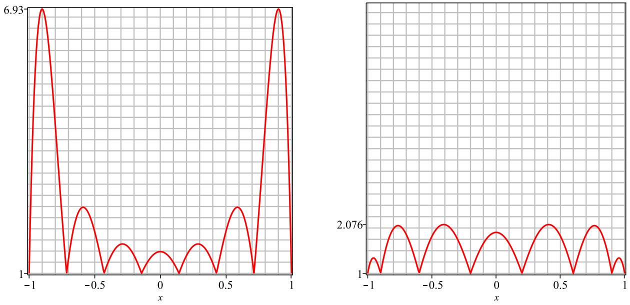


Polynomial interpolation with equidistant (□) and with mock-Chebyshev points (□)

Mock-Chebyshev interpolation

Lebesgue function

$$L_n(x) = \sum_{i=0}^{n} |\ell_i(x)|, \quad -1 \le x \le 1$$



Lebesgue function for equidistant and for mock-Chebyshev points, $n = 7$

Mock-Chebyshev interpolation

Lebesgue constants

Lebesgue constants of Chebyshev-Lobatto points and mock-Chebyshev points on $[-1, 1]$

n	Chebyshev-Lobatto points	mock-Chebyshev points
5	1.99	2.25
10	2.42	2.58
20	2.87	2.87
40	3.31	3.33
100	3.89	3.80

Bivariate polynomial interpolation

Problem statement

Given: (x_i, y_i, f_i), $i = 0, \ldots, N$, $\quad (x_i, y_i) \in [-1, 1]^2$

$$N + 1 = \dim\left(\Pi_n^2\right) = \frac{(n+1)(n+2)}{2}$$

Find:

$$p_n(x, y) = \sum_{k_1+k_2=0}^{n} c_{k_1 k_2} x^{k_1} y^{k_2}, \quad p_n(x_i, y_i) = f_i$$

Bivariate polynomial interpolation

Problem statement

Unisolvence: $\det(\tau_n) \neq 0$

$$\mathbf{x} = (x, y) \in \{\mathbf{x} \,:\, \|\mathbf{x}\|_\infty \le 1\}, \quad \mathbf{x}_i = (x_i, y_i), \quad i = 0, \dots, N$$

$$\varphi_n(x, y) = \left(x^0 y^0, \dots, x^{j_1} y^{j_2}, \dots, x^0 y^n\right), \quad 0 \le j_1 + j_2 \le n$$

$$\tau_n(\mathbf{x}_0, \dots, \mathbf{x}_N) = \begin{bmatrix} \varphi_n(x_0, y_0) \\ \vdots \\ \varphi_n(x_N, y_N) \end{bmatrix}$$

$$\ell_i(\mathbf{x}) = \frac{\det(\tau_n(\mathbf{x}_0, \dots, \mathbf{x}_{i-1}, \mathbf{x}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_N))}{\det(\tau_n(\mathbf{x}_0, \dots, \mathbf{x}_N))}$$

$$p_n(x, y) = \sum_{i=0}^{N} f_i \ell_i(\mathbf{x}), \quad \det(\tau_n) \neq 0$$

Bivariate polynomial interpolation

Problem statement

Error estimate:

$$\max_{\|(x,y)\|_\infty \le 1} |f(x,y) - p_n(x,y)| \le (1+\Lambda_n) \max_{\|(x,y)\|_\infty \le 1} |f(x,y) - p_n^*(x,y)|,$$

$$\Lambda_n(\mathbf{x}_0,\dots,\mathbf{x}_N) = \max_{\mathbf{x}\in[-1,1]^2} \sum_{i=0}^{N} |\ell_i(\mathbf{x})|$$

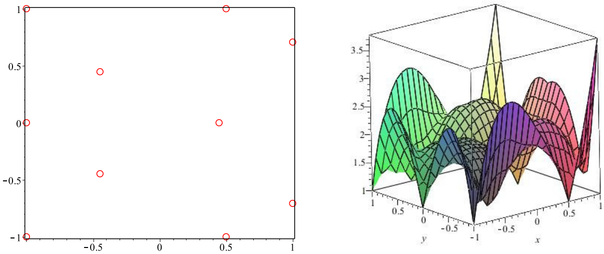
$p_n^* \in \Pi_n^2$ is the best uniform polynomial approximation to f

Order of growth:

- $[-1,1]^2$: Minimal order of growth $\mathcal{O}(\ln^2(n+1))$
  - Padua points [Caliari, De Marchi, Vianello, 2005]
  - Xu points [Bos, De Marchi, Vianello, 2006]

Bivariate polynomial interpolation

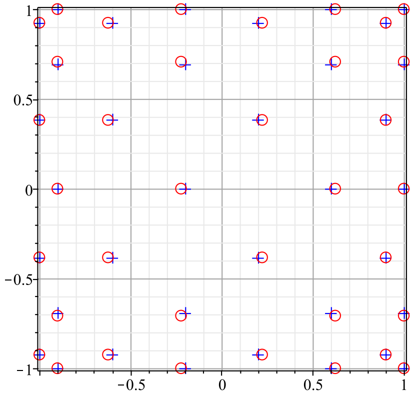
Lebesgue constant on the square [Caliari, De Marchi, Vianello, 2005]



Lebesgue function for 10 Padua points, n=3

Bivariate polynomial interpolation

Mock-Padua points



Graph of 36 Padua points ($\circ$) with their mock-Padua points ($+$) for degree $n = 7$

Bivariate polynomial interpolation

Mock-Padua points

Lebesgue constants of Padua points and mock-Padua points in the unit square

Degree Number of points	1 3	4 15	7 36	10 66	13 105	16 153	19 210
Padua points	2.00	4.41	5.84	6.88	7.71	8.41	9.01
mock-Padua points	2.00	4.56	6.10	6.89	8.07	8.55	9.18

References

1. J.P. Boyd, F. Xu, Divergence (Runge Phenomenon) for least-squares polynomial approximation on an equispaced grid and Mock-Chebyshev subset interpolation. Appl.Math.Comput. 210, 158–168, 2009.
2. L. Bos, S. De Marchi and M. Vianello, On the Lebesgue constant for the Xu interpolation formula, J. Approx. Theory 141,134–141, 2006.
3. M. Caliari, S. De Marchi, and M. Vianello, Bivariate polynomial interpolation on the square at new nodal sets, Appl. Math. Comp. 165, 261–274, 2005.
4. Vértesi, P., Optimal Lebesgue constant for Lagrange interpolation, SIAM J. Numer. Anal. 27, 1322–1331, 1990.

THANK YOU FOR YOUR ATTENTION!