Volume 1, Issue 2

Journal of Approximation Software

06 September 2024

SIRIO@UniTO

Journal of Approximation Software

Volume 1, Issue 2
06 September 2024

Special issue dedicated to Len Bos on the occasion of his 70th birthday

Preface

This second issue of the newborn Journal of Approximation Software is dedicated to the 70th birthday and the retirement of Len Bos. Though software has not been the main scientific interest of Len, he has worked on many codes and has inspired directly or indirectly a lot of numerical software in the framework of polynomial, rational, trigonometric and RBF approximation.

The present contributions range from approximation on the sphere to complex polynomial approximation, multinode Shepard approximation by scattered data, and to exponential integrators for optimal control problems. In the spirit of the journal, the papers describe the theory behind the methods and use of the codes, which are freely available on Github.

The Editors-in-Chief of JAS

Len walking towards the Sella Group in the Dolomites.

Len giving an unusual talk, at a Dolomites Workshop dinner.

Journal Managers

J. Approx. Softw., Vol. 1, No. 2 (2024) 1

Journal of Approximation Software

The applicability of equal area partitions of the unit sphere

P. Leopardi [1,2,⋆]

[1] *ACCESS-NRI*
[2] *The Australian National University*

Received: 23/04/2024 – Published: 06/09/2024
Communicated by: M. Vianello

Abstract

This paper addresses the idea of the applicability of mathematics, using, as a case study, a construction and software package that partition the unit sphere into regions of equal area. The paper assesses the applicability of this construction and software by examining citing works, including papers, dissertations and software.

Keywords: mathematics, applicability, applications, sphere, partitions, approximation

1 Introduction

The unreasonable effectiveness of mathematics? A well known paper by Wigner [139] claims that there is something mysterious or even miraculous about the appropriateness of mathematics as a language for describing nature ...

> The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift which we neither understand nor deserve. We should be grateful for it and hope that it will remain valid in future research and that it will extend, for better or worse, to our pleasure even though perhaps also to our bafflement, to wide branches of learning. [139]

Hamming [54] make some attempts at explanation, but comes to a similar conclusion to Wigner ...

> Some partial explanations ...
>
> 1. We see what we look for.

⋆ Corresponding author: paul.leopardi@anu.edu.au

2. We select the kind of mathematics to use.
3. Science in fact answers comparatively few problems.
4. The evolution of man provided the model.

> ...From all this I am forced to conclude both that mathematics is unreasonably effective and that all of the explanations I have given when added together simply are not enough to explain what I set out to account for. ...The logical side of the nature of the universe requires further exploration. [54]

On the other hand, Arnold [7] and Borovik [15] quote Gelfand's assertion that mathematics is unreasonably *in*effective in biology.

Arnold [7]:

> Here we can add a remark by I.M. Gel'fand: there exists yet another phenomenon which is comparable in its inconceivability with the inconceivable effectiveness of mathematics in physics noted by Wigner – this is the equally inconceivable ineffectiveness of mathematics in biology.

Borovik: [15]:

> This paper is an attempt to answer the question
>
> > Should we accept Israel Gelfand's assessment of the role of mathematics in biology?
>
> And my answer is
>
> > Yes, we should, for the time being: mathematics is still too weak for playing in biology the role it ought to play.

So what makes mathematics effective or ineffective in applications, and is the effectiveness or ineffectiveness reasonable or unreasonable?

A more pragmatic approach. The practice of applied mathematics usually takes a more pragmatic approach, especially when dealing with models and approximations. In scientific modelling, it is often stated that "All models are wrong but some are useful" [20], meaning that a completely faithful model may be unattainable, but it may be possible to build a parsimonious model that reflects the key phenomena or most important aspects of the system being modelled.

Much can and has been said about the construction of models of systems that are both fit for purpose and mathematically tractable [89]. Modelling often involves approximation, in the sense of neglecting some aspects of the systems, and idealization, that is making strictly incorrect assumptions that still preserve the important aspects to be understood [90, 102, 116]. The processes of approximation and idealization may then result in a model that can be described by known or at least constructable mathematics. The final step for a predictive model would then be to ensure that the mathematical formulation is tractable, in the sense that it results in a reasonable trade-off between computational effort and accuracy [1, 10, 115, 130].

Models also often involve approximations in the mathematical sense of the word. The approximation of functions from noisy and incomplete data [9], and the approximate solution of underdetermined systems of equations [63], including the solution of differential or integral equations with noisy and incomplete initial data, has long been a subject of study in statistics, applied mathematics, and machine learning. The subject of approximation theory deals

with the best approximation within a function space [131], and the theory of information-based complexity explicitly deals with the inherent trade-off between the cost of function evaluation versus the error in approximation [130, 143]. It is therefore usually the case that a new approximation method, algorithm or construction is accompanied by an analysis of its applicability to known abstract problems, often in terms of its asymptotic cost versus rate of convergence with respect to approximation error within a known setting (e.g. [25]).

A case study. This paper examines the applicability of a geometric construction: an equal area partition of a unit higher-dimensional sphere, and its associated distribution of points. The construction is described in a 2006 paper published in Electronic Transactions on Numerical Analysis [74], and is analyzed in more detail in a PhD thesis of 2007 [75], a paper of 2009 [76] and follow-up papers [77, 81]. As at 20 August 2024, the 2006 paper [74] has 346 citations listed in Google Scholar, 224 citations in Scopus, 179 in Web of Science, and 43 in MathSciNet.

Citations appear in Geophysical Journal International, Global Change Biology, IEEE Transactions on Audio Speech and Language Processing, Journal of Approximation Theory, Journal of the Atmospheric Sciences, Journal of Computational Chemistry, Journal of Differential Equations, Mathematics of Computation, Radio Science, RNA Journal, and elsewhere.

Note: some of the papers described below are accompanied by abbreviated 20 August 2024 citation counts of the form (G: g, S: s, W: w, M: m), for the Google Scholar, Scopus, Web of Science, and MathSciNet counts respectively. For example, the abbreviated counts for the 2006 paper [74] are (G: 346, S: 224, W: 179, M: 43).

The citations of the 2006 paper, the 2007 thesis, and the follow-up papers are generally of three types:

1. Application of the constructions to specific problems;
2. Evaluation of the constructions described by the paper, including comparisons with other constructions; and
3. Passing mention of the paper, sometimes with a short description.

This paper is mostly concerned with the first two types of citations.

2 Preliminaries

2.1 Some related problems

The problem of finding an equally distributed set of N points on a circle is solved easily: just use points arranged uniformly at an angle of $\frac{N}{2\pi}$. In contrast, on a unit sphere $\mathbb{S}^d \subset \mathbb{R}^{d+1}$ with $d > 1$, not only is the problem harder to solve, it is harder to pose precisely. There are a number of related problems, each of which gives rise to a different sense of equal distribution [14, 119]. These problems are often stated in terms of sequences of *spherical codes*, where each spherical code is a finite set of N points on the unit sphere $\mathbb{S}^d$, and we are often interested in some asymptotic value related to each sequence as N goes to infinity.

Asymptotic equidistribution in measure and related quadrature problems. Find a sequence of spherical codes that converges to the uniform distribution in the sense of convergence in measure (Figure 1). Such convergence is usually defined in terms of a discrepancy such as the spherical cap discrepancy [23, 46, 142]. If the spherical cap discrepancy of the spherical

codes in the sequence converges to zero, then the sequence is said to be *asymptotically uniformly distributed* [119], or *asymptotically equidistributed* [28, 92], or *weak-star convergent* [75, Definition 2.11.3].

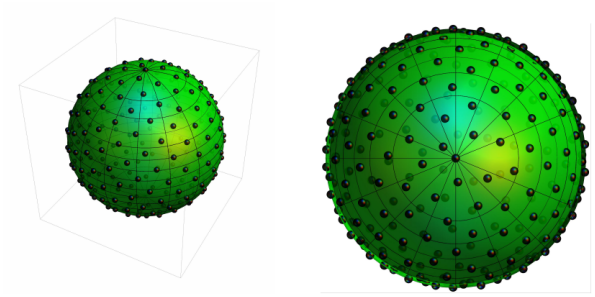


Figure 1: Brauchart, "Spherical Fibonacci lattices," [24].

Interpolation and related function approximation problems. Find a sequence of spherical codes such that a function approximation within a certain function space converges at a certain rate. Examples are Lagrange polynomial interpolation within continuous function spaces [6, 16, 141] and least squares approximation within L_∞ [61, 62, 123, 129] (Figure 2).

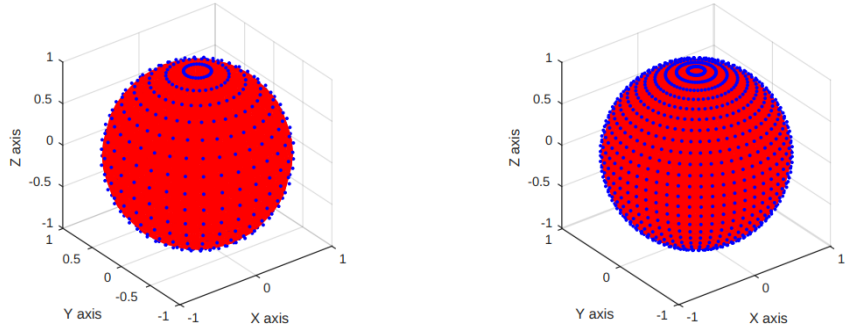


Figure 2: Themistoclakis and Van Barel, "Examples of the tensor product Gauss–Legendre quadrature nodes related to degrees of precision 31 and 51, i.e., having $N = 512$ (left) and $N = 1352$ (right) points," [129].

The Thomson and related energy minimization problems. Minimize the energy of N equally charged particles on a sphere, with respect to some potential (Figure 3).

The Tammes problem and packing of spherical caps. Given a fixed radius, how many non-overlapping spherical caps with that radius can be placed onto a unit sphere (Figure 4)? This radius is called the *packing radius* of the spherical code formed by the centres of the caps.

The covering problem. Given a fixed radius, how few overlapping spherical caps with that radius are needed to cover a unit sphere (Figure 5)? This radius is called the *covering radius* or *mesh norm* of the spherical code formed by the centres of the caps.

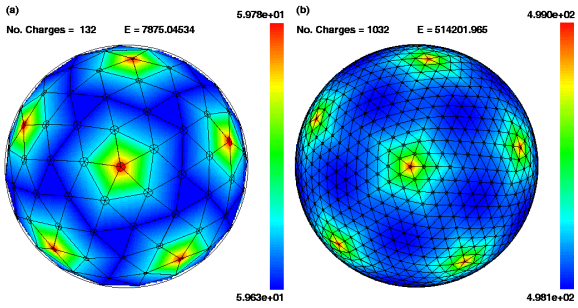


Figure 3: Atschuler et al. "Lattice configurations for 132 (a) and 1032 (b) charges," [5].

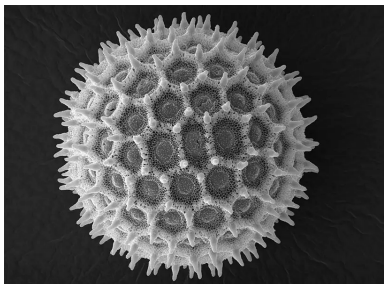


Figure 4: Dartmouth College Electron Microscope Facility, "A grain of pollen from Morning Glory flowers," [47].

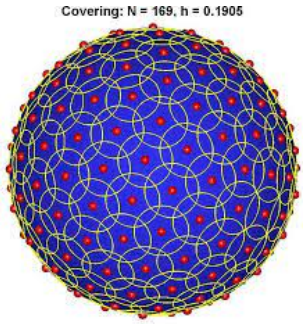


Figure 5: Saff and Womersley, "Covering of a sphere with 169 equal spherical caps," [118].

2.2 Some history

The history of constructions aimed at solving the problems posed in Section 2.1 is quite involved. See also the 2019 book by Borodachov et al. [14, Chapters 6 and 7].

Equidistribution without separation. Many constructions for $\mathbb{S}^2$ yield an asymptotic equidistribution, e.g. Hammersley, Halton, (t,s) etc. sequences mapped to the sphere [142] (Figure 6).

Separation without equidistribution. Hamkins [48] and Hamkins and Zeger [49] constructed $\mathbb{S}^d$ codes with asymptotically optimal packing density (Figure 7).

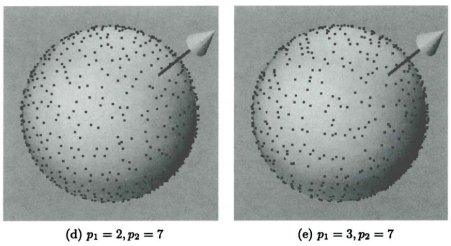


Figure 6: Wong and colleagues "Halton points with different bases on the sphere ($n = 1000$)," [142].

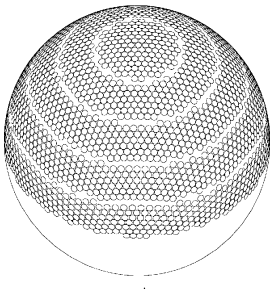


Figure 7: Hamkins and Zeger, "The wrapped spherical code $\mathscr{C}_W^{\Lambda_2}$ $(3;0:05)$," [49].

Equal area partitions. Alexander [3] asserts the existence of a diameter bounded set of equal area partitions of $\mathbb{S}^2$, sketching a construction based on the cubed sphere.

Stolarsky [126], Beck and Chen [11] and Bourgain and Lindenstrauss [17] each go on to assert the existence of a diameter bounded set of equal area partitions of $\mathbb{S}^d$ without giving an explicit construction.

Feige and Schechtman [41] describe a construction as part of an argument about the optimality of a solution of the Max-Cut problem in graph theory that can be modified into a construction of a diameter bounded set of equal area partitions of $\mathbb{S}^d$ (Figure 8).

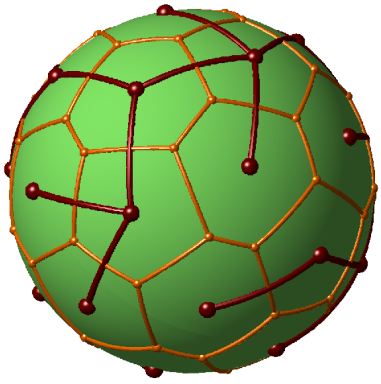


Figure 8: Leopardi. "Step 5 of the Feige-Schechtman construction" [41, 75].

The EQ(d,N) recursive zonal partition of the sphere $\mathbb{S}^d$ into N regions of equal area described in [74] and analyzed in [75, 76] is based on Zhou's 1994 construction for $\mathbb{S}^2$ [113, 148] (Figure 9) as modified by Saff, and Sloan's sketch of a partition of $\mathbb{S}^3$ [122] (Figure 10).

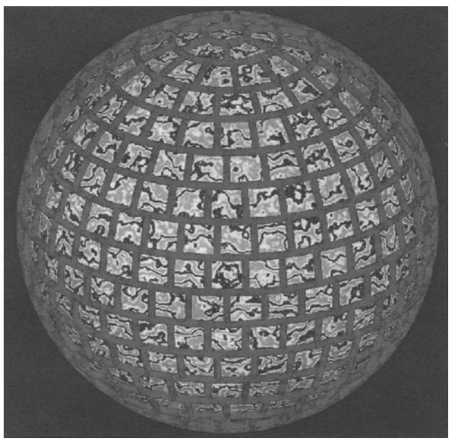


Figure 3. Partition of the sphere into 400 equal-area parts with diameters $\leq 7/\sqrt{400}$

Figure 9: Saff and Kuijlaars. "Partition of the sphere into 400 equal-area parts with diameters $\leq 7/\sqrt{400}$" [119].

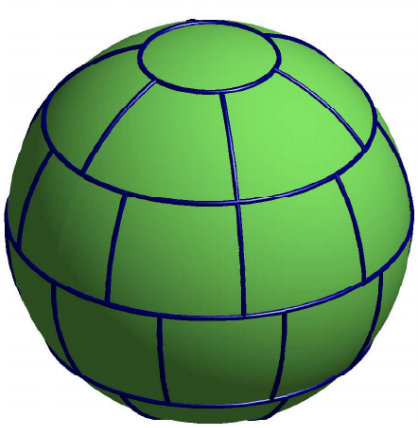


Figure 10: "Partition EQ$(2,33)$" [74].

The 2006 paper [74] describes the construction of the EQ(d,N) partition. The paper also provides estimates and numerical examples of the maximum diameter of each region in each partition of $\mathbb{S}^d$ into N regions for $N \leq 100\,000$ for $d = 2,3,4$, and also for $N = 2^k$ for $k = 1 \ldots 10$ and $d = 1 \ldots 8$. The maximum diameter is a good estimate for twice the covering radius.

The partition algorithm. The recursive zonal equal area partition algorithm is recursive in dimension d. For $d > 1$ it uses the idea of a "collar" – an annulus on the sphere arranged symmetrically about the North-South polar axis.

The 2006 paper [74] provides a detailed description of the partition algorithm EQ(d,N), but a brief pseudocode description is

if $N = 1$ **then**

There is a single region which is the whole sphere;

else if $d = 1$ **then**

Divide the circle into N equal segments;

else

Divide the sphere into zones, each the same area as an integer number of regions:

1. Determine the colatitudes of polar caps,
2. Determine an ideal collar angle,
3. Determine an ideal number of collars,
4. Determine the actual number of collars,
5. Create a list of the ideal number of regions in each collar,
6. Create a list of the actual number of regions in each collar,
7. Create a list of colatitudes of each zone;

Partition each spherical collar into regions of equal area, using the recursive zonal equal area partition algorithm for dimension $d-1$;

endif.

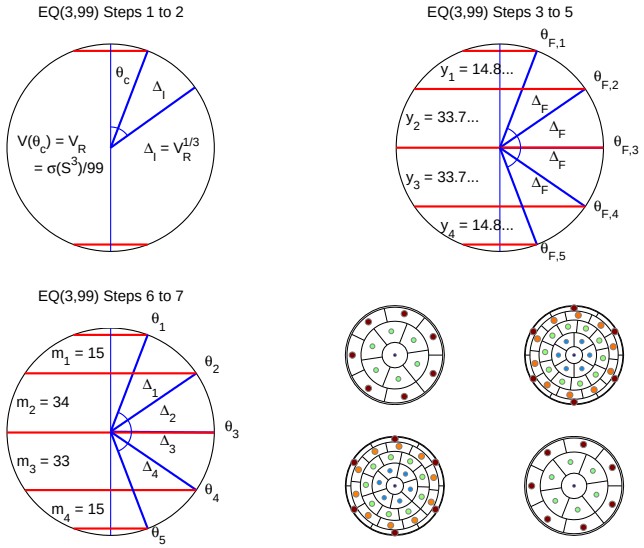


Figure 11: Partition algorithm for EQ$(3,99)$

Figure 11 is an illustration of the algorithm for EQ$(3,99)$, with step numbers corresponding to the step numbers in the pseudocode.

Spherical codes from equal area partitions. The 2007 thesis [75] describes the partition in more detail, describes the spherical codes EQP(d,N) consisting of a central point of each region of EQ(d,N), and proves that the sequences of these codes are asymptotically equidistributed for each d [75, Theorem 5.4.1] (Figure 12). Despite being unpublished, as at 20 August 2024 the thesis has 78 citations on Google Scholar.

Chapters 3 to 5 of the 2007 thesis also include the following statements, estimates and numerical examples:

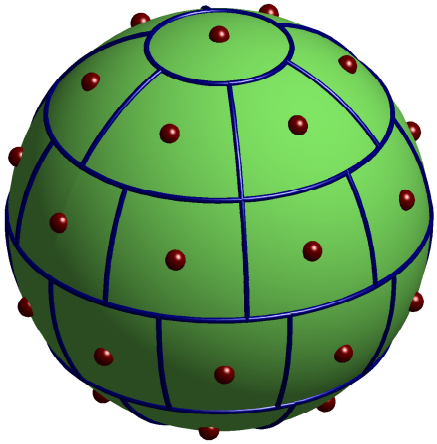


Figure 12: "EQ code EQP$(2,33)$, showing the partition EQ$(2,33)$" [75].

Chapter 3 contains estimates of the maximum diameter of each region, as per [74], with proofs. Section 3.10 lists numerical results on the maximum diameter of regions, as per [74].

Section 4.2 includes an explanation of why the spherical codes are *not good* for Lagrange polynomial interpolation, in terms of the condition number of the Gram matrix corresponding to each polynomial degree (see also [141]). The section remarks that the Gram matrix is often singular to machine precision, and also includes a statement and proof of the minimum polynomial degree for which Lagrange interpolation must fail. Section 4.3 estimates the minimum distance between code points, which is also a good estimate of twice the packing radius. Numerical examples are given for $d = 2,3,4$ and N up to 20 000. Section 4.4 gives estimates of the packing density, which is also related to the packing radius. This section also includes numerical examples for $d = 2,3,4$ and N up to 20 000.

Section 5.4 includes estimates of the spherical cap discrepancy and the Riesz energy, with numerical examples of the Riesz $d-1$ energy for $d = 2,3,4$ and N up to 20 000. Here the Riesz s-energy of a finite set $\mathscr{X} \subset \mathbb{S}^d$ of size N is defined as

$$\mathrm{E}_s(\mathscr{X}) = \frac{1}{N^2} \sum_{x \in \mathscr{X}} \sum_{\substack{y \in \mathscr{X} \\ y \neq x}} \|x-y\|^{-s},$$

using the usual Euclidean norm on $\mathbb{R}^{d+1}$.

Matlab code. The Recursive Zonal Equal Area Sphere Partitioning Toolbox [73, 79, 80] is a Matlab toolbox that was released in 2005 to accompany the paper [74] and PhD thesis [75]. The earlier history of the code, including the original Maple prototype, can be seen in the `CHANGELOG` file [80]. The following remarks refer to the situation at 20 August 2024.

Google Scholar lists four citations to the toolbox, excluding self-citations [31, 30, 112, 133]. The SourceForge URL of the toolbox [73] is mentioned in 17 other theses and papers indexed by Google Scholar [39, 55, 60, 69, 87, 88, 98, 109, 110, 112, 120, 124, 128, 132, 145, 144, 146], excluding self-citations. Of these theses and papers only six [69, 109, 110, 112, 124, 128] contain an attributed citation to the toolbox in their References section. Interestingly, one thesis [144] and two papers [145, 146] mistakenly call the toolbox "EASP" and do not cite the author.

The GitHub project for the toolbox [80] has 8 forks in GitHub, and is mentioned in one other paper indexed by Google Scholar [120]. Code from the toolbox is also included in at least 10 other GitHub projects [42, 59, 64, 68, 70, 86, 101, 105, 134, 135]. Unfortunately, SourceForge does not support code searches across its repositories, so the number of SourceForge projects that include code from the toolbox remains unknown. The same is true for GitLab public repository hosting.

The toolbox is also mentioned in the documentation for FERUM [18] but without attribution. The Matlab source code for FERUM 4.1 [19] contains a subset of the toolbox code. There is also a copy of the toolbox code at the Lamont-Doherty Earth Observatory `clifford.ldeo.columbia.edu` web site [125].

2.3 Follow-up papers and generalizations

The 2009 paper [76], based on the 2007 thesis [75], proves diameter bounds for both the EQ(d,N) sphere partition described in the 2006 paper [74], and a modified version of the construction of Feige and Schechtman as described in the thesis. Citations: (G: 34, S: 0, W: 13, M: 11).

A 2013 paper [77], following the arguments in Chapter 5 of the 2007 thesis [75], shows that a sequence of spherical codes with a well behaved upper bound on discrepancy and a well behaved lower bound on separation, such as the sequence of EQP(d,N) codes, satisfies an upper bound on the Riesz s-energy. Citations: (G: 21, S: 13, W: 12, M: 10).

A second 2013 paper [78] generalizes the paper [77] in the sense that it proves that, for a smooth compact connected d-dimensional Riemannian manifold M, if $0 \leq s \leq d$ then an asymptotically equidistributed sequence of finite subsets of M that is also well-separated yields a sequence of Riesz s-energies that converges to the energy double integral. In this case, the Riesz s-energy is defined using the geodesic distance on M. Citations: (G: 3, S: 0, W: 0, M: 0).

A 2017 joint paper with Gigante [45] generalizes the partition results of [75, 76] by combining the Feige and Schechtman construction with David's and Christ's dyadic cubes to yield a partition algorithm for connected Ahlfors regular metric measure spaces of finite measure. Citations: (G: 33, S: 19, W: 18, M: 18).

A second 2017 joint paper with Sommariva and Vianello [81] proves that good covering point configurations on the 2-sphere are optimal polynomial meshes, and extracts Caratheodory-Tchakaloff submeshes for compressed least squares fitting. This implies that the point sets generated by the construction of the 2006 paper [74] are optimal polynomial meshes. The paper also provides numerical examples where submeshes based on these point sets are used to construct positive weight quadrature rules. Citations: (G: 6, S: 4, W: 5, M: 1).

3 Evaluations and improvements

3.1 Evaluations

Many of papers citing the 2006 paper [74] and its related papers conduct one of two types of evaluation:

1. They evaluate methods that use the constructions described in [74] against one or more completely different methods as they apply to the problem being solved in the paper. This type of evaluation is most frequently seen in applications oriented papers and is treated in the Section 4.

2. They evaluate the constructions described in [74] against similar constructions, especially in relation to one or more of the related problems listed in Section 2.1. Some examples of this type of evaluation follow.

A 2009 paper by Marantis and colleagues [91] compares three different point distributions on $\mathbb{S}^2$, including EQP$(2,N)$, by using test samples of 240 points and using them to reconstruct a function defined by spherical harmonics up to degree 8: "it is subsequently sampled with the three proposed sample point distributions and the pattern is reconstructed using the estimated harmonic coefficients." Unfortunately the paper does not explicitly state the reconstruction method used. The EQP$(2,240)$ reconstruction fails badly. The paper makes an attempt to explain this.

A 2016 paper by Rachinger and colleagues [111] compares different "constellations" of points on hyperspheres in a complex vector space. The case described in the paper is a hypersphere in $\mathbb{C}^3$, equivalent to the sphere $\mathbb{S}^5 \subset \mathbb{R}^6$. The EQP$(5,64)$ and EQP$(5,512)$ codes are compared to constellations obtained via k-means clustering, potential minimization, and per-antenna phase shift keying. Curiously, the paper calls the the EQP codes "EQPA constellations." The constellations are compared in terms of construction complexity, capacity, minimum distance, and power efficiency: "...EQPA works in such a way that the distribution of points becomes more and more uniform as the constellation size increases. This algorithm profits from packing the hypersphere more densely."

A related 2016 paper by Sedaghat and colleagues [121] compares the EQP codes to codes created by spherical K-means clustering with respect to performance of a wireless communication scheme called Phase Modulation on the hypersphere: "...the codes obtained by the spherical K-means algorithm have much better performance than the EQ codes. Note that EQ codes have the advantage that they can be constructed much more easily than K-means codes."

One of the most comprehensive comparisons of constructions for spherical codes on $\mathbb{S}^2$ is found in the 2016 paper of Hardin, Michaels and Saff [55], which examines quadrature, energy, packing and covering properties of a number of such constructions. The paper shows that the EQ point sets generated by the construction described in [74] are not only equidistributed and well-separated, but they also perform well with respect to energy, with numerical behaviour comparable to empirically optimal point sets. For the logarithmic and Coulomb potentials, "the generalized spiral and zonal equal area points perform the best of the algorithmically generated points." For the Riesz s-energy with $s = 2$, "the generalized spiral, zonal equal area, and equal area icosahedral points perform the best." For the Riesz s-energy with $s = 3$, "the equal area icosahedral points outperform the spiral and zonal equal area points of the algorithmically generated configurations. This is expected because their Voronoi decomposition is closest to the regular hexagonal lattice." For more detailed proofs, see the 2017 PhD thesis of Michaels [96]. For more context, see the 2019 book by Borodachov and colleagues [14, Chapter 7].

3.2 Improvements

For the sphere $\mathbb{S}^2$ the *diamond ensemble* [12] is a construction for spherical codes resembling the EQP$(2,N)$ codes, where the code is constructed directly and not via an equal area partition. Similarly to Zhou's construction [113, 148] and the EQP$(2,N)$ codes, the code points are distributed amongst the north and south poles and a small number of parallels of latitude. On each parallel, the code points are equally spaced. Unlike Zhou's construction and the EQP$(2,N)$ codes, the code points on each parallel are offset by a random angle, and the number of code points per parallel are chosen to minimize the expected logarithmic energy. The diamond en-

semble can then be used to construct an equal area partition similar to Zhou's construction or the EQ$(2,N)$ partition [39].

The first paper [12] concentrates on logarithmic energy and cites Zhou's construction. It would be interesting to compare the results for logarithmic energy with the empirical logarithmic energy of the EQP$(2,N)$ codes, especially considering that the EQP$(2,N)$ codes have a non-random rotation offset on each parallel that maximizes the distance between the code points on adjacent parallels [75, Section 4.1.2].

The second paper [39] examines spherical cap discrepancy and also cites the PhD thesis [75], and the Matlab toolbox [79]. It is interesting to compare the proof of [39, Theorem 1.6] on the upper bound for spherical cap discrepancy of the diamond ensemble with the proof of [75, Theorem 5.4.1] on the same topic for the EQP(d,N) codes. As expected, the order of the bound in both proofs coincides for $d = 2$. The latter proof involves general $d > 1$ rather than just $d = 2$, but it uses a similar argument about the number of regions of an equal area partition that contain the boundary of a spherical cap. See, for example, [39, Figure 3].

4 Some applications

Biology and medicine

Biochemistry. The 2009 paper by Chu and colleagues [27] investigates RNA folding by simulating two simple cases where two helices are joined by a non-helix segment. The methods used in the paper include apparently using EQP$(3,16000)$ to produce "8000 equally spaced points on the upper half-sphere of the unit three-sphere $\mathbb{S}^3 \subset \mathbb{R}^4$, yielding a set of quaternions that sampled the space of rigid body rotations $SO(3)$ evenly." (Figure 13).

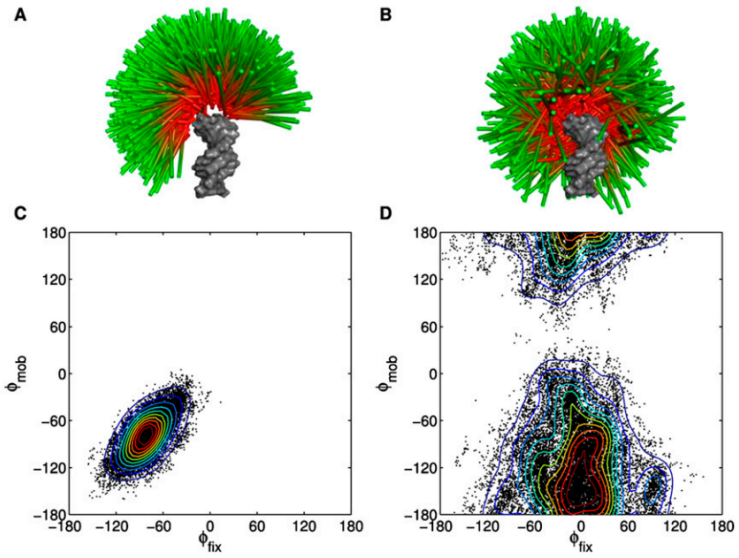


Figure 13: Chu et al. "Visualization of 1000 randomly selected conformers observed in the dPEG (A) and sPEG (B) simulations," [27].

Medical imaging. The 2020 paper by Lazarus and colleagues [72] extends the "SPARKLING (Spreading Projection Algorithm for Rapid K-space sampLING)" optimization algorithm for

efficient compressive sampling patterns for 3D magnetic resonance imaging (MRI). The 3D SPARKLING process uses the EQ$(2,100)$ partition to arrange MRI shots in a trajectory. The paper compares this process with two stacked SPARKLING processes and finds it to be inferior to a variable density stacked SPARKLING process (Figure 14).

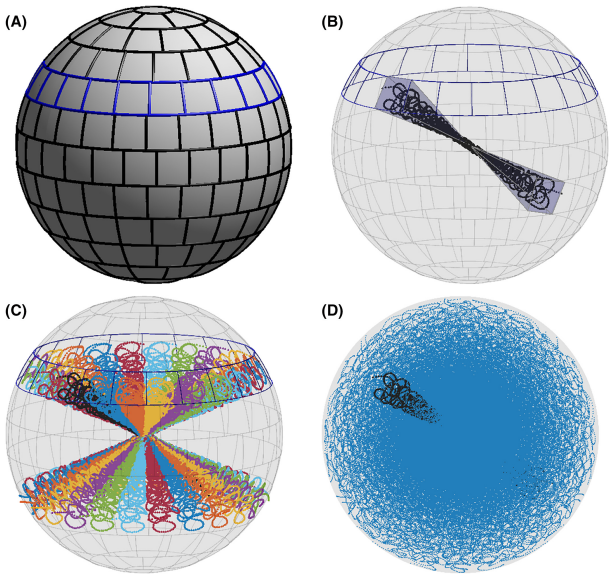


Figure 14: Lazarus et al. "3D SPARKLING process. A, Partition of the sphere into 100 regions of equal area. Regions along a constant elevation angle were highlighted in blue: they are identical up to a rotation. B, One 3D density sector containing a SPARKLING shot. C, The SPARKLING shot is then rotated along the considered latitude. D, the whole fully 3D SPARKLING trajectory. An individual segment is highlighted in black. ..." [72].

Neurobiology. The 2020 paper by Das and Maharatna [29] presents an "end-to-end toolchain that processes raw MRI data and generates network metrics for brain connectivity analysis using non-anatomical equal-area parcellation." The method presented in this paper is quite involved, but includes steps that use the EQ$(2,80)$ partition and EQP$(2,80)$ spherical code: "... we partition the spherical surface into equal sized areas by applying the equipartition algorithm of unit sphere [74]. We create a list of centre points of all the equal partitioned areas of unit sphere and scale them up to spherical surface ..." (Figure 15).

Climate and weather

Climate science. The 2018 paper by Werner and colleagues [138] presents "the first spatially resolved and millennium-length summer (June–August) temperature reconstruction over the Arctic and sub-Arctic domain (north of 60° N)."

The 2008 paper by Fauchereau and colleagues [40] applies Empirical Mode Decomposition (EMD) "in two dimensions over the sphere to demonstrate its potential as a data-adaptive method of separating the different scales of spatial variability in a geophysical (climatological/meteorological) field." The paper uses the EQ$(2,6500)$ partition and the EQP$(2,6500)$ spherical code (Figure 17).

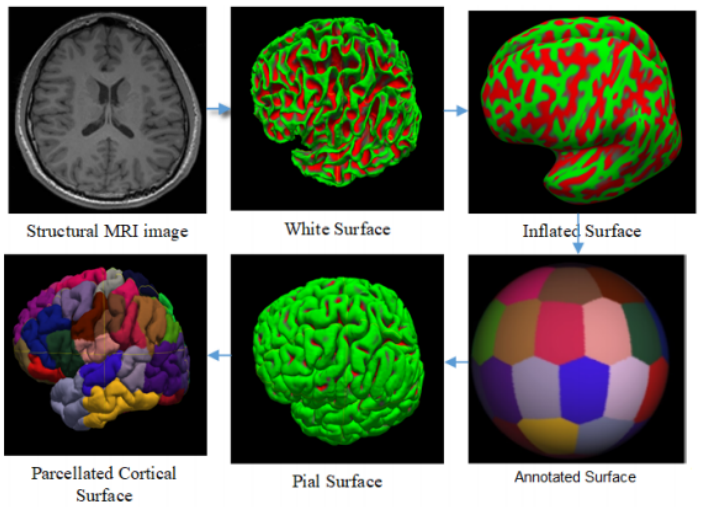


Figure 15: Das and Maharatna, "Raw MRI scans to parcellated segmented brain image," [29].

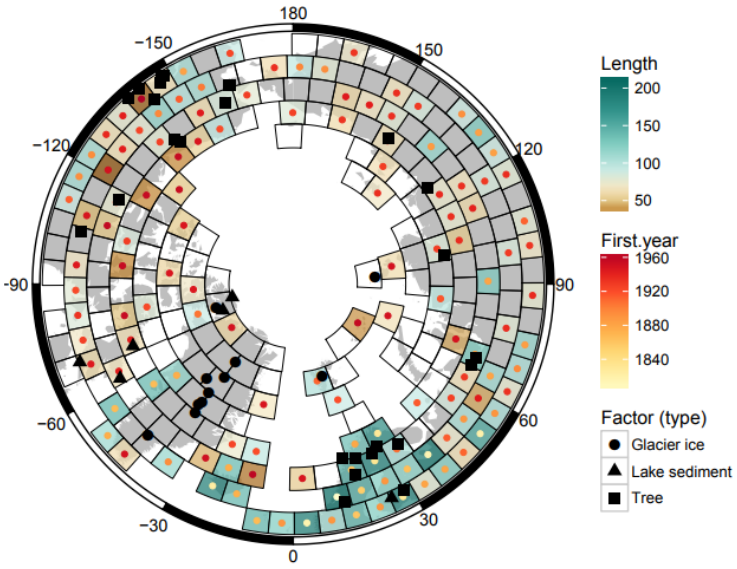


Figure 16: Werner et al. "Distribution of input data. Length (fill of quadrilaterals) and first year (coloured circles) of the regridded instrumental data. Symbols show the locations and type of proxy data used (PAGES 2k Consortium, 2017). The reconstruction target area is all grid cells marked with wire frames.". [138].

Numerical weather prediction. Papers by Mozdzynski and others at the European Centre for Medium Range Weather Forecasts (ECMWF) [32, 33, 99, 100, 137] describe the use of code derived from the EQSP Matlab Toolbox [73] to balance the parallel load of the ECMWF Integrated Forecasting System (IFS). The papers call this load balancing method *EQ_REGIONS partitioning* (Figure 18).

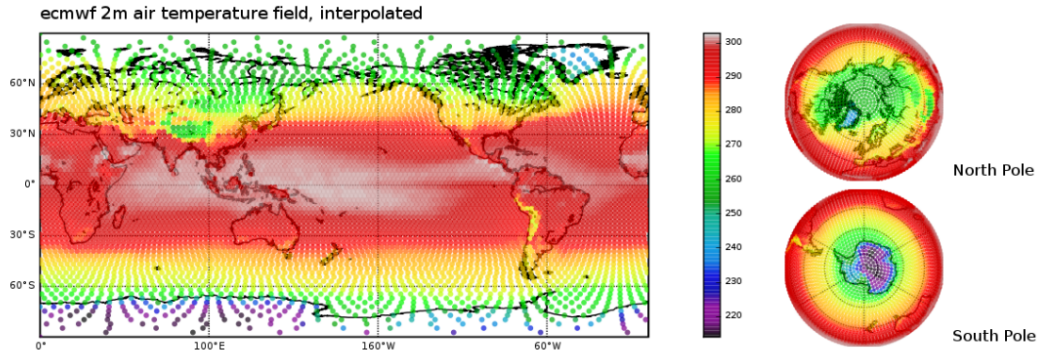


Figure 17: Fauchereau et al. "ERA 15 surface temperature long-term mean (1979–1993): interpolated onto a zonal equal area partitioning of the sphere using 6500 points." [40].

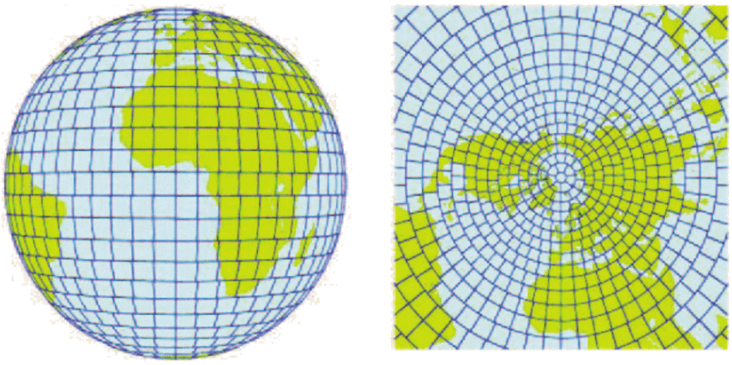


Figure 18: Mozdzynski et al. "EQ_REGIONS partitioning of grid-point space," [99].

Geology and geophysics

Papers and theses describing applications in geology and geophysics include [4, 36, 35, 38, 50, 52, 51, 53, 58, 66, 93, 103, 104]. For example, the paper by Matsuyama and colleagues [93] uses the EQ$(2,400)$ partition to sample tectonic patterns on the moon (Figure 19). The paper does not justify this choice of sampling method or estimate its accuracy.

The paper by Alken and colleagues [4] uses a robust Huber model based on 10 000 points obtained via the EQP$(2,10\,000)$ spherical code as a component of the evaluation of models of the Earth's magnetic field.

The thesis by Domingos [38], and the papers by Hammer and colleagues [52, 51, 53] Istas and colleagues [58], and Kloss and colleagues [66] use the EQP$(2,300)$ or EQP$(2,500)$ spherical codes to locate either 300 points or 500 points around the Earth, and use these points to locate geomagnetic virtual observatories (Figure 20). Each of these is effectively an approximate solution of the spherical cap packing problem.

Materials science

The 2021 paper by Sabiston and colleagues [117] presents and evaluates a micromechanics model for use in the fatigue characterization of injection moulded carbon fibre. The microstructure is characterized in terms of the orientation of carbon fibres, as an orientation distribution function (ODF). This function is approximated through the use of the EQ$(2,1200)$ partition and

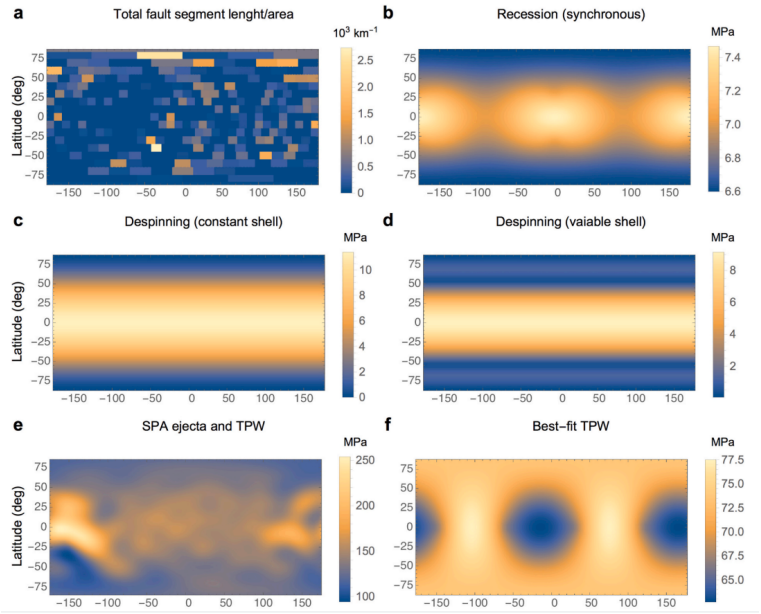


Figure 19: Matsuyama et al. "a, Total fault segment length per unit area computed by sampling the digitized fault segments in 400 equal area regions partitioned using the 'igloo' method of Leopardi (2006). b-f, Absolute difference between the maximum and minimum principal stresses (principal stresses difference), which quantifies the deviatoric stress, for a variety of mechanisms combined with isotropic contraction . . . " [93].

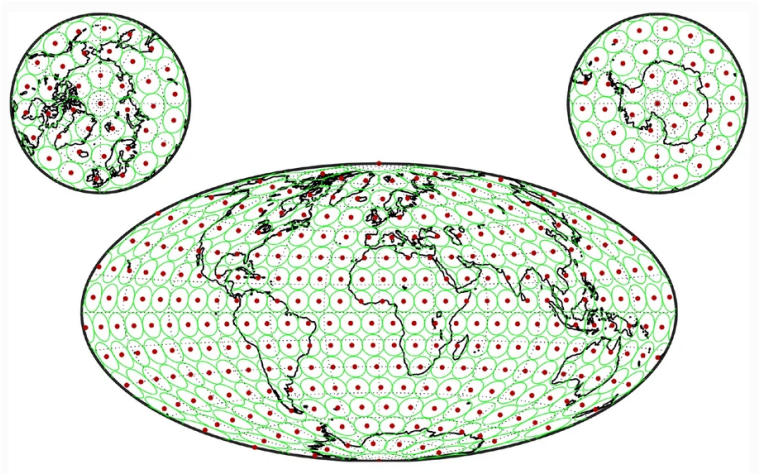


Figure 20: Hammer et al. "Distribution of the 300 GVOs (red dots) and associated cylinder bins (in green) using a Hammer projection," [51].

the EQP$(2, 1200)$ spherical code. "1200 was selected by performing a parametric study on the effect of number of orientations on the homogenized stress as well as the maximum interface stress" (Figure 21).

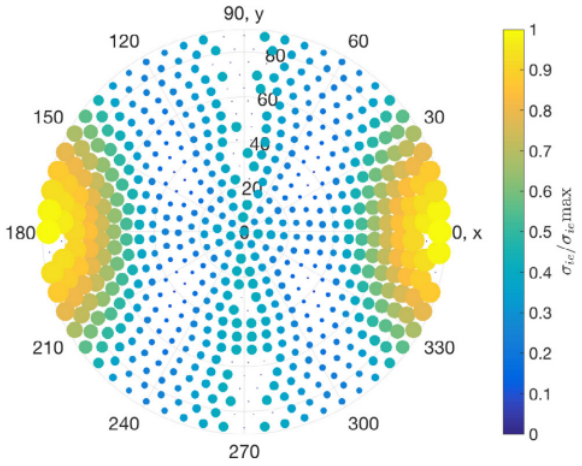


Figure 21: Sabiston et al. "Interface stress distribution over the fibre ODF for plaque location 1 considering the entire microstructure, excluding orientations that do not appear in the evaluated microstructural image," [117].

Mathematical physics

The 2021 paper by Benedikter and colleagues [13] rigorously derives "the leading order of the correlation energy of a Fermi gas in a scaling regime of high density and weak interaction." The paper uses a modified version of the EQ$(2,M)$ partition that (1) partitions the northern hemisphere and reflects this partition into the southern hemisphere; and (2) introduces corridors between the regions; (Figure 22).

Fig. 2 Patch decomposition of the northern half of the unit sphere: a spherical cap is placed at the pole; then collars along the latitudes are introduced and split into patches, separated by corridors. The vectors $\hat{\omega}_\alpha$ are picked as centers of the patches, marked in black. Finally patches are reflected by the origin to the southern half sphere

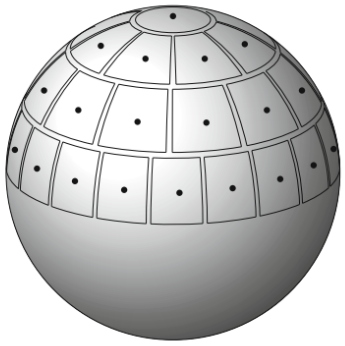
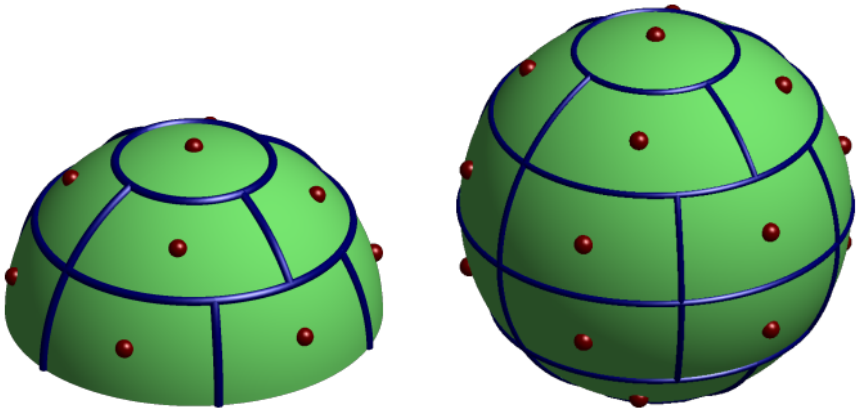


Figure 22: Benedikter et al. [13].

Robotics

The 2020 paper by Pfaff and colleagues [106]. proposes "a grid filter for arbitrary-dimensional unit hyperhemispheres and apply it to an orientation estimation task and another evaluation scenario." (Figure 23). It is one of a series of related papers [43, 71, 82, 83, 84, 85, 106, 107, 108] that each cite [74]. The partitions used in the paper differ from EQ(d,N) partitions in the following way: "we adjusted the algorithm so that it yields the best even integer number



(a) Partition of the hemisphere comprising 11 regions obtained using our novel partitioning scheme.

(b) Symmetric partition of $\mathbb{S}^2$ with 22 regions obtained by mirroring the partition of the hemisphere on the center of the full sphere.

Figure 23: Pfaff et al. "Illustration showing a partition of $\mathbb{H}^2$ with equally sized regions and a partition of $\mathbb{S}^2$ obtained by mirroring the partition of the hemisphere," [106].

of collars. Then, when subdividing from top to bottom, the boundary of one collar will run along the equator of the hypersphere." The paper does not explain how the modified algorithm generates regions of equal area, nor does it provide source code.

Visualization

The 2012 paper by Arrigo and colleagues [8] describes the R2G2 R CRAN package for the visualization of spatial data using Google Earth. The package uses the EQ$(2,50)$, *Partition*$(2,500)$, EQ$(2,5000)$, EQ$(2,10\,000)$, and EQ$(2,20\,000)$ partitions to calculate and plot histograms and other visualizations of data distributed on the Earth's surface (Figure 24).

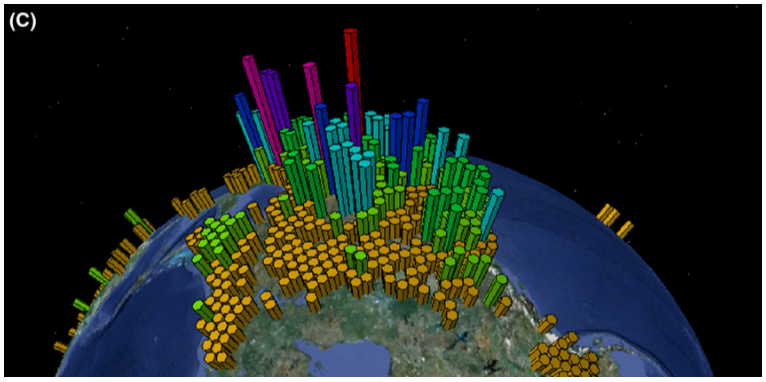


Figure 24: Arrigo, et al. "Species diversity of the *Selaginella* subgenus *Tetragonostachys* in North America using a grid of 20 000 cells with equal areas," [8].

Sphere vs hypersphere

The majority of application papers citing [74] focus only on applications on the sphere $\mathbb{S}^2$. Even so, only about 14 papers citing [74] also cite either the original work on zonal equal area partitioning of $\mathbb{S}^2$ by Rakhmanov, Saff and Zhou [113] or Zhou's PhD thesis [148]. Of these 14 papers ([2, 21, 22, 37, 55, 56, 57, 60, 91, 96, 127, 136, 140, 147, 150]) only six ([2, 57, 60, 91, 147, 150]) are applications-oriented.

Ahmadia's 2010 PhD thesis uses the EQ(d,N) partitions with d from 5 to 12 to solve a semiconductor lithography optimization problem [2]. The 2011 paper by Ishii uses the EQ$(2,N)$ partitions and EQP$(2,N)$ codes as possible solutions to a sampling problem used to calculate the total radiated power from radio equipment – a quadrature problem [57]). Jenkins' 2012 paper [60] concerns the construction of sparse spanners of unit ball graphs in $\mathbb{R}^3$, relating this problem to the covering problem on the sphere, and using EQ$(2,N)$ partitions to efficiently approximate coverings. The 2009 paper by Marantis and colleagues [91] is described in Section 3.1. The 2009 paper by Xie and colleagues [147] is a conference paper companion to [91]. The 2009 paper by Zotter [149] compares a number of different methods of approximating polynomial functions on the sphere $\mathbb{S}^2$ in order to analyze discrete spherical microphone and loudspeaker arrays. Unfortunately this paper seems to confuse interpolation using extremal fundamental systems on $\mathbb{S}^2$ with hyperinterpolation [123].

Of the papers that tackle applications on higher-dimensional spheres, some use the double covering of the $SO(3)$ group of rotations in $\mathbb{R}^3$ by the $SU(2)$ group, represented by the unit quaternions, modelled as the hypersphere $\mathbb{S}^3$, and therefore use the EQ$(3,N)$ partitions and the EQP$(3,N)$ codes [26, 27, 34, 44, 94, 95, 106, 108]. A few, such as the 2022 paper by Ramírez and Elvingson [114] address $\mathbb{S}^3$ and the EQP$(3,N)$ codes for other reasons. Others address applications in higher dimensions, including: Ahmadia's 2010 PhD thesis [2], as described above; a 2012 report by Kessler and colleagues [65] that describes an algorithm that uses the EQP codes to construct an approximately optimal path to extinction in systems of arbitrary dimensions; the 2016 paper by Rachsinger and colleagues [111], as described in Section 3.1; the related 2016 paper by Sedaghat and colleagues [121], also described in Section 3.1; a 2017 paper by Kurz and Hanebeck [69] that uses the EQP codes to construct linear regression Kalman filters; and finally a 2021 paper by Miyamoto and colleagues [97] that uses Hopf fibrations to construct spherical codes in $\mathbb{R}^{2^k}$, comparing these to EQP$(2^k-1,N)$ codes for k from 2 to 5.

Of the mathematical papers that use the EQ partitions and the EQP codes in higher dimensions, one stands out: the 2019 paper by Kunc and Fritzen [67], which uses the EQP codes as starting points for energy minimization.

5 Conclusion

Judging from the wide variety of applications of the EQ partitions and the EQP codes, these constructions appear to be widely applicable.

Closer inspection reveals that the constructions perform poorly on some problems. Chief among these is the reconstruction of functions via spherical harmonics [91], accomplished on $\mathbb{S}^2$, for example, by scattered data approximation [61, 62, 123]. The joint paper with Sommariva and Vianello [81] addresses this problem by using large EQP codes as norming sets, and constructing subsets that have approximation properties almost as good as the norming sets.

In the case covered in this paper, the applicability of the mathematical construction appears to be not so unreasonable, given the work that has been done in testing, for each relevant problem, the performance of the construction relative to alternatives, and the fitting of the construction into an overall solution that addresses each specific application.

Acknowledgments

This paper is based on a series of presentations given at Oak Ridge National Laboratory (ORNL) in 2014, The European Centre for Medium-Range Weather Forecasts (ECMWF) in 2016, The Bureau of Meteorology in 2017, and The Queensland Association of Mathematics Teachers (QAMT) State Conference in 2021.

Thanks to Kate Evans of ORNL, George Mozdzynski of ECMWF, and Monique Russell of QAMT for invitations to speak. Thanks to the Bureau of Meteorology for support to visit ECMWF in Reading in 2016.

The original research in the PhD thesis [75] was supported by a University Postgraduate Award from the University of New South Wales.

Declaration of interests

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Aaronson. Why philosophers should care about computational complexity. In B. Copeland, C. Posy, and O. Shagrir, editors, *Computability: Turing, Gödel, Church, and Beyond*, pages 261–327. MIT Press, 2013. ISBN 978-0-262-01899-9. doi: 10.7551/mitpress/8009.003.0011. URL https://doi.org/10.7551/mitpress/8009.003.0011.

[2] A. J. Ahmadia. *Parallel strategies for nonlinear mask optimization in semiconductor lithography*. PhD thesis, Columbia University, 2010.

[3] R. Alexander. On the sum of distances between N points on a sphere. *Acta Mathematica*, 23:443–448, 1972. doi: 10.1007/BF01896964. URL https://doi.org/10.1007/BF01896964.

[4] P. Alken, E. Thébault, C. Beggan, J. Aubert, J. Baerenzung, W. J. Brown, S. Califf, A. Chulliat, G. Cox, C. C. Finlay, et al. Evaluation of candidate models for the 13th generation International Geomagnetic Reference Field. Technical report, Springer, 2021. URL https://doi.org/10.1186/s40623-020-01281-4.

[5] E. L. Altschuler, T. J. Williams, E. R. Ratner, R. Tipton, R. Stong, F. Dowla, and F. Wooten. Possible global minimum lattice configurations for Thomson's problem of charges on a sphere. *Physical Review Letters*, 78(14):2681, 1997. doi: 10.1103/PhysRevLett.78.2681. URL https://doi.org/10.1103/PhysRevLett.78.2681.

[6] C. An, X. Chen, I. H. Sloan, and R. S. Womersley. Well conditioned spherical designs for integration and interpolation on the two-sphere. *SIAM J. Numer. Anal.*, 48(6):2135–2157, 2010. ISSN 0036-1429. doi: 10.1137/100795140. URL https://doi.org/10.1137/100795140.

[7] V. Arnold. On teaching mathematics. *Resonance Journal of Science Education*, 19 (9):851–861, September 2014. URL https://www.ias.ac.in/article/fulltext/reso/019/09/0851-0861.

[8] N. Arrigo, L. P. Albert, P. G. Mickelson, and M. S. Barker. Quantitative visualization of biological data in Google Earth using R2G2, an R CRAN package. *Molecular Ecology Resources*, 12(6):1177–1179, 2012. doi: 10.1111/1755-0998.12012. URL https://doi.org/10.1111/1755-0998.12012.

[9] M. Aste, M. Boninsegna, A. Freno, and E. Trentin. Techniques for dealing with incomplete data: a tutorial and survey. *Pattern Analysis and Applications*, 18:1–29, 2015. doi: 10.1007/s10044-014-0411-9. URL https://doi.org/10.1007/s10044-014-0411-9.

[10] J. Azzouni. Applying mathematics: An attempt to design a philosophical problem. *The Monist*, 83(2):209–227, 2000. URL http://www.jstor.org/stable/27903679.

[11] J. Beck and W. Chen. *Irregularities of distribution*. Cambridge University Press, 1987. doi: 10.1017/CBO9780511565984. URL https://doi.org/10.1017/CBO9780511565984.

[12] C. Beltrán and U. Etayo. The diamond ensemble: a constructive set of spherical points with small logarithmic energy. *Journal of Complexity*, 59:101471, 2020. doi: 10.1016/j.jco.2020.101471. URL http://hdl.handle.net/10902/20784.

[13] N. Benedikter, P. T. Nam, M. Porta, B. Schlein, and R. Seiringer. Correlation energy of a weakly interacting Fermi gas. *Inventiones mathematicae*, 225(3):885–979, 2021. doi: 10.1007/s00222-021-01041-5. URL https://doi.org/10.1007/s00222-021-01041-5.

[14] S. V. Borodachov, D. P. Hardin, and E. B. Saff. *Discrete energy on rectifiable sets*. Springer, 2019. ISBN 978-0-387-84807-5; 978-0-387-84808-2. doi: 10.1007/978-0-387-84808-2. URL https://doi.org/10.1007/978-0-387-84808-2.

[15] A. Borovik. A mathematician's view of the unreasonable ineffectiveness of mathematics in biology. *Biosystems*, 205:104410, 2021. doi: 10.1016/j.biosystems.2021.104410. URL https://doi.org/10.1016/j.biosystems.2021.104410.

[16] L. Bos and S. De Marchi. Limiting values under scaling of the Lebesgue function for polynomial interpolation on spheres. *Journal of approximation theory*, 96(2):366–377, 1999. doi: 10.1006/jath.1998.3245. URL https://doi.org/10.1006/jath.1998.3245.

[17] J. Bourgain and J. Lindenstrauss. Distribution of points on spheres and approximation by zonotopes. *Israel Journal of Mathematics*, 64:25–32, 1988. doi: 10.1007/BF02767366. URL https://doi.org/10.1007/BF02767366.

[18] J.-M. Bourinet. Ferum 4.1 user's guide. *Institute Français de Mécanique Avancée (IFMA), Clermont-Ferrand, France*, 2010. URL https://www.sigma-clermont.fr/sites/default/files/atoms/files/FERUM4.1_Users_Guide.pdf.

[19] J.-M. Bourinet, C. Mattrand, and V. Dubourg. A review of recent features and improvements added to FERUM software. In H. Furuta, D. M. Frangopol, and M. Shinozuka, editors, *Proc. 10th International Conference on Structural Safety and Reliability (ICOSSAR 2009), Osaka, Japan, September 13–17, 2009*. CRC Press, 2009. URL https://hal.science/hal-04502313.

[20] G. E. Box. Robustness in the strategy of scientific model building. In *Robustness in statistics*, pages 201–236. Elsevier, 1979. doi: 10.1016/B978-0-12-438150-6.50018-2. URL https://doi.org/10.1016/B978-0-12-438150-6.50018-2.

[21] J. Brauchart. Optimal logarithmic energy points on the unit sphere. *Mathematics of Computation*, 77(263):1599–1613, 2008. doi: 10.1090/S0025-5718-08-02085-1. URL https://doi.org/10.1090/S0025-5718-08-02085-1.

[22] J. Brauchart, D. P. Hardin, and E. B. Saff. The next-order term for optimal Riesz and logarithmic energy asymptotics on the sphere. *Recent advances in orthogonal polynomials, special functions, and their applications*, 578:31–61, 2012. doi: 10.1090/conm/578/11483. URL https://doi.org/10.1090/conm/578/11483.

[23] J. S. Brauchart. Punkvertverteilungen extremaler diskreter Energien auf Sphären. Diplomarbeit, Institut für Mathematik A, Technische Universität Graz, Graz, Austria, 2001.

[24] J. S. Brauchart. Low-discrepancy point sets lifted to the unit sphere. Presentation at MCQMC, 2012. URL http://www.mcqmc2012.unsw.edu.au/slides/MCQMC2012_Brauchart.pdf.

[25] J. S. Brauchart, J. Dick, E. B. Saff, I. H. Sloan, Y. G. Wang, and R. S. Womersley. Covering of spheres by spherical caps and worst-case error for equal weight cubature in Sobolev spaces. *Journal of Mathematical Analysis and Applications*, 431(2):782–811, 2015. doi: 10.1016/j.jmaa.2015.05.079. URL https://doi.org/10.1016/j.jmaa.2015.05.079.

[26] V. B. Chu. *Probing RNA Folding Through Electrostatic and Coarse-grained Simulations*. PhD thesis, Stanford University, 2010. URL https://purl.stanford.edu/st973xv8543.

[27] V. B. Chu, J. Lipfert, Y. Bai, V. S. Pande, S. Doniach, and D. Herschlag. Do conformational biases of simple helical junctions influence RNA folding stability and specificity? *RNA*, 15(12):2195–2205, 2009. doi: 10.1261/rna.1747509. URL https://doi.org/10.1261/rna.1747509.

[28] S. B. Damelin and P. J. Grabner. Energy functionals, numerical integration and asymptotic equidistribution on the sphere. *Journal of Complexity*, 19(3):231–246, June 2003. doi: 10.1016/S0885-064X(02)00006-7. URL https://doi.org/10.1016/S0885-064X(02)00006-7. (Postscript) Corrigendum, Journal of Complexity, 20 (2004), pp. 883–884.

[29] S. Das and K. Maharatna. An automated toolchain for quantitative characterisation of structural connectome from MRI based on non-anatomical cortical parcellation. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 5653–5656. IEEE, 2020. doi: 10.1109/EMBC44109.2020.9176642. URL https://doi.org/10.1109/EMBC44109.2020.9176642.

[30] D. Deboy. *Acoustic centering and rotational tracking in surrounding spherical microphone arrays*. PhD thesis, University of Music and Performing Arts Graz, Austria, 2010. URL https://ambisonics.iem.at/Members/zotter/index_html/publications/2010_Deboy_AcousticCenteringRotationTracking_DA.pdf.

[31] D. Deboy and F. Zotter. Acoustic center and orientation analysis of sound-radiation recorded with a surrounding spherical microphone array. In *Proceedings of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, volume 21, pages 6–7, 2010. URL https://api.semanticscholar.org/CorpusID:2326259.

[32] W. Deconinck, M. Hamrud, C. Kühnlein, G. Mozdzynski, P. K. Smolarkiewicz, J. Szmelter, and N. P. Wedi. Accelerating extreme-scale numerical weather prediction. In *Parallel Processing and Applied Mathematics: 11th International Conference, PPAM 2015, Krakow, Poland, September 6-9, 2015. Revised Selected Papers, Part II*, pages 583–593. Springer, 2016. doi: 10.1007/978-3-319-32152-3_54. URL https://doi.org/10.1007/978-3-319-32152-3_54.

[33] W. Deconinck, P. Bauer, M. Diamantakis, M. Hamrud, C. Kühnlein, P. Maciel, G. Mengaldo, T. Quintino, B. Raoult, P. K. Smolarkiewicz, et al. Atlas: A library for numerical weather prediction and climate modelling. *Computer Physics Communications*, 220: 188–204, 2017. doi: 10.1016/j.cpc.2017.07.006. URL https://doi.org/10.1016/j.cpc.2017.07.006.

[34] M. J. Del Razo, M. Dibak, C. Schütte, and F. Noé. Multiscale molecular kinetics by coupling Markov state models and reaction-diffusion dynamics. *The Journal of Chemical Physics*, 155(12), 2021. doi: 10.1063/5.0060314. URL https://doi.org/10.1063/5.0060314.

[35] S. G. Devriese. *Detecting and imaging time-lapse conductivity changes using electromagnetic methods*. PhD thesis, University of British Columbia, 2016. URL https://dx.doi.org/10.14288/1.0340342.

[36] S. G. Devriese and D. W. Oldenburg. Enhanced imaging of SAGD steam chambers using broadband electromagnetic surveying. In *SEG International Exposition and Annual Meeting*, pages SEG–2014. SEG, 2014. doi: 10.1190/segam2014-1247.1. URL https://doi.org/10.1190/segam2014-1247.1.

[37] A. Dickstein and F. Zapolsky. Approximation of quasi-states on manifolds. *Journal of Applied and Computational Topology*, 3(3):221–248, 2019. doi: 10.1007/s41468-019-00030-1. URL https://doi.org/10.1007/s41468-019-00030-1.

[38] J. M. R. Domingos. *Geomagnetic and space weather variability modes in satellite data*. PhD thesis, Université Grenoble Alpes; Universidade de Coimbra, 2018. URL https://theses.hal.science/tel-01828171/file/ROSA_DOMINGO_2018_archivage.pdf.

[39] U. Etayo. Spherical cap discrepancy of the diamond ensemble. *Discrete & Computational Geometry*, 66(4):1218–1238, 2021. doi: 10.1007/s00454-021-00305-4. URL https://doi.org/10.1007/s00454-021-00305-4.

[40] N. Fauchereau, G. Pegram, and S. Sinclair. Empirical mode decomposition on the sphere: application to the spatial scales of surface temperature variations. *Hydrology*

and Earth System Sciences, 12(3):933–941, 2008. doi: 10.5194/hess-12-933-2008. URL https://doi.org/10.5194/hess-12-933-2008.

[41] U. Feige and G. Schechtman. On the optimality of the random hyperplane rounding technique for MAX CUT. *Random Structures and Algorithms*, 20(3):403–440, 2002. doi: 10.1002/rsa.10036. URL https://doi.org/10.1002/rsa.10036. Special Issue: Probabilistic Methods in Combinatorial Optimization.

[42] A. Frey. https://github.com/sashafrey/latex, 2016.

[43] D. Frisch and U. D. Hanebeck. Deterministic Von Mises–Fisher sampling on the sphere using Fibonacci lattices. In *2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*, pages 1–8. IEEE, 2023. doi: 10.1109/SDF-MFI59545.2023.10361396. URL https://doi.org/10.1109/SDF-MFI59545.2023.10361396.

[44] A. Fusiello. Exact affine histogram matching by cumulants transformation. In *International Conference on Image Analysis and Processing*, pages 199–210. Springer, 2022. doi: 10.1007/978-3-031-06430-2_17. URL https://doi.org/10.1007/978-3-031-06430-2_17.

[45] G. Gigante and P. Leopardi. Diameter bounded equal measure partitions of Ahlfors regular metric measure spaces. *Discrete & Computational Geometry*, 57:419–430, 2017. doi: 10.1007/s00454-016-9834-y. URL https://doi.org/10.1007/s00454-016-9834-y.

[46] P. J. Grabner and R. F. Tichy. Spherical designs, discrepancy and numerical integration. *Mathematics of Computation*, 60:327–336, 1993. doi: 10.2307/2153170. URL https://doi.org/10.2307/2153170.

[47] M. Guniel. Electron microscopy images. 2024. URL https://www.dartmouth.edu/emlab/gallery/index.php.

[48] J. Hamkins. *Design and Analysis of Spherical Codes*. PhD thesis, Univ. of Illinois at Urbana-Champaign, 1996. URL https://hdl.handle.net/2142/20964.

[49] J. Hamkins and K. Zeger. Asymptotically dense spherical codes. I. Wrapped spherical codes. *IEEE Transactions on Information Theory*, 43(6):1774–1785, November 1997. doi: 10.1109/18.641544. URL https://doi.org/10.1109/18.641544.

[50] M. D. Hammer. *Local estimation of the Earth's core magnetic field*. PhD thesis, Technical University of Denmark, 2018. URL https://ftp.space.dtu.dk/pub/cfinl/theses/PhDThesis_Magnus_Hammer.pdf.

[51] M. D. Hammer, G. A. Cox, W. J. Brown, C. D. Beggan, and C. C. Finlay. Geomagnetic Virtual Observatories: monitoring geomagnetic secular variation with the Swarm satellites. *Earth, Planets and Space*, 73(1):1–22, 2021. doi: 10.1186/s40623-021-01357-9. URL https://doi.org/10.1186/s40623-021-01357-9.

[52] M. D. Hammer, C. C. Finlay, and N. Olsen. Applications for CryoSat-2 satellite magnetic data in studies of Earth's core field variations. *Earth, Planets and Space*, 73:1–22, 2021. doi: 10.1186/s40623-021-01365-9. URL https://doi.org/10.1186/s40623-021-01365-9.

[53] M. D. Hammer, C. C. Finlay, and N. Olsen. Secular variation signals in magnetic field gradient tensor elements derived from satellite-based geomagnetic virtual observatories. *Geophysical Journal International*, 229(3):2096–2114, 2022. doi: 10.1093/gji/ggac004. URL https://doi.org/10.1093/gji/ggac004.

[54] R. W. Hamming. The unreasonable effectiveness of mathematics. *The American Mathematical Monthly*, 87(2):81–90, 1980. doi: 10.2307/2321982. URL https://doi.org/10.2307/2321982.

[55] D. P. Hardin, T. Michaels, and E. B. Saff. A comparison of popular point configurations on S^2. *Dolomites Research Notes on Approximation*, 9(1):16–49, 2016. doi: 10.14658/PUPJ-DRNA-2016-1-2. URL https://drna.padovauniversitypress.it/2016/1/2.

[56] A. Holhoş and D. Roşca. An octahedral equal area partition of the sphere and near optimal configurations of points. *Computers & Mathematics with Applications*, 67(5):1092–1107, 2014. doi: 10.1016/j.camwa.2014.01.003. URL https://doi.org/10.1016/j.camwa.2014.01.003.

[57] N. Ishii. Comparison of sampling methods for total radiated power estimation from radio equipment integrated with antennas. *IEICE transactions on communications*, 94(5):1174–1183, 2011. doi: 10.1587/transcom.E94.B.1174. URL https://doi.org/10.1587/transcom.E94.B.1174.

[58] M. Istas, N. Gillet, C. Finlay, M. Hammer, and L. Huder. Transient core surface dynamics from ground and satellite geomagnetic data. *Geophysical Journal International*, 233(3):1890–1915, 2023. doi: 10.1093/gji/ggad039. URL https://doi.org/10.1093/gji/ggad039.

[59] R. Ivanov. https://github.com/rivapp/CAV21_repeatability_package, 2021.

[60] J. P. Jenkins, I. A. Kanj, G. Xia, and F. Zhang. Local construction of spanners in the 3D space. *IEEE Transactions on Mobile Computing*, 11(7):1140–1150, 2012. doi: 10.1109/TMC.2011.142. URL https://doi.org/10.1109/TMC.2011.142.

[61] K. Jetter, J. Stöckler, and J. D. Ward. Norming sets and spherical cubature formulas. In Z. Chen, Y. Li, C. A. Micchelli, and Y. Xu, editors, *Advances in Computational Mathematics (Guangzhou, 1997)*, volume 202, pages 237–244. Marcel Dekker, New York, 1999. ISBN 0-8247-1946-8.

[62] K. Jetter, J. Stöckler, and J. D. Ward. Error estimates for scattered data interpolation on spheres. *Mathematics of Computation*, 68:733–747, 1999. doi: 10.1090/S0025-5718-99-01080-7. URL https://doi.org/10.1090/S0025-5718-99-01080-7.

[63] S. Jokar and M. E. Pfetsch. Exact and approximate sparse solutions of underdetermined linear equations. *SIAM Journal on Scientific Computing*, 31(1):23–44, 2008. doi: 10.1137/070686676. URL https://doi.org/10.1137/070686676.

[64] T. Kam. https://github.com/phantomachine/HARDPIG, 2018.

[65] A. Kessler, L. B. Shaw, and I. B. Schwartz. On the construction of optimal paths to extinction. Technical report, US Naval Research Laboratory Washington DC Beam Physics Branch, 2012. URL https://apps.dtic.mil/sti/pdfs/ADA556290.pdf.

[66] C. Kloss and C. C. Finlay. Time-dependent low-latitude core flow and geomagnetic field acceleration pulses. *Geophysical Journal International*, 217(1):140–168, 2019. doi: 10.1093/gji/ggy545. URL https://doi.org/10.1093/gji/ggy545.

[67] O. Kunc and F. Fritzen. Generation of energy-minimizing point sets on spheres and their application in mesh-free interpolation and differentiation. *Advances in Computational Mathematics*, 45(5-6):3021–3056, 2019. doi: 10.1007/s10444-019-09726-5. URL https://doi.org/10.1007/s10444-019-09726-5.

[68] O. Kunc and Others. https://github.com/EMMA-Group/MinimumEnergyPoints, 2020.

[69] G. Kurz and U. D. Hanebeck. Linear regression Kalman filtering based on hyperspherical deterministic sampling. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 977–983. IEEE, 2017. doi: 10.1109/CDC.2017.8263785. URL https://doi.org/10.1109/CDC.2017.8263785.

[70] G. Kurz and Others. https://github.com/libDirectional/libDirectional, 2023.

[71] G. Kurz, F. Pfaff, and U. D. Hanebeck. Discretization of $SO(3)$ using recursive tesseract subdivision. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 49–55. IEEE, 2017. doi: 10.1109/MFI.2017.8170406. URL https://doi.org/10.1109/MFI.2017.8170406.

[72] C. Lazarus, P. Weiss, L. El Gueddari, F. Mauconduit, A. Massire, M. Ripart, A. Vignaud, and P. Ciuciu. 3D variable-density SPARKLING trajectories for high-resolution T2*-weighted magnetic resonance imaging. *NMR in Biomedicine*, 33(9):e4349, 2020. doi: 10.1002/nbm.4349. URL https://doi.org/10.1002/nbm.4349.

[73] P. Leopardi. Recursive zonal equal area sphere partitioning toolbox. Matlab software package available via SourceForge, 2005. URL http://eqsp.sourceforge.net.

[74] P. Leopardi. A partition of the unit sphere into regions of equal area and small diameter. *Electronic Transactions on Numerical Analysis*, 25(12):309–327, 2006. URL http://eudml.org/doc/129860.

[75] P. Leopardi. *Distributing points on the sphere: partitions, separation, quadrature and energy*. PhD thesis, UNSW Sydney, 2007. URL https://maths-people.anu.edu.au/~leopardi/Leopardi-Sphere-PhD-Thesis.pdf.

[76] P. Leopardi. Diameter bounds for equal area partitions of the unit sphere. *Electronic Transactions on Numerical Analysis*, 35:1–16, 2009. URL https://elibm.org/article/10006211.

[77] P. Leopardi. Discrepancy, separation and Riesz energy of finite point sets on the unit sphere. *Advances in Computational Mathematics*, 39:27–43, 2013. doi: 10.1007/s10444-011-9266-4. URL https://doi.org/10.1007/s10444-011-9266-4.

[78] P. Leopardi. Discrepancy, separation and Riesz energy of finite point sets on compact connected Riemannian manifolds. *Dolomites Research Notes on Approximation*, 6(2):120–129, 2013. doi: 10.14658/PUPJ-DRNA-2013-Special_Issue-12. URL https://drna.padovauniversitypress.it/2013/Special_Issue/12.

[79] P. Leopardi. Recursive zonal equal area sphere partitioning toolbox. Matlab software package available via GitHub, 2017. URL https://github.com/penguian/eq_sphere_partitions.

[80] P. Leopardi. Recursive zonal equal area sphere partitioning toolbox. Matlab software package available via Matlab Central, 2024. URL https://au.mathworks.com/matlabcentral/fileexchange/13356-eq_sphere_partitions.

[81] P. Leopardi, A. Sommariva, and M. Vianello. Optimal polynomial meshes and Caratheodory-Tchakaloff submeshes on the sphere. *Dolomites Research Notes on Approximation*, 10(2):18–24, 2017. doi: 10.14658/PUPJ-DRNA-2017-Special_Issue-4. URL https://drna.padovauniversitypress.it/2017/Special_Issue/4.

[82] K. Li, F. Pfaff, and U. D. Hanebeck. Grid-based quaternion filter for SO(3) estimation. In *2020 European Control Conference (ECC)*, pages 1738–1744. IEEE, 2020. doi: 10.23919/ECC51009.2020.9143723. URL https://doi.org/10.23919/ECC51009.2020.9143723.

[83] K. Li, F. Pfaff, and U. D. Hanebeck. Nonlinear von Mises–Fisher filtering based on isotropic deterministic sampling. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 108–113. IEEE, 2020. doi: 10.1109/MFI49285.2020.9235260. URL https://doi.org/10.1109/MFI49285.2020.9235260.

[84] K. Li, F. Pfaff, and U. D. Hanebeck. Hyperspherical Dirac mixture reapproximation, 2021. URL https://arxiv.org/abs/2110.10411.

[85] K. Li, F. Pfaff, and U. D. Hanebeck. Progressive von Mises–Fisher filtering using isotropic sample sets for nonlinear hyperspherical estimation. *Sensors*, 21(9):2991, 2021. doi: 10.3390/s21092991. URL https://doi.org/10.3390/s21092991.

[86] S. Li. https://github.com/shenlirobot/CORA, 2021.

[87] S.-B. Lin, X. Sun, and D. Wang. Distributed uncertainty quantification of kernel interpolation on spheres, 2023. URL https://arxiv.org/abs/2310.16384.

[88] S.-B. Lin, D. Wang, and D.-X. Zhou. Sketching with spherical designs for noisy data fitting on spheres. *SIAM Journal on Scientific Computing*, 46(1):A313–A337, 2024. doi: 10.1137/22M1484377. URL https://doi.org/10.1137/22M1484377.

[89] C. Lisciandra. Robustness analysis and tractability in modeling. *European Journal for Philosophy of Science*, 7:79–95, 2017. doi: 10.1007/s13194-016-0146-0. URL https://doi.org/10.1007/s13194-016-0146-0.

[90] M. MacLeod. The applicability of mathematics in computational systems biology and its experimental relations. *European Journal for Philosophy of Science*, 11(3):84, 2021. doi: 10.1007/s13194-021-00403-3. URL https://doi.org/10.1007/s13194-021-00403-3.

[91] L. Marantis, E. De Witte, and P. Brennan. Comparison of various spherical antenna array element distributions. In *2009 3rd European Conference on Antennas and Propagation*, pages 2980–2984. IEEE, 2009. URL https://ieeexplore.ieee.org/document/5068232.

[92] J. Marzo and J. Ortega-Cerdà. Equidistribution of Fekete points on the sphere. *Constructive Approximation*, 32(3):513–521, 2010. ISSN 0176-4276,1432-0940. doi: 10.1007/s00365-009-9051-5. URL https://doi.org/10.1007/s00365-009-9051-5.

[93] I. Matsuyama, J. Keane, A. Trinh, M. Beuthe, and T. Watters. Global tectonic patterns of the Moon. *Icarus*, 358:114202, 2021. doi: 10.1016/j.icarus.2020.114202. URL https://doi.org/10.1016/j.icarus.2020.114202.

[94] S. S. Mehta. *A daisy-chaining approach for vision-based control and estimation*. PhD thesis, 2009. URL https://ncr.mae.ufl.edu/dissertations/mehta_s.pdf.

[95] S. S. Mehta, P. Barooah, W. E. Dixon, E. L. Pasiliao, and J. W. Curtis. PEGUS: An image-based robust pose estimation method. In *2012 Ninth Conference on Computer and Robot Vision*, pages 78–85. IEEE, 2012. doi: 10.1109/CRV.2012.18. URL https://doi.org/10.1109/CRV.2012.18.

[96] T. Michaels. *Node generation on surfaces and bounds on minimal Riesz energy*. Vanderbilt University, 2017. URL http://hdl.handle.net/1803/14461.

[97] H. K. Miyamoto, S. I. Costa, and H. N. S. Earp. Constructive spherical codes by Hopf foliations. *IEEE Transactions on Information Theory*, 67(12):7925–7939, 2021. doi: 10.1109/tit.2021.3114094. URL https://doi.org/10.1109/tit.2021.3114094.

[98] G. Mozdzynski. A new partitioning approach for ECMWF's integrated forecasting system (IFS). In *Use Of High Performance Computing In Meteorology*, pages 148–166. 2007. doi: 10.21957/2m1i7n0v3a. URL https://doi.org/10.21957/2m1i7n0v3a.

[99] G. Mozdzynski, M. Hamrud, N. Wedi, J. Doleschal, and H. Richardson. A PGAS implementation by co-design of the ECMWF Integrated Forecasting System (IFS). In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, pages 652–661. IEEE, 2012. doi: 10.1109/SC.Companion.2012.90. URL https://doi.org/10.1109/SC.Companion.2012.90.

[100] G. Mozdzynski, M. Hamrud, and N. Wedi. A partitioned global address space implementation of the European Centre for Medium Range Weather Forecasts Integrated Forecasting System. *The International Journal of High Performance Computing Applications*, 29(3):261–273, 2015. doi: 10.1177/1094342015576773. URL https://doi.org/10.1177/1094342015576773.

[101] H. Na. https://github.com/htna/HCsbLib, 2024.

[102] J. D. Norton. Approximation and idealization: Why the difference matters. *Philosophy of Science*, 79(2):207–232, 2012. doi: 10.1086/664746. URL https://doi.org/10.1086/664746.

[103] N. Olsen, D. Ravat, C. C. Finlay, and L. K. Kother. LCS-1: a high-resolution global model of the lithospheric magnetic field derived from CHAMP and Swarm satellite observations. *Geophysical Journal International*, 211(3):1461–1477, 2017. doi: 10.1093/gji/ggx381. URL https://doi.org/10.1093/gji/ggx381.

[104] N. Olsen, D. Ravat, and M. E. Purucker. On the determination and interpretation of the lithospheric induced magnetisation. *Earth and Planetary Science Letters*, 606: 118038, 2023. doi: 10.1016/j.epsl.2023.118038. URL https://doi.org/10.1016/j.epsl.2023.118038.

[105] Q. Pan. https://github.com/qpanpony/ita-toolbox, 2020.

[106] F. Pfaff, K. Li, and U. D. Hanebeck. A hyperhemispherical grid filter for orientation estimation. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2020. URL https://isas.iar.kit.edu/pdf/FUSION20_Pfaff.pdf.

[107] F. Pfaff, K. Li, and U. D. Hanebeck. The spherical grid filter for nonlinear estimation on the unit sphere. *IFAC-PapersOnLine*, 53(2):2354–2360, 2020. doi: 10.1016/j.ifacol.2020.12.031. URL https://doi.org/10.1016/j.ifacol.2020.12.031.

[108] F. Pfaff, K. Li, and U. D. Hanebeck. The state space subdivision filter for $SE(3)$. In *2022 25th International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE, 2022. doi: 10.23919/FUSION49751.2022.9841384. URL https://doi.org/10.23919/FUSION49751.2022.9841384.

[109] H. Pomberger, F. Zotter, R. Höldrich, and S. Brandl. Estimating uncertainty in pressure-based sound power measurement due to spatial sampling and near fields. In *Proceedings of 6th Congress of the Alps Adria Acoustics Association*, 2014. URL https://iem.kug.ac.at/en/publications/publication/39997.

[110] E. Prats i Aymerich. *Configuracions de punts d'energia mínima a l'esfera*. PhD thesis, Universitat de Barcelona, 2017. URL http://hdl.handle.net/2445/122289.

[111] C. Rachinger, R. R. Müller, and J. B. Huber. Phase shift keying on the hypersphere: Peak power-efficient MIMO communications, 2016. URL https://arxiv.org/abs/1611.01009.

[112] A. Rajan and P. Tallapragada. Analysis of inter-event times in linear systems under region-based self-triggered control. *IEEE Transactions on Automatic Control*, 2023. doi: 10.1109/TAC.2023.3314136. URL https://doi.org/10.1109/TAC.2023.3314136.

[113] E. A. Rakhmanov, E. B. Saff, and Y. M. Zhou. Minimal discrete energy on the sphere. *Mathematical Research Letters*, 1(6):647–662, 1994. doi: 10.4310/MRL.1994.v1.n6.a3. URL https://doi.org/10.4310/MRL.1994.v1.n6.a3.

[114] E. V. Ramírez and C. Elvingson. An efficient linked list for molecular simulations on a spherical surface. *Journal of Physics A: Mathematical and Theoretical*, 55(38):385001, 2022. doi: 10.1088/1751-8121/ac852f. URL https://doi.org/10.1088/1751-8121/ac852f.

[115] J. R. Rice. The algorithm selection problem. In *Advances in computers*, volume 15, pages 65–118. Elsevier, 1976. doi: https://doi.org/10.1016/S0065-2458(08)60520-3. URL https://www.sciencedirect.com/science/article/pii/S0065245808605203.

[116] D. P. Rowbottom. Approximations, idealizations and 'experiments' at the physics–biology interface. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 42(2):145–154, 2011. ISSN 1369-8486. doi: https://doi.org/10.1016/j.shpsc.2010.11.021. URL https://www.sciencedirect.com/science/article/pii/S1369848610001147.

[117] T. Sabiston, B. Li, J. Kang, D. Wilkinson, and C. Engler-Pinto. Accounting for the microstructure in the prediction of the fatigue life of injection moulded composites for automotive applications. *Composite Structures*, 255:112898, 2021. doi: 10.1016/j.compstruct.2020.112898. URL https://doi.org/10.1016/j.compstruct.2020.112898.

[118] E. Saff and R. Womersley. Optimal polarization configurations for the sphere. Presentation at South Pacific Optimization Meeting, Newcastle, 2013. URL https://carmamaths.org/meetings/spom/pdf/spom2013-womersley.pdf.

[119] E. B. Saff and A. B. J. Kuijlaars. Distributing many points on a sphere. *Mathematical Intelligencer*, 19:5–11, 1997. doi: 10.1007/BF03024331. URL https://doi.org/10.1007/BF03024331.

[120] G. Schouten, W. Jansen, and J. Steckel. Simulation of pulse-echo radar for vehicle control and SLAM. *Sensors*, 21(2):523, 2021. doi: 10.3390/s21020523. URL https://doi.org/10.3390/s21020523.

[121] M. A. Sedaghat, R. R. Müller, and C. Rachinger. (continuous) phase modulation on the hypersphere. *IEEE Transactions on Wireless Communications*, 15(8):5763–5774, 2016. doi: 10.1109/TWC.2016.2569091. URL https://doi.org/10.1109/TWC.2016.2569091.

[122] I. H. Sloan. Equal area partition of S^3. Notes, July 2003.

[123] I. H. Sloan and R. S. Womersley. Constructive polynomial approximation on the sphere. *Journal of Approximation Theory*, 103(1):91–118, 2000. doi: 10.1006/jath.1999.3426. URL https://doi.org/10.1006/jath.1999.3426.

[124] P. R. Smale. Diffusion tensor imaging of motor connectivity in selected subjects with stroke. Technical report, University of Canterbury. Physics and Astronomy, 2007. URL http://hdl.handle.net/10092/1446.

[125] N. Steiger. https://clifford.ldeo.columbia.edu/nsteiger/recon_output/junkcode/hydro-recon/, 2021.

[126] K. B. Stolarsky. Sums of distances between points on a sphere II. *Proceedings of the American Mathematical Society*, 41:575–582, 1973. doi: 10.2307/2039137. URL https://doi.org/10.2307/2039137.

[127] X. Sun and Z. Chen. Spherical basis functions and uniform distribution of points on spheres. *Journal of approximation theory*, 151(2):186–207, 2008. doi: 10.1016/j.jat.2007.09.009. URL https://doi.org/10.1016/j.jat.2007.09.009.

[128] A. Szwajcowski, D. Krause, and A. Snakowska. Error analysis of sound source directivity interpolation based on spherical harmonics. *Archives of Acoustics*, 46(1), 2021. doi: 10.24425/aoa.2021.136564. URL https://doi.org/10.24425/aoa.2021.136564.

[129] W. Themistoclakis and M. V. Barel. Optimal lebesgue constants for least squares polynomial approximation on the (hyper)sphere, 2018. URL https://arxiv.org/abs/1808.03530.

[130] J. Traub and H. Wozniakowski. Information-based complexity: New questions for mathematicians. *The Mathematical Intelligencer*, 13(2), 1991. doi: 10.1007/BF03024085. URL https://doi.org/10.1007/BF03024085.

[131] L. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, Society for Industrial and Applied Mathematics, 2019. ISBN 978-1-611975-93-2.

[132] B. Van Dyke. *Directional Direct-search Optimization Methods with Polling Directions Based on Equal Angle Distributions*. PhD thesis, Washington State University, 2012. URL https://hdl.handle.net/2376/4697.

[133] M. Vianello. Global polynomial optimization by norming sets on sphere and torus. *Dolomites Research Notes on Approximation*, 11(1), 2018. doi: 10.14658/PUPJ-DRNA-2018-1-2. URL https://drna.padovauniversitypress.it/2018/1/2.

[134] R. Viard. https://github.com/ci2c/code, 2020.

[135] D. Wang. https://github.com/18357710774/SIAM_Sketching, 2023.

[136] Y. G. Wang, R. S. Womersley, H.-T. Wu, and W.-H. Yu. Numerical computation of triangular complex spherical designs with small mesh ratio. *Journal of Computational and Applied Mathematics*, 421:114796, 2023. doi: 10.1016/j.cam.2022.114796. URL https://doi.org/10.1016/j.cam.2022.114796.

[137] N. Wedi, P. Bauer, W. Denoninck, M. Diamantakis, M. Hamrud, C. Kuhnlein, S. Malardel, K. Mogensen, G. Mozdzynski, and P. Smolarkiewicz. *The modelling infrastructure of the Integrated Forecasting System: Recent advances and future challenges*. European Centre for Medium-Range Weather Forecasts, 2015. doi: 10.21957/thtpwp67e. URL https://doi.org/10.21957/thtpwp67e.

[138] J. P. Werner, D. V. Divine, F. Charpentier Ljungqvist, T. Nilsen, and P. Francus. Spatio-temporal variability of arctic summer temperatures over the past 2 millennia. *Climate of the Past*, 14(4):527–557, 2018. doi: 10.5194/cp-14-527-2018. URL https://doi.org/10.5194/cp-14-527-2018.

[139] E. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Communications in Pure and Applied Mathematics*, 13(1):1–14, 1960.

[140] R. S. Womersley. *Efficient spherical designs with good geometric properties*, pages 1243–1285. Springer, 2018. ISBN 978-3-319-72456-0. doi: 10.1007/978-3-319-72456-0_57. URL https://doi.org/10.1007/978-3-319-72456-0_57.

[141] R. S. Womersley and I. H. Sloan. How good can polynomial interpolation on the sphere be? *Advances in Computational Mathematics*, 14:195–226, 2001. doi: 10.1023/A:1016630227163. URL https://doi.org/10.1023/A:1016630227163.

[142] T.-T. Wong, W.-S. Luk, and P.-A. Heng. Sampling with Hammersley and Halton points. *Journal of graphics tools*, 2(2):9–24, 1997. doi: 10.1080/10867651.1997.10487471. URL https://doi.org/10.1080/10867651.1997.10487471.

[143] H. Woźniakowski. What is information-based complexity? In *Essays on the complexity of continuous problems*, pages 89–95. Eur. Math. Soc., Zürich, 2009. doi: 10.4171/069-1/5. URL https://doi.org/10.4171/069-1/5.

[144] X. Wu. *Four Dimensional Hybrid Wireless Communication via Coherent Dual-Polarized Antennas*. PhD thesis, University of Notre Dame, 2019. URL https://doi.org/10.7274/cc08hd79s71.

[145] X. Wu, T. Pratt, and T. Fuja. Four dimensional hybrid constellations for dual-polarized wireless communications. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018. doi: 10.1109/ICC.2018.8422468. URL https://doi.org/10.1109/ICC.2018.8422468.

[146] X. Wu, T. G. Pratt, and T. E. Fuja. Hybrid constellations for dual-polarized wireless communications. *IEEE Transactions on Wireless Communications*, 19(8):5321–5332, 2020. doi: 10.1109/TWC.2020.2991991. URL https://doi.org/10.1109/TWC.2020.2991991.

[147] J. Xie, L. Marantis, and P. V. Brennan. Spherical ESPRIT-based 2-D direction finding with different distributions of a spherical array antenna. In *2009 Loughborough Antennas & Propagation Conference*, pages 661–664. IEEE, 2009. doi: 10.1109/LAPC.2009.5352524. URL https://doi.org/10.1109/LAPC.2009.5352524.

[148] Y. M. Zhou. *Arrangements of points on the sphere*. PhD thesis, University of South Florida, Tampa, FL, 1995.

[149] F. Zotter. *Analysis and synthesis of sound-radiation with spherical arrays*. PhD thesis, University of Music and Performing Arts, Austria, 2009. URL https://ambisonics.iem.at/Members/zotter/index_html/publications/2009_Zotter_Diss_SoundRadiationSphericalArrays.pdf.

[150] F. Zotter. Sampling strategies for acoustic holography/holophony on the sphere. *NAG-DAGA, Rotterdam*, pages 1–4, 2009. URL https://old.iem.at/projekte/publications/paper/sampling/sampling.pdf.

J. Approx. Softw., Vol. 1, No. 2 (2024) 2

Journal of Approximation Software

CPOLYMESH: Matlab and Python codes for complex polynomial approximation by Chebyshev admissible meshes

D.J. Kenne [1], A. Sommariva [2], and M. Vianello [2,⋆]

[1] *Doctoral School of Exact and Natural Sciences, Jagiellonian University, Krakow, Poland*
[2] *Department of Mathematics, University of Padova, Italy*

Received: 02/07/2024 – Published: 06/09/2024
Communicated by: R. Cavoretto

Abstract

We provide Matlab and Python codes for polynomial approximation on complex compact sets with connected complement, by Chebyshev-like admissible polynomial meshes on boundaries with piecewise (trigonometric) polynomial parametrization. Such meshes have lower cardinality with respect to those previously known. They are used for polynomial least-squares, for the extraction of extremal interpolation sets of Fekete and Leja type, as well as for the computation of the uniform norms (Lebesgue constants) of polynomial projection operators.

Keywords: complex polynomial approximation, interpolation, least-squares, admissible polynomial meshes, discrete extremal sets, Approximate Fekete Points, Discrete Leja Points, Pseudo Leja Points, Lebesgue constant. (MSC2020: 65D05,65E05)

1 Introduction

In this paper we are concerned with (admissible) *polynomial meshes* $\{Z_n\}_{n\geq 1}$ and polynomial approximation on a complex compact set $K \subset \mathbb{C}$ with connected complement. By the famous Mergelyan Theorem [15], these are sets where any continuous function $f : K \to \mathbb{C}$, with holomorphic restriction to $int(K)$, can be uniformly approximated by polynomials.

Polynomial meshes are sequences of finite subsets $Z_n \subset K$ such that

$$\|p\|_K \leq c\|p\|_{Z_n}\,, \quad \forall p \in \mathbb{P}_n(\mathbb{C})\,, \tag{1}$$

where $\|\cdot\|$ is the uniform norm on a continuous or discrete bounded subset, and p is any polynomial with complex coefficients with degree not exceeding n (we recall that c is usually termed

⋆ Corresponding author: marcov@math.unipd.it

the "constant" of the polynomial mesh). Since Z_n is $\mathbb{P}_n(\mathbb{C})$-determining, i.e. polynomials in $\mathbb{P}_n(\mathbb{C})$ vanishing on Z_n vanish everywhere in $\mathbb{C}$, clearly $card(Z_n) \geq n+1 = dim(\mathbb{P}_n(\mathbb{C}))$. The polynomial mesh is then called *optimal* when $card(Z_n) = O(n)$.

Starting from the seminal paper of 2008 by Calvi and Levenberg [9], polynomial meshes (that can be defined also in multivariate instances) have begun to play a relevant role in polynomial approximation. Among their numerous properties, we may recall that they are preserved by affine transformations, finite union and small perturbations, are well-suited for least-squares approximation and contain extremal subsets of Fekete and Leja type for polynomial interpolation with slowly increasing Lebesgue constant. Moreover, polynomial meshes can be conveniently used for polynomial optimization and Lebesgue constant computation with rigorous interval error bounds. Without any pretence of exhaustivity, we may quote e.g. [2, 4, 3, 6, 5, 16, 17] with the references therein.

Focalizing on the complex univariate case, it has been recently proved in [4] that optimal admissible meshes of Chebyshev type can be constructed on the boundary ∂K, provided that it lies on a union of curves in K having a piecewise polynomial or trigonometric polynomial parametrization. The construction uses the fact that $\|p\|_K = \|p\|_{\partial K}$ by the maximum principle for holomorphic functions (cf. e.g. [14]), and basic polynomial inequalities concerning Chebyshev points on the parameter real interval.

More precisely, let C_N be the set of N Chebyshev zeros in $(-1,1)$, namely $\cos((2j-1)\pi/(2N))$, $1 \leq j \leq N$, or the set of $N+1$ Chebyshev extrema in $[-1,1]$, namely $\cos(j\pi/N)$, $0 \leq j \leq N$. Consider the points

$$\mathscr{C}_\nu^m = \tau(C_N) \subset [a,b] \tag{2}$$

where

$$N = m\nu\ ,\ \ \tau(u) = \frac{b-a}{2}u + \frac{b+a}{2}\ ,\ \ u \in [-1,1]\ , \tag{3}$$

in the algebraic case, and

$$N = 2m\nu\ ,\ \ \tau(u) = 2\arcsin\left(u\sin\left(\frac{b-a}{4}\right)\right) + \frac{b+a}{2}\ ,\ \ u \in [-1,1]\ , \tag{4}$$

in the trigonometric case. Then, the following estimate holds [4]:

Proposition 1. Let $\partial K \subseteq \Gamma = \bigcup_{j=1}^s \Gamma_j \subseteq K$ with parametric algebraic or trigonometric arcs $\Gamma_j = \gamma_j([a_j,b_j])$ of degree $d_j = \max\{deg\,Re(\gamma_j), deg\,Im(\gamma_j)\}$, $1 \leq j \leq s$ (where the angle intervals possibly are sub-intervals of the period in the trigonometric case, namely $b_j - a_j \leq 2\pi$). Then for every $p \in \mathbb{P}_n(\mathbb{C})$, $n \geq 1$, $m > 1$

$$\|p\|_K = \|p\|_\Gamma \leq c_m \|p\|_{Z_n^m}\ ,\ \ Z_n^m = \bigcup_{j=1}^s \gamma_j(\mathscr{C}_{nd_j}^m)\ ,\ \ c_m = \frac{1}{\cos(\pi/(2m))}\ , \tag{5}$$

i.e. $\{Z_n^m\}_{n\geq 1}$ is an admissible polynomial mesh for K with constant c_m.

Estimate (5) is a cornerstone of our code for complex polynomial approximation. Notice that the class of domains with connected complement and such boundaries is very wide: it includes linear polygons, as well as curvilinear polygons with boundary tracked by splines, or by polar arcs like $\gamma_j(t) = z_0 + r_j(t)(\cos(t) + i\sin(t))$ with $r_j(t)$ a trigonometric polynomial. See the Figures below for some illustrative examples. The corresponding meshes have $O(mn)$ cardinality, that asymptotically improves the $O(n^2)$ cardinality of previously known meshes on any connected compact set of $\mathbb{C}$ whose boundary is a C^1 parametric curve with bounded tangent vectors, cf. [1].

Observe that $c_m \to 1$ as $m \to \infty$. This fact is at the base of a computable interval estimate of the Lebesgue constant (uniform operator norm) of any linear projection operator $L_n : C(K) \to \mathbb{P}_n(\mathbb{C})$ of the form

$$L_n f(z) = \sum_{j=1}^{M} f(\xi_j)\, \phi_j(z)\,, \tag{6}$$

where $\{\xi_j\} \subset K$ and $\{\phi_j\}$ is a set of generators of $\mathbb{P}_n(\mathbb{C})$. We recall that such an operator structure holds for polynomial interpolation at $M = n+1$ distinct nodes, where the $\phi_j(z)$ are the corresponding cardinal Lagrange polynomials, but also by polynomial least-squares at $M > n+1$ sampling nodes (cf. [4]). In both cases we simply have

$$\phi_j(z) = K_n(z, \xi_j) = \sum_{k=1}^{n+1} q_k(z)\overline{q_k(\xi_j)}\,, \tag{7}$$

where K_n is the reproducing kernel of the discrete scalar product with unit weights supported at the sampling nodes $\{\xi_j\}$ and $\{q_k\}$ a discrete orthogonal polynomial basis. Given any polynomial basis of $\mathbb{P}_n(\mathbb{C})$, say $\{p_k\}$, we recall that a discrete orthogonal basis can be computed in principle by a *QR* factorization of the corresponding Vandermonde-like interpolation matrix, as

$$[q_1(z), \ldots, q_{n+1}(z)] = [p_1(z), \cdots, p_{n+1}(z)]R^{-1}\,. \tag{8}$$

The following result concerning Lebesgue constants has been proved in [4]:

Proposition 2. Let $\lambda_n(z) = \sum_{j=1}^{M} |\phi_j(z)|$, $z \in K$, be the "Lebesgue function" of L_n in (6) and $\{Z_n^m\}$ the polynomial mesh of Proposition 1. Then for every $n \geq 1$, $m > 1$, the following inequalities hold

$$\|\lambda_n\|_{Z_n^m} \leq \|L_n\| \leq c_m \|\lambda_n\|_{Z_n^m}\,, \tag{9}$$

$$0 \leq \|L_n\| - \|\lambda_n\|_{Z_n^m} \leq (c_m - 1)\|L_n\|\,, \tag{10}$$

for the Lebesgue constant $\|L_n\| = \|\lambda_n\|_K = \|\lambda_n\|_\Gamma$.

We observe that

$$c_m - 1 = \frac{1 - \cos(\pi/(2m))}{\cos(\pi/(2m))} \sim \frac{\pi^2}{8m^2} \approx \frac{1.23}{m^2}\,, \tag{11}$$

that is $\|\lambda_n\|_{Z_n^m}$ by (10) is a $O(1/m^2)$ relative approximation of the Lebesgue constant: for $m = 4$ we already get the Lebesgue constant with an error less than 10%, i.e. we can substantially evaluate its actual order of magnitude.

2 Description of the code

After the above summary of the main theoretical results and estimates underlying the complex polynomial approximation algorithms, we can now briefly describe the code implemented in Matlab and Python and available at https://github.com/alvisesommariva/CPOLYMESH. All the main computations are performed by basic numerical linear algebra subroutines. The main functions are:

- **Polynomial Mesh Constructor**

  Function `Cpom`

This function, given the parametrization intervals $[a_j, b_j]$ and the corresponding curve components, namely the complex algebraic or trigonometric polynomials $\gamma_j(t)$ of degree d_j, computes the Chebyshev submeshes $\gamma_j(\mathscr{C}^m_{nd_j})$ and their union

$$Z^m_n = \bigcup_{j=1}^{s} \gamma_j(\mathscr{C}^m_{nd_j})$$

as in Proposition 1. Linear and curvilinear polygons defined by spline arcs are included, as well as trigonometric polar arcs like $\gamma_j(t) = z_0 + r_j(t)(\cos(t) + i\sin(t))$, with $r_j(t)$ a real trigonometric polynomial on a subinterval of the period.

- **Stabilized Vandermonde Matrix Constructor**

  Function `Cvand`

  Constructs a rectangular Vandermonde-like matrix

  $$V_n(X) = (p_j(z_i)) \ , \ 1 \le i \le card(X) \ , \ 1 \le j \le n+1 \ ,$$

  on a complex set $X = \{z_i\}$. In order to cope with the extreme ill-conditioning of the Vandermonde matrices with the standard monomial basis, we have chosen to work with a shifted and scaled basis

  $$p_j(z) = ((z - z_b)/\delta)^{j-1} \ , \ 1 \le j \le n+1 \ ,$$

  where $z_b = \frac{1}{card(X)} \sum_{z_i \in X} z_i$ is the barycenter of the points and $\delta = \max_{z_i \in X} |z_i - z_b|$ the radius of an enclosing disk. If an enclosing disk is already known, its center z_b and radius δ can be directly passed as input parameters.

- **Discrete Orthogonal Polynomials Constructor and Evaluator**

  Functions `Cdop` and `Cdopeval`

  `Cdop` computes a discrete orthogonal polynomial basis on a finite complex set X with $card(X) \ge n+1$, and `Cdopeval` evaluates the orthogonal basis on a target complex set Y. Orthogonalization is performed by applying twice a QR factorization with unitary Q and square triangular factor R, namely

  $$V_n(X) = Q_1 R_1 \ , \ V_n(X)/R_1 = Q_2 R_2$$

  following the well-known "twice is enough" orthogonalization rule in finite precision arithmetic [10]. The target matrix is

  $$W_n(Y) = (V_n(Y)/R_1)/R_2$$

  where the matrix operator / is preferred to `inv` in order to automatically cope with possible ill-conditioning of the triangular factors.

- **Discrete Extremal Sets Constructor**

  Function `Cdes`

  Computes three interpolation pointsets corresponding to a greedy maximization of the Vandermonde determinant modulus on the polynomial mesh Z^m_n. We do not discuss their

features in detail here and refer to the quoted literature for the underlying theoretical and computational issues.

- AFP (Approximate Fekete Points): after a call to `Cdop` with $X = Z_n^m$ to get a better conditioned matrix, AFP are obtained by a QR factorization with column pivoting of the adjoint Q_2^H, taking the points in Z_n^m corresponding to the first $n+1$ elements of the column permutation vector. They do not form a sequence, but typically have the lowest Lebesgue constant among the three sets; cf. [6, 5, 8, 19].

- DLP (Discrete Leja Points): again after a call to `Cdop` with $X = Z_n^m$, DLP are obtained by a LU factorization with row pivoting of Q_2, taking the points in Z_n^m corresponding to the first $n+1$ elements of the row permutation vector. They form a sequence and in the present univariate complex instance are substantially equivalent to the iteration

$$\xi_j = argmax_{z \in Z_n^m} \prod_{k=1}^{j-1} |z - \xi_k| \ , \quad j = 2, \ldots, n+1 \ ,$$

after choosing ξ_1 as the point that maximizes the element modulus in the first column of Q_2; cf. [5, 7].

- PLP (Pseudo Leja Points): these are a sequence obtained by the iteration

$$\xi_j = argmax_{z \in Z_{j-1}^m} \prod_{k=1}^{j-1} |z - \xi_k| \ , \quad j = 2, \ldots, n+1 \ ,$$

after choosing the first point ξ_1 arbitrarily, e.g. ξ_1 is one of the points in Z_1^m with largest imaginary component; cf. [1] (and [11] for a multivariate extension).

- **Polynomial Projectors** (either Interpolation or Least-Squares)

  Function `Cfit`

  Given a sample column array $\mathbf{f} = f(X)$ of a function at a finite complex set X with $card(X) \geq n+1$, computes the polynomial projector coefficients in an orthogonal polynomial basis at X and evaluates the projector $L_n f$ at a target complex set Y. In view of (7)-(8) the computation is simply

  $$L_n f(Y) = W_n(Y) Q_2^H \mathbf{f}$$

  after a call to `Cdop` on X and `Cdopeval` on Y.

- **Lebesgue Constant Evaluator**

  Function `Cleb`

  Computes on a control set Z the maximum of the Lebesgue function of interpolation on a set X with $card(X) = n+1$ or least-squares with $card(X) > n+1$, as

  $$\|\lambda_n\|_Z = \|W_n(Z) Q_2^H\|_\infty = \|((V_n(Z)/R_1)/R_2) Q_2^H\|_\infty$$

  by a call to `Cdop` on X and to `Cdopeval` with $Y = Z$. In view of Proposition 2, choosing $Z = Z_n^m$ produced by a call to `Cpom` one gets the certified interval estimate (9) for the Lebesgue constant of X.

3 Numerical tests and demos

In this section we present several numerical tests for the code CPOLYMESH. All the corresponding demos are available at [12, 13], along with a number of other examples.

The compact sets considered, say $K_1, \ldots, K_9$, with boundary defined by algebraic or trigonometric polynomial arcs, appear in Figure 2. For the sake of clarity and brevity we do not plot Leja-like interpolation points, whose structure however is not much different from that of Fekete-like points. The figures below have been obtained by the Matlab package [12].

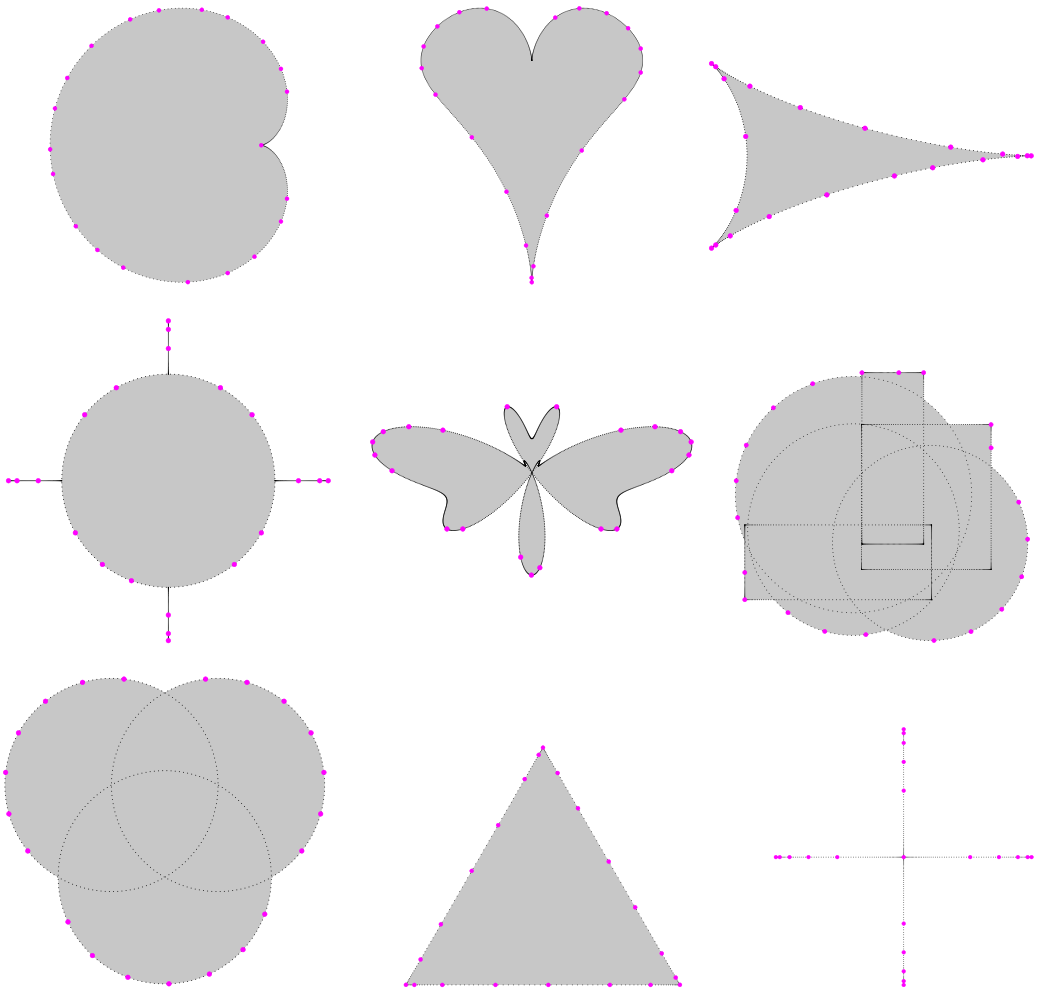


Figure 1: Exemplifying the variety of feasible (curvi)linear polygons: Approximate Fekete Points (magenta dots) for polynomial interpolation at degree $n = 20$ via Chebyshev admissible meshes (black dots) of the piecewise polynomial or trigonometric boundary, with $m = 2$ (cf. Proposition 1).

In particular, the compact sets are:

- a cardioid K_1 where ∂K_1 is defined parametrically as $z(t) = \cos(t)(1-\cos(t)+i(\sin(t)(1-\cos(t))))$, with $t \in [0, 2\pi]$;
- the "Laporte heart" K_2 where ∂K_2 is defined parametrically as $z(t) = \sin^3(t) + i(\cos(t) - \cos^4(t))$, with $t \in [0, 2\pi]$;
- the deltoid K_3 where ∂K_3 is defined parametrically as $z(t) = 10\exp(it) + 5\exp(-2it)$, with $t \in [0, 2\pi]$;

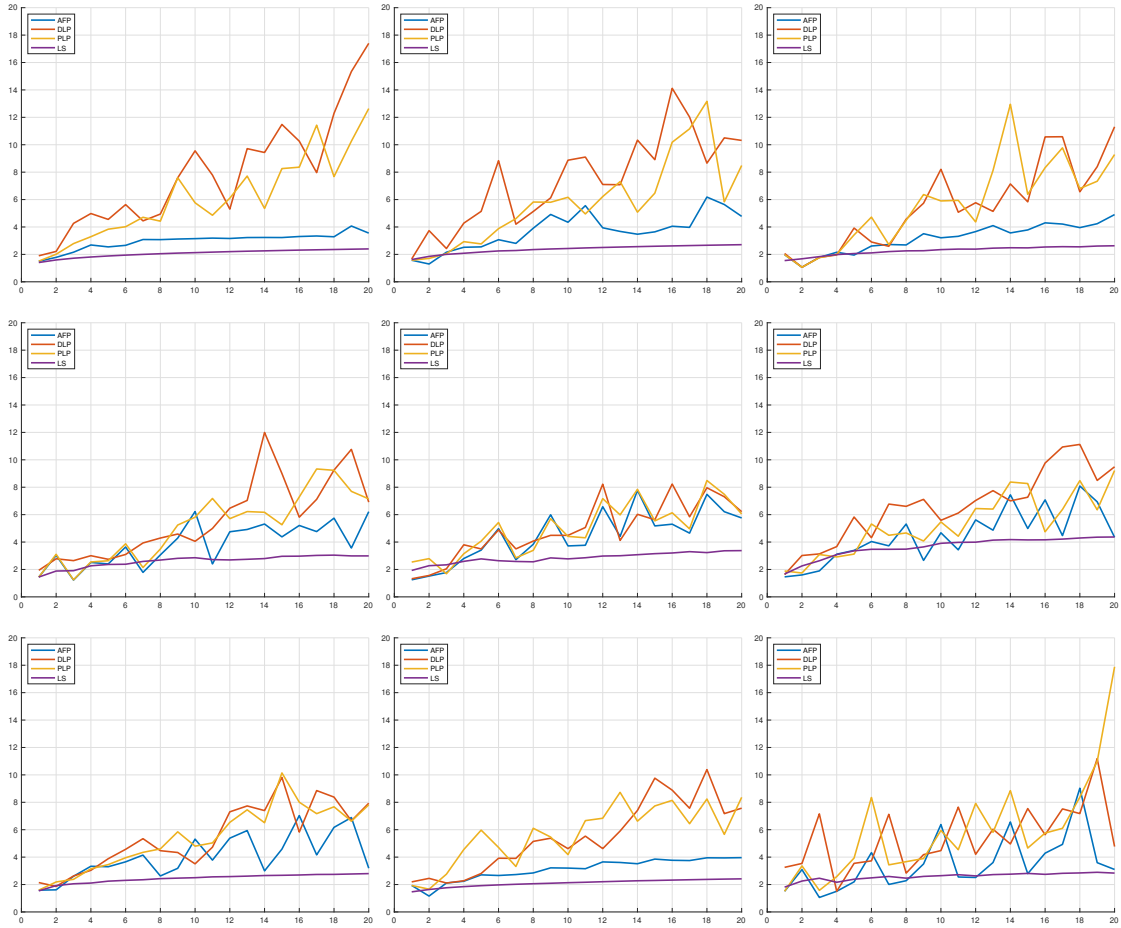


Figure 2: Lebesgue constants for the extremal interpolation sets AFP, DLP, PLP, and for LS approximation, with degrees $n = 1, 2, \ldots, 20$, on the domains of Fig. 1 in the same order.

- a sun shaped domain K_4 obtained as union of the unit-disk with 4 segments of length 0.5;
- the "Sautereau butterfly" K_5 where ∂K_5 is defined parametrically as $z(t) = (-3\cos(2t) + \sin(7t) - 1)\exp(it)$, with $t \in [0, 2\pi]$;
- the domain K_6 obtained as union of 3 random rectangles and 3 disks;
- the "Borromean-circles" domain K_7, that is the union of three disks with radii equal to $\sqrt{3}$ and centers $\exp(it_k)$ with $t_k = \frac{(4k-3)\pi}{6}$, $k = 1, 2, 3$;
- an equilateral triangle K_8 with vertices $P_k = \exp(it_k)$ with $t_k = \frac{(4k-1)\pi}{6}$, $k = 1, 2, 3$;
- a symmetric cross K_9 given as the union of 4 orthogonal unit segments with a common extremum.

We stress the variety of "(curvi)linear polygons" appearing in these examples: some domains are closure of open connected sets, other have components with no internal points. Some have a boundary corresponding to a single parametric curve, other have a complicated boundary being the union of simpler elements. In the latter case, there is no need to track accurately the boundary of the union (that in some situations could be a difficult task), since we can simply take a union admissible mesh for the union of the element boundaries. Notice also that the extremal interpolation points tend to concentrate on outward tips, cusps, angles or convex

portions of the boundary and to avoid inward and concave ones (a well-known electrostatic charges-like behavior, connected with their potential theoretic background, cf. [18]).

In Figure 2 we report the Lebesgue constants of interpolation at discrete extremal sets of Fekete and Leja type, and of Least Squares approximation. We can observe that all the Lebesgue constants exhibit an apparently sub-exponential average increase in the present degree range, in line with the corresponding continuous extremal sets, cf. e.g. [9, 20]. However, Leja-like points have tendentially a more erratic behavior with larger oscillations and tendentially higher values with respect to approximate Fekete points (a phenomenon already observed for example in the multivariate framework of [7]). On the contrary, Lebesgue constants of Least Squares approximation on the whole polynomial mesh have the lowest values, with an apparently logarithmic-like behavior.

3.1 Demos summary

3.1.1 Matlab package

The Matlab package includes two demos, that we briefly comment.

1. `demo_cdes_1`: by this routine we show how to
   - define the complex domain (several ways),
   - compute an admissible mesh (AM) of a fixed degree,
   - extract extremal sets,
   - compute a certified Lebesgue constant.
2. `demo_cdes_2`: by this routine we perform all batteries of numerical tests that are described above.

   In particular, varying the degrees, we
   - compute an admissible mesh (AM) of a fixed degree,
   - extract the AFP, DLP, PLP extremal sets,
   - compute for each of them a certified Lebesgue constant,
   - plot domain, extremal points and Lebesgue contants.

   Changing the value of the variable `domain_type`, from 1 to 22, one can test our routines also on other complex geometries, like hypocycloids, epicycloids, epitrochoids, limacons, rhodoneas, eggs, bifoliums, Talbot curves, tricuspoids, torpedos, ellipses and an alternative heart-shaped domain.

All the codes work also in GNU Octave. The only note is that to analyse curvilinear domains it is necessary to have installed the `spline` toolbox (see https://gnu-octave.github.io/packages/).

3.1.2 Python package

The Python package features a primary demo as well as four additional ones that are contained within subpackages. These subpackages are named according to the points outlined in Section 2. We briefly comment these demos.

1. `demo`: This is the primary routine of the Python package, performing identical tasks to `demo_cdes_2` available from the Matlab version.

2. `demo_cleb`: This demo is included in *"lebesgue_constant_evaluator"* subpackage and performs the same functions as `demo_cdes_1` from the Matlab package.

3. `demo_cfit`: This demo is included in *"polynomial_projectors"* subpackage and performs the following tasks by varying the degrees:

   - compute an admissible mesh (AM) for each degree,
   - extract the AFP, DLP, PLP extremal sets,
   - compute the polynomial of interpolation associated with each extremal set and compute the discrete least squares polynomial fitting the whole admissible mesh,
   - compute the errors of approximation in the supremum norm by varying the polynomial degrees,
   - plot the extremal points for the highest degree and the errors of approximation.

4. `demo_cdes`: This demo is included in *"discrete_extremal_sets_constructor"* subpackage and performs the following tasks:

   - compute an admissible mesh (AM) of a fixed degree,
   - extract the AFP, DLP, PLP extremal sets,
   - plot the extremal points on separate figures.

5. `demo_cpom`: This demo is included in the *"polynomial_mesh_constructor"* subpackage. It computes the admissible mesh for a fixed degree and plots the points on a figure.

When testing our routines, changing the values inside the python function `define_domain()`, from 0 to 31, will lead to the creation of complex polynomial curves such as: 0. Unit circle, 1. Segment [-1,1], 2. Polygon M, 3. Sun, 4. Ellipse, 5. Union of circles, 6. Lune, 7. Cardioid, 8. 4 lenses, 9. Curve polygon, 10. Limacon, 11. Lissajous, 12. Egg, 13. Rhodonea, 14. Habenicht clover, 15. Bifolium, 16. Torpedo, 17. Double egg, 18. Sautereau butterfly 1, 19. Sautereau Butterfly 2, 20. Borromean circles, 23. Laporte heart, 24. Epicycloid, 25. Epitrochoid, 26. Hypocycloid, 27. Nephroid, 28. Talbot curve, 29. Tricuspoid, 30. Rectangles+circles, 31. Equilateral triangle.

Acknowledgements

Work partially supported by the DOR funds of the University of Padova, by the INdAM-GNCS 2024 Project “Kernel and polynomial methods for approximation and integration: theory and application software” (A. Sommariva and M. Vianello) and by the National Science Center - Poland, grant Preludium Bis 1, N. 2019/35/O/ST1/02245 (D.J. Kenne). The research cooperation was funded by the program Excellence Initiative – Research University at the Jagiellonian University in Krakòw (A. Sommariva).

This research has been accomplished within the RITA “Research ITalian network on Approximation", the UMI Group TAA “Approximation Theory and Applications", and the SIMAI Activity Group ANA&A.

We thank Leokadia Białas-Cież for her precious help in developing this work.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Białas-Cież and J.-P. Calvi. Pseudo Leja sequences. *Ann. Mat. Pura Appl. (4)*, 191 (1):53–75, 2012. ISSN 0373-3114,1618-1891. doi: 10.1007/s10231-010-0174-x. URL https://doi.org/10.1007/s10231-010-0174-x.

[2] L. Białas-Cież and J.-P. Calvi. Invariance of polynomial inequalities under polynomial maps. *J. Math. Anal. Appl.*, 439(2):449–464, 2016. ISSN 0022-247X,1096-0813. doi: 10.1016/j.jmaa.2016.03.003. URL https://doi.org/10.1016/j.jmaa.2016.03.003.

[3] L. Białas-Cież, D. J. Kenne, A. Sommariva, and M. Vianello. Evaluating lebesgue constants by chebyshev polynomial meshes on cube, simplex and ball, 2023. URL https://arxiv.org/abs/2311.18656.

[4] L. Białas-Cież, D. J. Kenne, A. Sommariva, and M. Vianello. Chebyshev admissible meshes and Lebesgue constants of complex polynomial projections. *J. Comput. Appl. Math.*, 443:Paper No. 115766, 7, 2024. ISSN 0377-0427,1879-1778. doi: 10.1016/j.cam.2024.115766. URL https://doi.org/10.1016/j.cam.2024.115766.

[5] L. Bos, S. De Marchi, A. Sommariva, and M. Vianello. Computing multivariate Fekete and Leja points by numerical linear algebra. *SIAM J. Numer. Anal.*, 48(5):1984–1999, 2010. ISSN 0036-1429,1095-7170. doi: 10.1137/090779024. URL https://doi.org/10.1137/090779024.

[6] L. Bos, J.-P. Calvi, N. Levenberg, A. Sommariva, and M. Vianello. Geometric weakly admissible meshes, discrete least squares approximations and approximate Fekete points. *Math. Comp.*, 80(275):1623–1638, 2011. ISSN 0025-5718,1088-6842. doi: 10.1090/S0025-5718-2011-02442-7. URL https://doi.org/10.1090/S0025-5718-2011-02442-7.

[7] L. Bos, S. De Marchi, A. Sommariva, and M. Vianello. Weakly admissible meshes and discrete extremal sets. *Numer. Math. Theory Methods Appl.*, 4(1):1–12, 2011. ISSN 1004-8979,2079-7338. doi: 10.4208/nmtma.2011.m1003. URL https://doi.org/10.4208/nmtma.2011.m1003.

[8] L. P. Bos and N. Levenberg. On the calculation of approximate fekete points: the univariate case. *Electron. Trans. Numer. Anal.*, 30:377–397, 2008.

[9] J.-P. Calvi and N. Levenberg. Uniform approximation by discrete least squares polynomials. *J. Approx. Theory*, 152(1):82–100, 2008. ISSN 0021-9045,1096-0430. doi: 10.1016/j.jat.2007.05.005. URL https://doi.org/10.1016/j.jat.2007.05.005.

[10] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof. Rounding error analysis of the classical Gram-Schmidt orthogonalization process. *Numer. Math.*, 101(1):87–100, 2005. ISSN 0029-599X,0945-3245. doi: 10.1007/s00211-005-0615-4. URL https://doi.org/10.1007/s00211-005-0615-4.

[11] D. J. Kenne. Multidimensional pseudo-leja sequences, 2023. URL https://arxiv.org/abs/2303.11871.

[12] D. J. Kenne, A. Sommariva, and M. Vianello. Cpolymesh, matlab codes for complex polynomial approximation by chebyshev admissible meshes. https://github.com/alvisesommariva/CPOLYMESH, 2024.

[13] D. J. Kenne, A. Sommariva, and M. Vianello. Cpolymesh, python codes for complex polynomial approximation by chebyshev admissible meshes. https://github.com/DimitriKenne/CPOLYMESH, 2024.

[14] S. G. Krantz. *A guide to complex variables*, volume 32 of *The Dolciani Mathematical Expositions*. Mathematical Association of America, Washington, DC, 2008. ISBN 978-0-88385-338-2. MAA Guides, 1.

[15] S. N. Mergelyan. Uniform approximations to functions of a complex variable. *Amer. Math. Soc. Translation*, 1954(101):99, 1954.

[16] F. Piazzon and M. Vianello. Small perturbations of polynomial meshes. *Appl. Anal.*, 92(5):1063–1073, 2013. ISSN 0003-6811,1563-504X. doi: 10.1080/00036811.2011.649730. URL https://doi.org/10.1080/00036811.2011.649730.

[17] F. Piazzon and M. Vianello. A note on total degree polynomial optimization by Chebyshev grids. *Optim. Lett.*, 12(1):63–71, 2018. ISSN 1862-4472,1862-4480. doi: 10.1007/s11590-017-1166-1. URL https://doi.org/10.1007/s11590-017-1166-1.

[18] E. B. Saff and V. Totik. *Logarithmic potentials with external fields*, volume 316 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1997. ISBN 3-540-57078-0. doi: 10.1007/978-3-662-03329-6. URL https://doi.org/10.1007/978-3-662-03329-6. Appendix B by Thomas Bloom.

[19] A. Sommariva and M. Vianello. Computing approximate Fekete points by QR factorizations of Vandermonde matrices. *Comput. Math. Appl.*, 57(8):1324–1336, 2009. ISSN 0898-1221,1873-7668. doi: 10.1016/j.camwa.2008.11.011. URL https://doi.org/10.1016/j.camwa.2008.11.011.

[20] V. Totik. The Lebesgue constants for Leja points are subexponential. *J. Approx. Theory*, 287:Paper No. 105863, 15, 2023. ISSN 0021-9045,1096-0430. doi: 10.1016/j.jat.2023.105863. URL https://doi.org/10.1016/j.jat.2023.105863.

J. Approx. Softw., Vol. 1, No. 2 (2024) 3

Journal of Approximation Software

Exponential integrators for a mean-field selective optimal control problem

G. Albi [1], M. Caliari [1,⋆], E. Calzola [1], and F. Cassini [1]

[1] *Department of Computer Science, University of Verona, Italy*

Received: 04/07/2024 – Published: 06/09/2024
Communicated by: M. Vianello

Dedicated to Professor Leonard Peter Bos on the occasion of his 70th birthday

Abstract

In this paper we consider a mean-field optimal control problem with selective action of the control, where the constraint is a continuity equation involving a non-local term and diffusion. First order optimality conditions are formally derived in a general framework, accounting for boundary conditions. Hence, the optimality system is used to construct a reduced gradient method, where we introduce a novel algorithm for the numerical realization of the forward and the backward equations, based on exponential integrators. We illustrate extensive numerical experiments on different control problems for collective motion in the context of opinion formation, pedestrian dynamics, and mass transfer.

Keywords: mean-field control, multi-agent systems, PDE-constrained optimization, exponential integrators (MSC2020: 65M22, 49M41, 93A16)

1 Introduction

The study of collective motion of interacting agents systems is of paramount importance to understand the formation of coherent global behaviors at various scales, with applications to the study of biological, social, and economic phenomena. In recent years, there has been a surge of literature on the collective behavior of multi-agent systems, covering a wide range of topics such as cell aggregation and motility, coordinated animal motion [28, 30], opinion formation [36, 45, 55], coordinated human behavior [10, 27, 31, 48], and cooperative robots [26, 34, 46, 47]. These fields are vast and constantly evolving and we refer to the surveys [8, 29, 39] that provide a comprehensive overview of recent developments. Modeling such complex and diverse systems poses a significant challenge, since in general there are no first-principles

⋆ Corresponding author: marco.caliari@univr.it

as, for instance, in classical physics, or statistical mechanics. Nevertheless, the dynamics of the individuals have been successfully described by systems of Ordinary Differential Equations (ODEs) from Newton's laws designing basic interaction rules, such as attraction, repulsion, and alignments, or, alternatively, by considering an evolutive game where the dynamics is driven by the simultaneous optimization of costs by N players such as in References [42, 43]. In this context, of paramount importance for several applications is the design of centralized policies able to optimally enforce a desired state of the agents, see for instance References [5, 6, 23].

In this paper, we consider a constrained setting, where interacting individuals are influenced by a centralized control with selective action, i.e.,

$$\mathrm{d}x_i = \left(\frac{1}{N} \sum_{j=1}^{N} p(x_i, x_j)(x_j - x_i) + s(t, x_i, \rho^N) u_i \right) \mathrm{d}t + \sigma \mathrm{d}W_i^t, \tag{1}$$

with initial data $x^0 = [x_1^0, \dots, x_N^0]$. Here, each agent $x_i \in \Omega \subseteq \mathbb{R}^d$, for $i = 1, \dots, N$, accounts for pairwise interactions weighted by the function $p(\cdot,\cdot)$, and for disturbances modelled with a Brownian motion $\mathrm{d}W_i^t$. The action of the control $u = [u_1, \dots, u_N]$, $u_i \in \mathbb{R}^d$, is weighted by a selective function $s(t, x_i, \rho^N)$, with $\rho^N(x)$ the empirical measures associated to the interacting agent system, i.e., $\rho^N(t,x) = N^{-1} \sum_{i=1}^{N} \delta(x_i(t) - x)$. Then, the optimal control is obtained by minimizing the cost functional

$$J(u; x^0) = \mathbb{E}\left[\int_0^T \frac{1}{2N} \sum_{i=1}^{N} \left(\ell(t, x_i, \rho^N) + \gamma |u_i|^2 \right) \right], \tag{2}$$

where $\ell(t, x_i, f^N)$ is a running cost to be designed by the controller, with a quadratic penalization of the control for $\gamma \geq 0$.

For a large number of agents, we can write the mean-field optimal control problem corresponding to the finite dimensional optimal control problem (1)–(2) as follows (see References [11, 32, 33])

$$\min_{u \in U} \frac{1}{2} \int_0^T \int_\Omega \left(\ell(t, x, \rho) + \gamma |u|^2 \right) \rho \, dx dt. \tag{3a}$$

Here, U is the space of admissible controls and ρ is the density function satisfying the Partial Differential Equation (PDE)

$$\begin{cases} \partial_t \rho + \nabla \cdot \left(\left(\mathscr{P}(\rho) + s(t, x, \rho) u \right) \rho \right) - \dfrac{\sigma^2}{2} \Delta \rho = 0, \\ \rho(0, x) = \rho_0(x). \end{cases} \tag{3b}$$

The non-local interactions among agents are described by the integral term

$$\mathscr{P}(\rho)(t, x) = \int_\Omega p(x, y)(y - x) \rho(t, y) dy \tag{4}$$

and $\rho_0(x)$ is the initial distribution of the agents. Differently from mean-field games [1, 22, 43], in this context the goal is to compute a mean-field optimal strategy capable of driving the population density to a specific target, avoiding the curse of dimensionality induced by the large scale non-linear system of N agents. However, the numerical solution of the PDE-constrained optimization problem (3a)–(3b) requires careful treatment, see Reference [15]. In mean-field optimal control problems, various methodologies have been utilized to tackle the synthesis of high-dimensional systems, as seen in References [2, 24, 33, 49]. In this study, we adopt a

reduced gradient method strategy where the first order optimality system is solved iteratively for the realization of the control, as in References [7, 12]. Major challenges arise from the presence of the stiff diffusive and transport operators, and from the stability and storage requirements originated by the choice of the numerical solvers. For these kinds of problems, explicit time marching schemes usually require several time steps due to the lack of favorable stability properties, while implicit ones need possibly expensive solutions of non-linear systems (see References [9, 37], where implicit multistep or Runge–Kutta methods are employed) or linear systems (see Reference [40], where IMEX methods are proposed). A prominent and effective alternative way to numerically integrate stiff equations in time is to employ explicit *exponential integrators*, see Reference [41] for a seminal review. After semidiscretization in space, these schemes require to approximate the action of exponential and of exponential-like matrix functions (the so-called φ-functions), in contrast to the solution of (non-)linear systems.

The paper is structured as follows. In Section 2 we present a model of interest which generalizes the one in formulas (3), and we derive the formal optimality conditions using the associated Lagrangian function, obtaining a system of coupled PDEs. The first one is forward in time for the density function, while the second is backward in time for the adjoint variable. We numerically couple these equations using the steepest descent algorithm. In Section 3 we present the semidiscretization in space of the forward and of the backward PDEs, together with the numerical solution of the arising systems of ODEs using a pair of exponential integrators. For convenience of the reader, we also present there the derivation of the schemes and a brief discussion on common techniques to compute the involved matrix functions. Section 4 is devoted to some numerical validations and simulations in opinion formation (Sznajd, Hegselmann–Krause, and mass transfer) and pedestrian (see Reference [16]) models. We finally draw some conclusions in Section 5.

2 Mean-field selective optimal control problem

We consider the mean-field optimal control problem [7, 16, 32] defined by the functional minimization

$$\min_{u\in U} \mathscr{J}(u;\rho_0), \tag{5a}$$

where $\rho = \rho(t,x) \in \mathbb{R}$ is a probability density of agents satisfying

$$\begin{cases} \partial_t \rho + \nabla\cdot\left[\left(\mathscr{P}(\rho) + s(t,x,\rho)u\right)\rho\right] - \dfrac{\sigma^2}{2}\Delta\rho = 0, \\ \rho(0,x) = \rho_0(x), \\ \left(\left(\mathscr{P}(\rho) + s(t,x,\rho)u\right)\rho - \dfrac{\sigma^2}{2}\nabla\rho\right)\cdot\vec{n} = \begin{cases} \beta\rho & \text{on } \Gamma_F, \\ 0 & \text{on } \Gamma_Z \end{cases} \end{cases} \tag{5b}$$

and defined for each $(t,x) \in [0,T]\times\Omega$. The evolution of the density is driven by the non-local operator $\mathscr{P}(\rho)(t,x) \in \mathbb{R}^d$, as in equation (4), and by the control $u = u(t,x) \in \mathbb{R}^d$ weighted by the selective function $s(t,x,\rho) \in \mathbb{R}$. Here, we denoted by Γ_F the subset of the boundary in which there is a flux different from zero ($\beta \neq 0$) and by Γ_Z the part of $\partial\Omega$ with zero-flux boundary conditions. These two subsets are such that $\Gamma_F \cup \Gamma_Z = \partial\Omega$ and $\Gamma_F \cap \Gamma_Z = \emptyset$, and $\vec{n}$ is the outward normal vector to the boundary with norm equal to one. Finally, the functional in formula (5a) is given by

$$\mathscr{J}(u;\rho_0) = \frac{1}{2}\int_0^T \int_\Omega \left(e(t,x,\rho) + \gamma|u|^2\rho\right) dxdt + \frac{1}{2}\int_\Omega c(T,x,\rho(T,x))dx$$

for a general running cost $e(t,x,\rho) \in \mathbb{R}$ and a terminal cost $c(T,x,\rho(T,x)) \in \mathbb{R}$.

2.1 First order optimality conditions

We can derive the first order optimality conditions on a formal level using a Lagrangian approach. For a rigorous treatment we refer to References [7, 17]. We define the Lagrangian function with adjoint variable ψ as

$$\begin{aligned}\mathscr{L}(u,\rho,\psi) =& \frac{1}{2}\int_0^T \int_\Omega \left(e(t,x,\rho) + \gamma|u|^2\rho\right) dxdt + \frac{1}{2}\int_\Omega c(T,x,\rho(T,x))dx \\ &- \int_0^T \int_\Omega \psi \left(\partial_t\rho + \nabla\cdot\left[\left(\mathscr{P}(\rho) + s(t,x,\rho)u\right)\rho\right] - \frac{\sigma^2}{2}\Delta\rho\right) dxdt.\end{aligned} \tag{6}$$

The optimal solution (u^*,ρ^*,ψ^*) can be found by equating to zero the partial Fréchet derivatives of the Lagrangian function, i.e., by solving the following system

$$\begin{cases} D_u\mathscr{L}(u,\rho,\psi) = 0, \\ D_\psi\mathscr{L}(u,\rho,\psi) = 0, \\ D_\rho\mathscr{L}(u,\rho,\psi) = 0. \end{cases} \tag{7}$$

Before computing the partial derivatives in system (7), we integrate by parts the last term appearing in the Lagrangian function (6) and we get

$$\begin{aligned}\mathscr{L}(u,\rho,\psi) =& \frac{1}{2}\int_0^T \int_\Omega \left(e(t,x,\rho) + \gamma|u|^2\rho\right) dxdt + \frac{1}{2}\int_\Omega c(T,x,\rho(T,x))dx \\ &+ \int_0^T \int_\Omega \rho \left(\partial_t\psi + \frac{\sigma^2}{2}\Delta\psi + \left(\mathscr{P}(\rho) + s(t,x,\rho)u\right)\cdot\nabla\psi\right) dxdt \\ &- \int_0^T \int_{\Gamma_F} \rho \left(\frac{\sigma^2}{2}\nabla\psi\cdot\vec{n} + \beta\psi\right) dbdt \\ &- \int_\Omega (\psi(T,x)\rho(T,x) - \psi(0,x)\rho(0,x))dx,\end{aligned}$$

where we used the value of the boundary conditions appearing in equation (5b). Performing then the computations of the partial derivatives we obtain the gradient direction for the control variable u

$$D_u\mathscr{L}(u,\rho,\psi) = \gamma u + s(t,x,\rho)\nabla\psi, \tag{8}$$

the forward PDE for the density function ρ

$$\begin{cases} \partial_t\rho + \nabla\cdot\left[\left(\mathscr{P}(\rho) + s(t,x,\rho)u\right)\rho\right] - \dfrac{\sigma^2}{2}\Delta\rho = 0, \\ \rho(0,x) = \rho_0(x), \\ \left(\left(\mathscr{P}(\rho) + s(t,x,\rho)u\right)\rho - \dfrac{\sigma^2}{2}\nabla\rho\right)\cdot\vec{n} = \begin{cases}\beta\rho & \text{on } \Gamma_F, \\ 0 & \text{on } \Gamma_Z,\end{cases} \end{cases} \tag{9}$$

and the backward PDE for the adjoint variable ψ

$$\begin{cases} -\partial_t \psi = \dfrac{\sigma^2}{2}\Delta\psi + \big(\mathscr{P}(\rho) + (s(t,x,\rho) + \rho D_\rho s(t,x,\rho))u\big)\cdot\nabla\psi \\ \qquad\qquad + \mathscr{Q}(\rho,\psi) + \dfrac{1}{2}(D_\rho e(t,x,\rho) + \gamma|u|^2), \\ \psi(T,x) = \psi_T(x), \\ \dfrac{\sigma^2}{2}\nabla\psi\cdot\vec{n} = \begin{cases} -\beta\psi & \text{on } \Gamma_{\mathrm{F}}, \\ 0 & \text{on } \Gamma_{\mathrm{Z}}, \end{cases} \end{cases} \tag{10}$$

where

$$\mathscr{Q}(\rho,\psi)(t,x) = \int_\Omega p(y,x)(x-y)\cdot\nabla\psi(t,y)\rho(t,y)dy$$

and $\psi_T(x) = \frac{1}{2}D_\rho c(T,x,\rho(T,x))$. Now, in order to solve model (5), we employ a steepest descent approach (see References [7, 12]). Starting with an initial control u^0, at each iteration ℓ we insert u^ℓ into the forward equation (9) and solve it for $\rho = \rho^{\ell+1}$. We then insert u^ℓ and $\rho^{\ell+1}$ into the backward equation (10) and solve it for $\psi = \psi^{\ell+1}$. We finally update the control by using the gradient direction (8), i.e.,

$$u^{\ell+1} = u^\ell - \lambda^\ell(\gamma u^\ell + s(t,x,\rho^{\ell+1})\nabla\psi^{\ell+1})$$

and get $u^{\ell+1}$. We proceed iterating until $\mathscr{J}(u^{\ell+1})$ has stabilized within a given tolerance. For the numerical solution of equations (9) and (10) we use the method of lines: we first discretize in space and then use appropriate integrators for the obtained systems of ODEs.

3 Numerical integrators for the semidiscretized equations

In this section, we explain how to solve the forward and the backward PDEs in the steepest descent algorithm. By observing that both are semilinear parabolic equations, the idea is to use numerical schemes tailored for this type of problems. A prominent way is to apply explicit exponential integrators [41] to the systems of ODEs arising from the semidiscretization in space of the PDEs. By construction, these schemes solve exactly linear ODEs systems with constant coefficients and they allow for time steps usually much larger than those required by classical explicit methods, i.e., typically they do not suffer from a CFL restriction. On the other hand, this class of integrators requires the computation of the action of exponential-like matrix functions for which different efficient techniques have been developed in recent years (see Section 3.3).

3.1 Forward PDE

For the sake of clarity, and since we will present later on one-dimensional numerical examples, we consider $\Omega = [a,b]$ and we rewrite the forward PDE (9)

$$\begin{cases} \partial_t \rho(t,x) = \dfrac{\sigma^2}{2}\partial_{xx}\rho(t,x) - \partial_x\left((\mathscr{P}(\rho(t,\cdot))(t,x) + s(t,x,\rho(t,x))u(t,x))\rho(t,x)\right), \\ \rho(0,x) = \rho_0(x), \\ \left((\mathscr{P}(\rho(t,\cdot))(t,x) + s(t,x,\rho(t,x))u(t,x))\rho(t,x) - \dfrac{\sigma^2}{2}\partial_x\rho(t,x)\right)\Big|_a = \beta_a\rho(t,a), \\ \left((\mathscr{P}(\rho(t,\cdot))(t,x) + s(t,x,\rho(t,x))u(t,x))\rho(t,x) - \dfrac{\sigma^2}{2}\partial_x\rho(t,x)\right)\Big|_b = \beta_b\rho(t,b), \end{cases}$$

where $\beta_a, \beta_b \in \mathbb{R}$ can be selected so that it is possible to express both zero and nonzero fluxes. Notice that when we solve this equation we consider $u(t,x)$ a given function. We introduce a semidiscretization in space by finite differences on a grid of points x_i, with $i = 1,\dots,n$, in such a way that $\boldsymbol{\rho}(t) = [\rho_1(t),\dots,\rho_n(t)]^\mathsf{T}$ is the unknown vector whose components $\rho_i(t)$ approximate $\rho(t,x_i)$. Now, by denoting D_1 and D_2 the matrices which discretize ∂_x and ∂_{xx} at the grid points, respectively, and P the discretization of the linear integral operator $\mathscr{P}$ by a quadrature formula, the linear part of the right hand side of the equation is discretized by

$$\frac{\sigma^2}{2}D_2\boldsymbol{\rho}(t),$$

while the non-linear part becomes

$$\begin{aligned} &-(D_1P\boldsymbol{\rho}(t))\boldsymbol{\rho}(t) - (P\boldsymbol{\rho}(t))(D_1\boldsymbol{\rho}(t)) \\ &\qquad - (D_1\boldsymbol{s}(t,\boldsymbol{\rho}(t)))\boldsymbol{u}(t)\boldsymbol{\rho}(t) - \boldsymbol{s}(t,\boldsymbol{\rho}(t))(D_1\boldsymbol{u}(t))\boldsymbol{\rho}(t) - \boldsymbol{s}(t,\boldsymbol{\rho}(t))\boldsymbol{u}(t)(D_1\boldsymbol{\rho}(t)). \end{aligned}$$

Now, we also discretize the boundary conditions with finite differences by using virtual nodes, and we modify accordingly both the linear part and the non-linear one. The resulting non-linear system of ODEs is then

$$\begin{cases} \boldsymbol{\rho}'(t) = A_\mathrm{F}\boldsymbol{\rho}(t) + \boldsymbol{g}_\mathrm{F}(t,\boldsymbol{\rho}(t)), \quad t \in [0,T], \\ \boldsymbol{\rho}(0) = \boldsymbol{\rho}_0. \end{cases} \tag{11}$$

Given a time discretization $[t_0,\dots,t_k,\dots,t_m]$, with $t_0 = 0$ and $t_m = T$, the exact solution of system (11) at time t_{k+1} can be expressed using the variation-of-constants formula, i.e.,

$$\boldsymbol{\rho}(t_{k+1}) = \mathrm{e}^{\tau_k A_\mathrm{F}}\boldsymbol{\rho}(t_k) + \int_0^{\tau_k} \mathrm{e}^{(\tau_k - s)A_\mathrm{F}}\boldsymbol{g}_\mathrm{F}(t_k + s, \boldsymbol{\rho}(t_k + s))ds,$$

where $\tau_k = t_{k+1} - t_k$, for $k = 0,\dots,m-1$. In order to obtain an explicit first order numerical scheme, we denote by $\boldsymbol{\rho}_k$ the approximation of $\boldsymbol{\rho}(t_k)$ and approximate the non-linear function $\boldsymbol{g}_\mathrm{F}(t_k + s, \boldsymbol{\rho}(t_k + s))$ with $\boldsymbol{g}_\mathrm{F}(t_k, \boldsymbol{\rho}_k)$. Hence, we have

$$\begin{aligned} \boldsymbol{\rho}(t_{k+1}) \approx \boldsymbol{\rho}_{k+1} &= \mathrm{e}^{\tau_k A_\mathrm{F}}\boldsymbol{\rho}_k + \int_0^{\tau_k} \mathrm{e}^{(\tau_k - s)A_\mathrm{F}}\boldsymbol{g}_\mathrm{F}(t_k, \boldsymbol{\rho}_k)ds \\ &= \mathrm{e}^{\tau_k A_\mathrm{F}}\boldsymbol{\rho}_k + \left(\int_0^{\tau_k} \mathrm{e}^{(\tau_k - s)A_\mathrm{F}}ds\right)\boldsymbol{g}_\mathrm{F}(t_k, \boldsymbol{\rho}_k) \\ &= \mathrm{e}^{\tau_k A_\mathrm{F}}\boldsymbol{\rho}_k + \left(\tau_k\int_0^1 \mathrm{e}^{\tau_k(1-\theta)A_\mathrm{F}}d\theta\right)\boldsymbol{g}_\mathrm{F}(t_k, \boldsymbol{\rho}_k) \\ &= \mathrm{e}^{\tau_k A_\mathrm{F}}\boldsymbol{\rho}_k + \tau_k\varphi_1(\tau_k A_\mathrm{F})\boldsymbol{g}_\mathrm{F}(t_k, \boldsymbol{\rho}_k). \end{aligned} \tag{12}$$

Here, we introduced the exponential-like function

$$\varphi_1(X) = \int_0^1 \mathrm{e}^{(1-\theta)X} d\theta,$$

with $X \in \mathbb{C}^{n\times n}$ a generic matrix. This scheme is known as *exponential Euler*, it is an explicit method of first (stiff) order and it is A-stable by construction. Although its implementation does not need the solution of (non-)linear systems, at each time step it is required the evaluation of a linear combination of type $\mathrm{e}^{\tau_k X}\boldsymbol{v}_k + \tau_k\varphi_1(\tau_k X)\boldsymbol{w}_k$, where $\boldsymbol{v}_k, \boldsymbol{w}_k \in \mathbb{C}^n$ are suitable vectors, which we will address in Section 3.3.

3.1.1 Selective function independent of the density

A remarkable occurrence in the literature is $s(t,x,\rho(t,x)) = s(t,x)$, i.e., the selective function does not depend on the density (see Reference [7] for the case $s(t,x) = 1$, which we will also consider in the numerical examples). In this case, the linear part of the forward equation has time dependent coefficients

$$\left(\frac{\sigma^2}{2}D_2 - (D_1\boldsymbol{s}(t))\boldsymbol{u}(t) - \boldsymbol{s}(t)(D_1\boldsymbol{u}(t)) - \boldsymbol{s}(t)\boldsymbol{u}(t)D_1\right)\boldsymbol{\rho}(t),$$

while the non-linear part is now given by

$$-(D_1 P\boldsymbol{\rho}(t))\boldsymbol{\rho}(t) - (P\boldsymbol{\rho}(t))(D_1\boldsymbol{\rho}(t)).$$

By modifying accordingly the quantities to impose the boundary conditions, and using again the previous notation for simplicity, we end up with the system of ODEs

$$\begin{cases} \boldsymbol{\rho}'(t) = A_{\mathrm{F}}(t)\boldsymbol{\rho}(t) + \boldsymbol{g}_{\mathrm{F}}(t,\boldsymbol{\rho}(t)), \quad t \in [0,T], \\ \boldsymbol{\rho}(0) = \boldsymbol{\rho}_0. \end{cases} \tag{13}$$

At each t_k we can rewrite equivalently this system as

$$\begin{cases} \boldsymbol{\rho}'(t) = A_{\mathrm{F}}(t_k)\boldsymbol{\rho}(t) + (A_{\mathrm{F}}(t) - A_{\mathrm{F}}(t_k))\boldsymbol{\rho}(t) + \boldsymbol{g}_{\mathrm{F}}(t,\boldsymbol{\rho}(t)) \\ \qquad\quad = A_{\mathrm{F}}(t_k)\boldsymbol{\rho}(t) + \boldsymbol{g}^k_{\mathrm{F}}(t,\boldsymbol{\rho}(t)), \\ \boldsymbol{\rho}(0) = \boldsymbol{\rho}_0 \end{cases}$$

and apply the exponential Euler method, as done for system (11). Thus, we end up with the scheme

$$\begin{aligned} \boldsymbol{\rho}(t_{k+1}) \approx \boldsymbol{\rho}_{k+1} &= \mathrm{e}^{\tau_k A_{\mathrm{F}}(t_k)}\boldsymbol{\rho}_k + \tau_k\varphi_1(\tau_k A_{\mathrm{F}}(t_k))\boldsymbol{g}^k_{\mathrm{F}}(t_k,\boldsymbol{\rho}_k) \\ &= \mathrm{e}^{\tau_k A_{\mathrm{F}}(t_k)}\boldsymbol{\rho}_k + \tau_k\varphi_1(\tau_k A_{\mathrm{F}}(t_k))\boldsymbol{g}_{\mathrm{F}}(t_k,\boldsymbol{\rho}_k), \end{aligned} \tag{14}$$

for $k = 0,\dots,m-1$. As for the general case $s(t,x,\rho(t,x))$, we obtain in this way an explicit method of first order (which we call *exponential Euler–Magnus*) that requires again to evaluate a linear combination of actions of the matrix exponential and the matrix φ_1 function.

3.2 Backward PDE

We rewrite the backward PDE (10) in the one-dimensional case $\Omega = [a,b]$

$$\begin{cases} -\partial_t \psi(t,x) = \dfrac{\sigma^2}{2}\partial_{xx}\psi(t,x) + \mathscr{P}(\rho(t,\cdot))(t,x)\partial_x\psi(t,x) \\ \qquad\qquad + (s(t,x,\rho(t,x)) + \rho(t,x)s_\rho(t,x,\rho(t,x)))u(t,x)\partial_x\psi(t,x) \\ \qquad\qquad + \mathscr{Q}(\rho(t,\cdot),\psi(t,\cdot))(t,x) + \dfrac{1}{2}\left(e_\rho(t,x,\rho(t,x)) + \gamma u^2(t,x)\right), \\ \psi(T,x) = \psi_T(x), \\ \dfrac{\sigma^2}{2}\partial_x\psi(t,x)\big|_a = -\beta_a\psi(t,a), \\ \dfrac{\sigma^2}{2}\partial_x\psi(t,x)\big|_b = -\beta_b\psi(t,b), \end{cases}$$

where $s_\rho(t,x,\rho(t,x)) = D_\rho s(t,x,\rho(t,x))$ and $e_\rho(t,x,\rho(t,x)) = D_\rho e(t,x,\rho(t,x))$. Here, we assume that $\rho(t,x)$ and $u(t,x)$ are given functions. By applying a finite difference discretization on the same spatial grid as above and defining Q the discretization of the linear integral operator $\mathscr{Q}$, we obtain the time dependent coefficient linear part

$$\left(\frac{\sigma^2}{2}D_2 + P\boldsymbol{\rho}(t)D_1 + (\boldsymbol{s}(t,\boldsymbol{\rho}(t)) + \boldsymbol{\rho}(t)\boldsymbol{s}_{\boldsymbol{\rho}}(t,\boldsymbol{\rho}(t)))\boldsymbol{u}(t)D_1 + Q\boldsymbol{\rho}(t)D_1\right)\boldsymbol{\psi}(t)$$

and the source term

$$\frac{1}{2}\boldsymbol{e}_{\boldsymbol{\rho}}(t,\boldsymbol{\rho}(t)) + \gamma\boldsymbol{u}^2(t).$$

Finally, by taking into consideration boundary conditions, we end up with the inhomogeneous time dependent coefficient linear system of ODEs

$$\begin{cases} -\boldsymbol{\psi}'(t) = A_{\mathrm{B}}(t)\boldsymbol{\psi}(t) + \boldsymbol{g}_{\mathrm{B}}(t), \quad t \in [0,T], \\ \boldsymbol{\psi}(T) = \boldsymbol{\psi}_T. \end{cases} \tag{15}$$

By considering the same time discretization $[t_0,\dots,t_{k+1},\dots,t_m]$ as above, system (15) has a similar structure to system (13). Hence, taking into account that we are marching backward in time, we apply the exponential Euler–Magnus method and we obtain the time marching

$$\boldsymbol{\psi}(t_k) \approx \boldsymbol{\psi}_k = \mathrm{e}^{\tau_k A_{\mathrm{B}}(t_{k+1})}\boldsymbol{\psi}_{k+1} + \tau_k\varphi_1(\tau_k A_{\mathrm{B}}(t_{k+1}))\boldsymbol{g}_{\mathrm{B}}(t_{k+1}), \tag{16}$$

for $k = m-1, m-2, \dots, 0$.

3.3 Matrix functions evaluation

We have introduced two exponential integrators that require the evaluation of

$$\mathrm{e}^{\tau X}\boldsymbol{v} + \tau\varphi_1(\tau X)\boldsymbol{w}, \tag{17}$$

at each time step, where $\tau > 0$, $X \in \mathbb{R}^{n\times n}$, and $\boldsymbol{v},\boldsymbol{w} \in \mathbb{R}^n$. We stress that these quantities depend in general on the current time step, but for simplicity of notation we dropped the subscripts. If we choose a uniform time discretization, i.e., $\tau_k = \tau$, in the exponential Euler scheme (12) we can compute once and for all the matrices $\mathrm{e}^{\tau A_{\mathrm{F}}}$ and $\varphi_1(\tau A_{\mathrm{F}})$ and then multiply by the

corresponding vectors. In this case, for the matrix function approximations the most common techniques are Taylor expansions or Padé rational approximations with scaling and squaring (see, for instance, References [3, 18, 52, 53]). This approach is computationally attractive only for matrices of moderate size, taking into account also that the resulting matrix functions are full even if the original ones were sparse. When employing the exponential Euler–Magnus schemes (14) and (16), we can still pursue this approach. However, since here the matrices change at each time step, we need to recompute the matrix functions every time accordingly. It is also possible to compute linear combination (17) by using a *single* slightly augmented matrix function evaluation. In fact, thanks to [50, Proposition 2.1], we have that the first n rows of

$$\exp\left(\tau\begin{bmatrix} X & \boldsymbol{w} \\ 0\dots 0 & 0 \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v} \\ 1 \end{bmatrix}$$

coincide with vector (17). This is an attractive choice in a variable step size scenario, in which both the forward and the backward equations could be solved by a single matrix function evaluation at each time step.

When X is a large sized and sparse matrix, it may be convenient to compute directly vector (17) at each time step *without* explicitly computing the matrix exponential. State-of-the-art techniques follow this approach and are based on Krylov methods or direct interpolation polynomial methods (see, for instance, References [4, 20, 35, 44]).

4 Numerical experiments

We present in this section several numerical examples arising from different choices of parameters and functions in the continuous model (5). In particular, we consider numerical experiments for two different classes of multi-agent systems in opinion formation and pedestrian dynamics, and a mass transfer problem. In all cases, we discretize in space with second order centered finite differences and we employ the trapezoidal rule for the quadrature of the integral operators. All the numerical experiments have been performed on an Intel® Core™ i7-10750H CPU with six physical cores and 16GB of RAM, using MATLAB programming language. As a software, we use MathWorks MATLAB® R2022a. In order to compute the needed actions of exponential and φ_1-function, we employ the `kiops` function[1], which is based on the Krylov method and whose underlying algorithm is presented in Reference [35]. The code used for the simulations, accompanied by a detailed description, can be found in a GitHub repository[2]. In particular, it allows to reproduce the figures of all the following numerical experiments.

4.1 Control in opinion dynamics

In this section we consider two models for control of opinion dynamics, namely the Sznajd and the Hegselmann–Krause (bounded confidence) ones, similarly to References [7, 38, 54]. We set both models in the spatial domain $\Omega = [-1,1]$, whose boundaries represent the extremal opinions. The running cost is $e(t,x,\rho) = (x-x_d)^2\rho$ and the selective function $s(t,x,\rho)$ is set to the constant 1 (hence, we use the exponential Euler–Magnus scheme (14) for the forward equation). For both the problems we consider in model (5) zero-flux boundary conditions everywhere and null terminal cost function $c(T,x,\rho(T,x)) = 0$.

[1] https://gitlab.com/stephane.gaudreault/kiops/-/tree/master/, commit 94149844.
[2] https://github.com/cassinif/expint_mfsoc, commit a7b6748.

4.1.1 Sznajd model

In the first numerical experiment we present an example of Sznajd model for opinion formation taken from Reference [7]. In particular, we consider the interaction function $p(x,y) = x^2 - 1$, representing a repulsive interaction, and the target point in the running cost $x_d = -0.5$. Moreover, we set the penalization parameter $\gamma = 0.5$ and the diffusion coefficient $\sigma = \sqrt{0.02}$. The initial density function is of bimodal type

$$\rho_0(x) = C(\rho_+(x+0.75;0.05,0.5) + \rho_+(x-0.5;0.15,1)),$$

where

$$\rho_+(x;a,b) = \max\left\{-\left(\frac{x}{b}\right)^2 + a, 0\right\}$$

and C defined so that $\int_\Omega \rho_0(x)dx = 1$.

First of all, we show that the expected temporal rate of convergence of the exponential integrators is preserved also after a complete solution of the model. In fact, for a semidiscretization in space with $n = 200$ uniform grid points, we solve several times model (5) by the steepest descent method described at the end of Section 2 by employing an increasing sequence of time steps, ranging from $m = 300$ to $m = 700$. Each time, after the stabilization of the functional $\mathscr{J}$, we measure the error at the final time $T = 4$ for $\boldsymbol{\rho}(t)$ and at initial time for $\boldsymbol{\psi}(t)$ with respect to reference solutions. We display in Figure 1 the obtained relative errors, which confirm the expected accuracy and rate of convergence.

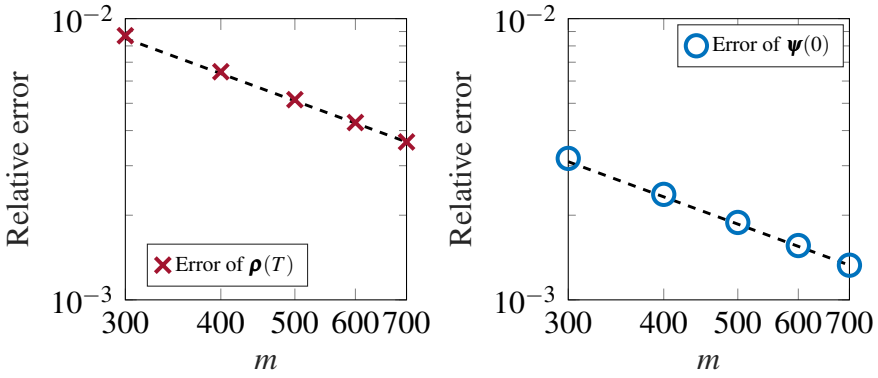


Figure 1: Relative errors in the infinity norm of $\boldsymbol{\rho}(T)$ ($T = 4$, left plot) and $\boldsymbol{\psi}(0)$ (right plot), with respect to a reference solution, for the Sznajd model described in Section 4.1.1 with $n = 200$ spatial discretization points and varying number of time steps m. A black dashed reference line of slope -1 is also displayed.

Then, we show the behavior of the Sznajd model in opinion formation. For this purpose we use a spatial discretization of $n = 800$ points and $m = 200$ time steps. Notice that we decide to employ a relatively large number of discretization points in space to highlight the fact that the exponential integrators do not exhibit any CFL restriction, in contrast to standard explicit methods. In Figures 2 and 3 we show the evolution of the density $\rho(t,x)$ and of the control $u(t,x)$. The results have the expected behavior of concentration of the opinions around the target point $x_d = -0.5$ and qualitatively match the analogous simulation available in the literature [7]. Moreover, we show in Figure 4 the value of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method. We observe that the method needs 23 iterations to reach the input tolerance $2 \cdot 10^{-3}$. Finally, the overall computational time of this simulation is about 40 seconds.

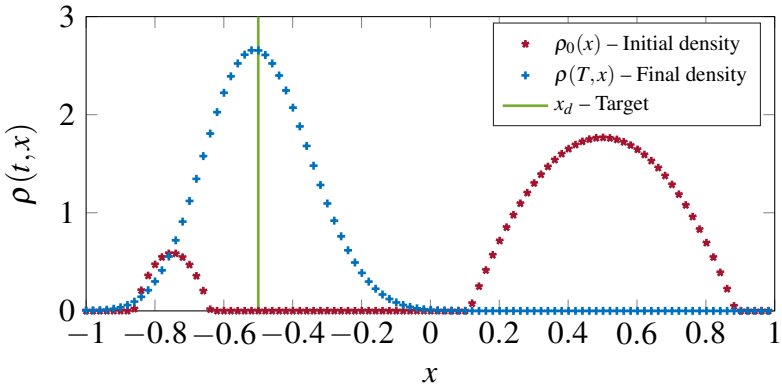


Figure 2: Density functions at initial time and at final time for the Sznajd model described in Section 4.1.1 with $n = 800$ spatial discretization points and $m = 200$ time steps. For visualization reasons, the densities are displayed each tenth point.

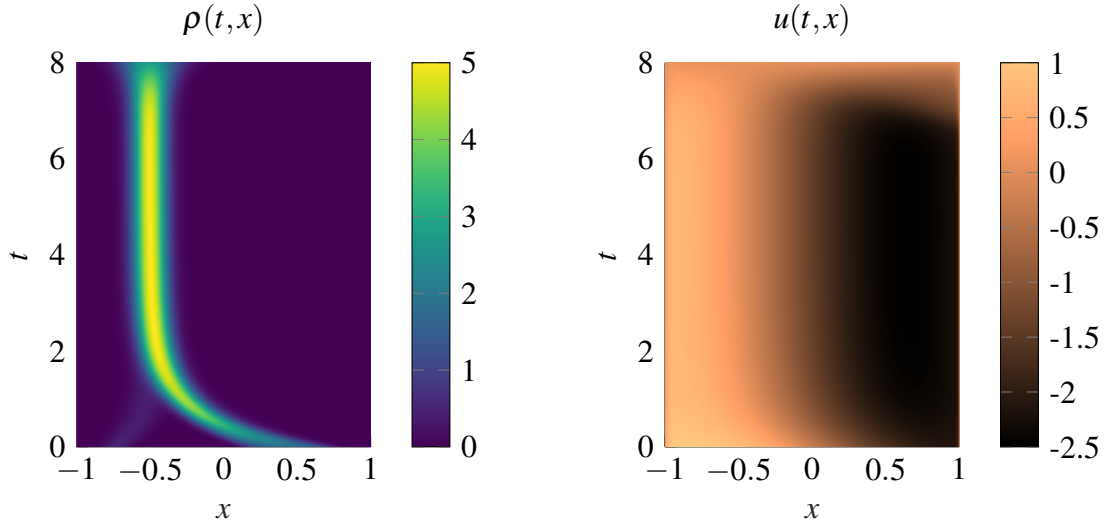


Figure 3: Evolution of the density $\rho(t,x)$ (left) and of the control $u(t,x)$ (right) up to the final time $T = 8$ for the Sznajd model described in Section 4.1.1 with $n = 800$ spatial discretization points and $m = 200$ time steps.

4.1.2 Hegselmann–Krause model

In the second numerical experiment we present an example of Hegselmann–Krause model for opinion formation taken from Reference [7]. In particular, we take the interaction function $p(x,y) = \chi_{\{|x-y|\leq\kappa\}}(x,y)$, with $\kappa = 0.15$, and the target point in the running cost $x_d = 0$. Moreover, we set the penalization parameter $\gamma = 2.5$ and the diffusion coefficient $\sigma = \sqrt{0.002}$. The initial density function is

$$\rho_0(x) = C(0.5 + \varepsilon(1 - x^2)),$$

where $\varepsilon = 0.01$ and C defined so that $\int_\Omega \rho_0(x)dx = 1$. For this model, we directly present the results using a spatial discretization of $n = 1000$ points and $m = 100$ time steps up to the final time $T = 10$. In Figures 5 and 6 we display the evolution of the density $\rho(t,x)$ and of the control $u(t,x)$. Similarly to the Sznajd model, the results match both the expectations and the outcomes in the literature. Then, we display in Figure 7 the value of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method. We observe that the method needs 15

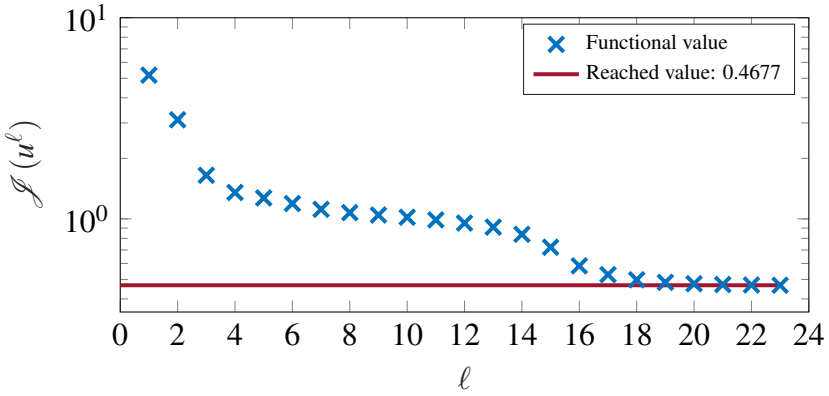


Figure 4: Value of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method for the Sznajd model described in Section 4.1.1 ($n = 800$ and $m = 200$).

iterations to reach the input tolerance $2 \cdot 10^{-3}$. Finally, this simulation takes roughly 15 seconds to run.

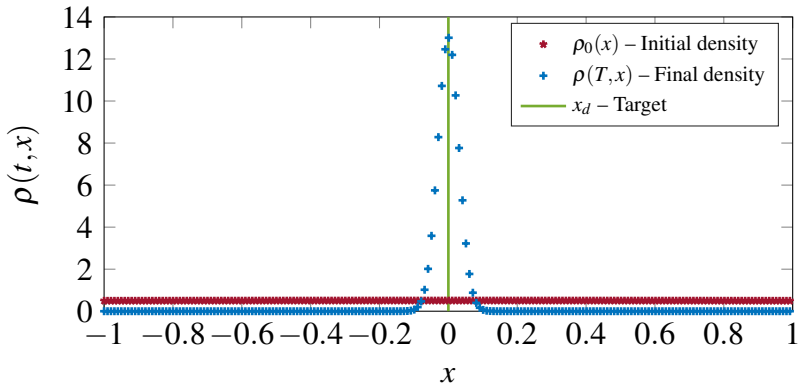


Figure 5: Density functions at initial time and at final time for the Hegselmann–Krause model described in Section 4.1.2 with $n = 1000$ spatial discretization points and $m = 100$ time steps. For visualization reasons, the densities are displayed each fifth point.

4.2 Crowd dynamics: fast exit scenario

In this section we consider a model for crowd dynamics taken from Reference [16]. We set the model in the spatial domain $\Omega = [-1,1]$, whose boundaries represent the exit doors. The non-local interaction kernel $p(x,y)$ is null and the selective function $s(t,x,\rho)$ is $1-\rho$ (hence, we employ the exponential Euler method (12) for the forward equation). The diffusion parameter is $\sigma = \sqrt{0.04}$, the exit intensity flux is $\beta = 10$, the penalization parameter is $\gamma = 1$, and the running cost is $e(t,x,\rho) = \rho$. The initial density function models the presence of two distinct groups, namely $\rho_0(x) = 0.9\mathrm{e}^{-100(x+0.4)^2} + 0.65\mathrm{e}^{-150x^2}$.

Similarly to the opinion dynamics case, we first show that the expected temporal rate of convergence of the exponential integrators is preserved after a complete solution of the model. To this purpose, we discretize this problem with $n = 200$ spatial discretization points and with different number of time steps, from $m = 300$ to $m = 700$, up to the final time $T = 2$. After the stabilization of the functional $\mathscr{J}$ in the steepest descent algorithm, we measure the error at

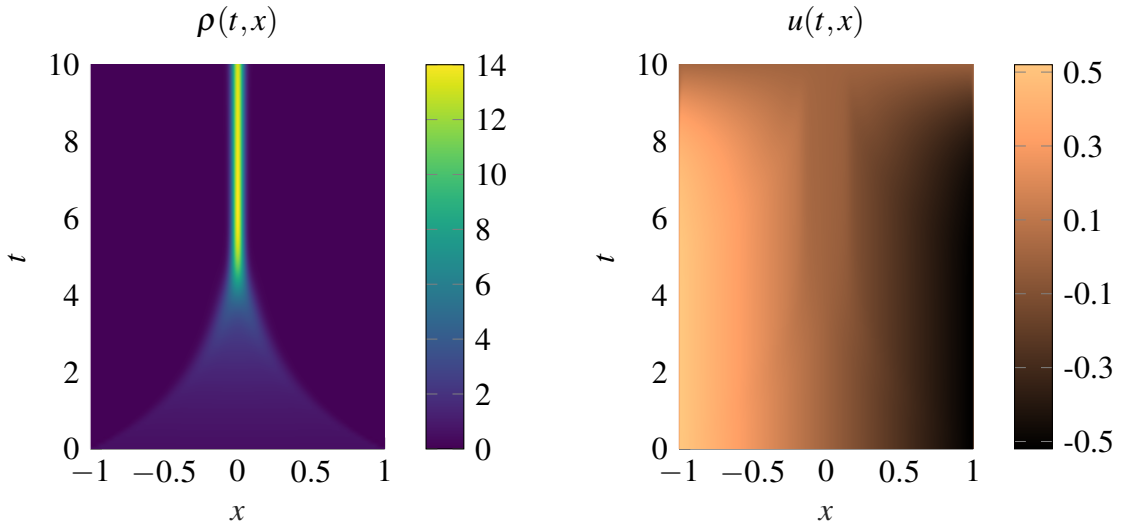


Figure 6: Evolution of the density $\rho(t,x)$ (left) and of the control $u(t,x)$ (right) up to the final time $T = 10$ for the Hegselmann–Krause model described in Section 4.1.2 with $n = 1000$ spatial discretization points and $m = 100$ time steps.

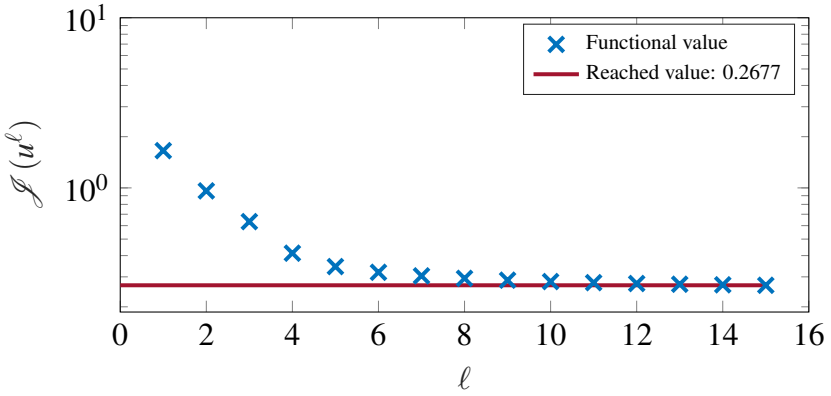


Figure 7: Value of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method for the Hegselmann–Krause model described in Section 4.1.2 ($n = 1000$ and $m = 100$).

final time for $\boldsymbol{\rho}(t)$ and at initial time for $\boldsymbol{\psi}(t)$ with respect to reference solutions. We display in Figure 8 the obtained relative errors which again confirm the expected accuracy and rate of convergence.

Then, we solve the same model up to the final time $T = 3$ and show its behavior. We discretize this problem with $n = 1000$ spatial discretization points and $m = 250$ time steps. We show the evolution of the density and of the control in Figures 9 and 10, where we can clearly see the exit of the crowd from the two doors. Moreover, we show in Figure 11 the value of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method. We observe that the method needs 13 iterations to reach the input tolerance $2 \cdot 10^{-3}$. Finally, the overall computational time of this simulation is about 45 seconds.

4.3 Mass transfer problem via optimal control

In this final example, we present an optimal control approach to a mass transfer problem, see for instance References [13, 51], where the particle density accounts for non-local interactions [14,

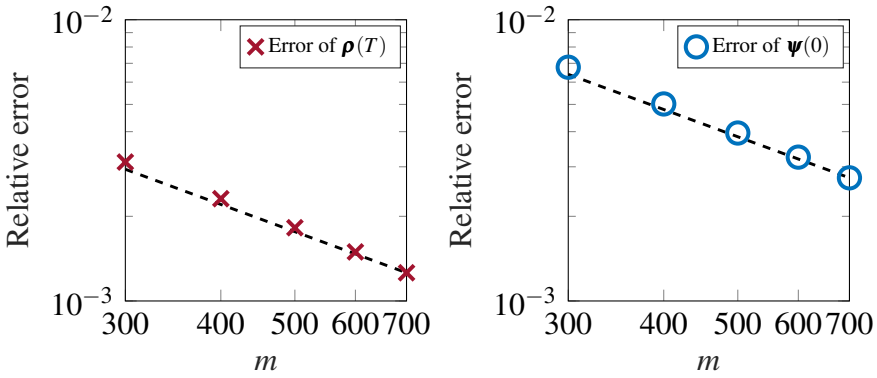


Figure 8: Relative errors in the infinity norm of $\boldsymbol{\rho}(T)$ ($T = 2$, left plot) and $\boldsymbol{\psi}(0)$ (right plot), with respect to a reference solution, for the pedestrian model described in Section 4.2 with $n = 200$ spatial discretization points and varying number of time steps m. A black dashed reference line of slope -1 is also displayed.

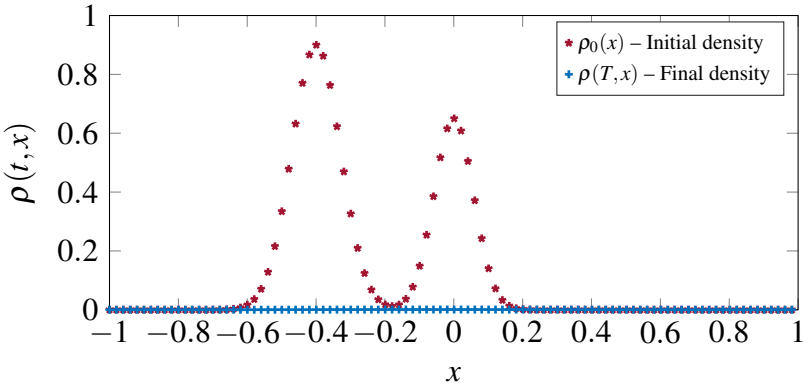


Figure 9: Density functions at initial time and at final time for the two-group crowd model described in Section 4.2 with $n = 1000$ spatial discretization points and $m = 250$ time steps. For visualization reasons, the densities are displayed each tenth point.

25]. Hence, the goal is to move the initial density function in the spatial domain $\Omega = [-1,1]$

$$\rho_0(x) = Ce^{-(x-\mu_0)^2/(2\sigma_0^2)},$$

where $\mu_0 = 0$, $\sigma_0 = 0.1$, and C is defined so that $\int_\Omega \rho_0(x)dx = 1$, to a target one

$$\bar{\rho}(x) = \bar{C}\left(e^{-(x-\mu_1)^2/(2\sigma_1^2)} + e^{-(x-\mu_2)^2/(2\sigma_2^2)}\right),$$

where $\mu_1 = 0.5$, $\sigma_1 = 0.1$, $\mu_2 = -0.3$, and $\sigma_2 = 0.15$, and $\bar{C}$ is defined so that $\int_\Omega \bar{\rho}(x)dx = 1$. The boundary conditions are of zero-flux type, the running cost is $e(t,x,\rho) = (\rho - \bar{\rho})^2$, the interaction kernel is of Sznajd type $p(x,y) = (x^2-1)/20$, and the selective function is $s(t,x,\rho) = 1$. The penalization parameter is $\gamma = 0.1$ and the diffusion parameter is $\sigma = \sqrt{0.02}$. We discretize the problem with $n = 1000$ spatial grid points and $m = 200$ time steps, and we run the simulation up to the final time $T = 3$. We consider a terminal cost given by $c(T,x,\rho(T,x)) = (\rho(T,x) - \bar{\rho}(x))^2$, which translates into $\psi_T(x) = \rho(T,x) - \bar{\rho}(x)$. In Figure 12 we plot the density functions at initial and final time, and we can observe that the initial density is correctly transported to the target one. In addition, in Figure 13 we present the evolution

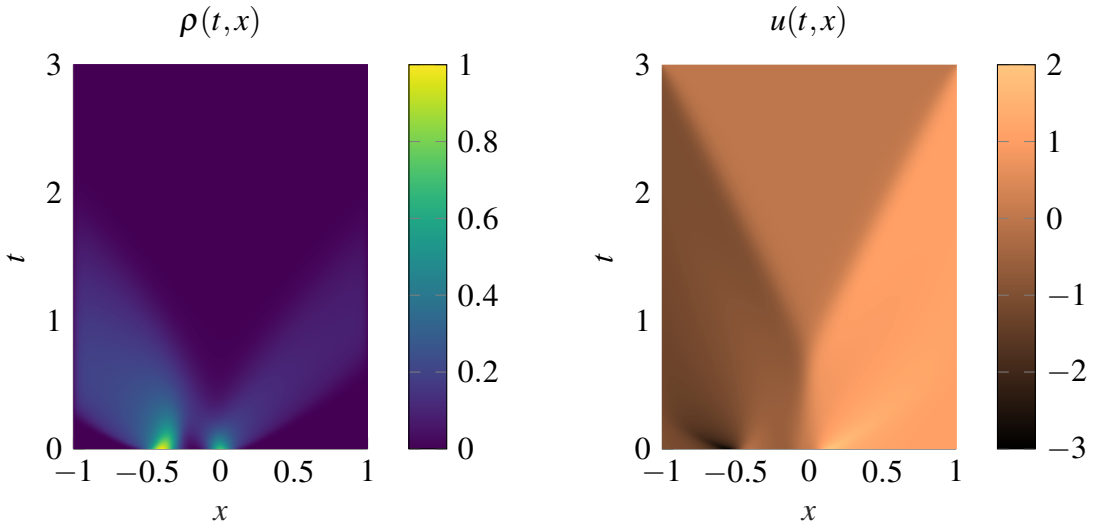


Figure 10: Evolution of the density $\rho(t,x)$ (left) and of the control $u(t,x)$ (right) up to the final time $T = 3$ for the two-group crowd model described in Section 4.2 with $n = 1000$ spatial discretization points and $m = 250$ time steps.

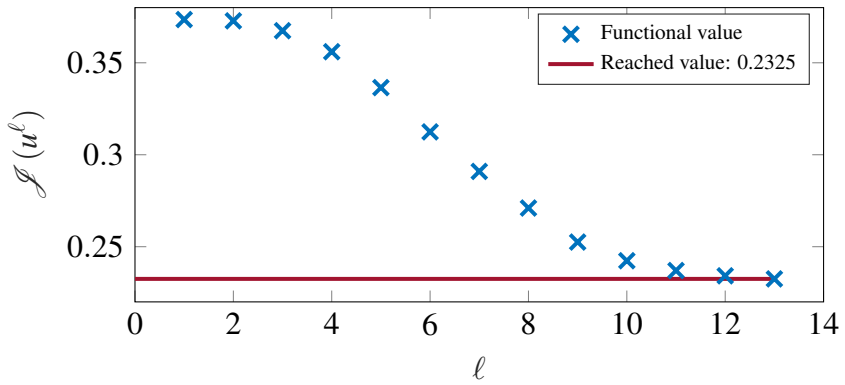


Figure 11: Value of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method for the two-group crowd model described in Section 4.2 ($n = 1000$ and $m = 250$).

of the density and of the control. Finally, we show in Figure 14 the values of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method. We observe that the method needs 33 iterations to reach the input tolerance $2 \cdot 10^{-3}$, with an overall computational time of this simulation of roughly 75 seconds.

5 Conclusions

We presented a mean-field optimal control model where the constraint is represented by a non-linear PDE with non-local interaction term and diffusion describing the evolution of a continuum of agents. We provide, at a formal level, first order optimality conditions, resulting in a forward–backward coupled system with associated boundary conditions. Thus, a reduced gradient method is derived for the synthesis of the mean-field control, where the primal and adjoint equations are efficiently solved by using exponential integrators. Our proposed approach has been successfully tested on various examples from the literature, including models of opinion formation and pedestrian dynamics in the one-dimensional setting. In future works we

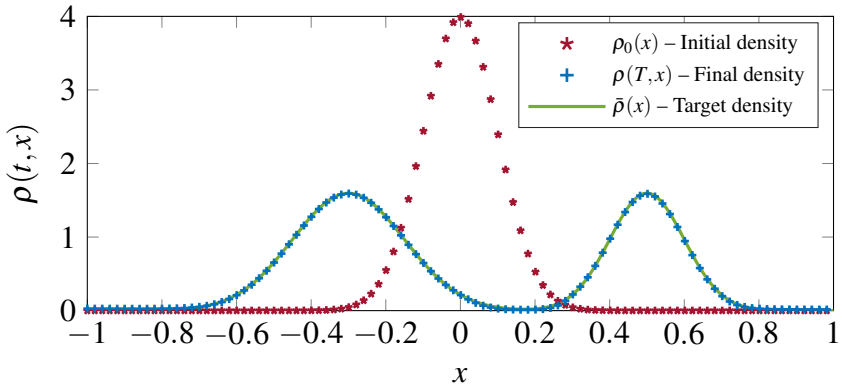


Figure 12: Density functions at initial time and at final time for the mass transfer problem described in Section 4.3 with $n = 1000$ spatial discretization points and $m = 200$ time steps.

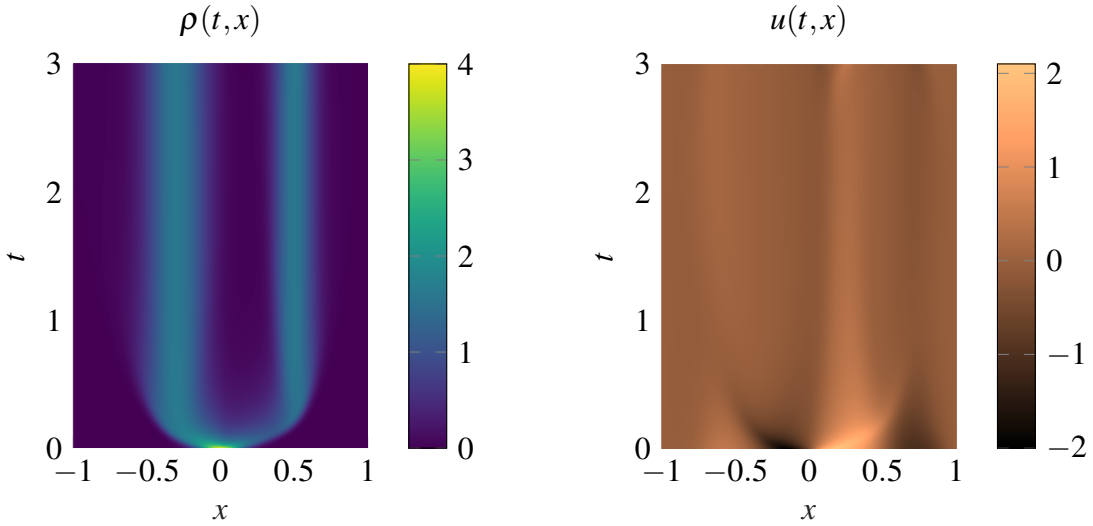


Figure 13: Evolution of the density $\rho(t,x)$ (left) and of the control $u(t,x)$ (right) up to the final time $T = 3$ for the mass transfer problem described in Section 4.3 with $n = 1000$ spatial discretization points and $m = 200$ time steps.

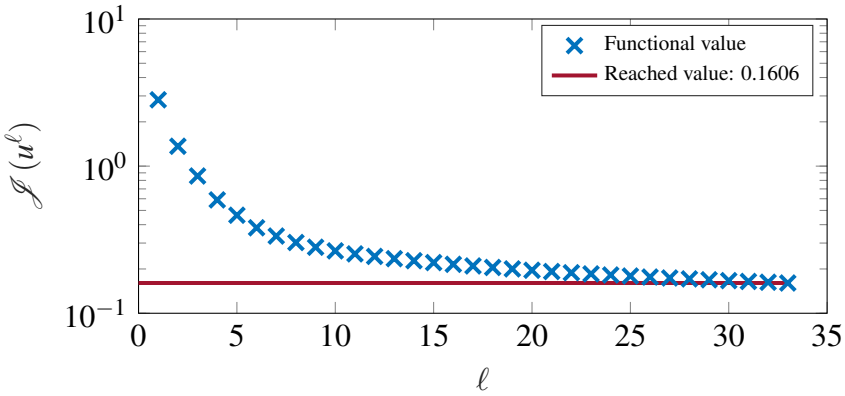


Figure 14: Value of the functional $\mathscr{J}(u^\ell)$ at the successive iterations of the steepest descent method for the mass transfer problem described in Section 4.3.

plan to exploit the efficiency of exponential integrators to tackle higher dimensional problems (possibly using ad hoc techniques for tensor structured problems [19, 20, 21]) and scenarios where a fine spatial discretization is required to correctly capture the behavior of the controlled dynamics.

Acknowledgments

The authors are members of the INdAM group GNCS, and acknowledge the support of the Italian Ministry of University and Research (MUR) through the MUR-PRIN Project 2022 No. 2022N9BM3N "Efficient numerical schemes and optimal control methods for time-dependent partial differential equations" and MUR-PRIN Project 2022 PNRR (No. P2022JC95T) "Data-driven discovery and control of multi-scale interacting artificial agent systems", financed by the European Union - Next Generation. Fabio Cassini holds a post-doc fellowship funded by INdAM.

The authors are former colleagues of Professor Leonard Peter Bos, and they would like to express their gratitude for his valuable presence in the Department of Computer Science at the University of Verona and all his precious advice.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Achdou and I. Capuzzo-Dolcetta. Mean Field Games: Numerical Methods. *SIAM J. Numer. Anal.*, 48(3):1136–1162, 2010. URL https://doi.org/10.1137/090758477.

[2] M. Aduamoah, B. D. Goddard, J. W. Pearson, and J. C. Roden. Pseudospectral methods and iterative solvers for optimization problems from multiscale particle dynamics. *BIT Numer. Math.*, 62:1703–1743, 2022. URL https://doi.org/10.1007/s10543-022-00928-w.

[3] A. H. Al-Mohy and N. J. Higham. A New Scaling and Squaring Algorithm for the Matrix Exponential. *SIAM J. Matrix Anal. Appl.*, 31(3):970–989, 2009. URL https://doi.org/10.1137/09074721X.

[4] A. H. Al-Mohy and N. J. Higham. Computing the Action of the Matrix Exponential with an Application to Exponential Integrators. *SIAM J. Sci. Comput.*, 33(2):488–511, 2011. URL https://doi.org/10.1137/100788860.

[5] G. Albi, M. Herty, and L. Pareschi. Kinetic description of optimal control problems and applications to opinion consensus. *Commun. Math. Sci.*, 13(6):1407–1429, 2015. URL https://doi.org/10.4310/CMS.2015.v13.n6.a3.

[6] G. Albi, M. Bongini, E. Cristiani, and D. Kalise. Invisible Control of Self-Organizing Agents Leaving Unknown Environments. *SIAM J. Appl. Math.*, 76(4):1683–1710, 2016. URL https://doi.org/10.1137/15M1017016.

[7] G. Albi, Y.-P. Choi, M. Fornasier, and D. Kalise. Mean Field Control Hierarchy. *Appl. Math. Optim.*, 76:93–135, 2017. URL https://doi.org/10.1007/s00245-017-9429-x.

[8] G. Albi, N. Bellomo, L. Fermo, S.-Y. Ha, J. Kim, L. Pareschi, D. Poyato, and J. Soler. Vehicular traffic, crowds, and swarms: From kinetic theory and multiscale methods to applications and research perspectives. *Math. Models Methods Appl. Sci.*, 29(10):1901–2005, 2019. URL https://doi.org/10.1142/S0218202519500374.

[9] G. Albi, M. Herty, and L. Pareschi. Linear multistep methods for optimal control problems and applications to hyperbolic relaxation systems. *Appl. Math. Comput.*, 354:460–477, 2019. URL https://doi.org/10.1016/j.amc.2019.02.021.

[10] G. Albi, F. Ferrarese, and C. Segala. Optimized leaders strategies for crowd evacuation in unknown environments with multiple exits. In *Crowd Dynamics, Volume 3: Modeling and Social Applications in the Time of COVID-19*, Modeling and Simulation in Science, Engineering and Technology, pages 97–131. Birkhäuser, Cham, 2021. URL https://doi.org/10.1007/978-3-030-91646-6_5.

[11] G. Albi, S. Almi, M. Morandotti, and F. Solombrino. Mean-field selective optimal control via transient leadership. *Appl. Math. Optim.*, 85:22, 2022. URL https://doi.org/10.1007/s00245-022-09837-4.

[12] R. Bailo, M. Bongini, J. A. Carrillo, and D. Kalise. Optimal consensus control of the Cucker-Smale model. *IFAC-PapersOnLine*, 51(13):1–6, 2018. URL https://doi.org/10.1016/j.ifacol.2018.07.245.

[13] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numer. Math.*, 84:375–393, 2000. URL https://doi.org/10.1007/s002110050002.

[14] M. Bongini and G. Buttazzo. Optimal control problems in transport dynamics. *Math. Models Methods Appl. Sci.*, 27(3):427–451, 2017. URL https://doi.org/10.1142/S0218202517500063.

[15] A. Borzì and V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations*. Computational Science & Engineering. SIAM, Philadelphia, PA, 2011. URL https://doi.org/10.1137/1.9781611972054.

[16] M. Burger, M. Di Francesco, P. A. Markowich, and M.-T. Wolfram. On a mean field game optimal control approach modeling fast exit scenarios in human crowds. In *52nd IEEE Conference on Decision and Control*, pages 3128–3133, 2013. URL https://doi.org/10.1109/CDC.2013.6760360.

[17] M. Burger, R. Pinnau, C. Totzeck, and O. Tse. Mean-Field Optimal Control and Optimality Conditions in the Space of Probability Measures. *SIAM J. Control Optim.*, 59(2):977–1006, 2021. URL https://doi.org/10.1137/19M1249461.

[18] M. Caliari and F. Zivcovich. On-the-fly backward error estimate for matrix exponential approximation by Taylor algorithm. *J. Comput. Appl. Math.*, 346:532–548, 2019. URL https://doi.org/10.1016/j.cam.2018.07.042.

[19] M. Caliari, F. Cassini, L. Einkemmer, A. Ostermann, and F. Zivcovich. A μ-mode integrator for solving evolution equations in Kronecker form. *J. Comput. Phys.*, 455:110989, 2022. URL https://doi.org/10.1016/j.jcp.2022.110989.

[20] M. Caliari, F. Cassini, and F. Zivcovich. A μ-mode BLAS approach for multidimensional tensor-structured problems. *Numer. Algorithms*, 92(4):2483–2508, 2023. URL https://doi.org/10.1007/s11075-022-01399-4.

[21] M. Caliari, F. Cassini, and F. Zivcovich. A μ-mode approach for exponential integrators: actions of φ-functions of Kronecker sums. *Calcolo*, 2024. URL https://doi.org/10.1007/s10092-024-00610-3. Accepted for publication.

[22] P. Cannarsa, R. Capuani, and P. Cardaliaguet. Mean field games with state constraints: from mild to pointwise solutions of the PDE system. *Calc. Var. Partial Diff. Equ.*, 60:108, 2021. URL https://doi.org/10.1007/s00526-021-01936-4.

[23] M. Caponigro, M. Fornasier, B. Piccoli, and E. Trélat. Sparse stabilization and control of alignment models. *Math. Models Methods Appl. Sci.*, 25(3):521–564, 2015. URL https://doi.org/10.1142/S0218202515400059.

[24] R. Carmona and M. Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II—the finite horizon case. *Ann. Appl. Probab.*, 32(6):4065–4105, 2022. URL https://doi.org/10.1214/21-AAP1715.

[25] J. A. Carrillo, M. Di Francesco, A. Figalli, T. Laurent, and D. Slepčev. Confinement in nonlocal interaction equations. *Nonlinear Anal. Theory Methods Appl.*, 75(2):550–558, 2012. URL https://doi.org/10.1016/j.na.2011.08.057.

[26] Y.-P. Choi, D. Kalise, J. Peszek, and A. A. Peters. A Collisionless Singular Cucker–Smale Model with Decentralized Formation Control. *SIAM J. Appl. Dyn. Syst.*, 18(4):1954–1981, 2019. URL https://doi.org/10.1137/19M1241799.

[27] E. Cristiani, B. Piccoli, and A. Tosin. *Multiscale Modeling of Pedestrian Dynamics*, volume 12 of *Modeling, Simulation & Applications*. Springer, Cham, 2014. URL https://doi.org/10.1007/978-3-319-06620-2.

[28] F. Cucker and S. Smale. Emergent Behavior in Flocks. *IEEE Trans. Automat. Control*, 52(5):852–862, 2007. URL https://doi.org/10.1109/TAC.2007.895842.

[29] P. Degond, J.-G. Liu, S. Motsch, and V. Panferov. Hydrodynamic models of self-organized dynamics: Derivation and existence theory. *Methods Appl. Anal.*, 20(2):89–114, 2013. URL https://doi.org/10.4310/MAA.2013.v20.n2.a1.

[30] M. R. D'Orsogna, Y. L. Chuang, A. L. Bertozzi, and L. S. Chayes. Self-Propelled Particles with Soft-Core Interactions: Patterns, Stability, and Collapse. *Phys. Rev. Lett.*, 96(10):104302, 2006. URL https://link.aps.org/doi/10.1103/PhysRevLett.96.104302.

[31] J. R. G. Dyer, A. Johansson, D. Helbing, I. D. Couzin, and J. Krause. Leadership, consensus decision making and collective behaviour in humans. *Philos. Trans. R. Soc. B*, 364(1518):781–789, 2009. URL https://doi.org/10.1098/rstb.2008.0233.

[32] M. Fornasier and F. Solombrino. Mean-Field Optimal Control. *ESAIM Control Optim. Calc. Var.*, 20(4):1123–1152, 2014. URL https://doi.org/10.1051/cocv/2014009.

[33] M. Fornasier, B. Piccoli, and F. Rossi. Mean-field sparse optimal control. *Philos. Trans. R. Soc. A*, 372(2028):20130400, 2014. URL https://doi.org/10.1098/rsta.2013.0400.

[34] G. Freudenthaler and T. Meurer. PDE-based multi-agent formation control using flatness and backstepping: Analysis, design and robot experiments. *Automatica*, 115:108897, 2020. URL https://doi.org/10.1016/j.automatica.2020.108897.

[35] S. Gaudreault, G. Rainwater, and M. Tokman. KIOPS: A fast adaptive Krylov subspace solver for exponential integrators. *J. Comput. Phys.*, 372:236–255, 2018. URL https://doi.org/10.1016/j.jcp.2018.06.026.

[36] J. Gómez-Serrano, C. Graham, and J.-Y. Le Boudec. The bounded confidence model of opinion dynamics. *Math. Models Methods Appl. Sci.*, 22(2):1150007, 2012. URL https://doi.org/10.1142/S0218202511500072.

[37] W. W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numer. Math.*, 87:247–282, 2000. URL https://doi.org/10.1007/s002110000178.

[38] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *J. Artif. Soc. Soc. Simul.*, 5(3), 2002. URL http://jasss.soc.surrey.ac.uk/5/3/2.html.

[39] M. Herty and L. Pareschi. Fokker-Planck asymptotics for traffic flow models. *Kinet. Relat. Models*, 3(1):165–179, 2010. URL https://doi.org/10.3934/krm.2010.3.165.

[40] M. Herty, L. Pareschi, and S. Steffensen. Implicit-Explicit Runge–Kutta Schemes for Numerical Discretization of Optimal Control Problems. *SIAM J. Numer. Anal.*, 51(4):1875–1899, 2013. URL https://doi.org/10.1137/120865045.

[41] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010. URL https://doi.org/10.1017/S0962492910000048.

[42] M. Huang, R. P. Malhamé, and P. E. Caines. Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Commun. Inf. Syst.*, 6(3):221–252, 2006. URL http://projecteuclid.org/euclid.cis/1183728987.

[43] J. M. Lasry and P. L. Lions. Mean field games. *Japanese J. Math.*, 2:229–260, 2007. URL https://doi.org/10.1007/s11537-007-0657-8.

[44] V. T. Luan, J. A. Pudykiewicz, and D. R. Reynolds. Further development of efficient and accurate time integration schemes for meteorological models. *J. Comput. Phys.*, 376:817–837, 2019. URL https://doi.org/10.1016/j.jcp.2018.10.018.

[45] S. Motsch and E. Tadmor. Heterophilious Dynamics Enhances Consensus. *SIAM Rev.*, 56(4):577–621, 2014. URL https://doi.org/10.1137/120901866.

[46] K.-K. Oh, M.-C. Park, and H.-S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015. URL https://doi.org/10.1016/j.automatica.2014.10.022.

[47] A. A. Peters, R. H. Middleton, and O. Mason. Leader tracking in homogeneous vehicle platoons with broadcast delays. *Automatica*, 50(1):64–74, 2014. URL https://doi.org/10.1016/j.automatica.2013.09.034.

[48] B. Piccoli and A. Tosin. Pedestrian flows in bounded domains with obstacles. *Continuum Mech. Thermodyn.*, 21(2):85–107, 2009. URL https://doi.org/10.1007/s00161-009-0100-x.

[49] L. Ruthotto, S. J. Osher, W. Li, L. Nurbekyan, and S. W. Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proc. Natl. Acad. Sci. U.S.A.*, 117(17):9183–9193, 2020. URL https://doi.org/10.1073/pnas.1922204117.

[50] Y. Saad. Analysis of Some Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992. URL https://doi.org/10.1137/0729014.

[51] F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser, Cham, 2015. URL https://doi.org/10.1007/978-3-319-20828-2.

[52] J. Sastre, J. Ibáñez, and E. Defez. Boosting the computation of the matrix exponential. *Appl. Math. Comput.*, 340:206–220, 2019. URL https://doi.org/10.1016/j.amc.2018.08.017.

[53] B. Skaflestad and W. M. Wright. The scaling and modified squaring method for matrix functions related to the exponential. *Appl. Numer. Math.*, 59(3–4):783–799, 2009. URL https://doi.org/10.1016/j.apnum.2008.03.035.

[54] K. Sznajd-Weron and J. Sznajd. Opinion evolution in closed community. *Int. J. Mod. Phys. C*, 11(6):1157–1165, 2000. URL https://doi.org/10.1142/S0129183100000936.

[55] G. Toscani. Kinetic models of opinion formation. *Commun. Math. Sci.*, 4(3):481–496, 2006. URL http://projecteuclid.org/euclid.cms/1175797553.

J. Approx. Softw., Vol. 1, No. 2 (2024) 4

Journal of Approximation Software

The Multinode Shepard Method: MATLAB Implementation

F. Dell'Accio [1], F. Di Tommaso [1,⋆], and F. Larosa [1]

[1] *Department of Mathematics and Computer Science, University of Calabria*

Received: 12/07/2024 – Published: 06/09/2024
Communicated by: R. Cavoretto

Abstract

The multinode Shepard method is an extension of inverse distance weighting, developed as a generalization of the triangular Shepard method to further improve interpolation accuracy in situations where the classic Shepard method results are limited. In particular, it considers multiple nodes for local interpolation, offering greater flexibility and improved accuracy in estimates. In this paper, we present two algorithms for computing the multinode Shepard interpolant, providing the related pseudocodes and MATLAB implementations.

Keywords: Multivariate Lagrange interpolation, Multinode Shepard method, Rational Interpolant, (MSC2020: 65D05, 41A05, 41A20)

1 Introduction

The approximation of scattered data is a crucial technique in modern science and engineering, enabling the extraction of significant information from incomplete or irregularly distributed data sets. One method for interpolating such data is Shepard's method.

The classical Shepard operator reconstructs a function through a weighted combination of its values at data points. The weights are the normalization of the inverse distances from the approximation point to the nodes. The nodes are scattered, that is, without any particular structure. More precisely, let $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ be a set of nodes in $\mathbb{R}^2$, and $f : \mathbb{R}^2 \to \mathbb{R}$ the function for which only the evaluations $f_i = f(\boldsymbol{x}_i)$ at the nodes are known. The Shepard operator [15] is defined as

$$S_\mu[f](\boldsymbol{x}) = \sum_{i=1}^{n} A_{\mu,i}(\boldsymbol{x}) f_i, \qquad \boldsymbol{x} \in \mathbb{R}^2, \tag{1}$$

⋆ Corresponding author: filomena.ditommaso@unical.it

where

$$A_{\mu,i}(\boldsymbol{x}) = \frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^{-\mu}}{\sum_{j=1}^{n} \|\boldsymbol{x}-\boldsymbol{x}_j\|^{-\mu}}, \qquad \boldsymbol{x} \in \mathbb{R}^2, \tag{2}$$

where $\mu > 0$ is a control parameter and $\|\cdot\|$ denotes the Euclidean norm.

The Shepard operator interpolates the data f_i at the nodes $\boldsymbol{x}_i$, even though it presents, for $u > 1$, flat spots in the neighborhood of all data point (which become cusps if $u \leq 1$). Moreover, it has a reproduction degree of zero, meaning it exclusively reproduces constant polynomials. This last property strongly influences the accuracy of the approximation provided by the Shepard interpolant, which converges to the function f with at most linear speed [13]. In order to overcome these limitations, several modifications have been proposed over the years [4].

The first significant modification of the Shepard operator is due to Little [14], who, based on the general idea of defining interpolants through convex combinations, proposed the following improvement. He considered a triangulation $T = \{t_j\}_{j=1}^{s}$ of the set X and replaced the values f_i in the expression of the Shepard operator (1) with the evaluations $L_j(\boldsymbol{x})$ of the linear polynomials interpolating at the vertices of each triangle t_j. Furthermore, he replaced the classical weight functions (2), obtained from the normalization of the inverse distances from individual nodes $\boldsymbol{x}_i$, with the product of the normalized inverse distances from the vertices $\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2}, \boldsymbol{x}_{j_3}$ of each triangle t_j. The resulting operator, known as the triangular Shepard operator, interpolates the data f_i at the nodes $\boldsymbol{x}_i$ and reproduces the polynomials up to degree 1. The formal definition is as follows

$$K_\mu[f](\boldsymbol{x}) = \sum_{j=1}^{s} B_{\mu,j}(\boldsymbol{x}) L_j(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^2, \tag{3}$$

where

$$B_{\mu,j}(\boldsymbol{x}) = \frac{\prod_{l=1}^{3} \|\boldsymbol{x}-\boldsymbol{x}_{j_l}\|^{-\mu}}{\sum_{k=1}^{s} \prod_{l=1}^{3} \|\boldsymbol{x}-\boldsymbol{x}_{k_l}\|^{-\mu}}, \qquad \boldsymbol{x} \in \mathbb{R}^2. \tag{4}$$

This deep modification of the Shepard operator aims to improve the accuracy and robustness of scattered data interpolation. The properties of the triangular Shepard operator have been extensively studied in [7]. In particular, the quadratic convergence of the triangular Shepard interpolant to the function f has been demonstrated, both in the case of regular triangulations, such as Delaunay, and in the case of compact triangulations, which do not exclude the intersection of triangles in common parts that are not only vertices or adjacent sides. This latter observation has paved the way for the not easy generalization of the classical Shepard operator, aiming to further increase the polynomial precision and, consequently, the accuracy of the approximation. This generalization, known as the multinode Shepard operator, has been proposed in a series of papers first dealing with the hexagonal case in $\mathbb{R}^2$ [6] and the tetrahedral case in $\mathbb{R}^3$ [3], relying on local barycentric coordinate systems, and then the general case in [8], exploiting a new representation of the local interpolation polynomial in Taylor form, centered at the barycenter of the local system of interpolation nodes [9].

In this paper, we present and discuss two algorithms for the computation of the multinode Shepard method. The paper is organized as follows. In Section 2 we recall the definition of the multinode Shepard operator and describe its properties. In Section 2.1 we present two algorithms for the implementation of the multinode Shepard method. In section 2.2 we give the Matlab implementation of the two codes. In Section 2.3 we report some numerical experiments

to test the effectiveness of the proposed algorithms. Finally, in Section 3 we summarize the benefits of the multinode Shepard method in terms of accuracy and computational efficiency.

2 Multinode Shepard method

Let $\Omega \subset \mathbb{R}^d$, $d \geq 2$, a non-empty connected open set, $\partial\Omega$ its boundary, $X = \{\boldsymbol{x}_i\}_{i=1}^n \subset \Omega \cup \partial\Omega$ a finite set of pairwise distinct scattered nodes and $f = \{f_i\}_{i=1}^n$ a set of function values associated to X. Let $r \in \mathbb{N}$ and $m = \binom{r+d}{d} = \dim\left(\mathscr{P}_r(\mathbb{R}^d)\right)$, where $\mathscr{P}_r(\mathbb{R}^d)$ denotes the space of polynomials of d variables of total degree $\leq r$.

We assume that a set $\{\sigma_j\}_{j=1}^s$ is given, such that for each $j = 1,\ldots,s$, $\sigma_j = \{\boldsymbol{x}_{j_k}\}_{k=1}^m \subset X$ is unisolvent for the polynomial space $\mathscr{P}_r(\mathbb{R}^d)$ and

$$\bigcup_{j=1}^{s} \sigma_j = X \tag{5}$$

(to shorten the notation, for each $j = 1,\ldots,s$, we are denoting with $j_k = \varphi_j(k)$ the image of $k \in \{1,\ldots,m\}$ by an injective map φ_j from $\{1,\ldots,m\}$ into $\{1,\ldots,n\}$).

A convenient way to represent the unique polynomial $P_j \in \mathscr{P}_r(\mathbb{R}^d)$, $j = 1,\ldots,s$, interpolating on $\sigma_j = \{\boldsymbol{x}_{j_1},\ldots,\boldsymbol{x}_{j_m}\}$ the data $\{f_{j_1},\ldots,f_{j_m}\}$ is given by the

$$P_j(\boldsymbol{x}) = \sum_{k=1}^{m} \ell_{j,k}(\boldsymbol{x}) f_{j_k}, \qquad \boldsymbol{x} \in \mathbb{R}^d,$$

where

$$\ell_{j,k}(\boldsymbol{x}) = \sum_{|\alpha| \leq r} a_{\alpha}^{(j,k)} \left(\boldsymbol{x} - \boldsymbol{x}_j^{(b)}\right)^{\alpha} \tag{6}$$

are the Lagrange fundamental polynomials written in the Taylor basis centered at the barycenter $\boldsymbol{x}_j^{(b)}$ of σ_j and $\alpha \in \mathbb{N}^m \cup \{(0,\ldots,0)\}$ is a multi-index (for more details see [9]). As well-known, the fundamental Lagrange polynomials satisfy the Kronecker delta property

$$\ell_{j,k}\left(\boldsymbol{x}_{j_l}\right) = \delta_{kl}, \quad j = 1,\ldots,s; \quad k,l = 1,\ldots,m. \tag{7}$$

The multinode inverse distance weighted functions based on the covering $\{\sigma_j\}_{j=1}^s$ are defined as follows [5]

$$W_{\mu,j}(\boldsymbol{x}) = \frac{\prod_{k=1}^{m} \left\|\boldsymbol{x} - \boldsymbol{x}_{j_k}\right\|^{-\mu}}{\sum_{l=1}^{s} \prod_{\lambda=1}^{m} \left\|\boldsymbol{x} - \boldsymbol{x}_{l_\lambda}\right\|^{-\mu}}, \qquad j = 1,\ldots,s, \quad \mu > 0. \tag{8}$$

The multinode functions form a partition of unity

$$\sum_{j=1}^{s} W_{\mu,j}(\boldsymbol{x}) = 1, \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{9}$$

and satisfy the following interpolation properties

$$W_{\mu,j}(\boldsymbol{x}_i) = 0 \text{ for all } \boldsymbol{x}_i \notin \sigma_j, \quad \sum_{j \in \mathscr{J}_i} W_{\mu,j}(\boldsymbol{x}_i) = 1, \tag{10}$$

where we set

$$\mathscr{J}_i = \{j \in \{1,\ldots,s\} : \boldsymbol{x}_i \in \sigma_j\}, \quad i = 1,\ldots,n.$$

In addition, if $\mu > 2$ the multinode functions $W_{\mu,j}(\boldsymbol{x})$ satisfy the following differential properties

$$\nabla W_{\mu,j}(\boldsymbol{x}_i) = \boldsymbol{0} \text{ for all } \boldsymbol{x}_i \notin \sigma_j, \quad \sum_{j\in\mathscr{J}_i} \nabla W_{\mu,j}(\boldsymbol{x}_i) = \boldsymbol{0}, \tag{11}$$

and

$$HW_{\mu,j}(\boldsymbol{x}_i) = \boldsymbol{0} \text{ for all } \boldsymbol{x}_i \notin \sigma_j, \quad \sum_{j\in\mathscr{J}_i} HW_{\mu,j}(\boldsymbol{x}_i) = \boldsymbol{0}, \tag{12}$$

where, as usual, $\nabla W_{\mu,j}(\boldsymbol{x})$ and $HW_{\mu,j}(\boldsymbol{x})$ denote the gradient and the Hessian matrix of $W_{\mu,j}(\boldsymbol{x})$, respectively. Finally, they are rational functions without real singularities if μ is an even integer (for more details, see [5]).

The multinode Shepard operator is a blend of the local interpolation polynomials realized by using multinode functions as follows

$$\mathscr{M}_\mu[f](\boldsymbol{x}) = \sum_{j=1}^{s} W_{\mu,j}(\boldsymbol{x}) P_j(\boldsymbol{x}) = \frac{\sum_{j=1}^{s}\prod_{k=1}^{m} \left\|\boldsymbol{x}-\boldsymbol{x}_{j_k}\right\|^{-\mu} P_j(\boldsymbol{x})}{\sum_{j=1}^{s}\prod_{k=1}^{m} \left\|\boldsymbol{x}-\boldsymbol{x}_{j_k}\right\|^{-\mu}}. \tag{13}$$

Since the property (9) $\mathscr{M}_\mu[\cdot]$ reproduces polynomials of d variables of total degree $\leq r$, while (10) imply that $\mathscr{M}_\mu[f]$ interpolates data f_i at $\boldsymbol{x}_i$, $i = 1,\ldots,n$. Moreover, by assuming that the set X is contained in a compact convex domain Ω and that the function f is of class $C^r(\Omega)$ with partial derivatives Lipschitz-continuous of order r, as proven in [5, Theorem 3.1], the multinode Shepard operator has an approximation accuracy of $O(h^{p+1})$ for each $\mu > \frac{d+p+1}{m}$. Here h denotes the fill distance of the set X.

From equation (13) it follows that the multinode Shepard operator is not uniquely defined, depending on the particular covering $\{\sigma_j\}_{j=1}^{s}$ of X. The existence of such a covering is *almost surely* guaranteed [11]. A straightforward determination of $\{\sigma_j\}_{j=1}^{s}$ can be done by considering, for each scattered point $\boldsymbol{x}_i$, the set of $m+q$, $q > 0$, nearest points and by choosing, among them, the subset of m points for which the local approximation to the function $f(\boldsymbol{x})$ provided by the polynomial $P_j(\boldsymbol{x})$ is near to the optimal one [10]. For $i \neq i'$, by denoting with $\sigma(\boldsymbol{x}_i)$ and $\sigma(\boldsymbol{x}_{i'})$ the subsets determined starting from the interpolation nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$, respectively, we have $\sigma_j = \sigma(\boldsymbol{x}_i)$ and $\sigma_{j'} = \sigma(\boldsymbol{x}_{i'})$, $j \neq j'$, if $\sigma(\boldsymbol{x}_i) \neq \sigma(\boldsymbol{x}_{i'})$, otherwise we set $\sigma_j = \sigma(\boldsymbol{x}_i) = \sigma(\boldsymbol{x}_{i'})$. While ensuring the covering condition, this procedure leads to an overly expensive definition of the multinode Shepard operator, due to the large number of terms in the sum (13).

In the following, we specialize in the case $d = 2$. Algorithms for general dimensions $d > 2$ can be obtained using the given approaches.

2.1 Pseudo-codes of two algorithms for the implementation of the multinode Shepard method

In this section, we present two algorithms for the computation of the multinode Shepard method. The Algorithm 1, based on the procedure mentioned above, chooses in the set of $m+q$, $q > 0$, nearby nodes to $\boldsymbol{x}_i$, the m discrete Leja points computed by the algorithm presented in [1]. This algorithm is based on the $PA = LU$ factorization of the Gram matrix. The set σ_j consists of the

points *related* to the first m rows after the $PA = LU$ factorization. Gaussian elimination with row pivoting performs a greedy optimization (i.e. not precise but still valid) of the Vandermonde determinant, iteratively searching for the new row (i.e. selecting the new interpolation point) such that the modulus of the increased determinant is maximized.

To speed up Algorithm 1 and reduce its computational cost, we introduce Algorithm 2. The key point of Algorithm 2 is to drastically reduce the number of subsets σ_j. Specifically, we create a copy $X' = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ of the node set X, whose first node is $\boldsymbol{x}_1$. In the process of determination of the covering $\{\sigma_j\}_{j=1}^{s}$, the set X remains fixed and is used to determine the subset N_i of the nearby nodes to $\boldsymbol{x}_i$. We reorder the nodes of N_i according to their increasing distances from $\boldsymbol{x}_i$, with $\boldsymbol{x}_i$ being the first node of X'. At the j-th step this rearrangement allows the identification of the subset σ_j by using the procedure stated in Algorithm 1, that is through the $PA = LU$ factorization of the Gram matrix. The subset σ_j is then subtracted from X', by maintaining the initial order of the nodes. Since at the step $j+1$ the new set $X' = X' \setminus \sigma_j$ will no longer contain $\boldsymbol{x}_i$ but a new first node $\boldsymbol{x}_{i'}$, the procedure ends when X' is empty.

To determine N_i, we set $M(\Omega)$ the Lebesgue measure of Ω, $\ell = \sqrt{\frac{\binom{r+1+2}{2} M(\Omega)}{n}}$ and $Q_i(\rho) = \left[x_i - \frac{\rho}{2}, x_i + \frac{\rho}{2}\right] \times \left[y_i - \frac{\rho}{2}, y_i + \frac{\rho}{2}\right]$, $\rho > 0$. Then

$$N_i = X \cap Q_i(\ell(1 + k/10)) \tag{14}$$

where k is the first non-negative integer such that $\#(N_i) \geq \binom{r+1+2}{2} = \dim\left(\mathscr{P}_{r+1}(\mathbb{R}^2)\right)$.

Algorithm 1 The multinode Shepard method with a m-tuple associated to each $\boldsymbol{x}_i$

Require: $X = \{\boldsymbol{x}_i\}_{i=1}^{n}$, the set of scattered points
$f = \{f_i\}_{i=1}^{n}$, the set of function values
r, the degree of the local polynomial interpolants
q, with $m+q$ the number of nearest neighbour points for determining $\{\sigma_j\}_{j=1}^{s}$
μ, the power parameter for the computation of (8)
$\boldsymbol{x}$, the set of approximation points
Ensure: $f(\boldsymbol{x})$, approximations of the function f at the points of $\boldsymbol{x}$

1: Set *Num*, the numerator of the multinode Shepard operator (13), equal to 0
2: Set *Den*, the denominator of the multinode Shepard operator (13), equal to 0
for $i = 1, \dots, n$ **do**
  3.1: Compute the vector *Dn* containing the distances between the i-th node and all other nodes
  3.2: Sort *Dn* according to the increasing distances from $\boldsymbol{x}_i$
  3.3: Consider the $m+q$ nearest points to $\boldsymbol{x}_i$ and choose, among them, the subset σ_j of m discrete Leja points by using the algorithm proposed in [1]
  if the subset σ_j has not been yet considered **then**
    4.1: Compute the polynomial interpolant of f based on the points of σ_j
    4.2: Compute the j-th term of the numerator and of the denominator of the multinode operator (13) and sum them up to *Num* and *Den*, respectively
  end if
end for
5: Compute the multinode Shepard operator as the ratio between *Num* and *Den*

Algorithm 2 The multinode Shepard method with minimized number of m-tuples

Require: $X = \{\boldsymbol{x}_i\}_{i=1}^n$, the set of scattered points
$f = \{f_i\}_{i=1}^n$, the set of function values
r, the degree of the local polynomial interpolants
$\boldsymbol{x}$, the set of approximation points
Ensure: $f(\boldsymbol{x})$, approximations of the function f at the points of $\boldsymbol{x}$

1: Compute $\ell = \sqrt{\frac{\binom{r+1+2}{2} M(\Omega)}{n}}$, the length of the side of the square to determine the set of nearby points
2: Create a copy X' of X.
3: Set *Num*, the numerator of the multinode Shepard operator (13), equal to 0
4: Set *Den*, the denominator of the multinode Shepard operator (13), equal to 0
5: Set i equal to 0
while $X' \neq \emptyset$ **do**
  6.1: Compute the set $N_i = X \cap Q_i(\ell)$
  6.2: Set k equal to 1
  while $\#(N_i) < \binom{r+1+2}{2}$ **do**
    7.1: Compute the set $N_i = X \cap Q_i(\ell(1+k/10))$
    7.2: Update k to $k+1$
  end while
  8: Sort N_i according to the increasing distances between the first node $\boldsymbol{x}'_i$ of X' and all nodes in N_i
  9: Choose the among the points in the ordered N_i, the subset σ_j of m discrete Leja points by using the algorithm proposed in [1]
  if the subset σ_j has not been yet considered **then**
    10.1: Compute the polynomial interpolant of f based on the points of σ_j
    10.2: Compute the j-th term of the numerator and of the denominator of the multinode operator (13) and sum them up to *Num* and *Den*, respectively
    11: Update i to $i+1$
  end if
end while
12: Compute the multinode Shepard operator as the ratio between *Num* and *Den*

2.2 MATLAB code

In this section we describe the MATLAB functions and demos available collected in a package named `Multinode_Shepard` freely available at [12].

2.2.1 Function `Multinode_Shepard1.m`

The function `Multinode_Shepard1.m` implements the Multinode Shepard method as described by the pseudocode Algorithm 1. In particular, we have the following Input and Output arguments:

INPUT:

- xn (double array): vector of the x-coordinates of the nodes
- yn (double array): vector of the y-coordinates of the nodes
- fn (double array): vector of the function values at the nodes (xn,yn)
- r (integer scalar): degree of the local polynomial interpolant
- q (integer scalar): number of additional points to select the m-tuple σ_j
- mu (integer scalar): power parameter
- x (double array): vector of the x-coordinate of the evalutation points
- y (double array): vector of the y-coordinate of the evalutation points

OUTPUT:

- MO (double array): values of the multinode Shepard operator at the points (x,y)
- s (integer scalar): number of the m-tuples σ_j

The function `Multinode_Shepard1.m` makes use of the following auxiliary functions and MATLAB functions:

- `powers`: function which computes the powers of the bivariate monomial basis of total degree d by using the routine mono_next_grlex.m
- `length`: computes the length of a vector
- `sort`: sorts in ascending order the elements of v, where v is the input vector and returns also the sort index I which specifies how the elements of v were rearranged to obtain the sorted output vector
- `ismember`: checks if a vector has the same entrances of the row of a matrix
- `sum`: computes the sum of the elements of a vector
- `BivVand`: function which computes the bivariate Vandermonde matrix
- `lu`: computes the LU factorization of a matrix and returns unit lower triangular matrix L, upper triangular matrix U, and permutation matrix P

- `backslash`: computes the solution of the linear system $Ax = b$, where A is a matrix and b is the known term
- `eps`: spacing of floating point numbers
- `prod`: computes the product of the over each column of a matrix

2.2.2 Function `Multinode_Shepard2.m`

The function `Multinode_Shepard2.m` implements the Multinode Shepard method as described by the pseudocode Algorithm 2. The output arguments are the same as the function `Multinode_Shepard1.m` described in Section 2.2.1 with the addition of the use of the MATLAB function `isempty.m` which allows to establish if a vector is empty. The input arguments are:

INPUT:

- xn (double array): vector of the x-coordinates of the nodes
- yn (double array): vector of the y-coordinates of the nodes
- fn (double array): vector of the function values at the nodes (xn,yn)
- r (integer scalar): degree of the local polynomial interpolant
- mu (integer scalar): power parameter
- x (double array): vector of the x-coordinate of the evaluation points
- y (double array): vector of the y-coordinate of the evaluation points

2.2.3 Demos

The demo `demo_trial` illustrates the numerical experiments provided in Section 2.4. The demo `demo_Stromboli` illustrates the numerical experiments provided in Section 2.3 by selecting, through a menu, the degree r of the local polynomial interpolant.

2.3 Numerical experiments

In this section we present some numerical experiments in order to show the efficiency of the multinode Shepard method. All the tests have been performed on a laptop with a 11th Gen Intel(R) Core(TM) i7-1165G7 2.80GHz 1.69 GHz processor and 16.00 GB RAM.

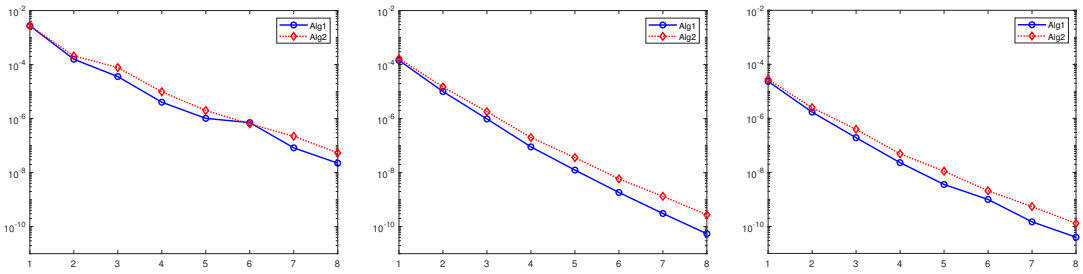
2.4 Reconstrution of smooth surfaces

For the first series of experiments, we make use of a set of 10000 Halton interpolation points and we use them to approximate the Franke function

$$\begin{aligned} f_1(x,y) = \; & 0.75e^{-\frac{(9x-2)^2+(9y-2)^2}{4}} + 0.50e^{-\frac{(9x-7)^2+(9y-3)^2}{4}} \\ & +0.75e^{-\frac{(9x+1)^2}{49}-\frac{(9y+1)^2}{10}} - 0.20e^{-(9x-4)^2-(9y-7)^2} \end{aligned}$$

in the unit square $[0,1]\times[0,1]$. We compute the pointwise error

$$e_i = |f(\boldsymbol{x}_i^*) - \mathscr{M}_4[f](\boldsymbol{x}_i^*)|,$$



MAE *MEAE* *RMSE*

Figure 1: Semilog plot of the maximum absolute errors (*MAE*), mean absolute errors (*MEAE*) and root mean square absolute errors (*RSME*) by varying the degree r of the local polynomial approximant from 1 to 8.

	number of m-tuples		CPU time (sec)	
r	Alg. 1	Alg. 2	Alg. 1	Alg. 2
1	9890	5173	7	3
2	10000	3354	13	3
3	10000	2406	15	3
4	9999	1783	31	4
5	10000	1362	44	4
6	10000	1183	58	4
7	10000	947	74	4
8	10000	802	88	4

Table 1: Number of m-tuples and CPU time (in seconds) for the experiments in Figure 1.

at a regular grid of $n_e = 100 \times 100$ points $\boldsymbol{x}_i^* \in [0,1] \times [0,1]$. We compare Algorithm 1 and Algorithm 2 by computing the maximum, mean and root mean square absolute errors

$$MAE = \max_{1 \le i \le ne} e_i; \quad MEAE = \frac{1}{ne} \sum_{i=1}^{ne} e_i; \quad RMSE = \sqrt{\frac{1}{ne} \sum_{i=1}^{ne} e_i^2} \tag{15}$$

and the CPU time (in seconds) for the computation of the approximate surface. In Figure 1 we display the semilog plot of the errors for the two algorithms by varying the degree of the local polynomial interpolant from 1 to 8 and in Table 1 we report the associated number s of m-tuples.

2.5 Reconstruction of surfaces from real-world data

For the second series of experiments, we make use of a set of real-world data related to the Stromboli Volcano including the Sciara del Fuoco of 2002 lava flow [2] which is constituted by 422710 DEM (Digital Elevation Model) data from which we extract the set of 97020 mock-Halton data. More precisely, we consider the set of 100000 Halton data mapped into the rectangle $[0,2060] \times [0,818]$ and we extract from the DEM data the ones that are closer to the Halton data by discarding the duplicates. To test the effectiveness of the multinode Shepard method, we extract from the mock-Halton dataset a subset $Z_{ne} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{ne}\}$ of $ne = 2000$ data to use as evaluation points.

In Table 2 we compare Algorithm 1 and Algorithm 2 by computing the maximum, mean and root mean square absolute errors as in (15) and in Table 3 we compare the maximum, mean

	MAE		*MEAE*		*RMSE*	
r	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2
1	$1.03e+1$	$7.90e+0$	$2.73e-1$	$2.90e-1$	$1.34e-2$	$1.38e-2$
2	$7.08e+0$	$2.52e+1$	$2.39e-1$	$2.97e-1$	$1.14e-2$	$2.02e-2$
3	$1.23e+1$	$1.15e+1$	$2.64e-1$	$3.61e-1$	$1.36e-2$	$1.96e-2$
4	$7.28e+0$	$2.21e+1$	$2.37e-1$	$3.42e-1$	$1.16e-2$	$2.00e-2$
5	$1.05e+1$	$1.62e+1$	$2.87e-1$	$4.19e-1$	$1.45e-2$	$2.23e-2$
6	$1.45e+1$	$2.49e+1$	$2.88e-1$	$3.97e-1$	$1.50e-2$	$2.27e-2$

Table 2: Maximum, mean and root mean square absolute errors.

	MAE_{rel}		$MEAE_{rel}$		$RMSE_{rel}$	
r	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2
1	$2.28e-1$	$1.90e-1$	$1.68e-3$	$1.70e-3$	$1.74e-4$	$1.69e-4$
2	$1.24e-1$	$1.72e+0$	$1.45e-3$	$2.55e-3$	$1.38e-4$	$8.74e-4$
3	$7.60e-1$	$1.46e+0$	$2.04e-3$	$2.79e-3$	$4.55e-4$	$7.62e-4$
4	$3.29e-1$	$5.67e-1$	$1.51e-3$	$2.47e-3$	$2.11e-4$	$4.02e-4$
5	$1.94e-1$	$5.07e-1$	$1.78e-3$	$2.75e-3$	$2.03e-4$	$3.95e-4$
6	$3.07e-1$	$2.18e-1$	$1.71e-3$	$2.15e-3$	$2.16e-4$	$2.02e-4$

Table 3: Maximum, mean and root mean square relative errors.

and root mean square relative errors

$$MAE_{rel} = \max_{1\le i\le ne} \frac{e_i}{f(\boldsymbol{x}_i)}; \quad MEAE_{rel} = \frac{1}{ne}\sum_{i=1}^{ne} \frac{e_i}{f(\boldsymbol{x}_i)}; \quad RMSE_{rel} = \sqrt{\frac{1}{ne}\sum_{i=1}^{ne}\left(\frac{e_i}{f(\boldsymbol{x}_i)},\right)^2}$$

by varying the degree r from 1 to 6.

In Table 4 we report the number s of m-tuples by varying the degree r of the local polynomial interpolant and the CPU time (in seconds) to reconstruct the surface in Figure 2 by using 412×164 grid points in the rectangle $[0,2060]\times[0,818]$ as evaluation points.

In Figure 2 we display the surface obtained by using the 95020 mock-Halton points and by evaluating the multinode Shepard operator on 412×164 grid points in the rectangle $[0,2060]\times[0,818]$.

The numerical results clearly show the convenience of the use of Algorithm 2 instead of Algorithm 1 since it produces approximations of the same accuracies of Algorithm 1 with

	number of m-tuples		CPU time (sec)	
r	Alg. 1	Alg. 2	Alg. 1	Alg. 2
1	94065	56592	808	262
2	95020	34826	1350	236
3	95018	25733	1644	241
4	95019	20200	2313	265
5	95019	16248	2891	271
6	95018	13339	3679	281

Table 4: Number of m-tuples varying the degree r of the local polynomial interpolant and CPU time in seconds to reconstruct the surface in Figure 2 by using, as evaluation points, 412×164 grid points in the rectangle $[0,2060]\times[0,818]$.

significantly reduced computation cost and CPU time. More precisely, Table 4 shows that Algorithm 2 reduces the number s of subsets σ_j as the degree r increases of a factor approximately equal to $\frac{n}{r+1}$, where n is the number of interpolation nodes.

3 Conclusions

We encourage scientists involved in image reconstruction to explore our MATLAB package `Multinode_Shepard` for enhanced accuracy and efficiency in scattered data interpolation. This package implements the multinode Shepard method, a significant advancement over the classical Shepard's method, providing robust solutions for interpolating complex datasets, with reduced computational cost and improved accuracy.

The multinode Shepard method, as demonstrated in our numerical experiments, significantly improves interpolation accuracy by utilizing local polynomial interpolants and inverse distance weighting functions. This method has been rigorously tested on real-world data, such as the Stromboli Volcano dataset, showing remarkable efficiency and precision.

By using our MATLAB package, you can benefit from:

- high accuracy in data interpolation, even with irregularly distributed datasets;
- efficient computation through optimized algorithms, reducing processing time and computational costs;
- robust performance validated through extensive numerical experiments.

The `Multinode_Shepard` package is freely available, and we invite you to access the source code and demonstrations via our GitHub repository: `Multinode_Shepard` GitHub [12]. This tool is designed to assist in your research, enabling more precise and reliable image reconstructions.

We welcome feedback and suggestions from the research community and are open to collaborations that can further enhance the capabilities and applications of the multinode Shepard method. Please feel free to reach out to us for any inquiries or potential joint projects.

Acknowledgments

This research has been achieved as part of RITA "Research ITalian network on Approximation", as part of the UMI group "Teoria dell'Approssimazione e Applicazioni" and was supported by INDAM-GNCS project 2024. The authors are members of the INdAM Research group GNCS.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

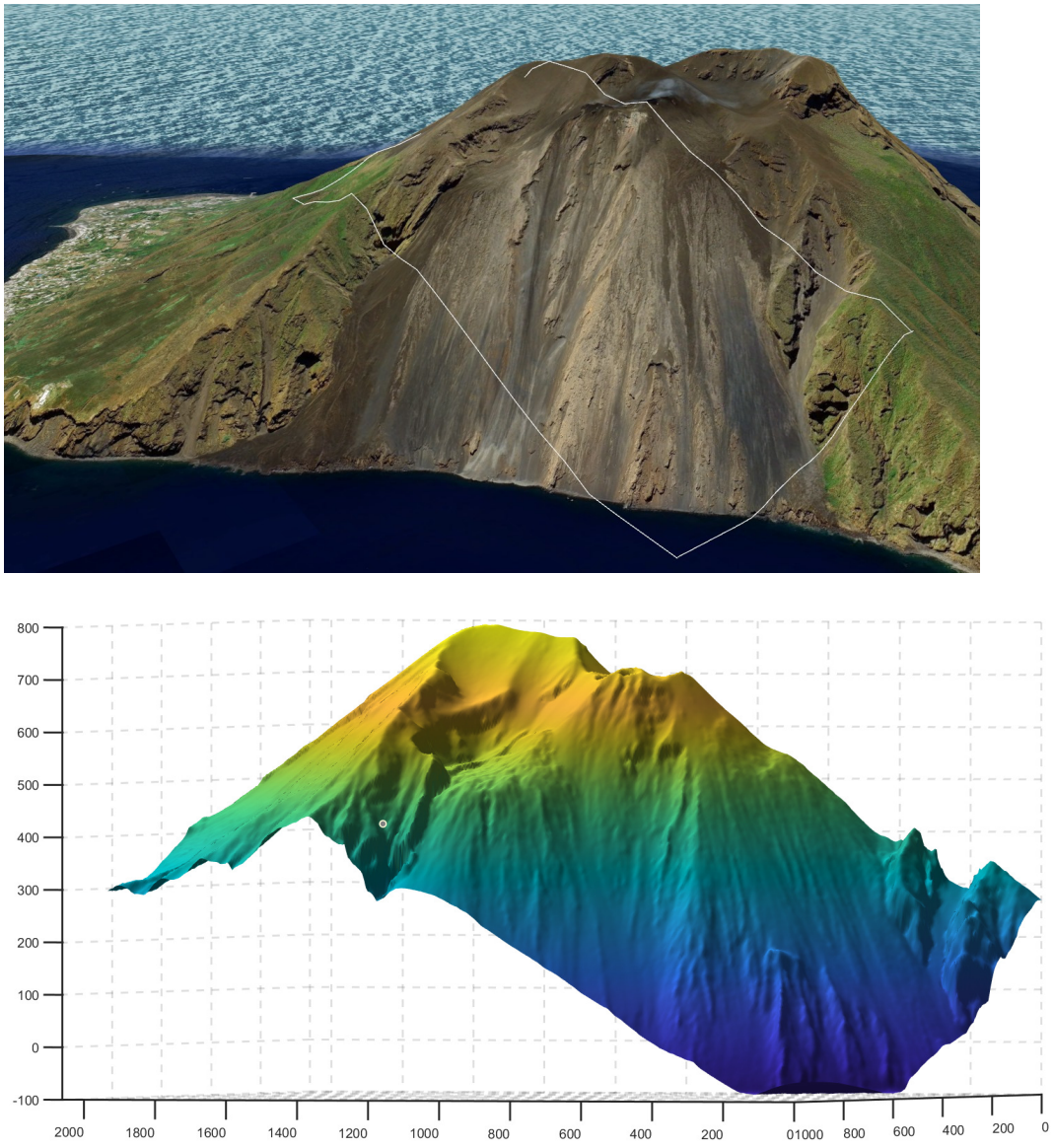


Figure 2: The Google Heart image of the Stromboli Volcano with the evidence of the zone from which the DEM data come from (top) and the reconstructed surface of the Stromboli Volcano (bottom) evaluated at a regular grid of 412×164 points in the rectangle $[0,2060] \times [0,818]$.

References

[1] L. Bos, S. De Marchi, A. Sommariva, and M. Vianello. Computing multivariate Fekete and Leja points by numerical linear algebra. *SIAM J. Numer. Anal.*, 48(5):1984–1999, 2010. ISSN 0036-1429,1095-7170. doi: 10.1137/090779024. URL https://doi.org/10.1137/090779024.

[2] B. Carr. Stromboli Volcano (Vents & Sciara del Fuoco): Italy, September 10, 2018. https://doi.org/10.5069/G9VX0DNG, 2019. Accessed: 2024-07-09.

[3] R. Cavoretto, A. De Rossi, F. Dell'Accio, and F. Di Tommaso. An efficient trivariate algorithm for tetrahedral Shepard interpolation. *J. Sci. Comput.*, 82(3):Paper No. 57, 15, 2020. ISSN 0885-7474,1573-7691. doi: 10.1007/s10915-020-01159-3. URL https://doi.org/10.1007/s10915-020-01159-3.

[4] F. Dell'Accio and F. Di Tommaso. Scattered data interpolation by Shepard's like methods: classical results and recent advances. *Dolomites Res. Notes Approx.*, 9(2):32–44, 2016. ISSN 2035-6803. doi: 10.14658/PUPJ-DRNA-2016-Special_Issue-5. URL https://drna.padovauniversitypress.it/2016/Special_Issue/5.

[5] F. Dell'Accio and F. Di Tommaso. Rate of convergence of multinode Shepard operators. *Dolomites Res. Notes Approx.*, 12(1):1–6, 2019. ISSN 2035-6803. doi: 10.1016/j.physletb.2018.10.073. URL https://doi.org/10.1016/j.physletb.2018.10.073.

[6] F. Dell'Accio and F. Di Tommaso. On the hexagonal Shepard method. *Appl. Numer. Math.*, 150:51–64, 2020. ISSN 0168-9274,1873-5460. doi: 10.1016/j.apnum.2019.09.005. URL https://doi.org/10.1016/j.apnum.2019.09.005.

[7] F. Dell'Accio, F. Di Tommaso, and K. Hormann. On the approximation order of triangular Shepard interpolation. *IMA J. Numer. Anal.*, 36(1):359–379, 2016. ISSN 0272-4979,1464-3642. doi: 10.1093/imanum/dru065. URL https://doi.org/10.1093/imanum/dru065.

[8] F. Dell'Accio, F. Di Tommaso, O. Nouisser, and N. Siar. Solving Poisson equation with Dirichlet conditions through multinode Shepard operators. *Comput. Math. Appl.*, 98:254–260, 2021. ISSN 0898-1221,1873-7668. doi: 10.1016/j.camwa.2021.07.021. URL https://doi.org/10.1016/j.camwa.2021.07.021.

[9] F. Dell'Accio, F. Di Tommaso, and N. Siar. On the numerical computation of bivariate Lagrange polynomials. *Appl. Math. Lett.*, 112:Paper No. 106845, 9, 2021. ISSN 0893-9659,1873-5452. doi: 10.1016/j.aml.2020.106845. URL https://doi.org/10.1016/j.aml.2020.106845.

[10] F. Dell'Accio, F. Di Tommaso, N. Siar, and M. Vianello. Numerical differentiation on scattered data through multivariate polynomial interpolation. *BIT*, 62(3):773–801, 2022. ISSN 0006-3835,1572-9125. doi: 10.1007/s10543-021-00897-6. URL https://doi.org/10.1007/s10543-021-00897-6.

[11] F. Dell'Accio, A. Sommariva, and M. Vianello. Random sampling and unisolvent interpolation by almost everywhere analytic functions. *Appl. Math. Lett.*, 145:Paper No. 108734, 6, 2023. ISSN 0893-9659,1873-5452. doi: 10.1016/j.aml.2023.108734. URL https://doi.org/10.1016/j.aml.2023.108734.

[12] F. Dell'Accio, F. Di Tommaso, and F. Larosa. Multinode Shepard method: the MATLAB Implementation. https://github.com/FilomenaDiTommaso85/Multinode_Shepard.git, 2024.

[13] R. Farwig. Rate of convergence of Shepard's global interpolation formula. *Math. Comp.*, 46(174):577–590, 1986. ISSN 0025-5718,1088-6842. doi: 10.2307/2007995. URL https://doi.org/10.2307/2007995.

[14] F. F. Little. Convex combination surfaces. In R. E. Barnhill and W. Boehm, editors, *Surfaces in computer aided geometric design - Proceedings of a Conference held at Oberwolfach, April 25–30, 1982*, volume 1479, pages 99–108. North-Holland Publishing Co., Amsterdam-New York, 1983.

[15] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *ACM '68: Proceedings of the 1968 23rd ACM National Conference*, pages 517–524, New York, NY, United States, 1968. Association for Computing Machinery. doi: 10.1145/800186.810616.