Qsurf: compressed QMC integration on parametric surfaces

G. Elefante [1],   A. Sommariva [1],   and   M. Vianello [1,⋆]

[1]*Department of Mathematics, University of Padova, Italy*

Abstract

We discuss a "bottom-up" algorithm for Tchakaloff-like compression of QMC (Quasi-MonteCarlo) integration on surfaces that admit an analytic parametrization. The key tools are Davis-Wilhelmsen theorem on the so-called "Tchakaloff sets" for positive linear functionals on polynomial spaces, and Lawson-Hanson algorithm for NNLS. This algorithm shows remarkable speed-ups with respect to Caratheodory-like subsampling, since it is able to work with much smaller matrices. We provide the corresponding Matlab code *Qsurf*, together with integration tests on regions of different surfaces such as sphere, torus, and a smooth Cartesian graph.

Keywords: Quasi-MonteCarlo formulas, surface integrals, analytic parametrization, low-discrepancy sequences, rejection sampling, Tchakaloff sets, quadrature compression, Davis-Wilhelmsen theorem, NonNegative Least Squares. (MSC2020: 65C05, 65D32)

1   Introduction

In the recent study [14], we have considered the compression problem for Quasi-MonteCarlo (QMC) surface integration on multibubbles (the surface of a ball union in $\mathbb{R}^3$), which can have a quite complicated structure. Indeed, numerical modelling with multibubbles is relevant in several applications, but compression of QMC integration seemed an overlooked approach, especially in the case of surface integrals.

In this paper, we extend such an approach to compressed QMC formulas for general integration problems on compact subsets of surfaces in $\mathbb{R}^3$, admitting an analytic parametrization. Such formulas preserve the approximation power of QMC up to the best uniform polynomial approximation error of a given degree to the integrand, but using a much lower number of sampling points.

⋆ Corresponding author: marcov@math.unipd.it

The key tools are Davis-Wilhelmsen theorem on the so-called "Tchakaloff sets" for positive linear functionals and Lawson-Hanson algorithm for NNLS, which allows to extract a set of "equivalent" re-weighted nodes from a huge uniformly distributed sequence with respect to the surface measure, by working in a "bottom-up" mode. Such a sequence can be obtained for example from a bivariate Halton sequence by an area-preserving map, when available, or by the probabilistic method of rejection sampling, which has been extended to the low-discrepancy deterministic setting, cf. e.g. [20, 31]. On the other hand, there are other relevant QMC point sequences on manifolds, see e.g. [2, 3].

The "bottom-up" approach shows remarkable speed-ups with respect to Caratheodory-like subsampling (cf. e.g. [16, 19, 21, 25, 29]), since it is able to work with much smaller matrices. We stress that one of the main difficulties consists in adapting the compression algorithm to work on the appropriate spaces of trivariate polynomials restricted to the surface, since the dimension of trivariate polynomial spaces can collapse in the case of algebraic surfaces.

The paper is organized as follows. In Section 2 we briefly discuss the theoretical background and the main idea of the "bottom-up" compression algorithm. Then, we sketch the algorithm, that has been implemented in Matlab, and comment on the main computational issues. Finally, in Section 3 we present some numerical examples concerning regions of sphere and torus, and the Cartesian graph of an analytic function. All the codes and demos are all freely available at [15].

2   QMC compression on surfaces

The possibility of compressing QMC integration rests on a somehow overlooked but relevant result of quadrature theory, originally proved by Davis [5] and then extended by Wilhelmsen [30]. Only recently this theorem has been rediscovered as a basic tool for positive cubature via adaptive NNLS moment-matching, cf. [13, 18, 26, 27].

Theorem 1. (Davis, 1967 - Wilhelmsen, 1976) Let $\{f_j\}_{1 \leq j \leq N}$ be continuous, real-valued, linearly independent functions defined on a compact set $\Omega \subset \mathbb{R}^d$, and $\mathscr{F} = span(f_1, \ldots, f_N)$. Assume that $\mathscr{F}$ satisfies the Krein condition (i.e. there is at least one $f \in \mathscr{F}$ which does not vanish on Ω) and that L is a positive linear functional on $\mathscr{F}$, i.e. $L(f) > 0$ for every $f \in \mathscr{F}$, $f \geq 0$ not vanishing everywhere in Ω.

If $\{P_i\}_{i=1}^{\infty}$ is an everywhere dense subset of Ω, then for sufficiently large m, the set $X_m = \{P_i\}_{i=1,\ldots,m}$ is a "Tchakaloff set", i.e. there exist weights $w_k > 0$, $k = 1, \ldots, \nu$, and nodes $\{Z_k\}_{k=1,\ldots,\nu} \subset X_m \subset \Omega$, with $\nu = card(\{Z_k\}) \leq N$, such that

$$L(f) = \ell(f) = \sum_{k=1}^{\nu} w_k f(Z_k) \,, \ \forall f \in \mathscr{F} \,. \tag{1}$$

Davis-Wilhelmsen theorem is a constructive generalization of the well-known Tchakaloff theorem [28] on the existence of positive quadrature formulas. But, just in view of its generality, it can be directly applied to a discrete functional like a QMC formula on $\Omega = \mathscr{J}$, $\mathscr{J}$ being a compact region of a surface $\mathscr{S} \subset \mathbb{R}^3$

$$L(f) = L_{\mathrm{QMC}}(f) = \frac{\sigma(\mathscr{J})}{M} \sum_{i=1}^{M} f(P_i) \approx \int_{\mathscr{J}} f \, d\sigma \,, \ f \in C(\mathscr{J}) \,, \tag{2}$$

where

$$X_M = \{P_i\}_{i=1,\ldots,M} \,, \ M > N \,,$$

is a uniformly distributed sequence on $\mathscr{J}$ and σ is the surface measure. Typically one generates a uniformly distributed sequence of cardinality say M_0 on the bounding surface $\mathscr{S} \supseteq \mathscr{J}$, from which sequence on $\mathscr{J}$ is extracted by a suitable in-domain algorithm. We observe that if $\sigma(\mathscr{J})$ is unknown or difficult to compute, it can be approximated as $\sigma(\mathscr{J}) \approx \sigma(\mathscr{S})M/M_0$.

Positivity of the functional for $f \in \mathscr{F} = \mathbb{P}_n^3(\mathscr{J})$ (the space of trivariate polynomials of total degree not exceeding n restricted to $\mathscr{J}$), is ensured whenever the set X_M is $\mathbb{P}_n^3(\mathscr{J})$-determining, i.e. a polynomial vanishing there vanishes everywhere on $\mathscr{J}$, or equivalently $dim(\mathbb{P}_n^3(X_M)) = N = dim(\mathbb{P}_n^3(\mathscr{J}))$, or even

$$rank(V_M) = N \ , \ \ V_M = V^{(n)}(X_M) = [f_j(P_i)] \in \mathbb{R}^{M \times N} \tag{3}$$

where V_M is the corresponding rectangular Vandermonde-like matrix. Notice that, X_M being a sequence, for every $k \leq M$ we have that

$$V_k = V^{(n)}(X_k) = [(V_M)_{ij}] \ , \ \ 1 \leq i \leq k \ , \ 1 \leq j \leq N \ . \tag{4}$$

We stress that the full rank requirement for V_M is not restrictive, in practice, when $\mathscr{S}$ is a surface that admits an analytic parametrization, the subset $\mathscr{J}$ is $\mathbb{P}_n^3(\mathscr{S})$-determining and the points are uniformly distributed with respect to the surface measure. Indeed, the probability that $det(V_N) = 0$ dealing with uniformly distributed points is null, as is ensured by the following proposition which is a special case of a general result proved in [7] in the case of continuous random point distributions.

Proposition 1. Let $\mathscr{S} \subset \mathbb{R}^3$ be a surface that admits an analytic parametrization $P = \Psi(u,v)$ from a connected open set $D \subset \mathbb{R}^2$, i.e. $\Psi = (\Psi_1, \Psi_2, \Psi_3)$ where $\Psi_i : D \to \mathbb{R}^3$ are analytic and $\Psi(D) = \mathscr{S}$. Moreover, let $\{f_j\}_{1 \leq j \leq N}$ be a basis of $\mathbb{P}_n^3(\mathscr{S})$ and $\{(u_i, v_i)\}_{i \geq 1}$ an equidistributed sequence on D with respect to any given probability density $\phi(u,v)$.

Then, the points $\{P_i = \Psi(u_i, v_i)\}_{1 \leq i \leq N}$ are almost surely unisolvent for polynomial interpolation in $\mathbb{P}_n^3(\mathscr{S})$.

Remark 1. *We can apply this proposition to the case where the parametrization is regular (so that the surface area element $\|\partial_u\Psi \times \partial_v\Psi\|_2/\sigma(\mathscr{J})$ is well-defined), $dim(\mathbb{P}_n^3(\mathscr{S})) = N = dim(\mathbb{P}_n^3(\mathscr{J}))$, and $d\sigma = \phi(u,v)\,du\,dv$ with density*

$$\phi(u,v) = I_{\mathscr{J}}(\Psi(u,v))\,\|\partial_u\Psi \times \partial_v\Psi\|_2/\sigma(\mathscr{J}) \ , \tag{5}$$

$I_{\mathscr{J}}$ denoting the indicator function of $\mathscr{J}$.

Remark 2. *To be rigorous, we should notice that Proposition 1 concerns random sequences, whereas here we deal with quasi-random sequences, where we can expect, and we have indeed verified experimentally, that the full-rank property of V_M in practice holds. In order to construct a sequence $\{P_i\}_{1 \leq i \leq N}$ that be uniformly distributed on $\mathscr{J}$ with respect to the surface measure, we can adopt the classical probabilistic method of* rejection sampling *on D applied to the density (5), that has been extended to low-discrepancy sequences; cf. [20, 31] with the references therein. Clearly, a suitable "in-domain" algorithm for $\mathscr{J}$ has to be at hand.*

Remark 3. *We recall that polynomial spaces can collapse on algebraic surfaces, i.e. it happens that $dim(\mathbb{P}_n^3(\mathscr{J})) = dim(\mathbb{P}_n^3(\mathscr{S})) < dim(\mathbb{P}_n^3(\mathbb{R}^3)) = (n+1)(n+2)(n+3)/6$. For example, if $\mathscr{J}$ is a subset with internal points w.r.t. the topology of the sphere S^2 (e.g. a spherical polygon as in the first example below), we have that $dim(\mathbb{P}_n^3(\mathscr{J})) = dim(\mathbb{P}_n^3(S^2)) = (n+1)^2$; we refer the reader, e.g., to [4] concerning the delicate matter of determining polynomial spaces dimension on algebraic varieties.*

In view of the results quoted above, when $M \gg N$ we can then try to find a Tchakaloff set $X_m \subset X_M$, with $N \leq m < M$, such that there exists a sparse nonnegative solution vector u to the underdetermined *moment-matching system*

$$V_m^t u = \lambda = V_M^t e \, , \quad e = \frac{\sigma(\mathscr{J})}{M} (1, \ldots, 1)^t \, . \tag{6}$$

In practice, we solve (6) via Lawson-Hanson active-set method [17] applied to the NNLS problem

$$\min_{u \geq 0} \|V_m^t u - \lambda\|_2 \, , \tag{7}$$

accepting the solution when the residual size is small, say

$$\|V_m^t u - \lambda\|_2 < \varepsilon \tag{8}$$

where ε is a given tolerance. Then the nonzero components of u provide nodes and weights of a compressed QMC formula extracted from X_m, that is $\{w_k\} = \{u_i : u_i > 0\}$ and $\{Z_k\} = \{P_i : u_i > 0\}$, giving

$$\ell_{\text{QMC}}(f) = \sum_{k=1}^{\nu} w_k f(Z_k) \, , \quad \nu \leq N \ll M \, , \tag{9}$$

where $\ell_{\text{QMC}}(f) = L_{\text{QMC}}(f)$ for every $f \in \mathbb{P}_n^3(\mathscr{J})$.

It is worth recalling that, in the case $m = M$, Caratheodory theorem on finite-dimensional conic combinations (applied to the columns of V_M^t) would ensure directly the existence of a Tchakaloff-like representation of the QMC functional (cf. [21] for a discussion on this point in the general framework of discrete measure compression by "Caratheodory-Tchakaloff subsampling"). In such a way, however, working with say an order of $10^5 - 10^6$ nodes, we would have to manage a huge matrix, that is we would have to solve the huge NNLS problem

$$\min_{u \geq 0} \|V_M^t u - \lambda\|_2 \, . \tag{10}$$

On the contrary, we can substantially reduce the computation cost by solving an increasing sequence of much smaller problems like (7) with $m := m_1, m_2, m_3, \ldots$ and $m_1 < m_2 < m_3 < \cdots \leq M$,

$$\min_{u \geq 0} \|V_{m_j}^t u - \lambda\|_2 \, , \quad j = 1, 2, 3, \ldots, \, m_1 \geq N \, , \tag{11}$$

corresponding to increasingly dense subsets $X_{m_1} \subset X_{m_2} \subset \cdots \subseteq X_M$, until the residual becomes sufficiently low. We may call this procedure a "*bottom-up*" approach to QMC compression. Indeed, as shown in [13], with a suitable choice of the sequence $\{m_j\}$ the residual becomes extremely small in few iterations, with a substantial speed-up with respect to (10).

Now, following [13] it is easy to derive the following error estimate

$$\left| \ell_{\text{QMC}}(f) - \int_{\mathscr{J}} f \, d\sigma \right| \leq \mathscr{E}_{\text{QMC}}(f) + 2\mu(\mathscr{J}) E_n(f; X_M)$$

$$\leq \mathscr{E}_{\text{QMC}}(f) + 2\mu(\mathscr{J}) E_n(f; \mathscr{J}) \, , \tag{12}$$

valid for every $f \in C(\mathscr{J})$, where $\mathscr{E}_{\text{QMC}}(f) = |L_{\text{QMC}}(f) - \int_{\mathscr{J}} f \, d\sigma|$ and we define $E_n(f; K) = \inf_{p \in \mathbb{P}_n^3(K)} \|f - p\|_{\infty, K}$ with K discrete or continuous compact set.

The meaning of (12) is that the compressed QMC functional ℓ_{QMC} retains the approximation power of the original QMC formula, up to a quantity proportional to the best polynomial approximation error to f in the uniform norm on X_M (and hence by inclusion in the uniform

norm on $\mathscr{J}$). We recall that the latter can be estimated depending on the regularity of f by multivariate Jackson-like theorems, cf. e.g. [22] for volume integrals where $\mathscr{J}$ is the closure of a bounded open set and [23] for the case of the sphere. On the other hand, we do not deepen here the topic of QMC convergence and error estimates, in particular on manifolds, referring the reader to specific papers and monographs, like e.g. [2, 3, 12] .

2.1 Algorithm description and computational issues

In this section we sketch the method implementation in the form of a pseudo-code and discuss its main computational features.

Algorithm Qsurf: *Bottom-up compression of QMC integration on a compact subset $\mathscr{J}$ of a surface $\mathscr{S} \subset \mathbb{R}^3$ with a regular analytic parametrization on a domain $D \subset \mathbb{R}^2$*

- input: the bounding surface measure $\sigma(\mathscr{S})$, possibly the measure $\sigma(\mathscr{J})$, the cardinality M_0 of a uniformly distributed sequence on $\mathscr{S}$, the cardinality increase factor $\theta > 1$, the moment-matching tolerance ε, the residual decrease threshold $\tau > 1$

(i) generate M_0 low-discrepancy points on the bounding surface $\mathscr{S} \supseteq \mathscr{J}$ (for example by rejection sampling on D w.r.t. the surface measure density) and extract the points $X = X_M = \{P_i\}_{i=1,\dots,M}$ that lie on $\mathscr{J}$ (by a suitable "in-domain" algorithm)

(ii) if unknown, approximate $\sigma(\mathscr{J})$ as $\sigma(\mathscr{J}) := \sigma(\mathscr{S})M/M_0$

(iii) % *selecting a basis of* $\mathbb{P}_n^3(X)$

     $(iii1)$ take a polynomial basis $\{p_1,\dots,p_{\mathscr{V}}\}$ of $\mathbb{P}_n^3$, $\mathscr{V} = \frac{(n+1)(n+2)(n+3)}{6}$

     $(iii2)$ compute the Vandermonde-like matrix $C := [p_j(P_i)] \in \mathbb{R}^{M \times \mathscr{V}}$

     $(iii3)$ compute $N := rank(C_{\mathscr{V}})$ where $C_{\mathscr{V}} = [(C)_{ij}]$, $1 \le i,j \le \mathscr{V}$

     $(iii4)$ compute the *QR* factorization with column pivoting $C_{\mathscr{V}}^{\pi} = QR$ where $\pi = (\pi_1,\dots,\pi_{\mathscr{V}})$ is the column permutation vector

     $(iii5)$ set $V_M := [(C)_{ij}]$, $1 \le i \le M$, $j = \pi_1,\dots,\pi_N$

(iv) compute the QMC moments $\lambda := V_M^t e$, $e = \sigma(\mathscr{J})/M (1,\dots,1)^t$

(v) % *bottom-up QMC compression*

     $(v1)$ initialize m, $N \le m \ll M$ and *momtype* $:= 0$

     $(v2)$ set $V_m := [(V_M)_{ij}]$, $1 \le i \le m$, $1 \le j \le N$

     $(v3)$ compute the *QR* factorization $V_m = Q_m R_m$

     $(v4)$ if *momtype* $= 0$ then

         – compute the modified QMC moments $q_m = (R_m^{-1})^t \lambda$ by solving the system $R_m^t q_m = \lambda$ via Gaussian elimination with row pivoting

         – set $A_m = Q_m$

         else

- compute the modified QMC moments $q_m = (R_m^{-1})^t \lambda = (V_M R_m^{-1})^t e$ as $q_m := A_M^t e$, by solving the matrix equation $R_m^t A_M^t = V_M^t$ via Gaussian elimination with row pivoting
- set $A_m = [(A_M)_{i,j}]$, $1 \le i \le m$, $1 \le j \le N$

($v5$) compute a sparse solution u to the NNLS problem

$$\min_{u \ge 0} \|A_m^t u - q_m\|_2$$

by Lawson-Hanson active-set algorithm

($v6$) compute the relative residual $res := \|V_m^t u - \lambda\|_2 / \|\lambda\|_2$

($v7$) if $res_0/res > \tau$ & $m < M$ then
 ($v7a$) if $momtype = 0$ then
  - set $momtype := 1$ and `goto` ($v2$)
  `else`
  - set $m := M$ and `goto` ($v2$)
($v8$) if $res > \varepsilon$ & $\lceil \theta m \rceil \le M$ then
  - set $m := \lceil \theta m \rceil$, $res_0 := res$ and `goto` ($v2$)

(vi) select the indexes $J = \{i : u_i > 0\}$ and set $w := u(J)$ and $Z := X(J)$

- `output`: the weights $\{w_k\}$ and nodes $\{Z_k\} \subset X$ of a compressed QMC formula on $\mathscr{J}$ with moment-matching residual res

Now, some observations on delicate aspects are in order. Step (iii) is a key point in the case of surface integration. As for the starting polynomial basis, for conditioning problems we adopt the product Chebyshev total-degree basis of the smaller bounding box say $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \supset X$, namely

$$p_j(x,y,z) = T_{\alpha_1(j)}(\sigma_1(x)) \cdot T_{\alpha_2(j)}(\sigma_2(y)) \cdot T_{\alpha_3(j)}(\sigma_3(z)), \; j = 1, \ldots, \mathscr{V},$$

$$\sigma_i : [a_i, b_i] \mapsto [-1, 1], \; \sigma_i(t) = \frac{2t - b_i - a_i}{b_i - a_i}, \; i = 1, 2, 3,$$

where $j \mapsto \alpha(j)$ corresponds to the graded lexicographical ordering of the 3-indexes $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, $0 \le \alpha_1 + \alpha_2 + \alpha_3 \le n$.

Moreover, we recall that $dim(\mathbb{P}_n^3(X))$ is simply the rank of the corresponding rectangular Vandermonde-like matrix C. In step ($iii4$), instead, we work with the principal square submatrix $C_{\mathscr{V}}$. As already observed in Section 2, with $\mathscr{V} \ge dim(\mathbb{P}_n^3(\mathscr{J}))$ uniformly distributed points on $\mathscr{J}$, the probability that such a rank be lower than $dim(\mathbb{P}_n^3(\mathscr{J}))$ is null, so that "almost-surely" Wilhelmsen theorem applies. In Matlab, one can use directly the built-in function `rank` based on an economy-size version of SVD. Notice that we are using a numerical rank (obtained by discarding the singular values below a tolerance close to machine precision), not the true rank. Nevertheless, dealing with polynomials restricted to X this is numerically equivalent to work, up to very small errors, with the true polynomial space. We stress that when $\mathscr{V} \ll M$, using $C_{\mathscr{V}}$ instead of C gives experimentally a substantial speed-up to the rank computation, by a factor roughly of the order of $M/\mathscr{V}$.

The polynomial basis selection, i.e. the determination of a set of linearly independent polynomials on $\mathscr{J}$ within the starting basis, is performed in ($iii4$) by a QR factorization with column pivoting of the Chebyshev-Vandermonde matrix $C_{\mathscr{V}}$ (again, an economy-size version

can be used in Matlab that produces only the first N columns of Q and a column permutation vector). In such a way we select a polynomial basis of $\mathbb{P}_n^3(X)$ by the first N components $\pi_1, \dots, \pi_N$ of the column permutation, say $(f_1, \dots, f_N) = (p_{\pi_1}, \dots, p_{\pi_N})$.

We can now turn to the second key step of the algorithm, that is the extraction of a compressed QMC formula in (v). As already observed, this is based on Wilhelmsen theorem, using just X_M as extraction set, in a "bottom-up" fashion. This procedure avoids working directly on the complete matrix V_M (cf. (10)), as done instead in other previous approaches to QMC compression like [6], cf. also the discussion in [1, 13]. Indeed, the overall number of points, i.e. of rows of V_M, can be huge, up to the order of $10^5 - 10^6$. In practice, we proceed along increasingly dense subsequences of the overall sequence, solving the corresponding NNLS problems and stopping when the relative moment-matching residual becomes sufficiently small.

To this purpose the classical Lawson-Hanson iterative method turns out to be a good choice, since it automatically seeks a sparse solution with a number of nonzeros not exceeding N. The method is implemented in most numerical programming environments, e.g. in Matlab by the built-in function `lsqnonneg`. On the other hand, there are improvements of the algorithm, cf. for example [24] for a survey, and the recent implementation named LHDM based on the concept of "Deviation Maximization" instead of "column pivoting" for the underlying QR factorizations, cf. [8, 9]. Indeed, in the present framework we have adopted LHDM, since it gives experimentally a speed-up of at least 2 with respect to `lsqnonneg`.

In order to cope ill-conditioning of the matrices used in the sequence of NNLS problem, that worsens increasing the degree, we perform an orthogonalization of V_m by QR factorization, that corresponds to work with the discrete orthogonal basis $(f_1, \dots, f_N) R_m^{-1}$. Such a basis is orthogonal with respect the counting measure supported at X, i.e. with respect to the discrete scalar product $\langle f, g \rangle_{X_m} = \sum_{i=1}^m f(P_i) g(P_i)$. Consequently, the original QMC moments have to be modified as in $(v4)$.

It should be stressed that, due to the inherited ill-conditioning of the triangular factor R_m by V_m, that increases with the degree, explicit inversion of R_m in $(v4)$ is avoided by solving linear systems via Gaussian elimination with row pivoting (that is in Matlab simply by applying the backslash operator).

We also notice that the complete matrix V_M is used only to compute the QMC moments in (iv), unless $(v7a)$ has to be followed due to a residual decrease factor below the required threshold. Such a phenomenon turns out to occur seldom with high degrees and strong ill-conditioning. In such a case, computation of $A_M = V_M R_m^{-1}$ becomes the computational bulk slowing down the whole process.

3   Numerical tests and demos

In order to show the effectiveness of the bottom-up compression procedure of QMC surface integration, we present some numerical tests, where we compare "Caratheodory-Tchakaloff" compression of multivariate discrete measures as implemented in the general-purpose package *dCATCH* [10], with the bottom-up approach described above. The Matlab codes and demos, collected in a package named *Qsurf*, are freely available at [15].

In all the tests we have set the parameters of the algorithm to $\varepsilon = 10^{-10}$, $\theta = 2$, $\tau = 10$, and m has been initialized to $2N$. The tests have been performed with a CPU AMD Ryzen 5 3600 with 48 GB of RAM, running Matlab R2022a.

We point out that in all the numerical experiments we have adopted Halton points but alternatively the software provides the usage of other sequences (e.g. Sobol sets).

3.1 Sphere region

In the first example we consider a large region $\mathscr{J}$ of the sphere, namely a *spherical polygon* (a polygon whose vertices are on the sphere and whose sides are great circle arcs) representing an approximation of continental Africa (see Fig. 1). In this case it is convenient to choose a spherical cap (say $\mathscr{C} \supset \mathscr{J}$) centered at the polygon centroid as bounding surface, $\mathscr{S} = \mathscr{C}$, and we can apply a rotation to the sphere in such a way that the centroid is at the north pole (this does not clearly affect surface integration on the region).

The indicator function of $\mathscr{J}$ can be easily implemented by stereographic projection from the south pole on the tangent plane at the north pole, that generates a planar polygon for which the Matlab `inpolygon` works quite efficiently. Observe that this procedure can be applied to any rotated spherical polygon that does not contain the south pole.

Then, we can parametrize the polar cap by the area-preserving map (i.e., $\|\partial_u \Psi \times \partial_v \Psi\|_2 = 1$)

$$\Psi(u,v) = r(\sqrt{1-u^2}\cos(v), \sqrt{1-u^2}\sin(v), u) \,, \tag{13}$$

$(u,v) \in D = (c,1) \times (0,2\pi)$, where r is the sphere radius and c is the z-quote of the cap boundary (in practice, working with the open rectangle D we loose the Greenwich 0-meridian arc cutting the cap, that has null surface measure and thus surface integration is not affected).

Now, starting from low-discrepancy points in D, e.g. Halton points, we get low-discrepancy points on the cap $\mathscr{S}$ and finally on the spherical polygon $\mathscr{J}$. On the other hand, Proposition 1 substantially applies since the map Ψ is analytic and regular on D (see also Remarks 1-2), and hence we can resort to the bottom-up algorithm *Qsurf* in order to compress QMC integration on a huge number M of mapped low-discrepancy points in $\mathscr{J}$. To the purpose of illustration, in Figure 1 we show the distribution of 64 compressed QMC points extracted from about 2400 Halton points, still matching the QMC moments on $\mathscr{J}$, up to degree 7.

In Table 1 we report the results obtained by applying the QMC compression with more than one million points on the spherical polygon, taking degrees $n = 3, 6, 9, 12, 15$, and accepting (8) with a tolerance $\varepsilon = 10^{-10}$. In particular, we display the cardinalities and compression ratios, the cpu-times for the construction of the low-discrepancy sequence (cpu Halton seq.) and those for the computation of the compressed rules.

The advantage of the new approach is two-fold, since in all the tests an inferior cputime with respect to *dCATCH* is required to determine the compressed rule and, differently from *dCATCH*, the solution of (7) always satisfies the moment residual criterion (8). In addition, less memory is necessary due to the inherent structure of the bottom-up approach, which works on much smaller matrices.

Finally, in Table 2, we approximate the integrals $\int_{\mathscr{J}} g_k d\sigma$ on three test functions, namely setting $P = (x,y,z)$

$$\begin{align} g_1(P) &= \exp(-\|P-P_0\|_2) \tag{14}\\ g_2(P) &= \cos(x+y+z) \tag{15}\\ g_3(P) &= \|P-P_0\|_2^5 \tag{16} \end{align}$$

P_0 being the centroid of the spherical polygon $\mathscr{J}$. The reference values of the integrals have been computed by a QMC rule with very high cardinality (more than 20 million points). We display the relative errors of the QMC rule with more than one million points and of the two proposed compressions. As expected from estimate (12), by increasing the QMC moment-matching degree the errors tend to stabilize around the underlying QMC error.
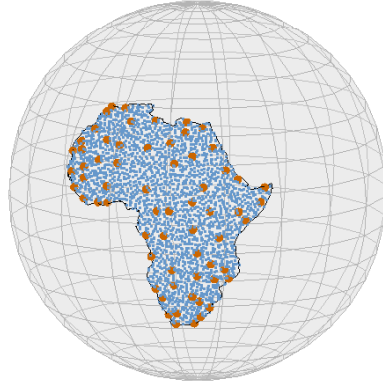
Figure 1: 64 compressed QMC points (red) at exactness degree $n = 7$, extracted from about 2400 mapped Halton points (blue) on the surface of a spherical polygon approximating continental Africa.

3.2   Torus region

The second example concerns surface integration on a region $\mathscr{J}$ of a torus $\mathscr{T}$, corresponding to a section by a plane, excluding the points that are internal to a ball intersecting the torus; see Fig. 2. In particular, we consider the torus with center $(0,0,0)$ and radii $r = 2$, $R = 3$, cut by the ball $B((0,4,0),\sqrt{6})$ and the upper half-space of $\mathbb{R}^3$ w.r.t. the plane of equation $-x/4 + y + 4z = 0$.

In this case it is not straightforward to apply a standard integrator, since one should track the domain $\Psi^{-1}(\mathscr{J})$ in D and then apply there a suitable cubature rule. On the contrary, QMC integration can be more easily constructed by rejection sampling in standard toroidal coordinates (here the bounding surface $\mathscr{S} = \mathscr{T}$ is the whole torus)

$$\Psi(u,v) = ((R + r\cos(u))\cos(v), (R + r\cos(u))\sin(v), r\sin(u)) , \qquad (17)$$

$(u,v) \in D = (0,2\pi) \times (0,2\pi)$, where R and r are the big and small torus radii respectively, and $\|\partial_u \Psi \times \partial_v \Psi\|_2 = r(R + r\cos(v))$. Observe that considering the open rectangle Ω we loose the possible intersection of $\mathscr{J}$ with two circles, that have null surface measure and do not affect surface integration. Moreover, the indicator function of $\mathscr{J}$ can be implemented by the simple inequalities that describe an half-space determined by the cutting plane, and the interior of the ball. Again, the map Ψ is analytic and regular so that Proposition 1 with Remarks 1-2 applies and algorithm *Qsurf* can be used. In Figure 2 we show the distribution of 64 compressed QMC points, extracted from about 8000 mapped Halton points after selection by rejection sampling w.r.t. the surface measure density, still matching the QMC moments on $\mathscr{J}$ up to degree 7.

In Table 3 we again report the results obtained by applying QMC compression with more than one million points on the region $\mathscr{J}$. As for the spherical polygon, we consider degrees $n = 3,6,9,12,15$, accepting (8) with a tolerance $\varepsilon = 10^{-10}$. In all the tests an inferior cputime is required by *Qsurf* to determine the compressed rule and, while *dCATCH* fails for degree $n = 15$. Moreover, the solution by the new approach to (7) always satisfies the moment residual criterion (8).

deg	3	6	9	12	15
card. QMC	$M = 1{,}184{,}341$				
card. dCATCH	16	49	98	165	239
card. Qsurf	16	49	100	169	256
compr. ratio	7.4e+04	2.4e+04	1.1e+04	7.0e+03	4.6e+03
cpu Halton seq.	4.53e+01s				
cpu dCATCH	4.1e+00s	1.5e+01s	4.6e+01s	1.3e+02s	3.1e+02s
cpu Qsurf	3.8e-01	1.2e+00s	3.1e+00s	6.4e+00s	1.3e+01
speed-up	10.8	12.5	14.8	20.3	23.8
mom. resid. dCATCH	4.3e-12	4.3e-12	⋆ 2.9e-04	⋆ 6.2e-04	⋆ 2.0e-03
mom. resid. Qsurf					
iter. 1	3.7e-16	8.5e-01	2.9e+01	7.9e+01	1.4e+01
iter. 2		6.5e-02	1.7e-04	3.8e-02	7.6e-01
iter. 3		6.3e-16	1.1e-15	1.3e-15	1.7e-02
iter. 4					2.8e-15

Table 1: QMC compression by with more than one million points on a spherical polygon approximating continental Africa.

deg	3	6	9	12	15
$E^{QMC}(g_1)$	3.0e-05				
$E^{dCATCH}(g_1)$	1.0e-03	3.1e-05	2.7e-05	3.2e-05	1.7e-05
$E^{Qsurf}(g_1)$	1.2e-04	3.0e-05	3.0e-05	3.0e-05	3.0e-05
$E^{QMC}(g_2)$	1.5e-05				
$E^{dCATCH}(g_2)$	8.9e-05	1.5e-05	2.5e-05	4.9e-07	3.7e-06
$E^{Qsurf}(g_2)$	1.4e-05	1.5e-05	1.5e-05	1.5e-05	1.5e-05
$E^{QMC}(g_3)$	8.6e-04				
$E^{dCATCH}(g_3)$	2.4e-02	1.3e-03	7.8e-04	8.2e-04	5.8e-04
$E^{Qsurf}(g_3)$	2.3e-02	7.7e-04	8.3e-04	8.6e-04	8.6e-04

Table 2: Relative integration errors for the three test functions (14)-(16) on a spherical polygon approximating continental Africa, by means of QMC, dCATCH and Qsurf.

Lastly, in Table 4 we approximate the value of $\int_{\mathscr{S}} g_k \, d\sigma$, $k = 1, 2, 3$, with the same functions defined in (14)-(16) and $P_0 = (0, -3, 2)$. The reference values of the integrals have been computed by means of a QMC rule with very high cardinality (more than 20 million points). We display the relative errors of the QMC rule with about one million points and of the two proposed compressions. Notice again that, as expected from estimate (12), by increasing the QMC moment-matching degree the errors tend to stabilize around the underlying QMC error.

3.3 Cartesian graph

In the third example we consider as a regular surface $\mathscr{S}$ the Cartesian graph of an analytic function, namely the popular Franke's surface, which is the graph of a linear combination of Gaussians

$$F(u,v) = \frac{3}{4} e^{-\frac{1}{4}((9u-2)^2 + (9v-2)^2)} + \frac{3}{4} e^{-\frac{1}{49}((9u+1)^2 + (9v+1)^2)}$$

$$+ \frac{1}{2} e^{-\frac{1}{4}((9u-7)^2 + (9v-3)^2)} - \frac{1}{5} e^{-((9u-4)^2 + (9v-7)^2)}, \tag{18}$$

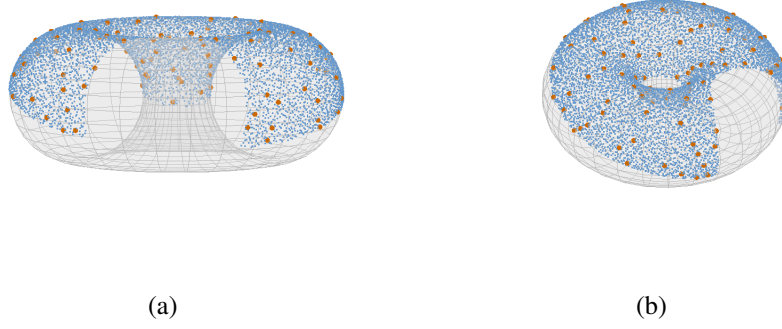(a)   (b)

Figure 2: 100 compressed QMC points (red) at exactness degree $n = 7$, extracted from about 8000 mapped Halton points (blue) by rejection sampling on a torus region, determined by a cutting ball and plane (view from different perspectives).

deg	3	6	9	12	15
card. *QMC*			$M = 1,006,200$		
card. *dCATCH*	20	74	164	290	450
card. *Qsurf*	20	74	164	290	452
compr. ratio	5.0e+04	1.3e+04	6.1e+03	3.5e+03	2.2e+03
cpu Halton seq.			1.0e+01s		
cpu *dCATCH*	2.8e+00s	1.6e+01s	4.4e+01s	1.2e+02s	3.0e+02s
cpu *Qsurf*	2.7e-01s	9.9e-01s	2.9e+00s	6.3e+00s	2.2e+01s
speed-up	10.4	16.2	15.2	19.0	13.6
mom. resid. *dCATCH*	1.2e-11	1.2e-11	1.2e-11	1.2e-11	⋆ 9.1e-07
mom. resid. *Qsurf*					
iter. 1	3.0e-16	8.9e-01	1.3e+00	6.4e+00	2.5e+01
iter. 2		1.1e-15	1.9e-15	2.6e-01	1.3e-01
iter. 3				3.3e-15	4.5e-15

Table 3: QMC compression with more than one million points on the torus region in Fig. 2.

$(u, v) \in D = (0,1) \times (0,1)$.

We take two regions of such a surface, the first determined by a cutting ball and plane, whereas the second is a disconnected one determined by three cutting balls; see Figs. 3 and 4. Again, the map Ψ is analytic and regular, since $\|\partial_u \Psi \times \partial_v \Psi\|_2 = \sqrt{1 + (\partial_u F)^2 + (\partial_v F)^2}$, so that Proposition 1 with Remarks 1-2 applies and algorithm *Qsurf* can be used.

The numerical tests are collected in Tables 5-8, and show results that are in line with those of the previous examples, apart from the fact that the numerically determined dimension of the trivariate polynomial spaces does not collapse on the surface (at least up to degree 9). This is expected since Franke's surface is a transcendental, i.e. not algebraic, surface. Notice in particular that at degrees 9, 12, 15, *dCATCH* fails to reach the required residual tolerance, whereas *Qsurf* always succeeds in at most 4-5 iterations.

Remark 4. *In the numerical code there are some parameters ε, θ and τ that must be specified.*

The cardinality increase factor θ, becomes important when compression is not achieved, using the current subset of the QMC nodes. A small θ would propose a new pointset that may fail at the next stage because not dense enough, while a large value would instead define a too dense set, possibly increasing the algorithm cputime. Thus we decided as reasonable choice to
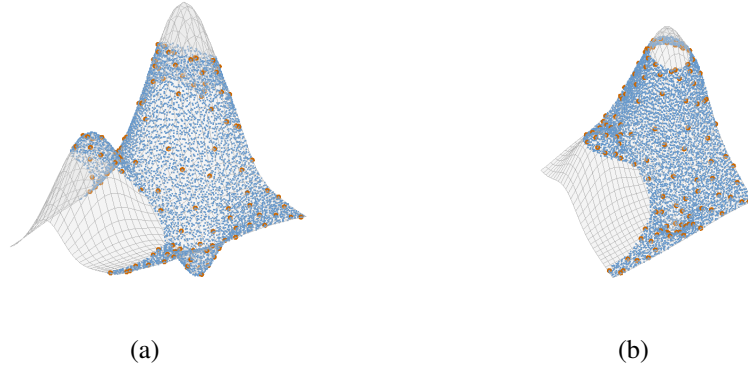
(a)   (b)

Figure 3: 120 compressed QMC points (red) at exactness degree $n = 7$, extracted from about 6500 mapped Halton points (blue) by rejection sampling on a Franke's surface region, determined by a cutting ball and plane (view from different perspectives).
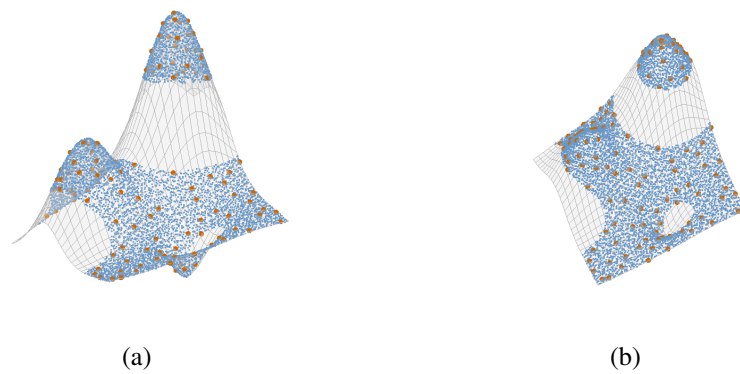

(a)   (b)

Figure 4: 120 compressed QMC points (red) at exactness degree $n = 7$, extracted from about 6500 mapped Halton points (blue) by rejection sampling on a Franke's surface disconnected region, determined by three cutting balls (view from different perspectives).

deg	3	6	9	12	15
$E^{QMC}(g_1)$	1.7e-04				
$E^{dCATCH}(g_1)$	3.5e-01	1.2e-02	2.5e-03	2.2e-04	2.2e-04
$E^{Qsurf}(f_1)$	5.5e-01	6.5e-02	2.4e-03	5.4e-04	1.5e-04
$E^{QMC}(g_2)$	2.4e-04				
$E^{dCATCH}(g_2)$	3.5e-01	2.5e-01	7.2e-03	1.3e-04	2.4e-04
$E^{Qsurf}(f_2)$	1.7e+00	1.3e-01	1.5e-03	1.8e-04	2.4e-04
$E^{QMC}(g_3)$	5.2e-06				
$E^{dCATCH}(f_3)$	4.3e-03	2.3e-06	5.2e-06	5.2e-06	5.2e-06
$E^{Qsurf}(g_3)$	8.0e-03	3.1e-06	5.2e-06	5.2e-06	5.2e-06

Table 4: Relative errors for the three test functions (14)-(16) on the torus region of Fig. 2, by means of QMC, *dCATCH* and *Qsurf*.

deg	3	6	9	12	15
card. *QMC*	$M = 1,293,600$				
card. *dCATCH*	20	84	212	407	586
card. *Qsurf*	20	84	220	442	701
compr. ratio	6.5e+04	1.5e+04	5.9e+03	2.9e+03	1.8e+03
cpu Halton seq.	1.8e+00s				
cpu *dCATCH*	3.6e+00s	2.1e+01s	5.7e+01s	1.8e+02s	4.9e+02s
cpu *Qsurf*	3.5e-01s	1.4e+00s	3.8e+00s	2.0e+01s	2.7e+01s
speed-up	10.3	15.0	15.0	9.0	18.1
mom. resid. *dCATCH*	7.6e-12	7.6e-12	$\star$ 8.9e-04	$\star$ 2.9e-03	$\star$ 5.9e-03
mom. resid. *Qsurf*					
iter. 1	1.9e-16	5.5e-01	1.5e+00	1.4e+01	3.4e+01
iter. 2		1.0e-15	1.8e-15	3.9e-01	1.9e+00
iter. 3				1.2e-02	4.2e-15
iter. 4				2.5e-15	

Table 5: QMC compression with more than one million points on on the Franke's surface region in Fig. 3.

set $\theta = 2$, so doubling the cardinality of the pointset in case of failure.

To illustrate the effect of this choice, we consider the torus region in Figure 2, for degree of exactness 6 and different values of θ, that is $\theta = 1.2, 1.4, \ldots, 5$. In Figure 5 we take as ordinate the median of the cputime of the bottom-up QMC compression over 10 tests. In such experiment, values of θ in the interval $[2, 3]$ show a good behavior, justifying our choice.

Concerning the relative moment matching tolerance ε, as default we used $\varepsilon = 10^{-10}$, in view of the typical approximation quality of the QMC approximation. However if the initial cardinality of the QMC rule is sufficiently large, there is numerical evidence that for mild degrees of precision the value of ε can be set smaller, e.g. $\varepsilon = 10^{-14}$.

Finally, recalling that the relative residual decrease threshold $\tau > 1$ is important to determine if the residual error made by successive stages of the algorithm, say res_{old} and res_{new}, is stagnating, we set as default $\tau = 10$, so detecting such an event when $res_{old} < 10\, res_{new}$. We decided for this option, since a much higher value would indicate stagnation even when it does not happen, while a much smaller value would rarely detect such a situation.

deg	3	6	9	12	15
card. *QMC*	$M = 1,305,444$				
card. *dCATCH*	20	84	212	405	612
card. *Qsurf*	20	84	220	448	735
compr. ratio	6.5e+04	1.6e+04	5.9e+03	2.9e+03	1.8e+03
cpu Halton seq.	1.74e+00s				
cpu *dCATCH*	3.8e+00 s	2.2e+01	5.2e+01	1.9e+02s	5.0e+02s
cpu *Qsurf*	3.6e-01	1.4e+00	9.0e+00	3.9e+01	6.3e+01
speed-up	10.6	15.7	5.8	4.9	7.9
mom. resid. *dCATCH*	1.8e-12	1.8e-12	⋆ 8.2e-04	⋆ 2.4e-03	⋆ 3.4e-03
mom. resid. *Qsurf*					
iter. 1	6.0e-16	7.4e-02	7.7e-01	3.2e+01	3.4e+01
iter. 2		1.2e-15	2.4e-01	4.0e-01	6.1e+00
iter. 3			2.4e-01	8.5e-02	6.1e+00
iter. 4			1.6e-11	8.5e-02	9.2e-12
iter. 5				7.9e-12	

Table 6: QMC compression with more than one million points on the on the Franke's surface disconnected region in Fig. 4.

deg	3	6	9	12	15
$E^{QMC}(g_1)$	1.2e-05				
$E^{dCATCH}(g_1)$	4.4e-03	1.1e-05	1.5e-06	8.1e-06	4.6e-05
$E^{Qsurf}(g_1)$	6.7e-04	9.6e-06	1.2e-05	1.2e-05	1.2e-05
$E^{QMC}(g_2)$	3.0e-07				
$E^{dCATCH}(g_2)$	7.8e-05	2.9e-07	3.6e-05	1.2e-04	6.3e-05
$E^{Qsurf}(g_2)$	4.0e-05	3.0e-07	3.0e-07	3.0e-07	3.0e-07
$E^{QMC}(g_3)$	6.0e-05				
$E^{dCATCH}(g_3)$	1.4e-01	3.8e-05	1.1e-04	2.8e-05	1.2e-04
$E^{Qsurf}(g_3)$	2.2e-02	1.1e-04	6.1e-05	6.0e-05	6.0e-05

Table 7: Relative integration errors for the three test functions (14)-(16) on the Franke's surface region of Fig. 3, by means of QMC, *dCATCH*, *Qsurf*.
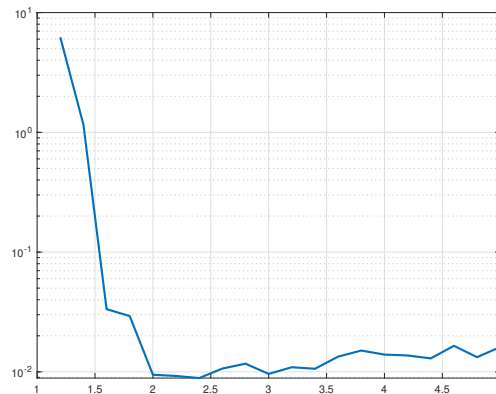

Figure 5: Median of the cputime of the bottom-up QMC compression with degree of exactness $n = 6$, over 10 tests, on a subregion of the torus, for several values of θ.

deg	3	6	9	12	15
$E^{QMC}(g_1)$			1.3e-06		
$E^{dCATCH}(g_1)$	4.6e-03	2.3e-06	1.7e-05	8.1e-06	2.1e-05
$E^{Qsurf}(g_1)$	2.1e-03	9.7e-07	1.3e-06	1.3e-06	1.3e-06
$E^{QMC}(g_2)$			6.6e-05		
$E^{dCATCH}(g_2)$	5.3e-05	6.6e-05	6.7e-05	9.1e-05	4.8e-05
$E^{Qsurf}(g_2)$	1.1e-04	6.6e-05	6.6e-05	6.6e-05	6.6e-05
$E^{QMC}(g_3)$			1.7e-06		
$E^{dCATCH}(g_3)$	2.3e-01	4.1e-05	1.6e-04	6.3e-05	8.1e-05
$E^{Qsurf}(g_3)$	1.1e-01	4.0e-05	1.3e-06	1.7e-06	1.7e-06

Table 8: Relative integration errors for the three test functions (14)-(16) on the Franke's surface disconnected region of Fig. 4, by means of QMC, *dCATCH*, *Qsurf*.

4 Software

We have implemented and tested in Matlab all the described routines.

The demos `demo_CQMC_sphpoly`, `demo_CQMC_torus`, `demo_CQMC_franke` illustrate the numerical experiments performed in the previous section. Their structure is essentially similar and can be modified to treat other subsets and/or parametric surfaces, adapting the function `pts_domain` to the new instance. This corresponds to items (*i*) and (*ii*) of *Algorithm Qsurf*.

The routine `cqmc_v2` implements its remaining items from (*iii*) to (*vi*). To this purpose, the basis selection in (*iii*) is obtained by means of the function `dCHEBVAND_v2`, while the computation of a sparse solution in (*v5*) is achieved by an user's choice implementation of the Lawson-Hanson algorithm (namely, the Matlab built-in function `lsqnonneg` or the alternative open-source codes `lawsonhanson` and `LHDM` proposed respectively in [24] and [9]). Moreover, having in mind to compare algorithm Qsurf with previous approaches, we also provide the routine `dCATCH` from [11], which implements Caratheodory-like compression via NNLS.

The open source software is available at [15].

Acknowledgements

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] C. Bittante, S. De Marchi, and G. Elefante. *A new quasi-Monte Carlo technique based on nonnegative least squares and approximate Fekete points*. Numer. Math. Theory Methods Appl., vol. 9, 4, (2016), 640–663. ISSN 1004-8979,2079-7338. URL http://dx.doi.org/10.4208/nmtma.2016.m1516

[2] L. Brandolini, C. Choirat, L. Colzani, G. Gigante, R. Seri, and G. Travaglini. *Quadrature rules and distribution of points on manifolds*. Ann. Sc. Norm. Super. Pisa Cl. Sci. (5), vol. 13, 4, (2014), 889–923. ISSN 0391-173X,2036-2145. URL http://dx.doi.org/10.2422/2036-2145.201103_007

[3] J. S. Brauchart, E. B. Saff, I. H. Sloan, and R. S. Womersley. *QMC designs: optimal order quasi Monte Carlo integration schemes on the sphere*. Math. Comp., vol. 83, 290, (2014), 2821–2851. ISSN 0025-5718,1088-6842. URL http://dx.doi.org/10.1090/S0025-5718-2014-02839-1

[4] D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics, (Springer, Cham2015), fourth edn. ISBN 978-3-319-16720-6; 978-3-319-16721-3. An introduction to computational algebraic geometry and commutative algebra, URL http://dx.doi.org/10.1007/978-3-319-16721-3

[5] P. J. Davis. *A construction of nonnegative approximate quadratures*. Math. Comp., vol. 21, (1967), 578–582. ISSN 0025-5718,1088-6842. URL http://dx.doi.org/10.2307/2005001

[6] S. De Marchi and G. Elefante. *Quasi-Monte Carlo integration on manifolds with mapped low-discrepancy points and greedy minimal Riesz s-energy points*. Appl. Numer. Math., vol. 127, (2018), 110–124. ISSN 0168-9274,1873-5460. URL http://dx.doi.org/10.1016/j.apnum.2017.12.017

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

[8] M. Dessole, M. Dell'Orto, and F. Marcuzzi. *The Lawson-Hanson algorithm with deviation maximization: finite convergence and sparse recovery*. Numer. Linear Algebra Appl., vol. 30, 5, (2023), Paper No. e2490, 19. ISSN 1070-5325,1099-1506. URL http://dx.doi.org/10.1002/nla.2490

[9] M. Dessole, F. Marcuzzi, and M. Vianello. *Accelerating the Lawson-Hanson NNLS solver for large-scale Tchakaloff regression designs*. Dolomites Res. Notes Approx., vol. 13, 1, (2020), 20–29. ISSN 2035-6803. URL http://dx.doi.org/10.14658/PUPJ-DRNA-2020-1-3

[10] M. Dessole, F. Marcuzzi, and M. Vianello. *dCATCH: a numerical package for d-variate near G-optimal Tchakaloff regression via fast NNLS*. Mathematics, vol. 8, 7. ISSN 2227-7390. URL http://dx.doi.org/10.3390/math8071122

[11] M. Dessole, F. Marcuzzi, and M. Vianello. *dCATCH: numerical package for d-variate discrete measure compression, near-optimal design and polynomial fitting - v1.1.* https://www.math.unipd.it/~marcov/dCATCH.html (2020)

[12] J. Dick and F. Pillichshammer. *Digital nets and sequences*, (Cambridge University Press, Cambridge2010). ISBN 978-0-521-19159-3. Discrepancy theory and quasi-Monte Carlo integration, URL http://dx.doi.org/10.1017/CBO9780511761188

[13] G. Elefante, A. Sommariva, and M. Vianello. *CQMC: an improved code for low-dimensional compressed quasi-MonteCarlo cubature*. Dolomites Res. Notes Approx., vol. 15, 2, (2022), 92–100. ISSN 2035-6803. URL http://dx.doi.org/10.14658/PUPJ-DRNA-2022-2-8

[14] G. Elefante, A. Sommariva, and M. Vianello. *Compressed QMC volume and surface integration on union of balls* (2023). URL https://arxiv.org/abs/2303.01460

[15] G. Elefante, A. Sommariva, and M. Vianello. *Qsurf: a software package for compressed QMC integration on parametric surfaces (in Matlab).* https://github.com/alvisesommariva/Qsurf (2023)

[16] S. Hayakawa. *Monte Carlo cubature construction*. Jpn. J. Ind. Appl. Math., vol. 38, 2, (2021), 561–577. ISSN 0916-7005,1868-937X. URL http://dx.doi.org/10.1007/s13160-020-00451-x

[17] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, *Classics in Applied Mathematics*, vol. 15, (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA1995). ISBN 0-89871-356-0. Revised reprint of the 1974 original, URL http://dx.doi.org/10.1137/1.9781611971217

[18] G. Legrain. *Non-negative moment fitting quadrature rules for fictitious domain methods*. Comput. Math. Appl., vol. 99, (2021), 270–291. ISSN 0898-1221,1873-7668. URL http://dx.doi.org/10.1016/j.camwa.2021.07.019

[19] C. Litterer and T. Lyons. *High order recombination and an application to cubature on Wiener space*. Ann. Appl. Probab., vol. 22, 4, (2012), 1301–1327. ISSN 1050-5164,2168-8737. URL http://dx.doi.org/10.1214/11-AAP786

[20] N. Nguyen and G. Ökten. *The acceptance-rejection method for low-discrepancy sequences*. Monte Carlo Methods Appl., vol. 22, 2, (2016), 133–148. ISSN 0929-9629,1569-3961. URL http://dx.doi.org/10.1515/mcma-2016-0104

[21] F. Piazzon, A. Sommariva, and M. Vianello. *Caratheodory-Tchakaloff subsampling*. Dolomites Res. Notes Approx., vol. 10, 1, (2017), 5–14. ISSN 2035-6803. URL http://dx.doi.org/10.14658/PUPJ-DRNA-2017-1-2

[22] W. Pleśniak. *Multivariate Jackson inequality*. J. Comput. Appl. Math., vol. 233, 3, (2009), 815–820. ISSN 0377-0427,1879-1778. URL http://dx.doi.org/10.1016/j.cam.2009.02.095

[23] D. L. Ragozin. *Constructive polynomial approximation on spheres and projective spaces*. Trans. Amer. Math. Soc., vol. 162, (1971), 157–170. ISSN 0002-9947,1088-6850. URL http://dx.doi.org/10.2307/1995746

[24] M. Slawski. *Non-negative least squares: comparison of algorithms.* https://sites.google.com/site/slawskimartin/code

[25] A. Sommariva and M. Vianello. *Compression of multivariate discrete measures and applications.* Numer. Funct. Anal. Optim., vol. 36, 9, (2015), 1198–1223. ISSN 0163-0563,1532-2467. URL http://dx.doi.org/10.1080/01630563.2015.1062394

[26] A. Sommariva and M. Vianello. *Computing Tchakaloff-like cubature rules on spline curvilinear polygons.* Dolomites Res. Notes Approx., vol. 14, 1, (2021), 1–11. ISSN 2035-6803. URL http://dx.doi.org/10.14658/PUPJ-DRNA-2021-1-1

[27] A. Sommariva and M. Vianello. *Low cardinality positive interior cubature on NURBS-shaped domains.* BIT, vol. 63, 2, (2023), [Paper No. 22], 20. ISSN 0006-3835,1572-9125. URL http://dx.doi.org/10.1007/s10543-023-00958-y

[28] V. Tchakaloff. *Formules de cubatures mécaniques à coefficients non négatifs.* Bull. Sci. Math. (2), vol. 81, (1957), 123–134. ISSN 0007-4497

[29] M. Tchernychova. *Caratheodory cubature measures.* Ph.D. thesis, University of Oxford (2015). URL https://ora.ox.ac.uk/objects/uuid:a3a10980-d35d-467b-b3c0-d10d2e491f2d

[30] D. R. Wilhelmsen. *A nearest point algorithm for convex polyhedral cones and applications to positive linear approximation.* Math. Comp., vol. 30, 133, (1976), 48–57. ISSN 0025-5718,1088-6842. URL http://dx.doi.org/10.2307/2005429

[31] H. Zhu and J. Dick. *Discrepancy bounds for deterministic acceptance-rejection samplers.* Electron. J. Stat., vol. 8, 1, (2014), 678–707. ISSN 1935-7524. URL http://dx.doi.org/10.1214/14-EJS898