Department of Mathematics, University of Utah
Topics in Numerical Solutions of PDE
MTH6630 – Section 001 – Spring 2019

Project 0
Basics of submission

Write a program that uses a Monte Carlo technique to estimate π.

Let $\rho : \mathbb{R}^d \to \mathbb{R}$ be a given probability density and let $f : \mathbb{R}^d \to \mathbb{R}$ be a given function. If X is a random variable distributed according to ρ, then a Monte Carlo estimate of $\mathbb{E}[f(X)]$ using a size-M ensemble is the approximation,

$$\mathbb{E}[f(X)] \coloneqq \int_{\mathbb{R}^d} f(x)\rho(x)\,\mathrm{d}x \approx F_M =: \frac{1}{M}\sum_{m=1}^{M} f(X_m), \tag{1}$$

where $(X_m)_{m=1}^M$ are independent random variables distributed according to the density ρ. Note that F_M itself is a random variable.

Let $R = \left\{(x_1, x_2) \in \mathbb{R}^2 \,\middle|\, x_1^2 + x_2^2 \le 1\right\}$ be the unit ball in $\mathbb{R}^2$. Define ρ as the uniform (probability) density on the square $[-1,1]^2 \subset \mathbb{R}^2$, i.e., $\rho(x_1, x_2) = 1/4$ if both $|x_1|$ and $|x_2|$ are at most 1, and $\rho(x_1, x_2) = 0$ otherwise. Define $f(x)$ as the characteristic function of R, i.e.,

$$f(x) = \mathbb{1}_R(x) = \begin{cases} 1, & x \in R, \\ 0, & x \notin R \end{cases}$$

Since

$$\mathbb{E}[f(X)] = \int_{[-1,1]^2} f(x)\frac{1}{4}\,\mathrm{d}x\,\mathrm{d}y = \frac{1}{4}\int_R \mathrm{d}x\,\mathrm{d}y = \frac{\pi}{4},$$

then F_M defined in (1) is an estimator for $\pi/4$.

Using your programming language of choice, write a program that computes $4F_M$ and hence estimates. (Note that from (1), to generate one instance of F_M you need only the ability to generate instances of the uniform random variable X, and to implement the function f.) Report the convergence of $4F_M$ to π as a function of M. Since F_M is random, you should report ensemble statistics of F_M (e.g., the deviation of a computed mean from π, an ensemble-computed standard deviation, etc.) Use your code to verify the central limit theorem for a sufficiently large M.

Submission instructions

You are required to submit all projects via the version control system Git. You will do so by "pushing" updates to a remote repository. You will need to

1. Complete the assignment above.
2. Write a report regarding your completion of the assignment above. Your report should briefly discuss the assignment, your solution, and document results based on the requested simulations above.

3 Create a git repository on your computer (the "local" repository) that contains both a LaTeX report with figures, and the source code that reproduces the figures as they appear in the report.
4 Create an account on bitbucket (`bitbucket.org`). Do not submit your work through github.
5 Create a (private) repository named `math6630-project-0` on your bitbucket account (the "remote" repository). In your local repository, create a pointer to the remote repository with the `git remote` command.
6 Submit your project to the remote repository with a `git push` command from your local repository. (You will need to create an SSH public/private keypair and upload your public key to bitbucket to accomplish this.)
7 Give me (username `akilnarayan`) write access to your `math6630-project-0` bitbucket repository.

Submission expectations

There are two components of the submission: a LaTeX report and source code. Both are mandatory.

General repository etiquette

- If there are unusual things about the way you've organized your files, provide a `README` text file that documents the general structure of the repository.
- Provide a `makefile` that allows me to easily compile your LaTeX report. If your code is in a compiled language, provide a makefile in the code folder.
- Your repository should in general only track ASCII-type text files (images are an exception). It should not track compiled binaries, pdf output from tex compilation, or datasets generated by your code. In general your repositories should be *small* in disk space, with the largest files probably being the pictures used in your reports.
- General rule of thumb: if I cannot figure out how to generate your LaTeX report and/or use your computer code within 60 seconds of looking at your repository, I will return it to you for you to document things properly.

Submission structure

The base folder of your repository should contain the LaTeX file (`.tex` file) for your report. You may structure all dependent files (images, bib files, secondary tex files, etc.) in any way you choose. The base folder should contain a subfolder called `code` that contains *all* code that reproduces figures in your LaTeX document.

LaTeX report expectations

- Write your report so that someone can understand the assignment without having seen the assignment sheet.
- Extreme verbosity is unnecessary (and should be avoided), but enough detail should be provided so that the general implementation (or your solution for the implementation) is reasonably clear to someone educated in the course material.
- Do not include any source code text in your report.

- Your report is largely an explanation of the mathematics required for your solution, along with a compilation of numerical results.
- Your report should describe the solution in a programming language-agnostic manner. I.e., you should not make reference to any special data structures, code modularity, memory allocation, or syntactical solutions that you had to implement in order to complete the assignment.
- Number all figures and tables, e.g., `Figure 2`, `Table 1`, etc.

Code expectations

- All computer code should be located in the `code` subfolder.
- Include a README text file that explains the purpose of the main files in the subfolder. You should write your README file with the intention of giving a clear guide to a recipient about what they are supposed to do once they receive your code.
- If you use a compiled language, include a `makefile` that generates all necessary binaries.
- Each figure/table in your report should be paired with a file in the `code` subfolder that reproduces that figure/table. E.g., if you are coding in Matlab and your report contains `Figure 2` and `Table 1`, files `figure_2.m` and `table_1.m` should exist in the `code` subdirectory, and running those files should reproduce the figures. (If using a compiled language, binaries named, e.g., `figure_2.o` and `table_1.o` should be generated by your makefile.)

Example submission

I have provided an example submission (for a different assignment) online:
`https://bitbucket.org/akilnarayan/math6630-project-minus1/`
Please check this for a demonstration of the above procedures.