Quadrature at fake nodes

Dolomites Research Notes on Approximation

Special Issue *Multivariate Approximation: Theory and Application, MATA2020*, Volume 14 · 2021 · Pages 39–45

Quadrature at *fake nodes*

Stefano De Marchi[a,d] · Giacomo Elefante[b] · Emma Perracchione[c] · Davide Poggiali[d]

Abstract

We investigate the use of the so-called mapped bases or *fake nodes* approach in the framework of numerical integration. We show that such approach is able to mitigate the Gibbs phenomenon when integrating functions with steep gradients. Moreover, focusing on the *optimal* properties of the Chebyshev-Lobatto nodes, we are able to analytically compute the quadrature weights of the *fake* Chebyshev-Lobatto nodes. Such weights, quite surprisingly, coincide with the composite trapezoidal rule. Numerical experiments show the effectiveness of the proposed method especially for mitigating the Gibbs phenomenon without the need of resampling the given function.

1 Introduction

As well-known the Newton-Cotes formulae, that rely on univariate polynomial interpolation at equispaced points suffer from instability even for low degree polynomials (Runge and Gibbs phenomena are indeed two consequences of this [5, 6, 24, 25]). Such instability, which is common to many approximation methods (see e.g. [4, 12, 16, 17]), directly follows from the computation of the polynomial interpolant that - after being integrated - defines the quadrature rules. To overcome this problem, for instance, in [19] the authors discuss new approaches, based also on least-squares, aimed to construct stable Newton-Cotes-like formulae for high order polynomials. Moreover, in this direction, many efforts are devoted to study conformal maps that lead to resampling the function at different data in order to improve the approximation or mitigate the instability, see e.g. [1, 3]. Simply choosing *ad hoc* sets of nodes leads to resampling the (unknown) function and to avoid this drawback, here we present the use of the so-called mapped bases or *fake nodes* approach [13] in the framework of numerical integration.

To fix the notation, we need to recall the basics of univariate polynomial interpolation (for more details the unfamiliar reader can refer e.g. to the book [8]).

Let $I=[a,b]\subset\mathbb{R}$ and $X_n=\{x_i,\ i=0,\dots,n\}\subset I$ be a set of distinct nodes (also called data points). In classical univariate interpolation we reconstruct a function $f:I\longrightarrow\mathbb{R}$ given its samples at the node set X_n, that is $F_n=\{f_i:=f(x_i),\ i=0,\dots,n\}$. Thus, let $\Pi_n=\text{span}\{1,x,\dots,x^n\}$ be the linear space of polynomials of degree less or equal than n, the interpolating polynomial $P_{n,f}\in\Pi_n$ of f at X_n can be expressed as

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

being $\boldsymbol{a}=(a_0,\dots,a_n)^\intercal$ the vector of the coefficients which is determined by solving the system

$$V\boldsymbol{a}=\boldsymbol{f},$$

where $V_{i,j}:=V_{i,j}(X_n)=x_i^j$, $i,j=0,\dots,n$, is the well-known Vandermonde matrix for the point set X_n and $\boldsymbol{f}=(f_0,\dots,f_n)^\intercal$.

Alternatively, $P_{n,f}$ can be expressed in the *Lagrange basis* $\mathscr{L}=\{l_0,\dots,l_n\}$, so that

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

where the *elementary Lagrange polynomial* can be written in determinant form as

$$l_i(x)=\frac{\det(V_i(X_n))}{\det(V(X_n))}=\prod_{\substack{0\le j\le n\\ j\ne i}}\frac{x-x_j}{x_i-x_j},$$

where $V_i(X_n)=V(x_0,\dots,x_{i-1},x,x_{i+1},\dots,x_n)$; refer to [9].

Based on papers studying mapped bases (cf. e.g. [18, 21]) and inspired by the kernel (discontinuous) framework [11, 22, 23], in [13] the authors introduced a method that changes the given interpolation problem, and in particular the node distribution, without the need of resampling the (unknown) function f. This turns out to be helpful in applications, such as rational approximation [2], imaging [10] and, and in what we will discuss in this note, i.e. numerical quadrature.

The paper is organized as follows. In Section 2, for making the paper self-contained, we review the basics of quadrature rules and of the mapped bases approach. Then, in Section 3, we study the quadrature method we propose. Numerical experiments are presented in Section 4. The last section deals with conclusions and future work.

[a] Dipartimento di Matematica “Tullio Levi-Civita”, Università di Padova, Italy; demarchi@math.unipd.it
[b] Département de mathématiques, Université de Fribourg, Switzerland; giacomo.elefante@unifr.ch
[c] Department of Mathematics DIMA, University of Genova; perracchione@dima.unige.it
[d] PNC - Padova Neuroscience Center, Università di Padova, Italy; poggiali.davide@gmail.com

2 Preliminaries

We start by recalling the main ideas of the *fake nodes* approach and then we briefly review the basics of numerical integration.

2.1 Mapped bases approach

Consider a bijective map $S : I \longrightarrow \mathbb{R}$. For any $\tilde{x} \in J := S(I)$, we may construct the polynomial $P_{n,g} : J \longrightarrow \mathbb{R}$ interpolating the set of function values F at the *fake nodes* $S(X_n) = \{S(x_i) = \tilde{x}_i\ i = 0,\dots,n\} \subset J$, for some function $g : J \longrightarrow \mathbb{R}$, so that it belongs to $C^r(\tilde{J})$, for a closed set $\tilde{J}$ so that $J \subseteq \tilde{J}$, $r \geq 0$ and

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

Hence, for $x \in I$ the interpolant $R^S_{n,f} \in \mathrm{span}\{(S(x))^i,\ i = 0,\dots,n\}$ is given by

$$R^S_{n,f}(x) := P_{n,g}(S(x)) = \sum_{i=0}^{n} a^S_i S(x)^i.$$

Trivially, $\boldsymbol{a}^S = (a^S_0,\dots,a^S_n)^\intercal$ are determined by solving the system

$$V^S\,\boldsymbol{a}^S = \boldsymbol{f},$$

where $V^S = V(S(X_n))$ and $\boldsymbol{f} = (f_0,\dots,f_n)^\intercal$.

As for classical polynomial interpolation, we are able to define the mapped Lagrange basis so that for $x \in I$ the approximant $R^S_{n,f}$ reads as follows

$$R^s_{n,f}(x) = \sum_{i=0}^{n} f(x_i) l^S_i(x),$$

where

$$l^S_i(x) := l_i(S(x)) = \frac{\det(V_i(S(X_n)))}{\det(V(S(X_n)))} = \prod_{\substack{0\leq j\leq n\\ j\neq i}} \frac{S(x) - S(x_j)}{S(x_i) - S(x_j)}.$$

We are here interested in studying such approach applied to the approximation of integrals of univariate functions. To this aim, mainly for notation convenience, we briefly recall the basics of quadrature formulae.

2.2 Numerical quadrature

Given $f : I = [a,b] \to \mathbb{R}$, we are interested in approximating

$$\mathscr{I}(f,I) := \int_I f(x)\mathrm{d}x,$$

via quadrature formulae of interpolation type. To this aim we take a set of distinct (quadrature) nodes $X_n = \{x_i,\ i = 0,\dots,n\}$ and we assume for simplicity that $x_0 = a$, $x_n = b$ and $n \geq 1$. Then, the classical quadrature formula obtained via $P_{n,f}$ assumes the form:

$$\mathscr{I}_n(f,I) := \boldsymbol{w}^\intercal \boldsymbol{f} \approx \mathscr{I}(f,I),$$

where the weights $\boldsymbol{w} = (w_0,\dots,w_n)^\intercal$ are computed by solving $V^\intercal\,\boldsymbol{w} = \boldsymbol{m}$, where $\boldsymbol{m} = (m_0,\dots,m_n)^\intercal$ is the vector of moments of the polynomial basis, i.e. $m_i = \mathscr{I}(x^i, I)$, $i = 0,\dots,n$.

We also recall that the interpolant can be written in cardinal form using the Lagrange basis. Therefore, in an equivalent way, the weights are the integrals

$$w_i = \mathscr{I}(l_i, I), \quad i = 0,\dots,n.$$

Provided that $f \in C^{n+1}(I)$, the quadrature error assumes the form [20, p. 303–304]

$$E_n(f) = \mathscr{I}(f - P_{n,f}, I) = \frac{1}{(n+1)!}\mathscr{I}\left((f^{(n+1)} \circ \xi)\,\omega_{n+1}, I\right),$$

where $\xi := \xi(x)$ is from the pointwise interpolation error and ω_{n+1} is the nodal polynomial

$$\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i).$$

Remark 1. We further point out that one can easily see that

$$|E_n(f)| \leq \max_{\eta\in I} \frac{\left|f^{(n+1)}(\eta)\right|}{(n+1)!}\mathscr{I}\left(|\omega_{n+1}|, I\right).$$

We now drive our attention towards the use of mapped bases as an alternative to standard quadrature rules.

3 Quadrature via fake nodes

In this section at first we investigate the quadrature problem via fake nodes for a generic map and then, we drive our attention towards two specific maps introduced for mitigating non-physical oscillations due to the typical instability of the interpolant at equispaced nodes or in the presence of discontinuities.

In the framework presented in [13] and briefly recalled in the previous section, using $R^S_{n,f}$ for constructing the quadrature formulae, we have that

$$\mathscr{I}(f,I) \approx \mathscr{I}^S_n(f,I) := (\boldsymbol{w}^S)^\intercal \boldsymbol{f},$$

where the weights $\boldsymbol{w}^S = (w^S_0,\ldots,w^S_n)^\intercal$ are computed by solving the system of moments

$$(V^S)^\intercal\, \boldsymbol{w}^S = \boldsymbol{m}^S,$$

with $\boldsymbol{m}^S = (m^S_0,\ldots,m^S_n)^\intercal$, and $m^S_i = \mathscr{I}(S(x)^i, I)$, $i = 0,\ldots,n$.

Equivalently, for $x \in I$, the interpolant $R^S_{n,f}$ can be viewed as a standard polynomial interpolant $P_{n,g}$ on the mapped set of nodes $J = S(I)$. Moreover, if the map S is at least $C^1(I)$, letting $t = S(x)$, for $x \in I$, we get

$$\tilde{S}(t) := \frac{\mathrm{d}S^{-1}(t)}{\mathrm{d}t} = \frac{1}{S'(S^{-1}(t))}, \tag{1}$$

that, by considering the Lagrange form of the interpolant, brings to,

$$\mathscr{I}(R^S_{n,f}, I) = \mathscr{I}\left(\sum_{i=0}^{n} f_i\, l^S_i, I\right) = \mathscr{I}\left(\sum_{i=0}^{n} f_i\,(l_i \circ S), I\right) = \mathscr{I}\left(\sum_{i=0}^{n} f_i l_i \tilde{S}, J\right) = \mathscr{I}(P_{n,g}\tilde{S}, J). \tag{2}$$

Thus, the S-mapped weights become

$$w^S_i = \mathscr{I}(l^s_i, I) = \mathscr{I}(l_i\tilde{S}, J). \tag{3}$$

For the quadrature error we can easily prove the following.

Proposition 3.1. Let S be such that g is at least $C^{n+1}(\tilde{J})$. Then,

$$E^S_n(f) = \frac{1}{(n+1)!}\mathscr{I}((g^{(n+1)} \circ \xi \circ S)(\omega^S_{n+1} \circ S), I),$$

being

$$\omega^S_{n+1}(t) := \prod_{i=0}^{n}(t - t_i),$$

the nodal polynomial at the fake nodes and where $\xi := \xi(t)$ is from the pointwise interpolation error. We have also that

$$|E^S_n(f)| \le \frac{\max_{\eta\in J}|g^{(n+1)}(\eta)|}{(n+1)!}\mathscr{I}(|\omega_{n+1}\tilde{S}|, J).$$

Proof. Letting $t = S(x)$, we have that

$$E^S_n(f) = \mathscr{I}(f - R^S_{n,f}, I) = \mathscr{I}((g - P_{n,g})\tilde{S}, J).$$

Then, by using the pointwise interpolation error, we obtain

$$E^S_n(f) = \frac{1}{(n+1)!}\mathscr{I}((g^{(n+1)} \circ \xi)\,\omega^S_{n+1}\,\tilde{S}, J) = \frac{1}{(n+1)!}\mathscr{I}((g^{(n+1)} \circ \xi \circ S)(\omega^S_{n+1} \circ S), I),$$

which in particular implies that

$$|E^S_n(f)| \le \frac{1}{(n+1)!}\max_{\eta\in J}|g^{(n+1)}(\eta)|\mathscr{I}(|\omega_{n+1}\tilde{S}|, J).$$

■

Remark 2. If S is so that g is a polynomial of degree n, then the quadrature has exactness n. Moreover, we observe that, having defined the error as in Proposition 3.1, all considerations made in Remark 1, hold for the mapped basis approach too.

As in [13], we now focus on the computation of the weights for two specific maps, called S-Gibbs and S-Runge. We will show that while the S-Gibbs leads to a robust and novel method for mitigating the Gibbs effect, the use of the S-Runge map coincides with a truly popular composite algorithm in the case of equispaced nodes.

3.1 S-Gibbs

Letting $\xi_0 = a$ and $\xi_d = b$, suppose that the function f shows jump discontinuities, whose positions and magnitudes are encoded in the set

$$D_d := \{(\xi_i, \alpha_i)\ \xi_i < \xi_{i+1},\ \alpha_0 = \alpha_d = 0,\ \alpha_i := |f(\xi_i^+) - f(\xi_i^-)|,\ i = 1, \ldots, d-1\}.$$

To construct a reliable approximation $R^S_{n,f}$ of f, as in [13], we introduce the so-called shift *parameter* $k \in \mathbb{R}^+$. Therefore, letting $\beta_i := k\alpha_i,\ i = 0, \ldots, d$, and $B_i = \sum_{j=0}^{i} \beta_j$, we define S as follows:

$$S(x) = x + B_i, \quad \text{for } x \in [\xi_i, \xi_{i+1}[,\ 0 \le i \le d-1.$$

Finally, letting $I_i = [\xi_i^+, \xi_{i+1}^-[$ and $J_i = S(I_i)$, we approximate the integral as

$$w_j^S = \mathscr{I}(l_j^s, I) = \sum_{i=0}^{d-1} \mathscr{I}(l_j \tilde{S}, J_i) = \sum_{i=0}^{d-1} \mathscr{I}(l_j, J_i),$$

where the last equality follows from (2) and the fact that $\tilde{S}$ is bijective and constantly equal to 1 on each sub-interval and where l_i is the Lagrange polynomials at the fake nodes.

The interpolant constructed via the S-Gibbs map naturally encodes the discontinuities in the mapped bases. For that reason, we expect to achieve accurate approximations for discontinuous functions.

3.2 S-Runge

When studying the S-Runge map, we consider two data distributions: equispaced nodes and scattered data. Indeed, while any extension to scattered points turns out to be trivial for the for the S-Gibbs, the S-Runge map requires some investigation.

3.2.1 Equispaced nodes

In order to prevent the appearance of the Runge phenomenon, the natural way is to consider the set of (increasingly ordered) *Chebyshev-Lobatto* (CL) points on $K = [-1, 1]$ (also known as *Chebyshev extremal points*)

$$C_n = \left\{ c_i = -\cos\left(\frac{i\pi}{n}\right),\ i = 0, \ldots, n \right\}.$$

Their Lebesgue constant grows logarithmically with respect to n (cf. e.g. [5]). Then, using the map $S : I \longrightarrow K$ given by [13]

$$S(x) = -\cos\left(\frac{x-a}{b-a}\pi\right), \tag{4}$$

we can prove the following result for the quadrature rule based on the *fake nodes* approach. While the result is theoretically interesting, since it tells us how to compute the weights, numerically it shows that the proposed method simply coincides, quite surprisingly, with the composite trapezoidal rule.

Theorem 3.1. *Given $X_n = \{x_k = a + kh,\ h = (b-a)/n,\ k = 0, \ldots, n\} \subseteq I$ and the map (4), the S-mapped quadrature weights are then given by*

$$w_k^S = \begin{cases} \dfrac{h}{2}, & \textit{for } k \in \{0, n\}, \\ h, & \textit{for } k = 1, \ldots, n-1. \end{cases}$$

Proof. Let $\nu : K \longrightarrow \mathbb{R}$

$$\nu(x) = (1 - x^2)^{-1/2}$$

be the Chebyshev weight function and c_i, $i = 0, \ldots, n$, the CL nodes, then

$$\mathscr{I}(f\nu, K) \approx \sum_{i=0}^{n} f(c_i) w_i^C = \sum_{i=0}^{n} f(c_i) \frac{\pi}{z_i n}, \tag{5}$$

where w_i^C are the classical Chebyshev weights (see e.g. [7]), $z_i = 2$ if $i \in \{0, n\}$ and 1 otherwise.

Let us now take on I the set of equispaced nodes X_n and the S-map (4). Thus, $S(X_n) = C_n$. Letting $t = S(x)$, then $l_i(S(x)) = l_i(t)$, $i = 0, \ldots, n$, where $l_i(t)$ is the Lagrange basis at the CL nodes in K. Therefore, taking into account (1), (2) and the fact that

$$\mathrm{d}x = \tilde{S}(t)\mathrm{d}t = \nu(t)\frac{b-a}{\pi}\mathrm{d}t,$$

we have

$$w_i^S = \frac{(b-a)}{\pi} \mathscr{I}(l_i \nu, K).$$

By observing that the quadrature rule (5) is exact for polynomials of degree less or equal to n, we obtain

$$w_i^S = \frac{(b-a)}{\pi} \sum_{j=0}^{n} l_i(c_j) w_j^C = \frac{(b-a)}{\pi} \sum_{j=0}^{n} \delta_{\mathrm{i,j}} \frac{\pi}{z_j n} = \frac{b-a}{z_i n}.$$

■

Remark 3. The weights for the *fake nodes* approach with CL nodes coincide with the trapezoidal composite rule and up to a constant with the Gaussian quadrature formulae for approximating the integrals of weighted Chebyshev functions.

The procedure here outlined can be extended to any set of scattered data. To achieve this, we only need to replace the analytical map (4) with an interpolating function that maps scattered nodes into the CL points. For instance, we investigate the use of Radial Basis Functions (RBFs) and linear piecewise interpolant.

3.2.2 Scattered nodes

Let us take a set of scattered nodes $X_n \subset [a,b] \subset \mathbb{R}$

$$X_n = \{x_0 = a < x_1 < \cdots < x_{n-1} < x_n = b\}.$$

The aim is to define a function S that maps on the Chebyshev nodes. In principle, any interpolant S so that $S(x_i) = c_i$, $i = 0, \ldots, n$, could be used. Indeed, once an interpolant S is defined, then the weights can be computed as in (3). As first example, we take S as a RBF interpolant (see e.g. [15]), i.e.

$$S(x) = \sum_{k=0}^{n} \gamma_k \phi(|x - x_k|), \quad x \in I,$$

where ϕ is the so-called RBF. Here we will use the Matérn C^2 function given by

$$\phi(r) = \mathrm{e}^{-\varepsilon r}(1 + \varepsilon r),$$

where ε is the so-called shape parameter and r denotes the Eucledian distance. The coefficients are characterized by solving the linear system of equations [15]

$$K\boldsymbol{\gamma} = \boldsymbol{c},$$

where $\boldsymbol{\gamma} = (\gamma_0, \ldots, \gamma_n)^\intercal$, $\boldsymbol{c} = (c_0, \ldots, c_n)^\intercal$ and $K_{ik} = \phi(|x_i - x_k|)$, $i, k = 0, \ldots, n$. Since K is a symmetric and strictly positive definite kernel, this system has exactly one solution. Once the coefficients are computed, the weights are determined as in (3).

As a second example, we propose to use the linear piecewise spline interpolant. On each interval $I_k = [x_k, x_{k+1}]$, the map is defined as

$$S_k(x) = m_k(x - x_k) + q_k,$$

where

$$m_k = \frac{c_{k+1} - c_k}{x_{k+1} - x_k},$$

and $q_k = c_k$. Therefore, for $x \in I$ the map $S : I \longrightarrow \mathbb{R}$ assumes the form

$$S(x) = \sum_{k=0}^{n-1} \chi_{I_k}(x) S_k(x),$$

$\chi_{I_k}(x) = 1$ if $x \in I_k$ and 0 elsewhere.

Then we can compute the weights for the quadrature by splitting the integral on each sub-interval $I_k = [x_k, x_{k+1}]$. Indeed, taking into account the change of variable in (1),

$$w_i^s = \mathscr{I}\left(l_i^s, I\right) = \mathscr{I}(l_i \tilde{S}, J) = \sum_{k=0}^{n-1} \frac{1}{m_k} \mathscr{I}\left(l_i \chi_{J_k}, J_k\right) = \sum_{k=0}^{n-1} \frac{1}{m_k} \mathscr{I}\left(l_i, J_k\right),$$

with $J_k = S(I_k) = [c_k, c_{k+1}]$ and l_i the Lagrange polynomials at the Chebyshev nodes.

We now focus on the numerical experiments that are devoted to investigate the possible benefits coming from the use of the fake nodes in the context of quadrature rules.

4 Numerical experiments

The experiments are carried out in Python 3.7 using Numpy 1.15. The Python code is available at `https://github.com/pog87/FakeQuadrature`.

We consider the following test functions (a discontinuous and a Runge function, respectively)

$$f_1(x) = \begin{cases} \sin(x), & \text{for } x \le 0, \\ \log(x^4 + 4) + 7, & \text{for } x > 0, \end{cases} \quad \text{and} \quad f_2(x) = \frac{1}{4x^2 + 1}.$$

We compute the integral of the functions over the interval $I = [-2, 2]$. In Figure 1 the absolute error between the true value of the integral and its approximation is displayed for several polynomials degrees. In Figure 1 (left) we compare the computation of the integral with equispaced points, CL nodes and fake nodes that make use of the S-Gibbs map. For such map, we fix the shift parameter $k = 1.5$. We can note that encoding the discontinuity directly into the basis via the S-Gibbs map leads to a truly effective tool which outperforms the other considered methods.

In Figure 1 (right) we sample the function f_2 at the Halton points, which are a set of low-discrepancy points [14], and we compare the behaviour of the classical computation of the weights and the one obtained via the fake nodes. Specifically, the computation of the quadrature weights for the mapped bases approach is obtained with two different maps: linear piecewise spline and RBF interpolant. For the latter we take the Matérn C^2 kernel and we fix the shape parameter as $\varepsilon = 0.5$. We note that the *fake* quadrature formulae outperform the computation of the integral on Halton nodes.

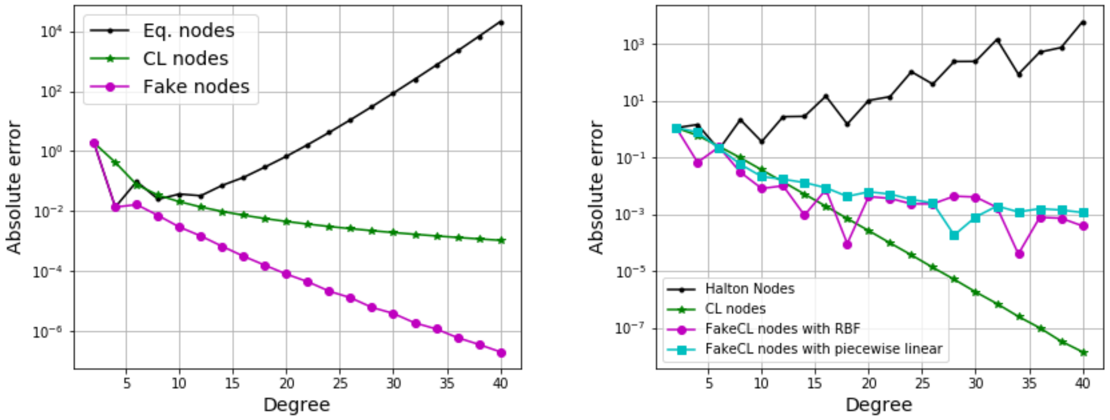

Figure 1: Approximation of the integral over $I = [-2, 2]$ of the functions f_1 and f_2, respectively. Increasing the node numbers, the Fake node approach shows to be competitive in both cases.

5 Conclusions

We investigated the application of the mapped bases approach, without resampling, to quadrature. Numerical evidence shows that, by using this technique, we can reduce the Runge and Gibbs phenomena. In particular, the S-Gibbs map outperforms the other considered methods. Work in progress consists in extending such a simple but effective procedure to higher dimensions.

Acknowledgements

This research has been accomplished within Rete ITaliana di Approssimazione (RITA), partially funded by GNCS-INδAM, the NATIRESCO project BIRD181249 and through the European Union's Horizon 2020 research and innovation programme ERA-PLANET, Grant Agreement No. 689443, via the GEOEssential project.

References

[1] B. Adcock, R.B. Platte, *A mapped polynomial method for high-accuracy approximations on arbitrary grids*, SIAM J. Numer. Anal. **54** (2016), 2256–2281.

[2] J.P. Berrut, S. De Marchi, G. Elefante, F. Marchetti, *Treating the Gibbs phenomenon in barycentric rational interpolation via the S-Gibbs algorithm*, Appl. Math. Letters 103, (2020), 106196

[3] J.P. Berrut, G. Elefante, *A periodic map for linear barycentric rational trigonometric interpolation*, Appl. Math. Comput. 371 (2020), 124924.

[4] J.P. Boyd, F. Xu, *Divergence (Runge Phenomenon) for least-squares polynomial approximation on an equispaced grid and Mock-Chebyshev subset interpolation*, Appl. Math. and Comput. **210** (2009), 158–168.

[5] L. Brutman, *On the Lebesgue function for polynomial interpolation*, SIAM J. Numer. Anal. **15** (1978), 694–704.

[6] L. Brutman, *Lebesgue functions for polynomial interpolation – a survey*, Ann. Numer. Math. **4** (1997), 111–127.

[7] P.L. Chebyshev, *Sur les quadratures*, J. Math. Pures Appl. **19**:2 (1874), 19–34. http://sites.mathdoc.fr/JMPA/PDF/JMPA_1874_2_19_A2_0.pdf

[8] P.J. Davis, *Interpolation and Approximation*, Dover Publications, 1975.

[9] S. De Marchi, *Polynomials arising in factoring generalized Vandermonde determinants: an algorithm for computing their coefficients*, Math. Comput. Modelling **34** (2001), 271–281.

[10] S. De Marchi, W. Erb, E. Francomano, F. Marchetti, E. Perracchione, D. Poggiali, *Fake nodes approximation for magnetic particle imaging*, in: IEEE, Proc. of Melecon 2020.

[11] S. De Marchi, W. Erb, F. Marchetti, E. Perracchione, M. Rossini, *Shape-Driven Interpolation with Discontinuous Kernels: Error Analysis, Edge Extraction and Applications in Magnetic Particle Imaging*, SIAM J. Sci. Comput. **42** (2020), B472–B491.

[12] S. De Marchi, F. Marchetti, E. Perracchione, *Jumping with Variably Scaled Discontinuos Kernels (VSDKs)*, BIT Numer. Math., **60** (2020), 441–463.

[13] S. De Marchi, F. Marchetti, E. Perracchione, D. Poggiali, *Polynomial interpolation via mapped bases without resampling*, J. Comput. Appl. Math., **364** (2020), 112347–12.

[14] J. Dick, F. Pillichshammer, *Digital Nets and Sequences: Discrepancy Theory and Quasi–Monte Carlo Integration*, Cambridge University Press, 2010.

[15] G.E. Fasshauer, *Meshfree Approximations Methods with* MATLAB, World Scientific, Singapore, 2007.

[16] B. Fornberg, N. Flyer, *The Gibbs Phenomenon in Various Representations and Applications*, chapter The Gibbs phenomenon for radial basis functions. Sampling Publishing, Potsdam, NY, 2008.

[17] D. Gottlieb, C.W. Shu, *On the Gibbs phenomenon and its resolution*, SIAM Review **39** (1997), pp. 644–668.

[18] N. HALE, L.N. TREFETHEN, *New quadrature formulas from conformal maps*, SIAM J. Numer. Anal. **46** (2008), 930–948.

[19] P. HOLOBORODKO, *Stable Newton-Cotes formulas*, http://www.holoborodko.com/pavel/numerical-methods/numerical-integration/stable-newton-cotes-formulas-open-type/

[20] E. ISAACSON, H.B. KELLER, *Analysis of Numerical Methods*, Dover Publications, 1994.

[21] D. KOSLOFF, H. TAL-EZER, *A modified Chebyshev pseudospectral method with an $O(N^{-1})$ time step restriction*, J. Comput. Phys. **104** (1993), 457–469.

[22] L. ROMANI, M. ROSSINI, D. SCHENONE, *Edge detection methods based on RBF interpolation*, J. Comput. Appl. Math. **349** (2019), 532–547.

[23] M. ROSSINI, *Interpolating functions with gradient discontinuities via variably scaled kernels*, Dolom. Res. Notes Approx. **11** (2018), 3–14.

[24] C. RUNGE, *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*, Zeit. Math. Phys. **46** (1901), 224–243.

[25] A.H. TURETSKII, *The bounding of polynomials prescribed at equally distributed points*, Proc. Pedag. Inst. Vitebsk Vol. 3 (1940), 117–127 (in Russian).