Interpolation and approximation of discontinuities via mapped polynomials and discontinous kernels[1]

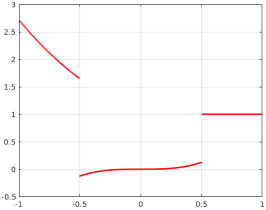
Stefano De Marchi
Department of Mathematics “Tullio Levi-Civita” &
Padua Neuroscience Center
University of Padova (Italy)
MACMAS19 - Granada, 11/09/2019

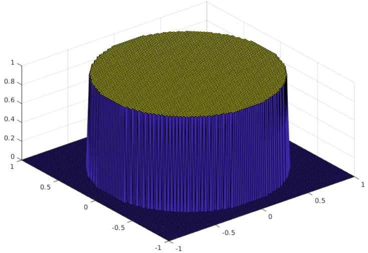
DIPARTIMENTO MATEMATICA

[1] Joint work with W. Erb (Padova), F. Marchetti (Padova), E. Perracchione (Padova) and M. Rossini (Milano-Bicocca)

Data interpolation/approximation

of discontinuous functions

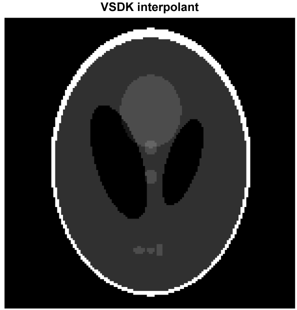


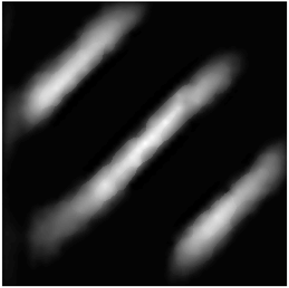


Approaches: optimal choice of interpolation points, rational approximation, sinc-approx, filtering, extrapolation,....

Data interpolation in (medical) imaging

CT, MRI, PET, SPECT, MPI, satellite images





Approaches: geometric alignement, registration, reconstruction (CT, SPECT, MPI, satellite) , ...

Let's start

- Interpolation by polynomials and rational functions of discontinuous functions is an historical approach and well-studied. Two well-known phenomena are the Runge and Gibbs effects [Runge 1901, Gibbs 1899].

Let's start

- Interpolation by polynomials and rational functions of discontinuous functions is an historical approach and well-studied. Two well-known phenomena are the Runge and Gibbs effects [Runge 1901, Gibbs 1899].
- Interpolation by kernels, mainly Radial Basis Functions, are suitable for high-dimensional scattered data problems [Hardy 1971, MJD Powell 1977, Schaback 1993 and many more], solution of PDES [e.g. Kansa 1990], machine learning [Samuel 1950, Fasshauer&McCourt 1995], image registration [e.g. Gómez-Garcia et al 2008], etc...

Let's start

- Interpolation by polynomials and rational functions of discontinuous functions is an historical approach and well-studied. Two well-known phenomena are the Runge and Gibbs effects [Runge 1901, Gibbs 1899].
- Interpolation by kernels, mainly Radial Basis Functions, are suitable for high-dimensional scattered data problems [Hardy 1971, MJD Powell 1977, Schaback 1993 and many more], solution of PDES [e.g. Kansa 1990], machine learning [Samuel 1950, Fasshauer&McCourt 1995], image registration [e.g. Gómez-Garcia et al 2008], etc...

Interpolation of discontinuos functions by mapped polynomials and discontinuous kernels is what we discuss today

1. S. De Marchi, F. Marchetti, E. Perracchione and D. Poggiali: Polynomial interpolation via mapped bases without resampling, *J. Comp. Appl. Math. 2019 (online)*
2. S. De Marchi, W. Erb, F. Marchetti, E. Perracchione and M. Rossini: Shape-driven interpolation with discontinuous kernels: error analysis, edge extraction and applications in MPI. *Submitted/reviewed*

PART I

Polynomial interpolation with mapped bases: the ”fake nodes approach”

Outline PART I

Inspiring ideas

1. In applications: samples are given. To resample (in 1d at Chebyshev points, or extract mock Chebyshev or other sets of good interpolation points that depend on applications (such as Padua pts (2005), Approximate Fekete Pts, Discrete Leja Sequences (2010), Lissajous points (2015), (P or S)-greedy (2003/05), minimal energy (199?)...)

Inspiring ideas

1. In applications: samples are given. To resample (in 1d at Chebyshev points, or extract mock Chebyshev or other sets of good interpolation points that depend on applications (such as Padua pts (2005), Approximate Fekete Pts, Discrete Leja Sequences (2010), Lissajous points (2015), (P or S)-greedy (2003/05), minimal energy (199?)...)
2. In [Adcock and Platte 2016] a similar idea was investigated for analytic functions on compact intervals by weighted least-squares of mapped polynomial basis via [Kosloff, Tal-Azer 1993] map

$$m_\alpha(x) = \frac{\sin(\alpha\pi x/2)}{\sin(\alpha\pi/2)}, \ \alpha \in (0, 1]$$

giving rise to the space

$$\mathbb{P}_n^\alpha = \{p \circ m_\alpha, \ p \in \mathbb{P}_n\}.$$

Some notations

- $I = [a, b] \subset \mathbb{R}$, $X_n \subset I$ (interpolation points), $f : I \to \mathbb{R}$ and $F_n := \{f(X_n)\}$ (f fnct to be reconstructed)
- $P_{n,f}$: interpolating polynomial, $P_{n,f} \in M_n := \mathrm{span}\{1, x, \ldots, x^n\}$ or using $L_n := \mathrm{span}\{l_0, l_1, \ldots, l_n\}$ Lagrange basis

$$P_{n,f}(x) = \sum_{i=0}^{n} l_i(x) f(x_i), \quad x \in I$$

with

$$\boxed{l_i(x) = \frac{V_i(x_0, ..., x_{i-1}, x, x_i, \ldots, x_n)}{V(x_0, \ldots, x_n)}}$$

ratio of two Vandermonde determinants

- Lebesgue constant: $\Lambda_n := \max_{x \in I} \sum_{i=0}^{n} |l_i(x)|$ which is the stability constant, norm of the projection on M_n or conditioning of the interpolation problem.

Mapping the basis

without resampling

Let $S : I \to \mathbb{R}$ be the map and $P_{n,g} : S(I) \to \mathbb{R}$ the interpolating polynomial at the mapped or “fake” nodes $S(X_n)$, that is

$$\boxed{P_{n,g}(\bar{x}) = \sum_{i=0}^{n} c_i \bar{x}^i, \quad \bar{x} \in S(I)}$$

for some $g : S(I) \to \mathbb{R} \in \mathcal{C}^r(I)$ s.t.

$$\boxed{g_{|S(X_n)} = f_{|X_n} .}$$

$\hookrightarrow$ i.e. no resampling

Cont'

without resampling

We are then interested to the function

$$\boxed{R^S_{n,f}(x) := P_{n,g}(S(x)) = \sum_{i=0}^{n} c_i S_i(x)\,,} \tag{1}$$

$R^S_{n,f}$ is the interpolant at the original nodes X_n and function values F_n spanned by mapped basis $S_n := \{S_i(\cdot),\ i = 0, \ldots, n\}$.

Cont'

without resampling

We are then interested to the function

$$\boxed{R^S_{n,f}(x) := P_{n,g}(S(x)) = \sum_{i=0}^{n} c_i S_i(x)\,,} \qquad (1)$$

$R^S_{n,f}$ is the interpolant at the original nodes X_n and function values F_n spanned by mapped basis $S_n := \{S_i(\cdot),\ i = 0, \ldots, n\}$.

Equivalence

- mapped bases approach on I: interpolate f on the set X_n via $R^s_{n,f}$ in the function space S_n.
- The "fake" nodes approach on $S(I)$: interpolate g on the set $S(X_n)$ via $P_{n,g}$ in the polynomial space M_n.

Admissible S maps

Definition

S is admissible if the resulting interpolation process has unique solution, that is the Vandermonde-like $V^S := V(S(x_0), \dots, S(x_n)) \neq 0$

Admissible S maps

Definition

S is admissible if the resulting interpolation process has unique solution, that is the Vandermonde-like $V^S := V(S(x_0), \ldots, S(x_n)) \neq 0$

- Necessity: $S(x_i) \neq S(x_j),\ \forall x_i \neq x_j$, i.e. S is injective on X_n.
- $\boxed{V^S = \sigma(X_n, S) \cdot V(x_0, \ldots, x_n)}$ with

$$\sigma(S, X_n) = \prod_{0 \le i < j \le n} \frac{S_j - S_i}{x_j - x_i} . \tag{2}$$

here $S_r = S(x_r)$.

Admissible S maps: cont'

Proposition [DeM et al. JCAM 2019]

Let l_i be the classical i-th Lagrange polynomial and let l_i^S be the S-Lagrange. Then,

$$l_i^S(x) = \gamma_i(x) l_i(x), \; x \in I, \tag{3}$$

where $\gamma_i(x) := \dfrac{\det(V_i^s(x))}{\sigma(S, X_n)\det(V_i(x))}$, with $\sigma(S, X_n)$ as before.

Actually, $\gamma_i(x) = \dfrac{\beta_i(x)}{\alpha_i}$, with

$$\boxed{\beta_i(x) := \prod_{\substack{0\le j\le n \\ j\neq i}} \frac{S(x) - S_j}{x - x_j}, \quad \alpha_i := \prod_{\substack{0\le j\le n \\ j\neq i}} \frac{S_i - S_j}{x_i - x_j}.}$$

Mapped Interpolant

$$\boxed{R^S_{n,f}(x) = \sum_{i=0}^{n} f_i l_i^S(x),\ x \in I}$$

We can define the S-Lebesgue constant

$$\Lambda_n^S := \max_{x \in S(I)} \sum_{i=0}^{n} |l_i^S(x)|\,. \tag{4}$$

so that

$$\|f - R^S_{n,f}\|_\infty \leq (1 + \Lambda_n^s) E_n^{s,\star}(f), \tag{5}$$

Observations

Obs 1

[Gross and Richards 1986] gave a remarkable formula for points in the unit disk of $\mathbb{C}$ for analytic map $S_c(z) = (1 - z)^{-c},\ c \geq 1$.

Lettting $s \in [-a, a]^n$ the determinant of the coefficient matrix can be factored as

$$\mathtt{det} M_n(s, s) = c_n V(s) V(s) \int_{U(n)} \mathtt{det}(I - susu^{-1})^{-(c+n-1)} du$$

where $U(n)$ the group of unitary complex matrices (see also [Bos,DeM,Levenberg 2014] where we discussed Fekete points for ridge functions).

Obs 2

Generalized Vandermonde determinants give rise to similar factorization [DeM 2001, 2002]

Lebesgue bound

Theorem [DeM et al. JCAM 2019]

We have that

$$\boxed{\Lambda_n^S \le \left(\frac{L}{D}\right)^n \Lambda_n,}$$

where

$$L = \max_j \max_{x \in I} \left| \frac{S(x) - S_j}{x - x_j} \right|,$$

$$D = \min_i \min_{j \neq i} \left| \frac{S_i - S_j}{x_i - x_j} \right|,$$

Lebesgue bound

Theorem [DeM et al. JCAM 2019]

We have that

$$\boxed{\Lambda_n^S \leq \left(\frac{L}{D}\right)^n \Lambda_n,}$$

where

$$L = \max_j \max_{x\in I} \left|\frac{S(x) - S_j}{x - x_j}\right|,$$

$$D = \min_i \min_{j\neq i} \left|\frac{S_i - S_j}{x_i - x_j}\right|,$$

Sketch. We proceed by giving an upper bound for $|\beta_i|$: $|\beta_i(x)| \leq \prod_{\substack{0\leq j\leq n\\ j\neq i}} L_i^j$, where $L_i^j := \max_{x\in I}\left|\frac{S(x)-S_j}{x-x_j}\right|$. Thus, $|\beta_i(x)| \leq L_i^n$, We then give a lower bound for $|\alpha_i|$ obtaining $|\alpha_i| \geq D_i^n$, where $D_i := \min_{j\neq i}\left|\frac{S_i-S_j}{x_i-x_j}\right|$. We have that

$$|\ell_i^S(x)| \leq \left(\frac{L_i}{D_i}\right)^n |\ell_i(x)|.$$

Therefore, defining $L := \max_i L_i$, $D := \min_i D_i$ and considering the sum of the Lagrange polynomials, we obtain $\boxed{\Lambda_n^S \leq \left(\frac{L}{D}\right)^n \Lambda_n}$. □

The ”fake” nodes approach: summary

- Constructing $R^S_{n,f}$ is equivalent to build the polynomial interpolant at the ”fake” nodes.
- If l_i is the Lagrange polynomial at $S(X_n)$, then at $\bar{x} = S(x) \in S(I)$

$$l_i(\bar{x}) = l_i(S(x)) = \prod_{\substack{0 \le j \le n \\ j \ne i}} \frac{S(x) - S(x_j)}{S(x_i) - S(x_j)} = l_i^S(x), \quad x \in I.$$

- As a consequence, we obtain $\Lambda_n^S(I) = \Lambda_n(S(I))$.

The "fake" nodes approach: summary

- Constructing $R^S_{n,f}$ is equivalent to build the polynomial interpolant at the "fake" nodes.
- If l_i is the Lagrange polynomial at $S(X_n)$, then at $\bar{x} = S(x) \in S(I)$

$$l_i(\bar{x}) = l_i(S(x)) = \prod_{\substack{0 \le j \le n \\ j \ne i}} \frac{S(x) - S(x_j)}{S(x_i) - S(x_j)} = l_i^S(x)\,, \quad x \in I.$$

- As a consequence, we obtain $\Lambda_n^S(I) = \Lambda_n(S(I))$.

$\hookrightarrow$ Remark: find a suitable admissible map S. $\hookleftarrow$

S-Runge

AIM: find a S s.t the resulting set of fake nodes $S(X_n)$ guarantees a stable interpolation process. The "natural" choice: Chebyshev-Lobatto (CL) nodes on the interval I

Algorithm (S-Runge)

1. ***Inputs:*** *X_n (ordered left-right).*
2. ***Main procedure:*** *Let $\mathcal{C}_{n+1}$ be the CL nodes.*
   - *For $x \in [x_i, x_{i+1}]$, $i = 0, \ldots, n-1$, define S as the piecewise linear interpolant*

     $$S(x) = \beta_{1,i}(x - x_i) + \beta_{2,i},$$

     where

     $$\beta_{1,i} = \frac{c_{i+1} - c_i}{x_{i+1} - x_i}, \quad \beta_{2,i} = c_i.$$
3. ***Outputs:*** *S.*

S-Gibbs

Obs: jump discontinuities set of f is

$$\boxed{D_m := \left\{(\xi_i, d_i) \,|\, \xi_i \in (a,b),\ \xi_i < \xi_{i+1},\ \text{ and } d_i := |f(\xi_i^+) - f(\xi_i^-)|\right\}, i = 0, \dots, m.}$$

Remark: to identify jumps/discontinuities see e.g. [Canny IEEE 1986, Archibald et al. SINUM2005, Romani et al. JCAM 2019] and Sestini's talk, Pepe's poster.

Algorithm (S-Gibbs)

1. ***Inputs:*** *X_n, D_m and $k > 0$.*
2. ***Main procedure:***
   - *1. $\alpha_i := kd_i,\ i = 0, \dots, m$.*
     2. Letting $A_i = \sum_{j=0}^{i} \alpha_j$, define S as follows:

$$S(x) = \begin{cases} x, & \text{for } x \in [a, \xi_0[, \\ x + A_i, & \text{for } x \in [\xi_i, \xi_{i+1}[,\ 0 \le i < m, \text{ or } x \in [\xi_m, b]. \end{cases}$$

3. ***Outputs:*** *S.*

Remarks (S-Gibbs)

- Our strategy consists in constructing the map S in such a way that it sufficiently increases the gap between the node right before and the one right after the discontinuity via the real parameters α_i.
- About the *shifting parameter* $k > 0$. We experimentally observed that its selection is not critical. The resulting interpolation process is not sensitive to its choice, provided that it is sufficiently large, i.e. in such a way that in the mapped space the so-constructed function g has no steep gradients;

Cardinals

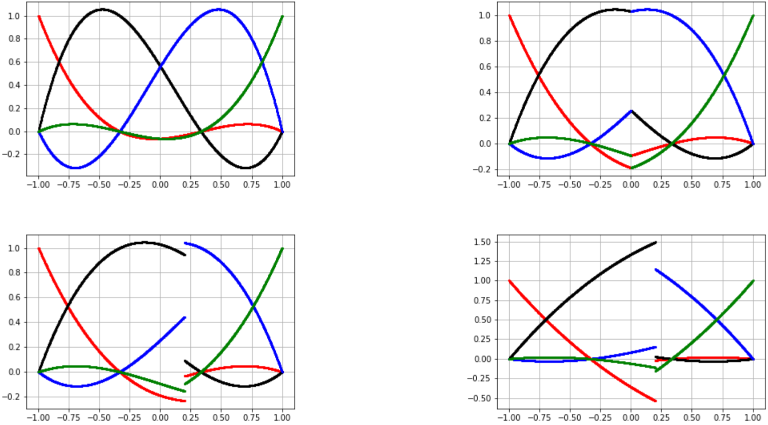


Figure: Left-right, up-down: the original cardinals on 4 nodes, the cardinals around $\xi = 0, k = 0$ the cardinals around $\xi = 0.2, k = 1$,the cardinals around $\xi = 0, k = 0.5$.

Runge example

$I = [-5, 5]$, $f_1 = 1/(1 + x^2)$, X_n: equal or random pts, E_n evaluation pts.
Relative Maximum Absolute Error (RMAE)

$$\mathrm{RMAE} = \max_{i=0,\ldots,m} \frac{|R^s_{n,f}(\bar{x}_i) - f(\bar{x}_i)|}{|f(\bar{x}_i)|}.$$

E_n are equally spaced pts.

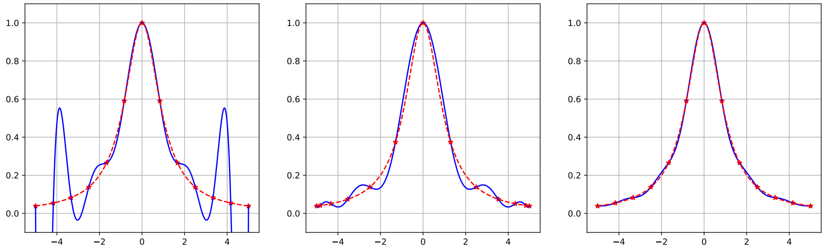


Figure: Interpolation with 13 points of the Runge function on $[-5, 5]$ using equispaced (left), CL (center) and fake nodes (right). The nodes are represented by stars, the original and reconstructed functions are plotted with continuous red and dotted blue lines, respectively.

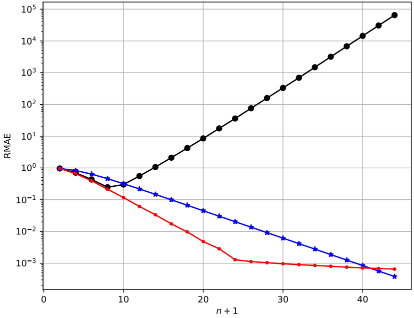


Figure: The RMAE for the Runge function varying the number of nodes. The results with equispaced, CL and fake nodes are represented by black circles, blue stars and red dots, respectively.

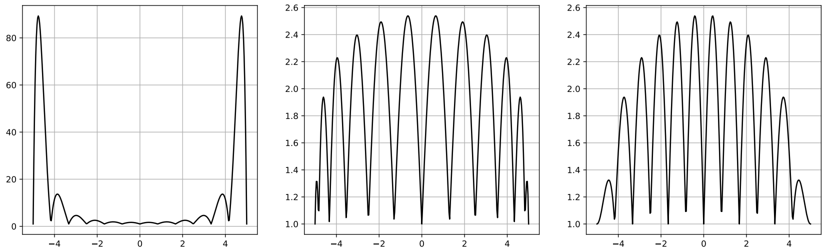


Figure: Lebesgue functions of equispaced (left), CL (center) and fake CL (right) nodes.

Gibbs example

$$f_2(x) := \begin{cases} \dfrac{x^2}{10}, & -5 \le x < -\frac{3}{2}, \\ \dfrac{1}{4}x + \dfrac{19}{8}, & -\frac{3}{2} \le x < \frac{5}{2}, \\ -\dfrac{x^3}{30} + 4, & \frac{5}{2} \le x \le 5. \end{cases}$$

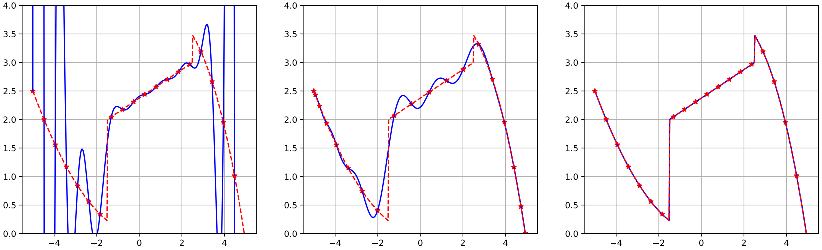


Figure: Interpolation at 20 points of the function f_2 on $[-5, 5]$, using equispaced (left), CL nodes (center) and the discontinuous map (right). The nodes are represented by stars, the original and reconstructed functions are plotted with continuous red and dotted blue lines, respectively.

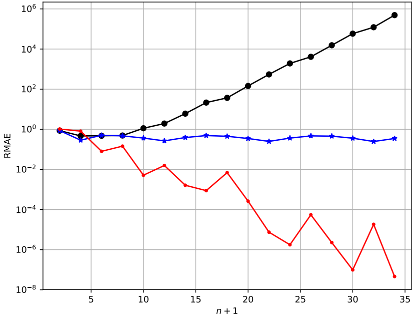


Figure: The RMAE for the function f_2 varying the number of nodes. The results with equispaced, CL and fake nodes are represented by black circles, blue stars and red dots, respectively.

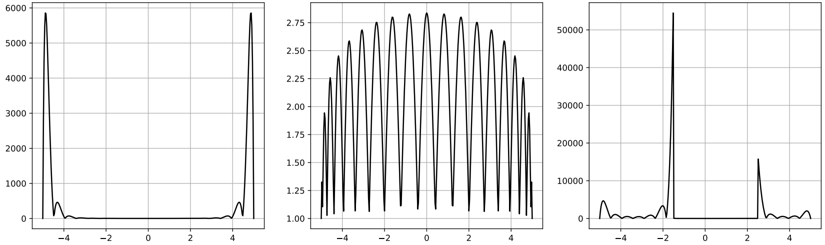


Figure: Lebesgue functions of equispaced (left), CL (center) and fake nodes (right).

The abs function

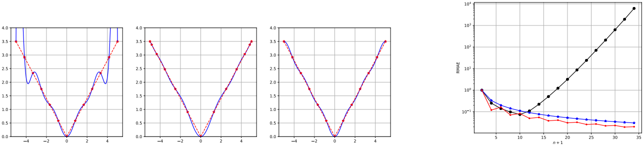


Figure: Runge for $f(x) = 7|x|$

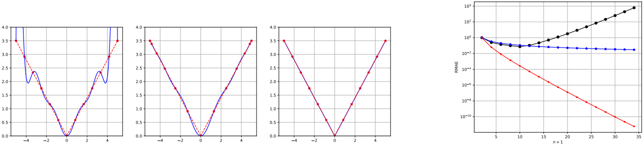


Figure: Gibbs for $f(x) = 7|x|$

Two gifs

What we are doing

- Extensions to rational functions, in particular the Floater-Hormann (FH) and trigonometric FH (for periodic signals)
- In the 2d case, we have results on discontinuous functions on the square, using polynomial approximation at the Padua points or tensor product meshes;
- in 2d and 3d we can extract Approximate Fekete Points on various domains (disk, sphere, polygons, spherical caps, lunes, ...)
- In higher dimensions we could consider to the so-called Lissajous points or Varyably Scaled Discontinuos Kernels (VSDK) for scattered data (see part II)

Links: https://www.math.unipd.it/~marcov/CAA.html,
https://en.wikipedia.org/wiki/Padua_points,
https://en.wikipedia.org/wiki/Runge%27s_phenomenon#S-Runge_algorithm_without_resampling

Cont

Stability and error analysis

- We are working in improving the error analysis and bounding the Lebesgue constant(s).
- Applications: Image registration in nuclear medicine, periodic signals,...

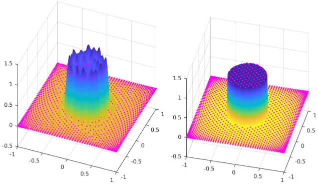


Figure: Left: interpolation with PD60 of a function with a circular jump. Right: the same by mapping circularly the PD points, and using least-squares fake-Padua

PART II

Variably Scaled Discontinuous Kernels (VSDK)

Outline PART II

Generality on kernel-based approximation

- $\phi : [0,\infty) \to \mathbb{R}$, Conditionally Positive Definite (CPD) of order ℓ or Strictly Positive Definite (SPD) and radial

  - globally supported:

name	ϕ	ℓ
Gaussian C^∞ (GA)	$e^{-\varepsilon^2 r^2}$	0
Generalized Multiquadrics C^∞ (GM)	$(1 + r^2/\varepsilon^2)^{3/2}$	2

  - locally supported:

name	ϕ	ℓ
Wendland C^2 (W2)	$(1-\varepsilon r)^4_+ \, (4\varepsilon r + 1)$	0
Buhmann C^2 (B2)	$2r^4 \log r - 7/2r^4 + 16/3r^3 - 2r^2 + 1/6$	0

Generality on kernel-based approximation

- $\phi : [0, \infty) \to \mathbb{R}$, Conditionally Positive Definite (CPD) of order ℓ or Strictly Positive Definite (SPD) and radial
  - globally supported:

name	ϕ	ℓ
Gaussian C^∞ (GA)	$e^{-\varepsilon^2 r^2}$	0
Generalized Multiquadrics C^∞ (GM)	$(1 + r^2/\varepsilon^2)^{3/2}$	2

  - locally supported:

name	ϕ	ℓ
Wendland C^2 (W2)	$(1 - \varepsilon r)_+^4 \, (4\varepsilon r + 1)$	0
Buhmann C^2 (B2)	$2r^4 \log r - 7/2r^4 + 16/3r^3 - 2r^2 + 1/6$	0

- we often consider $\phi(\varepsilon \cdot)$, with ε called shape parameter

Generality on kernel-based approximation

- $\phi : [0, \infty) \to \mathbb{R}$, Conditionally Positive Definite (CPD) of order ℓ or Strictly Positive Definite (SPD) and radial
  - globally supported:

name	ϕ	ℓ
Gaussian C^∞ (GA)	$e^{-\varepsilon^2 r^2}$	0
Generalized Multiquadrics C^∞ (GM)	$(1 + r^2/\varepsilon^2)^{3/2}$	2

  - locally supported:

name	ϕ	ℓ
Wendland C^2 (W2)	$(1 - \varepsilon r)_+^4 (4\varepsilon r + 1)$	0
Buhmann C^2 (B2)	$2r^4 \log r - 7/2r^4 + 16/3r^3 - 2r^2 + 1/6$	0

- we often consider $\phi(\varepsilon \cdot)$, with ε called shape parameter
- kernel notation $K(\mathbf{x}, \mathbf{y}) (= K_\varepsilon(\mathbf{x}, \mathbf{y})) = \Phi_\varepsilon(\mathbf{x} - \mathbf{y}) = \phi(\varepsilon \|\mathbf{x} - \mathbf{y}\|_2)$

Generality on kernel-based approximation

- $\phi : [0, \infty) \to \mathbb{R}$, Conditionally Positive Definite (CPD) of order ℓ or Strictly Positive Definite (SPD) and radial
  - globally supported:

name	ϕ	ℓ
Gaussian C^∞ (GA)	$e^{-\varepsilon^2 r^2}$	0
Generalized Multiquadrics C^∞ (GM)	$(1 + r^2/\varepsilon^2)^{3/2}$	2

  - locally supported:

name	ϕ	ℓ
Wendland C^2 (W2)	$(1 - \varepsilon r)_+^4 (4\varepsilon r + 1)$	0
Buhmann C^2 (B2)	$2r^4 \log r - 7/2r^4 + 16/3r^3 - 2r^2 + 1/6$	0

- we often consider $\phi(\varepsilon\cdot)$, with ε called shape parameter
- kernel notation $K(\mathbf{x}, \mathbf{y})(= K_\varepsilon(\mathbf{x}, \mathbf{y})) = \Phi_\varepsilon(\mathbf{x} - \mathbf{y}) = \phi(\varepsilon\|\mathbf{x} - \mathbf{y}\|_2)$
- native space $\mathcal{N}_K(\Omega)$ (where K is the reproducing kernel)
- finite subspace $\mathcal{N}_K(X) = \mathrm{span}\{K(\cdot, x) : x \in X, |X| = N\} \subset \mathcal{N}_K(\Omega)$.
- $P_{f,X} = \sum_{i=1}^{N} c_i K(\cdot, \mathbf{x}_i)$ is the kernel-based interpolant

Separation distance, fill-distance and power function

$$q_X := \frac{1}{2} \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|_2 \,, \quad (\textit{separation distance}) \qquad h_{X,\Omega} := \sup_{\mathbf{x} \in \Omega} \min_{\mathbf{x}_j \in X} \|\mathbf{x} - \mathbf{x}_j\|_2 \,, \quad (\textit{fill} - \textit{distance})$$

$$P_{\Phi_\varepsilon, X}(\mathbf{x}) := \sqrt{\Phi_\varepsilon(\mathbf{0}) - (\mathbf{u}^*(\mathbf{x}))^{\mathbf{T}} \, \mathbf{A} \, \mathbf{u}^*(\mathbf{x})} \,, \quad (\textit{power function}) \qquad A_{ij} = K(\mathbf{x}_i, \mathbf{x}_j), \ \mathbf{u}^* \ \textbf{vector of cardinal functions}$$

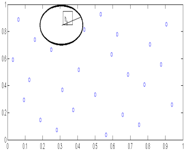


Figure: The fill-distance of 25 Halton points $h \approx 0.2667$

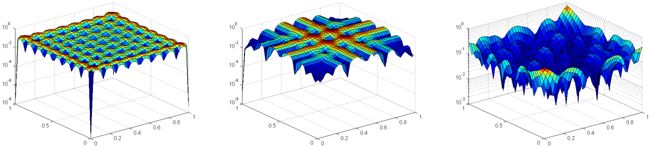


Figure: Power function for the Gaussian kernel with $\varepsilon = 6$ on a grid of 81 uniform, Chebyshev and Halton points, respectively.

Pointwise error estimates

see Wendland's (2005) or Fasshauer's (2007) books

Theorem

Let $\Omega \subset \mathbb{R}^d$ and $K \in \mathcal{C}(\Omega \times \Omega)$ be PD on $\mathbb{R}^d$. Let $X = \{\mathbf{x}_1, \ldots, , \mathbf{x}_n\}$ be a set of distinct points. Take a function $f \in \mathcal{N}_\Phi(\Omega)$ and denote with P_f its interpolant on X. Then, for every $\mathbf{x} \in \Omega$

$$|f(\mathbf{x}) - P_f(\mathbf{x})| \le P_{\Phi_\varepsilon, X}(\mathbf{x}) \|f\|_{\mathcal{N}_K(\Omega)}\,. \tag{6}$$

Theorem

Let $\Omega \subset \mathbb{R}^d$ and $K \in \mathcal{C}^{2\kappa}(\Omega \times \Omega)$ be symmetric and positive definite, $X = \{\mathbf{x}_1, \ldots, , \mathbf{x}_N\}$ a set of distinct points. Consider $f \in \mathcal{N}_K(\Omega)$ and its interpolant P_f on X. Then, there exist positive constants h_0 and C (independent of $\mathbf{x}$, f and Φ), with $h_{X,\Omega} \le h_0$, such that

$$|f(\mathbf{x}) - P_f(\mathbf{x})| \le C\, h_{X,\Omega}^{\kappa} \sqrt{C_K(\mathbf{x})} \|f\|_{\mathcal{N}_K(\Omega)}\,. \tag{7}$$

and $C_K(\mathbf{x}) = \max_{|\beta|=2\kappa} \quad \max_{\mathbf{w},\mathbf{z} \in \Omega \cup B(\mathbf{x}, c_2 h_{X,\Omega})} |D_2^\beta \Phi(\mathbf{w}, \mathbf{z})|$

Strategies for controlling the interpolation error

Obs

The choice of the shape parameter ε in order to get the smallest (possible) interpolation error is crucial.

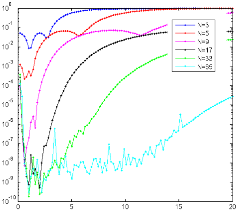
- Trial and Error
- Power function minimization
- Leave One Out Cross Validation (LOOCV)

Strategies for controlling the interpolation error

Obs

The choice of the shape parameter ε in order to get the smallest (possible) interpolation error is crucial.

- Trial and Error
- Power function minimization
- Leave One Out Cross Validation (LOOCV)



Trial and error strategy: interpolation of the 1d *sinc* function with Gaussian for $\varepsilon \in [0, 20]$, taking 100 values of ε and $N = 2^k + 1, k = 1, \ldots, 6$ equispaced data points.

Trade-off principle

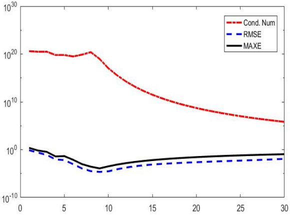


Figure: RMSE, MAXERR and 2-Condition Number with 30 values of $\varepsilon \in [0.1, 20]$, for interpolation of the Franke function on a grid of 40×40 Chebyshev points

Trade-off or uncertainty principle [Schaback 1995]

- Accuracy vs Stability
- Accuracy vs Efficiency
- Accuracy and stability vs problem size

From RBF to VSK interpolation

Main motivation

The shape parameter ε is a crucial computational issue in RBF interpolation (tuning strategies, cross validation,...)

From RBF to VSK interpolation

Main motivation

The shape parameter ε is a crucial computational issue in RBF interpolation (tuning strategies, cross validation,...)

VSK

To overcome such problems, [Bozzini et al 2015] introduced the so-called Variably Scaled Kernels (VSK)

From RBF to VSK interpolation

Main motivation

The shape parameter ε is a crucial computational issue in RBF interpolation (tuning strategies, cross validation,...)

VSK

To overcome such problems, [Bozzini et al 2015] introduced the so-called Variably Scaled Kernels (VSK)

Scale function

The classical tuning strategy of finding the optimal shape parameter, is now based on a scaling function which plays the role of a density function.

Variably Scaled Kernels (VSK)

[Bozzini et al. 2015]

Definition

Letting $\Sigma \subseteq (0, +\infty)$ *and* Φ *a positive definite radial kernel on* $\Omega \times \Sigma \subset \mathbb{R}^{d+1}$*, depending on the shape parameter* $\varepsilon > 0$*. Given a scaling function* $\psi : \Omega \longrightarrow \Sigma$*, we define a VSK* Φ_ψ *on* Ω *as*

$$\boxed{\Phi_\psi(\mathbf{x}, \mathbf{y}) := \Phi((\mathbf{x}, \psi(\mathbf{x})), (\mathbf{y}, \psi(\mathbf{y}))), \quad \forall\, \mathbf{x}, \mathbf{y} \in \Omega.} \tag{8}$$

Variably Scaled Kernels (VSK)

[Bozzini et al. 2015]

Definition

Letting $\Sigma \subseteq (0, +\infty)$ *and* Φ *a positive definite radial kernel on* $\Omega \times \Sigma \subset \mathbb{R}^{d+1}$*, depending on the shape parameter* $\varepsilon > 0$*. Given a scaling function* $\psi : \Omega \longrightarrow \Sigma$*, we define a VSK* Φ_ψ *on* Ω *as*

$$\boxed{\Phi_\psi(\mathbf{x}, \mathbf{y}) := \Phi((\mathbf{x}, \psi(\mathbf{x})), (\mathbf{y}, \psi(\mathbf{y}))), \quad \forall\ \mathbf{x}, \mathbf{y} \in \Omega.} \tag{8}$$

Obs

- *if* Φ *is radial* $\Phi_\psi(\mathbf{x}, \mathbf{y}) = \Phi(\|\mathbf{x} - \mathbf{y}\|^2 + (\psi(\mathbf{x}) - \psi(\mathbf{y}))^2)$,
- *in Machine Learning this is known as Kernel Trick.*

VSK interpolant

- Set $\Psi(\mathbf{x}) = (\mathbf{x}, \psi(\mathbf{x}))$.
  The interpolant on the node set
  $$\Psi(X) := \{(\mathbf{x}_k, \psi(\mathbf{x}_k)),\ \mathbf{x}_k \in X\},$$
  (we choose $\varepsilon = 1$) takes the form
  $$\boxed{P_f(\Psi(\boldsymbol{x})) = \sum_{k=1}^{N} c_k \Phi(\Psi(\boldsymbol{x}), \Psi(\boldsymbol{x}_k)), \quad \boldsymbol{x} \in \Omega,\ \boldsymbol{x}_k \in X}$$

- Given interpolant P_f on $\mathbb{R}^{d+1}$, we can project back on Ω the points $(x, \psi(x)) \in \mathbb{R}^{d+1}$. The VSK interpolant $\mathcal{V}_f$ on Ω , belongs to $\mathrm{span}\{\Phi_\psi(\cdot, \boldsymbol{x}_k),\ k = 1, \dots, N\}$ and by using (8)
  $$\boxed{\mathcal{V}_f(\mathbf{x}) := \sum_{k=1}^{N} c_k \Phi_\psi(\boldsymbol{x}, \boldsymbol{x}_k) = \sum_{k=1}^{N} c_k \Phi(\Psi(\boldsymbol{x}), \Psi(\boldsymbol{x}_k)) = P_f(\Psi(\boldsymbol{x})).} \tag{9}$$

Variably Scaled Discontinous Kernels (VSDK)

References

- VSK for discontinuous functions [Rossini 2017],
- VSK-PU for elliptic PDEs [DeM et al, 2018].
- Variably Scaled Discontinuos Kernels presented [DeM, Marchetti, Perracchione, 2019]

Learning 1d example [DeM, Marchetti, Perracchione, 2019]

Let $\Omega = (a, b) \subset \mathbb{R}$ be an open interval and let $\xi \in \Omega$. We consider a function f that has a jump discontinuity in ξ

$$f(x) := \begin{cases} f_1(x), & a < x < \xi, \\ f_2(x), & \xi \leq x < b, \end{cases}$$

where

$$\lim_{x \to \xi^-} f_1(x) \neq f_2(\xi) \, .$$

Problem/Solution

- Problem: approximating f on node set $X \subset \Omega$ originates the Gibbs phenomenon.
- Solution: consider VSK-like interpolants as follows.

Cont '

- Let $\alpha,\beta \in \mathbb{R}_{>0},\ \alpha \neq \beta$ and $\mathcal{S} = \{\alpha,\beta\}$. As scaling function consider

$$\psi(x) := \begin{cases} \alpha, & x < \xi, \\ \beta, & x \geq \xi. \end{cases}$$

  ψ is piecewise constant, having a jump discontinuity at ξ as the function f.

- The interpolant $\mathcal{V}_\psi$ on $X = \{x_k,\ k = 1, \ldots, N\}$ is then a linear combination of discontinuous functions $\Phi_\psi(\cdot, x_k)$ having a jump at ξ
  - if $a < x_k < \xi$

$$\Phi_\psi(x, x_k) = \begin{cases} \phi(|x - x_k|), & x < \xi, \\ \phi(\|(x,\alpha) - (x_k,\beta)\|_2), & x \geq \xi, \end{cases}$$

  - if $\xi \leq x_k < b$

$$\Phi_\psi(x, x_k) = \begin{cases} \phi(|x - x_k|), & x \geq \xi, \\ \phi(\|(x,\alpha) - (x_k,\beta)\|_2), & x < \xi. \end{cases}$$

Definition, 1d case

Definition

Let $\Omega = (a, b) \subset \mathbb{R}$ *be an open interval,* $\mathcal{S} = \{\alpha, \beta\}$ *with* $\alpha \neq \beta \in \mathbb{R}_{>0}$ *and let* $\mathcal{D} = \{\xi_j,\ j = 1, \dots, \ell\} \subset \Omega$ *be the set of the jumps,* $\xi_j < \xi_{j+1}$ *for all* j*. Define* $\psi : \Omega \longrightarrow \mathcal{S}$ *s.t.*

$$\psi(x) := \begin{cases} \alpha, & x \in (a, \xi_1) \text{ or } x \in [\xi_j, \xi_{j+1}), \quad \text{where } j \text{ is even,} \\ \beta, & x \in [\xi_j, \xi_{j+1}), \quad \text{where } j \text{ is odd,} \end{cases}$$

and

$$\psi(x)_{|[\xi_\ell, b)} := \begin{cases} \alpha, & \ell \text{ is even,} \\ \beta, & \ell \text{ is odd.} \end{cases}$$

Definition, 1d case

Definition

Let $\Omega = (a,b) \subset \mathbb{R}$ be an open interval, $\mathcal{S} = \{\alpha, \beta\}$ with $\alpha \neq \beta \in \mathbb{R}_{>0}$ and let $\mathcal{D} = \{\xi_j,\ j = 1, \ldots, \ell\} \subset \Omega$ be the set of the jumps, $\xi_j < \xi_{j+1}$ for all j. Define $\psi : \Omega \longrightarrow \mathcal{S}$ s.t.

$$\psi(x) := \begin{cases} \alpha, & x \in (a, \xi_1) \text{ or } x \in [\xi_j, \xi_{j+1}), \quad \text{where } j \text{ is even}, \\ \beta, & x \in [\xi_j, \xi_{j+1}), \quad \text{where } j \text{ is odd}, \end{cases}$$

and

$$\psi(x)_{|[\xi_\ell, b)} := \begin{cases} \alpha, & \ell \text{ is even}, \\ \beta, & \ell \text{ is odd}. \end{cases}$$

The corresponding kernel Φ_ψ is a VSDK on Ω.

Definition, 1d case

Definition

Let $\Omega = (a, b) \subset \mathbb{R}$ be an open interval, $\mathcal{S} = \{\alpha, \beta\}$ with $\alpha \neq \beta \in \mathbb{R}_{>0}$ and let $\mathcal{D} = \{\xi_j,\ j = 1, \ldots, \ell\} \subset \Omega$ be the set of the jumps, $\xi_j < \xi_{j+1}$ for all j. Define $\psi : \Omega \longrightarrow \mathcal{S}$ s.t.

$$\psi(x) := \begin{cases} \alpha, & x \in (a, \xi_1) \text{ or } x \in [\xi_j, \xi_{j+1}), \quad \text{where } j \text{ is even}, \\ \beta, & x \in [\xi_j, \xi_{j+1}), \quad \text{where } j \text{ is odd}, \end{cases}$$

and

$$\psi(x)_{|[\xi_\ell, b)} := \begin{cases} \alpha, & \ell \text{ is even}, \\ \beta, & \ell \text{ is odd}. \end{cases}$$

The corresponding kernel Φ_ψ is a VSDK on Ω.

Obs

Dealing with functions having jumps becomes "natural" to use linear combination of $\Phi_\psi(\cdot, x_k)$ functions having jumps at the same locations. But the error analysis requires another approach !

VSDK, analysis

DIPARTIMENTO MATEMATICA

Letting Ω and $\mathcal{D}$ as before. Take $n \in \mathbb{N}$ and $\psi_n : \Omega \longrightarrow \Sigma \subseteq (0, +\infty)$

$$\psi_n(x) := \begin{cases} \alpha, & x \in (a, \xi_1 - 1/n) \text{ or } x \in [\xi_j + 1/n, \xi_{j+1} - 1/n) \quad j \text{ is even}, \\ \beta, & x \in [\xi_j + 1/n, \xi_{j+1} - 1/n) \quad j \text{ is odd}, \\ \gamma_1(x), & x \in [\xi_j - 1/n, \xi_j + 1/n) \quad j \text{ is odd}, \\ \gamma_2(x), & x \in [\xi_j - 1/n, \xi_j + 1/n) \quad j \text{ is even}, \end{cases} \tag{10}$$

$$\psi_n(x)_{|[\xi_\ell + 1/n, b)} := \begin{cases} \alpha, & \ell \text{ is even}, \\ \beta, & \ell \text{ is odd}, \end{cases}$$

where γ_1, γ_2 are continuous, strictly monotonic functions and

$$\lim_{x \to \xi_{j+1} + 1/n} \gamma_1(x) = \gamma_2(\xi_j - 1/n) = \beta, \qquad \lim_{x \to \xi_{j+1} + 1/n} \gamma_2(x) = \gamma_1(\xi_j - 1/n) = \alpha.$$

VSDK, analysis

Letting Ω and $\mathcal{D}$ as before. Take $n \in \mathbb{N}$ and $\psi_n : \Omega \longrightarrow \Sigma \subseteq (0, +\infty)$

$$\psi_n(x) := \begin{cases} \alpha, & x \in (a, \xi_1 - 1/n) \text{ or } x \in [\xi_j + 1/n, \xi_{j+1} - 1/n) \quad j \text{ is even}, \\ \beta, & x \in [\xi_j + 1/n, \xi_{j+1} - 1/n) \quad j \text{ is odd}, \\ \gamma_1(x), & x \in [\xi_j - 1/n, \xi_j + 1/n) \quad j \text{ is odd}, \\ \gamma_2(x), & x \in [\xi_j - 1/n, \xi_j + 1/n) \quad j \text{ is even}, \end{cases} \tag{10}$$

$$\psi_n(x)_{|[\xi_\ell + 1/n, b)} := \begin{cases} \alpha, & \ell \text{ is even}, \\ \beta, & \ell \text{ is odd}, \end{cases}$$

where γ_1, γ_2 are continuous, strictly monotonic functions and

$$\lim_{x \to \xi_{j+1} + 1/n} \gamma_1(x) = \gamma_2(\xi_j - 1/n) = \beta, \qquad \lim_{x \to \xi_{j+1} + 1/n} \gamma_2(x) = \gamma_1(\xi_j - 1/n) = \alpha.$$

Scaling function as pointwise limit

From Definition above it is straightforward to verify that $\forall x \in \Omega$

$$\lim_{n \to \infty} \psi_n(x) = \psi(x).$$

Example to fix idea

Take a functions with discontinuities at $\xi_1 = -0.5$ and $\xi_2 = 0.5$. By using (10), we define the scaling function for the corresponding VSKs

$$\psi_n(x) := \begin{cases} 1, & x \in (-1, \xi_1 - 1/n) \text{ or } x \in [\xi_2 + 1/n, 1), \\ 2, & x \in [\xi_1 + 1/n, \xi_2 - 1/n), \\ (nx - \xi_1 n + 3)/2, & x \in [\xi_1 - 1/n, \xi_1 + 1/n), \\ (-nx + \xi_2 n + 3)/2, & x \in [\xi_2 - 1/n, \xi_2 + 1/n). \end{cases} \tag{11}$$

The $\lim_{n\to\infty} \psi_n(x)$ is the discontinous scaling function of the VSDKs

$$\psi(x) := \begin{cases} 1, & x \in (-1, \xi_1) \text{ or } x \in [\xi_2, 1), \\ 2, & x \in [\xi_1, \xi_2). \end{cases} \tag{12}$$

Example, cont’

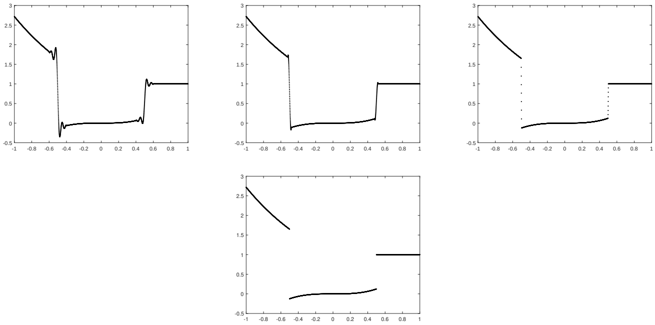


Figure: The VSK interpolant using ψ_{10}, ψ_{50}, ψ_{500} and the VSDK interpolant ψ. f_1 on X using Matérn kernel C^6.

VSDK, convergence and error bound

Theorem (DeM, Marchetti, Perracchione 2019)

For every $x, y \in \Omega$,

$$\lim_{n\to\infty} \Phi_{\psi_n}(x, y) = \Phi_\psi(x, y),$$

where Φ_ψ is the kernel considered in Definition.
The interpolant at the nodes $X = \{x_k,\ k = 1, \ldots, N\}$ on Ω is the limit

$$\lim_{n\to\infty} \mathcal{V}_{\psi_n}(x) = \mathcal{V}_\psi(x), \quad \forall\, x \in \Omega\,.$$

Power function and error bound

$$P_{\Phi_\psi, X}(x) = \lim_{n\to\infty} P_{\Phi_{\psi_n}, X}(x).$$

For all $f \in \mathcal{N}_{K_\psi(\Omega)}$ we have

$$|f(x) - \mathcal{V}_\psi(x)| \quad \leq \quad P_{\Phi_\psi, X}(x) \|f\|_{\mathcal{N}_{K_\psi(\Omega)}}, \quad x \in \Omega.$$

VSDK, multidimensional

Obs

VSDKs rely on classical RBFs and therefore they can be extended to any spatial dimension.

Partition

Let $\Omega \subset \mathbb{R}^d$ be an open subset with Lipschitz boundary and discontinuous function $f : \Omega \longrightarrow \mathbb{R}$ s.t. exists a disjoint partition $\mathcal{P} = \{R_1, \ldots, R_m\}$ of regions having Lipschitz boundaries with jumps along $(d-1)$-dimensional manifolds, say $\gamma_1, \ldots, \gamma_p$

$$\gamma_i \subseteq \bigcup_{i=1}^{m} \partial R_i \setminus \partial\Omega, \quad \forall i = 1, \ldots, p.$$

Scaling function

Let Ω as above, $\mathcal{S} = \{\alpha_1, \ldots, \alpha_m\}$ real distinct values and $\mathcal{P}$ the partition of Ω. A scaling functions ψ on Ω is $\psi(\mathbf{x})_{|R_i} := \alpha_i$.

VSDK, multidimensional

We may consider continuous scaling functions ψ_n on Ω s.t. $\forall \mathbf{x} \in \Omega$,

$$\lim_{n\to\infty} \psi_n(\mathbf{x}) = \psi(\mathbf{x}),$$

and

$$\lim_{n\to\infty} \mathcal{V}_{\psi_n}(\mathbf{x}) = \mathcal{V}_{\psi}(\mathbf{x}),$$

Theorem (Power function and error bound)

Let Φ_ψ be a VSDK as in Definition 49. Suppose that $X = \{\mathbf{x}_i, i = 1, \ldots, N\} \subseteq \Omega$ have distinct points. For all $f \in \mathcal{N}_{K_\psi(\Omega)}$ we have

$$|f(\mathbf{x}) - \mathcal{V}_\psi(\mathbf{x})| \leq P_{\Phi_\psi, X}(\mathbf{x}) \|f\|_{\mathcal{N}_{K_\psi(\Omega)}}, \quad \mathbf{x} \in \Omega.$$

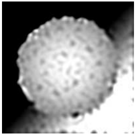
Note: error estimates in terms of the fill-distance are in [DeM, Erb et al, 2019].

An example

Let $\Omega = (-1, 1)^2$

$$f(x_1, x_2) = \begin{cases} e^{-(x_1^2+x_2^2)}, & x_1^2 + x_2^2 \leq 0.6, \\ x_1 + x_2, & x_1^2 + x_2^2 > 0.6, \end{cases}$$

We take 1089 Halton points on Ω as interpolation nodes and we evaluate the approximant on equispaced points with mesh size $1.00\mathrm{E}-2$.
We interpolate the function f via classical RBF interpolation on the set of nodes X, using the Gaussian kernel function and selecting the optimal shape parameter ε via LOOCV. Then, we compare this results with VSDKs.



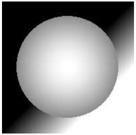


Figure: Classical RBF (left) and VSDK (right) interpolants.

What we are doing

Obs

Applications to Magnetic Particle Imaging on Lissajous points via VSDK are discussed in the recent paper [DeM, Erb et al, 2019].

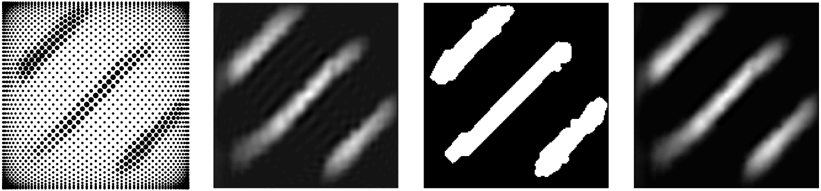


Figure: Comparison of different interpolation methods in MPI. The reconstructed data on the Lissajous nodes $\mathrm{LS}_2^{(33,32)}$ (left) is first interpolated using the polynomial scheme derived in [Erb et al 2016](middle left). Using a mask constructed upon a threshold strategy (middle right) the second interpolation is performed by the VSDK scheme (right).

Literature

1. Image compression by **CATCH** [Piazzon et al. 2017].
2. RBF-based Reduced Order Method (P-Greedy approach) [Wirtz et al. 2015].
3. Learning with kernels [Fasshauer & McCourt 2015].

◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇◇

F. Piazzon, A. Sommariva, M. Vianello, *Caratheodory-Tchakaloff Subsampling*, Dolom. Res. Notes Approx. **10** (2017), pp. 5–14.

D. Wirtz, N. Karajan, B. Haasdonk, *Surrogate Modelling of multiscale models using kernel methods*, Int. J. Numer. Met. Eng. **101** (2015), pp. 1–28.

G.E. Fasshauer, M.J. McCourt, *Kernel-based Approximation Methods Using Matlab*, World Scientific, Singapore, 2015.

Thank you, Gracias, Grazie

Rete ITaliana di Approssimazione

Italian Network on Approximation

https://sites.google.com/site/italianapproximationnetwork/