On the numerical compression of QMC rules.

Giacomo Elefante, **Alvise Sommariva**, Marco Vianello

Constructive approximation of functions 3
Krakow, Poland, 22-24 February 2023.

Thanks: Conference Organizers, co-workers, RITA, UMI group TAA, GNCS project *Methods and software for multivariate integral models*

Purpose

In this talk,

- we start introducing the well-known Tchakaloff theorem (and one of its variants)

  existence theorem of certain algebraic and *low* cardinality cubature rules with positive weights on a multivariate compact domain $\Omega \subset \mathbb{R}^d$;

- we compress cubature rules,

  we show how, from cubature rules on Ω (w.r.t. a positive measure) with positive weights and interior nodes (i.e. of PI-type), whose algebraic degree of precision *ADE* is equal to m and the number of nodes higher than the dimension "r" of the polynomial space $\mathbb{P}_m(\Omega)$ of total degree m, we can extract rules of PI-type but with at most "r" nodes, by means of Lawson-Hanson algorithm;

- application to Quasi-Montecarlo Cubature: rule compression

  given a Quasi-Montecarlo Cubature rule (acronym: QMC) and a certain degree of precision m, we compress the QMC rule with a positive one, that provides the same results over polynomials of degree at most m;

- some multivariate examples

  results over bivariate and trivariate domains, peculiarities of volumes and surfaces obtained by union of balls.

Important: all the Matlab routines used in this talk are available at the author's homepage.

Paper

Compressed cubature over polygons with applications to optical design (2020)

- purpose: cubature formula over a convex, nonconvex or even multiply connected polygons Ω.
- strategy: once a minimal triangulation of Ω is available, we obtain the rule by applying an almost-minimal rule of PI-type on each triangle with the wanted ADE, summing the contributions.

Remark

- *minimal triangulation: one can triangulate a general M-sides polygon via $M-2$ triangles (easy task in a convex polygon, not trivial for a general one),*
- *almost-minimal rule the number of its nodes is almost minimal between those having a certain degree of precision and requirement on the nodes (e.g. internal) and weights (e.g. positive).*

Example: quadrature over polygons (triang. based approach)

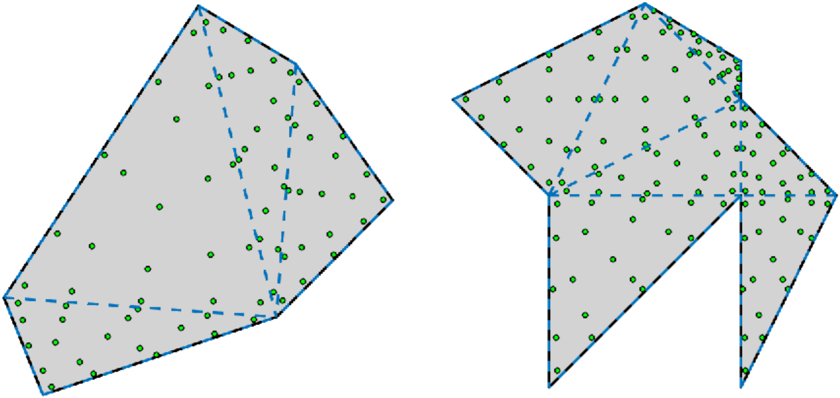


Figure: Examples of polygonal domains (ADE=9).

Left: a convex domain with 6 sides (76 nodes), Right: a non-convex domain with 9 sides (133 nodes).

Note: all the weights are positive.

- **Pros**: In general these rules always have internal nodes as well as positive weights.
- **Cons**: Rule still may have high cardinality if the polygon has many sides.

Observe that in the examples above for ADE$= 9$

- convex domain: the rule has 76 nodes,
- not convex domain: the rule has 133 nodes.

Remark

In both cases the number of nodes is higher than the dimension of the polynomial space $\mathbb{P}_9$ *that is equal to* $(9+1)(9+2)/2 = 55$.

Our project is to quickly extract from the previous one, another rule of PI-type with the same degree of precision but with a number of nodes at most equal to dim$(\mathbb{P}_9)$ *(i.e. a rule compression).*

Carathéodory-Tchakaloff Subsampling

Paper

M. Putinar, A Note on Tchakaloff's Theorem, Proc. of AMS, Vol. 125, No. 8 (1997), 2409–2414.

Theorem (Carathéodory-Tchakaloff, see more general Putinar theorem)

Let

1. μ *be a multivariate discrete measure supported at a finite set* $X = \{\mathbf{x}_k\}_{k=1,\ldots,N} \subset \mathbb{R}^d$, *with correspondent positive weights* $\{w_k\}_{k=1,\ldots,N}$,
2. $\Phi = span(\phi_1, \ldots, \phi_r)$ *a finite dimensional space of d-variate functions defined on* $\Omega \supseteq X$, *with* $dim(\Phi|_X) \leq r$.

Then there exist a quadrature formula with nodes $T = \{\mathbf{t}_k\}_{k=1,\ldots,N_c} \subseteq X$ *and positive weights* $\{u_k\}_{k=1,\ldots,N_c}$, *such that* $N_c \leq dim(\Phi|_X)$ *and*

$$\int_\Omega f(\mathbf{x}) d\mu := \sum_{k=1}^{N} w_k f(\mathbf{x}_k) = \sum_{i=1}^{N_c} u_i f(\mathbf{t}_i) := \int_\Omega f(\mathbf{x}) d\mu_C, \text{ for all } f \in \Phi|_X.$$

Example

If we have a rule of PI-type with cardinality N higher than $r = \dim(\mathbb{P}_m(\Omega))$ then we can extract one of PI-type with cardinality $N_c \leq r$, having the same *integration* values in $\mathbb{P}_m(\Omega)$.

Paper

Compression of multivariate discrete measures and applications (2015).

Given

- a formula of PI-type with ADE=m, nodes $X = \{\mathbf{x}_k\}_{k=1,\ldots,N} \subset \mathbb{R}^d$ and positive weights $\{w_k\}_{k=1,\ldots,N}$,
- a basis $\{\phi_1, \ldots, \phi_r\}$ of $\mathbb{P}_m(\Omega)$,

let

- $V_{i,j} = (\phi_j(\mathbf{x}_i))$ the Vandermonde matrix at the nodes,
- $\mathbf{b} = (b_j)_{j=1,\ldots,r}$ where $b_j = \int_\Omega \phi_j d\mu = \sum_{i=1}^N w_i \phi_j(\mathbf{x}_i)$, the vector of the μ moments.

The problem mentioned above resorts into computing a nonnegative solution with at most "r" nonvanishing components to the underdetermined linear system

$$V^T \mathbf{u} = \mathbf{b}.$$

The computation of a nonnegative solution with at most $r = \dim(\mathbb{P}_m(\Omega))$ nonvanishing components to the underdetermined linear system $V^T\mathbf{u} = \mathbf{b}$ can be perfomed finding a sparse solution to the quadratic minimum problem

$$\text{NNLS:} \begin{cases} \min_u \|V^T\mathbf{u} - \mathbf{b}\|_2 \\ \mathbf{u} \geq 0 \end{cases}$$

via Lawson-Hanson active set method for NonNegative Least Squares (NNLS).

In Matlab this can be done by means of the Matlab built-in routine `lsqnonneg` as well as by the more recent `LHDM` by Dessole, Marcuzzi and Vianello.

Remark

- *The approach mentioned above is effective for mild ADE, say on the order of ADE=20 for bivariate domains and ADE=10 for trivariate domains.*
- *There are also other approaches, e.g. by based on linear programming or by a different combinatorial algorithm (recursive Halving Forest), based on SVD.*

As example, we can consider the application of the technique mentioned above to extract a rule of PI-type, for computing a similar one on the polygonal domains treated above.

Algorithm

`input`: *the nodes* $\{\mathbf{x}_k\}_{k=1,\dots,N}$, *the weights* $\{w_k\}_{k=1,\dots,N}$ *of a PI-type rule with* $N > r$ *(r is the dimension of the polynomial space* $\mathbb{P}_m$*) and a polynomial basis* $\{\psi_j\}_{j=1,\dots,r}$*;*

1. *Vandermonde matrix: compute* $U = (\phi_k(\mathbf{x}_i))$*;*
2. *fight ill-conditioning: compute the QR factorization with column pivoting* $\sqrt{W}U(:,\pi) = QR$*, where* $\sqrt{W} = diag(\{w_k\})$ *and* π *is a permutation vector; this corresponds to a change of basis* $(\phi_1, \dots, \phi_r) = (\psi_1, \dots, \psi_r)R^{-1}$*, so obtaining an orthonormal basis w.r.t. the discrete measure defined by the nodes* $\{\mathbf{x}_k\}_{k=1,\dots,N}$ *and the weights* $\{w_k\}_{k=1,\dots,N}$*;*
3. *moments: evaluate the vector* $\mathbf{b} = Q^T\mathbf{w}$ *where* $\mathbf{w}_k = w_k$*;*
4. *compute a positive sparse solution: solve* $Q^T\mathbf{u} = \mathbf{b}$ *by Lawson-Hanson algorithm (or its alternatives).*

By this algorithm,

- adopting as basis $\{\psi_j\}$ the total-degree product Chebyshev basis of the smallest Cartesian rectangle $[a_1, b_1] \times [a_2, b_2]$ containing Ω, with the graded lexicographical ordering,
- from the PI-rules with ADE=9 obtained via triangulation,

we get the PI-rules below with cardinality $55 = \dim(\mathbb{P}_9)$.

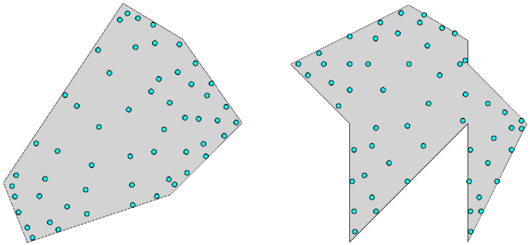


Figure: Examples of polygonal domains (ADE=9).

Left: a convex domain with 6 sides (55 nodes, the previous rule had 76 nodes), Right: a non-convex domain with 9 sides (55 nodes, the previous rule had 133 nodes).

Remark (When do not apply this technique)

We observe that this approach is useful only when the initial rule of PI-type with ADE=m has cardinality higher the dimension L of $\mathbb{P}_m(\Omega)$.

Thus it is worthless in the case of classical domains as the interval, the disk, simplex, cube, sphere, where there are explicit rules of PI-type with cardinality inferior to L.

Remark (Cputimes on the previous domains)

For mild ADE the computation of these compressed rules is fast. Running Matlab R2022a, on a computer with an Apple M1 processor and 16GB of RAM, we had average cputimes as in the table below:

Domain	tri. rule	compress.
Convex domain	*1.1e-3s*	*1.7e-2s*
Non-convex domain	*5.2e-3s*	*1.1e-2s*

Table: *Average cputime for computing rules with ADE=9 in the previous polygonal domains.*

This technique can be used to compress Quasi-Montecarlo cubature rules on $\Omega \subset \mathbb{R}^d$ obtained by set operations of compact domains $\Omega_1, \dots, \Omega_\nu \subset \mathbb{R}^d$:

- polynomial basis: product-type Chebyshev basis in $\mathbb{P}_m(\Omega)$ on the bounding box $\mathcal{R}$ of the domain Ω;
- mesh points: *sufficiently dense* low discrepancy points in the bounding box $\mathcal{R}$ (good choice for volumes) or on a suitable subdomain of $\mathcal{R}$ containing Ω (good for surfaces); notice that if the measure of the domain is not known, it must be approximated numerically in order to apply QMC;
- in-domain routine: in domain routine on each domain Ω_k, $k = 1, \dots, \nu$ followed by suitable set operations;
- moment computation: via Quasi-Montecarlo cubature.

This allows to achieve a rule with few nodes that equals the results of the QMC rule applied to polynomials in $\mathbb{P}_m$.

Purpose

Retaining the approximation power of the original QMC formula (up to a quantity proportional to the best polynomial approximation error of degree m to f, in the uniform norm on Ω), using much fewer nodes.

- Domains: this technique can be used to compress QMC on $\Omega \subset \mathbb{R}^d$ obtained by set operations of compact domains $\Omega_1, \ldots, \Omega_\nu \subset \mathbb{R}^d$.
- Alternatives: very often the detection of specific features (as its boundary $\partial\Omega$ or computation of the polynomial moments) may be not available or so difficult to make extremely complicated the usage of other techniques than QMC.
- In-domain routine:
  - Verification of certain inequalities: for example, the unit-ball $B(\mathbf{0}, 1)$ is defined as the set

$$B(\mathbf{0}, 1) := \{\mathbf{x} = (x, y, z) \in \mathbb{R}^3 \text{ such that } x^2 + y^2 + z^2 \leq 1\}.$$

  - Specific codes: Matlab built-in `inpolygon` (polygonal domains), `inpolyhedra` (polyhedral domains), `in-rs` (curved polygons with boundary defined by NURBS).

Paper

For details about `in-rs` *see: inRS: implementing the indicator function for NURBS-shaped planar domains, Applied Mathematics Letters, Volume 130, August 2022.*

Application to QMC compression: some bivariate examples

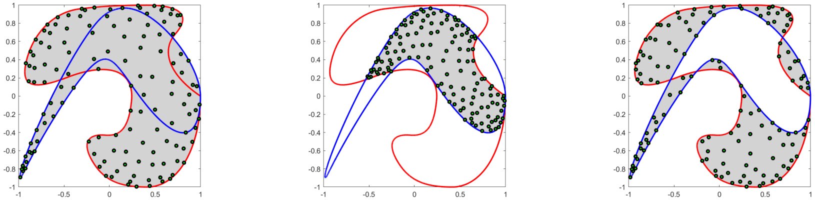


Figure: 231 compressed QMC nodes with exactness degree $n = 20$, on complex shapes arising from union (top-left), intersection (top-right) and symmetric difference (bottom) of two NURBS-shaped domains (extraction from a million Halton points of domain bounding boxes, basis Φ obtained by orthonormalization of a tensorial type basis in the bounding box $\mathcal{R}$ of the domain Ω).

- Domains: $\mathcal{R}$ is the smaller rectangle (with sides parallel to the axes) containing Ω;
- polynomial basis: subset of tensorial-type Chebyshev basis defining $\mathbb{P}_m$ on $\mathcal{R}$;
- In-domain routine: `in-rs`;

Application to QMC compression: some bivariate examples

deg	5	10	15	20
card. *CQMC*	21	66	136	231
compr. ratio	1.2e+04	3.9e+03	1.9e+03	1.1e+03
cpu *CQMC*	4.0e-02	1.2e-01	2.8e-01	5.8e+00
mom. resid. *CQMC*	5.8e-16	1.4e-15	2.4e-15	7.0e-15

Table: Compression parameters of QMC cubature with $N = 255923$ Halton points on the intersection of two NURBS-shaped domains as in Figure above top-right. By CQMC we intend results obtained via the new compression algorithm.

deg	5	10	15	20
$E(f_1)$	2.7e-04	1.4e-08	3.0e-13	4.5e-16
$E(f_2)$	2.3e-04	2.4e-05	1.1e-05	5.6e-06

Table: Relative CQMC errors $E(f_k)$, $k = 1, 2$ for the two test functions $f_1(P) = \exp(-|P - P_0|^2)$, $f_2(P) = |P - P_0|^5$ on the intersection of Fig. 1 top-right.

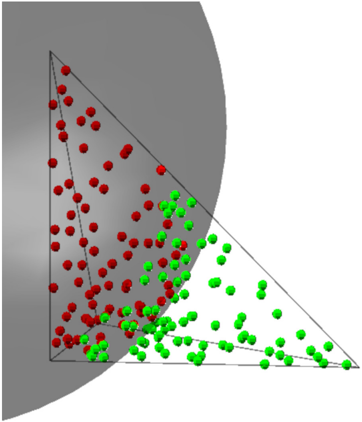


Figure: 84 compressed QMC nodes with exactness degree n = 6, on intersection (red bullets) and difference (green bullets) of a tetrahedral element with a ball (extraction from a million Halton points of domain bounding boxes, cputime: $\approx 5 \cdot 10^{-2} s$, basis Φ obtained by orthonormalization of a tensorial type basis in the bounding box $\mathcal{R}$ of the domain Ω).

Application to QMC compression: trivariate examples

deg	2	4	6
card. *CQMC*	10	35	84
compr. ratio	2.2e+04	6.2e+03	2.6e+03
cpu *CQMC*	4.1e-02	4.1e-02	1.8e-01
mom. resid. *CQMC*	1.7e-16	6.0e-16	1.2e-15

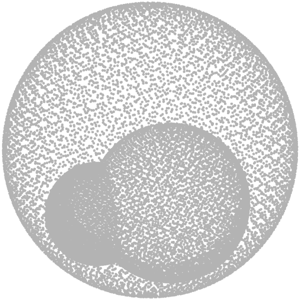
Table: Compression parameters of QMC cubature with $N = 216217$ Halton points on the intersection of a tetrahedral element with a ball as in the last figure. By CQMC we intend results obtained via the new compression algorithm.

deg	2	4	6
card. *CQMC*	10	35	84
compr. ratio	5.9e+03	1.7e+03	7.0e+02
cpu *CQMC*	3.3e-02	2.1e-02	5.5e-02
mom. resid. *CQMC*	5.0e-16	6.1e-16	1.2e-15

Table: Compression parameters of QMC cubature with $N = 58561$ Halton points on the difference of a tetrahedral element with a ball as in the last figure. By CQMC we intend results obtained via the new compression algorithm.

Let $B(C_j, r_j)$ be a ball with center $C_j \in \mathbb{R}^3$ and radius $r_j > 0$ and consider domains of the forms

1. $\Omega_V = \cup_{j=1}^{L} B(C_j, r_j)$ (volume);
2. $\Omega_S = \partial \cup_{j=1}^{L} B(C_j, r_j)$ (surface).



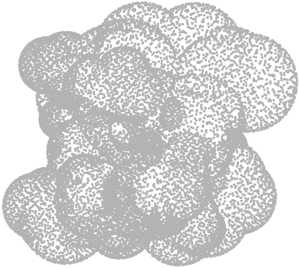


Figure: Left: union of 3 balls, Right: union of 100 balls.

Application to QMC compression: union of balls (volumes and surfaces)

Main difficulties:

- their geometry can be very complicated, since the balls may intersect, even creating cavities: hard to subdivide in manageable subregions;
- depending on the balls, the polynomial space $\mathbb{P}_m(\Omega_S)$ over the surface Ω_S may have a dimension inferior than $\mathbb{P}_m(\mathbb{R}^3)$ (spheres are algebraic surfaces), and it is not straightforward to determine exactly a well-conditioned basis (even the computation of $\dim(\mathbb{P}_m(\Omega_S))$ may be a tough problem).

Where they arise:

- molecular modelling, computational geometry, computational optics, wireless network analysis;

Problems:

- basic (but not trivial): exact computation of areas or volumes of such sets;
- more difficult: computing volume or surface integrals there by quadrature formulas.

Paper

Qbubble: a numerical code for compressed QMC volume and surface integration on union of balls, *submitted.*

Purpose

We intend to compress a rule, matching the QMC values of integrands in $\mathbb{P}_m$, in the case of the volumes, i.e. $\Omega_V = \cup_{j=1}^{L} B(C_j, r_j)$.

- full QMC rule: easy,
  - low discrepancy sequences in the bounding box $\mathcal{R}$ are available, and the restriction on Ω_V provides low discrepancy sequences;
  - easy approximation of domain volume via QMC and volume of the parallelepiped $\mathcal{R}$ (bounding box);
- polynomial basis: technical and new,
  starting from product Chebyshev basis a trick is used to reduce computations for determining a well-conditioned basis (using just a small subset of QMC nodes);
- moment evaluation: easy,
  via the full QMC rule;
- compressed QMC: technical and new,
  a trick is used to reduce computations (again using just a small subset of QMC nodes);

Purpose

We intend to compress a rule, matching the QMC values of integrands in $\mathbb{P}_m$, in the case of the surfaces, i.e.. $\Omega_S = \partial \cup_{j=1}^{L} B(C_j, r_j)$:

- full QMC rule: quite easy,
  - low discrepancy sequences X_j in each sphere $S_j = \partial B(C_j, r_j)$ are available, hence one can determine after some technicalities low discrepancy sequences over Ω_S;
  - easy approx. of Ω_S area via QMC and area of each sphere S_j, $j = 1, \ldots, L$;
- polynomial basis: very technical and new,
  - usage of Matlab numerical `rank revealing algorithms` to determine the dimension of the polynomial space anda well-conditioned basis (the dimensions of $\mathbb{P}_m(\Omega_S)$ and $\mathbb{P}_m(\mathbb{R}^3)$ may be different);
  - starting from product Chebyshev basis a trick is used to reduce computations for determining a well-conditioned basis (using just a small subset of QMC nodes);
- moment evaluation: easy,
  via the full QMC rule;
- compressed QMC: technical and new,
  a trick is used to reduce computations (again using a small subset of QMC nodes);

Application to QMC compression: union of 3 balls (example on a volume)

deg	3	6	9	12
card. *QMC*	1128709			
card. *CQMC*	20	84	220	455
compr. ratio	5.6e+04	1.3e+04	5.1e+03	2.5e+03
cpu *QMC*	9.0e-01			
cpu *CQMC*	2.5e-01	8.6e-01	2.2e+00	5.5e+00
mom. resid. *CQMC* iter. 1	4.2e-16	1.2e-15	1.9e-15	5.3e-15

Table: Example with the union of 3 balls, in a bounding box with 2400000 low-discrepancy points. Compressed codes used the acronym CQMC.

Remark

New codes are from 13.6 to 25.4 times faster than the old ones

Application to QMC compression: union of 100 balls (example on a volume)

deg	3	6	9	12
card. *QMC*	1195806			
card. *CQMC*	20	84	220	455
compr. ratio	5.6e+04	1.3e+04	5.1e+03	2.8e+03
cpu *QMC*	1.3e+00			
cpu *CQMC*	2.6e-01	9.1e-01	2.4e+00	5.8e+00
mom. resid. *CQMC* iter. 1	1.3e-16	7.2e-16	1.6e-15	7.3e-15

Table: Example with the union of 100 balls, in a bounding box with 2400000 Halton points. Compressed codes used the acronym CQMC.

Remark

New codes are from 13.1 to 27.9 times faster than the old ones.

deg	3	6	9	12
card. *QMC*	1024179			
card. *CQMC*	20	83	200	371
compr. ratio	5.1e+04	1.2e+04	5.1e+03	2.8e+03
cpu *QMC*	8.8e-01			
cpu *CQMC*	2.8e-01	1.1e+00	2.8e+00	5.9e+00
speed-up	10.7	16.4	17.9	23.7
cpu Q_c^{full}	2.7e+00	1.3e+01	2.9e+01	5.9e+01
speed-up	9.6	11.8	10.4	10.0
mom. resid. *CQMC* iter. 1	7.2e-16	1.1e-15	2.3e-15	4.0e-15

Table: Compression of surface QMC integration on the union 3 balls, starting from 500000 low-discrepancy points on each sphere. Compressed codes used the acronym CQMC.

Remark

New codes are from 10.7 to 23.7 times faster than the old ones.

Application to QMC compression: union of 100 balls (example on a surface)

deg	3	6	9	12
card. *QMC*	1032718			
card. *CQMC*	20	84	220	455
compr. ratio	5.2e+04	1.2e+04	4.7e+03	2.3e+03
cpu *QMC*	1.5e+01			
cpu *CQMC*	3.0e-01	1.2e+00	3.0e+00	6.6e+00
mom. resid. *CQMC* iter. 1	2.7e-16	1.0e-15	2.3e-15	4.5e-15

Table: Compression of surface QMC integration on the union 100 balls, starting from 60000 low-discrepancy points on each sphere. Compressed codes used the acronym CQMC.

Remark

New codes are from 9.3 to 16.7 times faster than the old ones.

Next we show the integration errors on three test functions with different regularity, namely

- $f_1(P) = |P - P_0|^5$ (class C^4 with discontinuous fifth derivatives);
- $f_2(P) = \cos(x + y + z)$ (analytic);
- $f_3(P) = \exp(-|P - P_0|^2)$ (analytic);

where $P_0 = (0, 0, 0) \in \Omega$.

Remark

- *It is easy to see that for every $f \in C(\Omega)$, the following error estimate holds*

$$|I_{CQMC}(f) - I(f)| \leq \mathcal{E}_{QMC}(f) + 2\,\mu(\Omega)\,E_n(f;\Omega)\ ,$$

where $\mathcal{E}_{QMC}(f) = |I_{QMC}(f) - I(f)|$ and $E_n(f;\Omega)$ is the best approximation error of f w.r.t. $\mathbb{P}_n$, in Ω, w.r.t. the sup-norm.

- *The reference values of the integrals have been computed by a QMC formula starting from 10^8 Halton points in the bounding box.*

Application to QMC compression: on the numerical integration of some functions, 3 balls volumes

deg	3	6	9	12
$E^{QMC}(f_1)$	3.5e-04			
$E^{new}(f_1)$	4.8e-02	3.0e-04	3.5e-04	3.5e-04
$E^{QMC}(f_2)$	7.3e-04			
$E^{new}(f_2)$	3.5e+00	7.6e-02	2.0e-03	7.3e-04
$E^{QMC}(f_3)$	8.7e-05			
$E^{new}(f_3)$	5.6e-01	1.2e-01	1.4e-02	2.7e-03

Table: Example with 3 balls (the reference values are computed via QMC starting from 10^8 Halton points in the bounding box).

Application to QMC compression: on the numerical integration of some functions, 100 balls volumes

deg	3	6	9	12
$E^{QMC}(f_1)$	1.1e-04			
$E^{new}(f_1)$	7.7e-03	8.9e-05	1.1e-04	1.1e-04
$E^{QMC}(f_2)$	1.7e-04			
$E^{new}(f_2)$	4.5e-03	6.5e-05	1.7e-04	1.7e-04
$E^{QMC}(f_3)$	2.2e-04			
$E^{new}(f_3)$	2.4e-02	1.4e-02	3.5e-05	2.2e-04

Table: Example with 100 balls (the reference values are computed via QMC starting from 10^8 Halton points in the bounding box).

deg	3	6	9	12
$E^{QMC}(f_1)$	3.9e-06			
$E^{new}(f_1)$	1.1e-04	6.3e-07	4.0e-06	3.9e-06
$E^{QMC}(f_2)$	8.6e-05			
$E^{new}(f_2)$	6.7e-01	1.0e-02	5.9e-04	8.6e-05
$E^{QMC}(f_3)$	5.8e-06			
$E^{new}(f_3)$	3.0e-01	2.5e-03	6.9e-04	4.8-05

Table: Compression of surface QMC integration on the union 3 balls (the reference values are computed via QMC starting from 10^6 points on each sphere).

Application to QMC compression: on the numerical integration of some functions, 100 balls surfaces

deg	3	6	9	12
$E^{QMC}(f_1)$	4.0e-05			
$E^{new}(f_1)$	2.3e-03	2.9e-05	4.0e-05	4.0e-05
$E^{QMC}(f_2)$	2.0e-04			
$E^{new}(f_2)$	5.2e-01	3.6e-04	1.9e-04	2.0e-04
$E^{QMC}(f_3)$	1.6e-04			
$E^{new}(f_3)$	4.1e-01	4.8e-03	1.3e-04	1.6e-04

Table: Compression of surface QMC integration on the union 100 balls (the reference values are computed via QMC starting from 10^6 points on each sphere).

Bibliography

1. M. Dessole, F. Marcuzzi, M. Vianello, Accelerating the Lawson-Hanson NNLS solver for large-scale Tchakaloff regression designs, DRNA 13, 20–29 (2020).
   Fast Lawson-Hanson algorithm and Matlab codes.
2. 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. TMA, Vol 9(4), pp. 640–663 (2016).
   Compression of Quasi-Montecarlo rules.
3. S. De Marchi, G. Elefante: Quasi-Monte Carlo integration on manifolds with mapped low-discrepancy points and greedy minimal Riesz s-energy points, Applied Numerical Mathematics 127, 110–124 (2018).
   Quadrature points on manifolds via the Quasi-Monte Carlo (QMC) method
4. G. Elefante, A. Sommariva, and M. Vianello, CQMC: an improved code for low-dimensional Compressed Quasi-MonteCarlo cubature, DRNA 15 (2), 92–1000 (2022).
   Compression of Quasi-Montecarlo rules.
5. G. Elefante, A. Sommariva, and M. Vianello, Qbubble: a numerical code for compressed QMC volume and surface integration on union of balls, submitted.
   Compression of Quasi-Montecarlo rules over suitable volumes and surfaces.
6. A. Sommariva, Matlab codes used in the numerical experiments
7. A. Sommariva, M. Vianello, Compression of multivariate discrete measures and applications, *Numer. Funct. Anal. Optim.*, 36, 1198–1223 (2015).
   Details on cubature compression