Generalizing the Tolerance for Guaranteed QMC Algorithms

Lluís Antoni Jiménez Rugama
Joint work with Doctor Fred J. Hickernell
Department of Applied Mathematics, Illinois Institute of Technology
ILLINOIS INSTITUTE OF TECHNOLOGY

Motivation

The high dimensional integration is used for many applications: option pricing, ion transport models, statistical physics ...
Some common techniques are:

Method	Convergence
Trapezoidal rule:	$\mathcal{O}(N^{-2/d})$
Simpson's rule:	$\mathcal{O}(N^{-4/d})$
Monte Carlo:	$\mathcal{O}(N^{-1/2})$
Quasi-Monte Carlo:	$\mathcal{O}(N^{-1+\varepsilon})$

with d the dimension and N the number of points.
Many methods suffer from what is called Curse of dimensionality. When d is big, the convergence becomes really slow. Nevertheless, among all the methods, Monte Carlo and Quasi-Monte Carlo have a nice property: *they do not depend on d*.
Then, it looks fair to study the multidimensional numerical integration with Quasi-Monte Carlo:

$$\int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{t}_i)$$

for a fixed sequence $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots$ in the unit cube $[0,1)^d$.

The Koksma-Hlawka inequaltiy

$$\left|\int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{t}_i)\right| \leqslant D^*(n,\{\boldsymbol{t}_i\})\,V(f)$$

This inequality could be an interesting start for bounding the error. However, $V(f)$ -"roughness" of the function- is hard to compute and in this inequality, the sequence and $V(f)$ are unrelated a priori. This is the reason why we develop a new approach for the problem.

Generalizing the Tolerance

Consider the solution as $I := \int_{[0,1)^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, our QMC algorithm as $\widehat{I}_m := \frac{1}{b^m}\sum_{i=0}^{b^m-1} f(\boldsymbol{t}_i)$ and $\widehat{\varepsilon}_m$ its bound on the absolute error found in (1). Let also $\mathrm{tol}(a,b)$ be defined Lipschitz $L=1$ in b and non-decreasing in both arguments, i.e. $\mathrm{tol}(a,b) \leqslant \mathrm{tol}(a',b')$ for $a \leqslant a'$ and $b \leqslant b'$. Ideally, we will use $\mathrm{tol}(\varepsilon_a, \varepsilon_r|I|)$.
Define

$$\Delta_{m,\pm} := \frac{1}{2}\left[\mathrm{tol}\left(\varepsilon_a, \varepsilon_r\left|\widehat{I}_m - \widehat{\varepsilon}_m\right|\right) \pm \mathrm{tol}\left(\varepsilon_a, \varepsilon_r\left|\widehat{I}_m + \widehat{\varepsilon}_m\right|\right)\right],$$

$$\widetilde{I}_m := \widehat{I}_m + \Delta_{m,-}$$

Lemma 1 *We claim that if $\widehat{\varepsilon}_m \leqslant \Delta_{m,+}$, then*

$$\left|I - \widetilde{I}_m\right| \leqslant tol(\varepsilon_a, \varepsilon_r|I|)$$

The new algorithm consists on increasing m until $\widehat{\varepsilon}_m \leqslant \Delta_{m,+}$.

An important remark is that $\Delta_{m,-}$ is always shrinking $\widehat{I}_m$ into $\widetilde{I}_m$ because

$$\widehat{I}_m > 0 \Longleftrightarrow \Delta_{m,-} < 0.$$

This means that $\widetilde{I}_m$ is biased, always closer to 0 than $\widehat{I}_m$. This is helpful for the relative error because given $\widehat{I}_m$, if I is above $\widehat{I}_m$, then $\left|\frac{I-\widehat{I}_m}{I}\right|$ can be smaller. Therefore, in order to minimize $\left|\frac{I-\widehat{I}_m}{I}\right|$, we enlarge I with respect to $\widehat{I}_m$ by shrinking $\widehat{I}_m$. We then build the biased estimator $\widetilde{I}_m$ as desired. Thus, for the new algorithm it will be easier to satisfy the condition $\varepsilon_r \geqslant \left|\frac{I-\widetilde{I}_m}{I}\right|$.

Upper Bound on the Computational Complexity

To talk about the upper bound on the complexity, we need the following Lemma,
Lemma 2 *If*

$$\widehat{\varepsilon}_m \leqslant \frac{tol(\varepsilon_a, \varepsilon_r|I|)}{1+\varepsilon_r}$$

then,

$$\widehat{\varepsilon}_m \leqslant \Delta_{m,+}$$

Now, let's define $M(\varepsilon, S)$ such that, $m \geqslant M(\varepsilon,S) \Rightarrow \widehat{\varepsilon}_m \leqslant \varepsilon$. Here, S represents the set of functions belonging to $\mathcal{C}$ and the integer M only depends on S and ε. In our guaranteed algorithm, we find the minimum m for which $\widehat{\varepsilon}_m \leqslant \varepsilon$, given a particular function.
Consider $M^* = M\left(\frac{\mathrm{tol}(\varepsilon_a,\varepsilon_r|I|)}{1+\varepsilon_r}, S\right)$. Then, M^* is an upper bound on the computational complexity since,

$$\begin{aligned} m \geqslant M^* &\overset{Mdef}{\Longrightarrow} \widehat{\varepsilon}_m \leqslant \frac{\mathrm{tol}(\varepsilon_a, \varepsilon_r|I|)}{1+\varepsilon_r} \\ &\overset{lem(2)}{\Longrightarrow} \widehat{\varepsilon}_m \leqslant \Delta_{m,+} \\ &\overset{lem(1)}{\Longrightarrow} \left|I - \widetilde{I}_m\right| \leqslant \mathrm{tol}(\varepsilon_a, \varepsilon_r|I|) \end{aligned}$$

From it, we can obtain a bound on the computational cost for our guaranteed QMC algorithm,

$$\mathrm{cost} \leqslant cM^* b^{M^*} + \$(f) b^{M^*}$$

Guaranteed QMC Algorithm: Cones of Functions

Once the mapping that gives us the ordering of the wavenumbers is fixed, we can build our algorithm as follows. First we define the cone $\mathcal{C}$ of functions:

Cone assumption

There exist non-increasing $\widehat{\omega}$ and $\breve{\omega}$ such that for all $\ell_* \leqslant \ell \leqslant m$

$$\widehat{S}(\ell,m) := \sum_{\kappa=\lfloor b^{\ell-1}\rfloor}^{b^\ell-1}\sum_{\lambda=1}^{\infty}\left|\hat{f}_{\kappa+\lambda b^m}\right|, \quad \breve{S}(m) := \sum_{\kappa=b^m}^{\infty}\left|\hat{f}_\kappa\right| \quad S(\ell) := \sum_{\kappa=\lfloor b^{\ell-1}\rfloor}^{b^\ell-1}\left|\hat{f}_\kappa\right|$$

$$\widehat{S}(\ell,m) \leqslant \widehat{\omega}(m-\ell)\breve{S}(m), \qquad \breve{S}(m) \leqslant \breve{\omega}(m-\ell)S(\ell).$$

This cone is characterized for having the property that if $f \in \mathcal{C} \Longrightarrow af \in \mathcal{C}$ for all $a \in \mathbb{R}$. Below there is an example for a 2 dimensional function. On the left we have the surface plot of it and on the right, the interpretation of what some specific sums would correspond to:

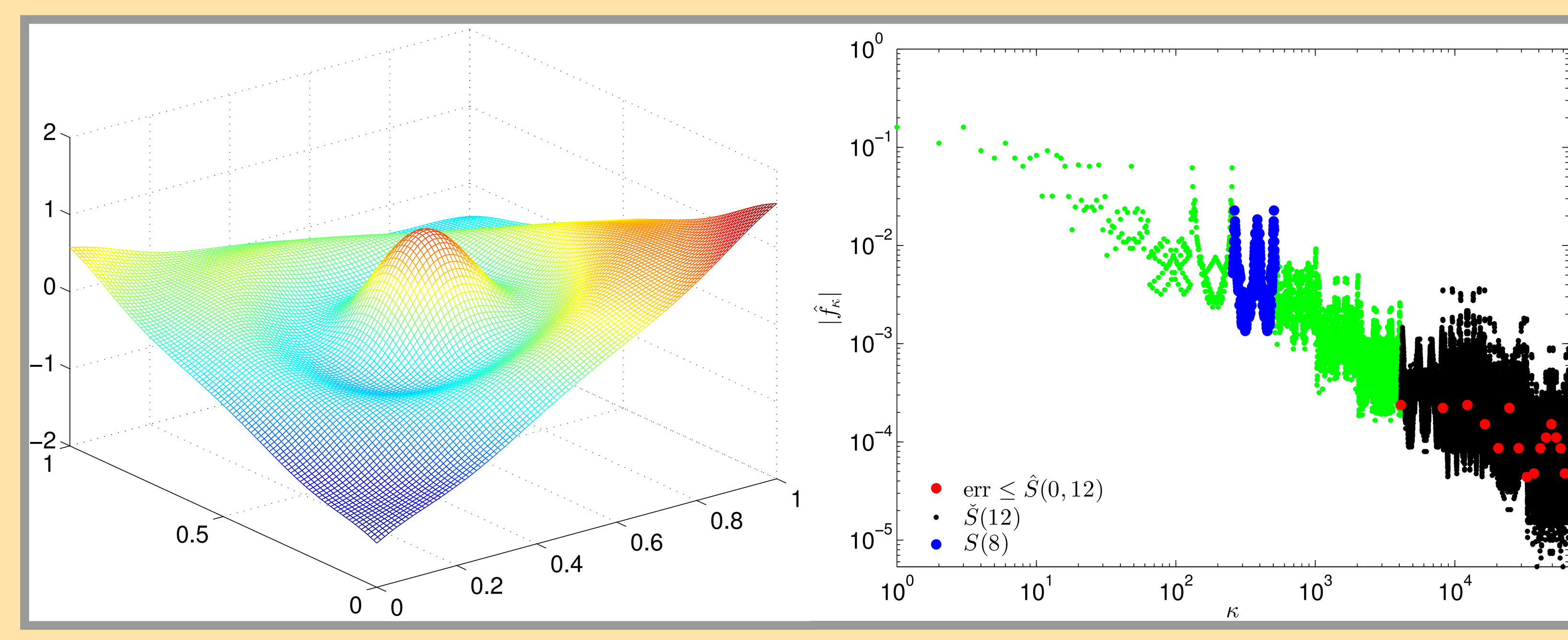

Fig. 1: Example for the function $f(x,y) = \frac{\sin\left(20\sqrt{(x-0.5)^2+(y-0.5)^2}\right)}{20\sqrt{(x-0.5)^2+(y-0.5)^2}} - 4(x-0.5)(y-0.75)$.

In addition, for all the functions lying in $\mathcal{C}$, we can show that our error is bounded by sums of our approximated coefficients $\widetilde{S}(\ell,m) := \sum_{\kappa=\lfloor b^{\ell-1}\rfloor}^{b^\ell-1}\left|\tilde{f}_{m,\kappa}\right|$,

$$\left|\int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{b^m}\sum_{i=0}^{b^m-1} f(\boldsymbol{t}_i)\right| \leqslant \frac{\widehat{\omega}(m)\breve{\omega}(m-\ell)}{1-\widehat{\omega}(m-\ell)\breve{\omega}(m-\ell)}\widetilde{S}(\ell,m) \qquad (1)$$

Guaranteed Automatic Adaptive algorithm

Given an error tolerance $\varepsilon > 0$ and an integrand f satisfying the cone conditions on its coefficients, fix $r \in \mathbb{N}$ and let $\mathfrak{C}(m) = \frac{\widehat{\omega}(m)\breve{\omega}(r)}{1-\widehat{\omega}(r)\breve{\omega}(r)}$. Initialize $m = r \in \mathbb{N}$ and do,

Step 1. Compute the sum of the approximated coefficients $\widetilde{S}(m-r,m)$.
Step 2. If the error tolerance is satisfied, i.e.

$$\mathfrak{C}(m)\widetilde{S}(m-r,m) \leqslant \varepsilon,$$

then return the answer.
Step 3. Otherwise, increase m by one, and return to Step 1.

References

[1] S.-C. T. Choi, Y. Ding, F. J. Hickernell, L. Jiang, and Y. Zhang, *GAIL: Guaranteed Automatic Integration Library (version 1)*. MATLAB software, 2013.
[2] J. Dick, F. Kuo, and I. H. Sloan, *High dimensional integration — the Quasi-Monte Carlo way*, Acta Numer., (2014), pp. 1–153.
[3] F. J. Hickernell and L. A. Jiménez Rugama, *Reliable adaptive cubature using digital sequences*, 2014. in preparation.
[4] F. J. Hickernell and H. Niederreiter, *The existence of good extensible rank-1 lattices*, J. Complexity, 19 (2003), pp. 286–300.
[5] L. A. Jiménez Rugama and F. J. Hickernell, *Adaptive multidimensional integration based on rank-1 lattices*, 2014. in preparation.
[6] H. Niederreiter and F. Pillichshammer, *Construction algorithms for good extensible lattice rules*, Constr. Approx., 30 (2009), pp. 361–393.
[7] I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration*, Oxford University Press, Oxford, 1994.

Acknowledgements

I would like to specially thank my advisor Doctor Fred J. Hickernell together with my coworkers Yuhan Ding, Lan Jiang, Xuan Zhou, Yizhi Zhang, Sou-Cheng Choi and Xin Tong. Furthermore, also thank the people in MCQMC 2014 that provided additional support: Dirk Nuyens, Alan Gentz, Frances Kuo and Pierre l'Ecuyer. This work was funded by the US National Science Foundation DMS-1357690 grant and "La Caixa" US graduate studies fellowship.