From: **Robbe, Pieterjan Magda M (-EXP)** pmrobbe@sandia.gov
Subject: Re: [Ext] [EXTERNAL] Questions about MCQMC software
Date: July 3, 2022 at 9:45 PM
To: hickernell@iit.edu

PR

Hi Fred,

Thank you so much for your message! I'm looking forward to meeting you again at MCQMC, too. Let me try and answer your questions:

1) I started creating the Julia packages during my PhD, mainly out of necessity, as there were no QMC packages in Julia yet. They have kind of grown organically depending on the state and focus of the research, I would say. I think the oldest package is the one for multilevel Monte Carlo, which I started during the second year of my PhD, 5 years ago already.

2)

- At Sandia Livermore, we mainly use the uncertainty quantification toolkit (UQTk), which was initiated by Bert Debusschere. It's a lightweight C++/python package with a focus on polynomial chaos surrogate construction and bayesian inference. There are some rudimentary implementations of faster sampling methods, such as latin hypercube sampling, but it's mostly focussed on random sampling. The applications I have been working on at the moment haven't really cried for a pure QMC method just yet: they are only moderate in dimensionality, and code failures is an issue, which makes random sampling attractive again. I've had some discussions with a colleague about adding a QMC option to UQTk, but we haven't done anything in that direction yet.

- Sandia also has Dakota, but that's mostly maintained at the New Mexico site. I believe Dakota does have some basic QMC methods implemented (like Halton or Hammersly), but no lattice rules and no higher-order, so there's definitely room for improvement!

3)

- My personal experience is that it's more useful for end users to make your QMC methods available in their favorite programming language and/or package, instead of simply referring to your own package. Most users don't want to learn a new language or package if all they want to do is simply try out your method, and compare the results to what they already use. This means that you should not only develop and maintain your own package, but also any potential wrapper that you create to link your own (QMC) package to the package of the user.

  As an example, in the Julia community, there is a lot of interest for scientific machine learning. The main application for QMC here is parameter space exploration, and so the developers created a package with a uniform

API for different QMC methods. A couple of years ago, it was nothing more than an empty placeholder with only Sobol sequences implemented. Over the years, many QMC methods have been added, including the Julia code for lattice rules. For this, I had to create some additional wrapper code myself to make sure that the code would fit into the framework.

I feel that this approach creates more user awareness and engagement, but it also comes with a downside. You have to be prepared to maintain your package for a long time. Users will often create feature requests, report bugs or ask questions. This is not always very convenient, as I often forget the things I wrote 3+ years ago.

- Another thing that I find difficult is that it is hard to get reliable statistics on how many users are actually using your package. For example, if the package is on GitHub, you can see the number of clones/downloads of your package over time, but I feel like that's not the right metric for the number of active users, as it's probably an overestimation. On the other hand, if a user has a question or signals an error, then at least you are sure that someone else is using your code. Yet, I rarely reach out to the developer if I have an issue with someone else's code, so counting the number of people that contact you may be an underestimation of the actual number of users. If the group of users is large enough, you could set up a mailing list, but I feel like this is more of an outdated tool.

- Several other issues that we can discuss in more detail:

    - a uniform database with examples to test your methods on, something which is common in the ML community (QMCPy is a step in the right direction!)
    - code maintenance and continuity
    - how to best support code availability in different programming languages
    - ...

I hope this helps you in preparing your talk! Please, let me know if you would need more elaborate answers on some of these points.

Also, a lunch meeting on Tuesday during the conference sounds good!

Happy 4th of July,
Pieterjan

From: Fred Hickernell <hickernell@iit.edu>
Sent: Sunday, July 3, 2022 14:17
To: Robbe, Pieterjan Magda M (-EXP)
Subject: [EXTERNAL] Questions about MCQMC software

Dear Pieterjan

Dear Pieterjan,

Hope that you are doing well this summer and enjoying your time in Sandia or wherever your travels take you. Looking forward to our sessions on MCQMC software.

To prepare my MCQMC 2022 talk on *Challenges in Developing Great MCQMC Software,* I would like update myself on

- When your work on publicly available QMC software, such as your Julia package, began,
- Recent updates to any QMC software that you are involved with, and
- Any challenges in Monte Carlo and quasi-Monte Carlo software development that you would like me to share with the audience.

Any other thoughts relevant to the topic would be appreciated as well.

I am thinking of inviting folk interested in discussing MCQMC software to gather at MCQMC 2022 on Tuesday lunch to discuss where we see the development going and how we can promote the long-term development of such software. It would be an open meeting, but I would like to assess interest. Would you like to join?

If you could reply within a week, that would be most helpful.

Thanks,
Fred

Fred J. Hickernell
Vice Provost for Research, research.iit.edu
www.iit.edu/research/about/magazine
Professor, Department of Applied Mathematics
Center for Interdisciplinary Scientific Computation (CISC) cos.iit.edu/cisc/
Illinois Institute of Technology
Galvin Tower, Room 10F9-1, 10 W 35th St, Chicago, IL 60616
hickernell@iit.edu, +1 312 567 3470

Working in part remotely. Please contact me by email or phone.