DIPLOMAMUNKA

A GUARDYAN kód számítási hatékonyságának növelése

Pukler Márton

Témavezető: Dr. Légrády Dávid
egyetemi docens
BME NTI

BME NTI
2022. dátum

Diplomamunka témakiírás

Az NTI-ben fejlesztett, az időfüggést direkt szimulációval kezelő reaktordinamikai célú Monte-Carlo alapú GUARDYAN (GPU Assisted Reactor Dynamic Analysis) kód a fél évtizedes fejlesztés eredményeképpen alkalmassá vált atomerőművek tranziens jelenségeinek szimulációjára. Szemben a bevett reaktordinamikai számítási módszerekkel, a GUARDYAN számítási sémája minimalizálja a fizikai folyamatok modellezésénél alkalmazott közelítéseket. A reaktorok gyors reaktortranzienseit érintő biztonsági elemzéseit így validálni lehet a GUARDYAN kód segítségével. A kód hátránya, a futási idő: néhány reaktormásodperc szimulációja néhány tucat óra számítási időt jelent. A hallgató feladata a kód számítási hatékonyságának növelése szóráscsökkentési eljárások segítségével.

A hallgató feladata az értékességfüggvény (adjungáltfüggvény) alapján történő kölcsönhatási törvények torzítási sémájának kidolgozása és alkalmazása az adjungált alapú Wodcock-módszer és a fürkészminták (Sampling Importance Resampling) szögsorsolási technikák segítségével. A hallgató feladata elemezni, hogy az adjungáltfüggvény ismeretének hiányosságai hogyan befolyásolják a gyorsítás mértékét. A hallgató feladata determinisztikusan előállított adjungáltfüggvény alkalmazása, akár eltérő térbeli részletezettségű, egymásba ágyazott adjungált-térképek alapján. A hallgató további feladata az adjungáltfüggvények alkalmazása a GUARDYAN kódban már implementált populáció-alapú fésülés szóráscsökkentési eljárásában is, és az eredményeket saját eredményeivel a hatékonyság növelése szempontjából összevetni. A hallgató feladata emellett a szimuláció hatékonysága szempontjából döntő fontosságú egyéb faktorok azonosítása és lehetőség szerinti javítása.

Önállósági nyilatkozat

Alulírott Pukler Márton a Budapesti Műszaki és Gazdaságtudományi Egyetem fizikus MSc szakos hallgatója kijelentem, hogy ezt a diplomamunkát meg nem engedett segédeszközök nélkül, önállóan, a témavezető irányításával készítettem, és csak a megadott forrásokat használtam fel.

Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból vettem, a forrás megadásával jelöltem.

Budapest, 2022. jú

Pukler Márton

Abstract

:)

Acknowledgements

:)

Contents

1 List of Symbols

$g(x)$	detector function
$\mathbb{p}(x)$	probability density function
$\mathbb{E}$	expected value
$\mathbb{D}$	variance
Σ	cross section
Σ_{maj}	majorant cross section
Σ_{samp}	sampling cross section
Q	sampling probability
q_z	ideal (zero variance) sampling probability
o	oversampling factor
s	stretching factor
χ	ingoing adjoint
ψ	outgoing adjoint

Table 1: a

2 Diplomamunka

2.1 Monte Carlo estimation of integrals

Let us take a look at the following 1D integral:

$$I = \int_{-\infty}^{\infty} f(x)g(x)\,\mathrm{d}x\,, \tag{1}$$

where $g(x)$ is bounded and has a compact support, limiting the integral to a finite interval, this function will be regarded as the pay-off of the integral and will be called detector function. If $f(x)$ is non-negative, bounded and integrable on the same finite interval, we can define a pdf (probability density function) $\mathbb{p}(x)$, by normalizing $f(x)$:

$$\mathbb{p}(x) = \frac{f(x)}{\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x}. \tag{2}$$

If x_i $(i = (1, 2, ..., N)$ samples are drawn from $\mathbb{p}(x)$, the estimate $\hat{I}$ of integral I is the following:

$$I \approx \hat{I} = \frac{1}{N} \sum_{i=1}^{N} \left(g(x_i) \int_{-\infty}^{\infty} f(x) \, \mathrm{d}x \right) = \frac{1}{N} \sum_{i=1}^{N} c_i, \tag{3}$$

where c_i is the contribution of the ith particle to the integral. If we take into consideration that the samples are drawn form the same distribution, the formula for the expected value of a function of a random variable can be applied to equation 3:

$$\mathbb{E}\left[g(x)\right] = \int_{-\infty}^{\infty} g(x) \mathbb{P}(x) \, \mathrm{d}x = \int_{-\infty}^{\infty} g(x) \frac{f(x)}{\int_{-\infty}^{\infty} f(x) \, \mathrm{d}x} \, \mathrm{d}x \,, \tag{4}$$

we obtain that the expected value of the estimator equals the integral:

$$\begin{aligned} \mathbb{E}[\hat{I}] = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} \left(g(x_i) \int_{-\infty}^{\infty} f(x) \, \mathrm{d}x \right) \right] = \frac{1}{N} \mathbb{E}\left[\sum_{i=1}^{N} g(x_i) \right] \int_{-\infty}^{\infty} f(x) \, \mathrm{d}x = \\ = \frac{N}{N} \int_{-\infty}^{\infty} g(x) \frac{f(x)}{\int_{-\infty}^{\infty} f(x)} \, \mathrm{d}x \int_{-\infty}^{\infty} f(x) \, \mathrm{d}x = \int_{-\infty}^{\infty} f(x) g(x) \, \mathrm{d}x = I, \end{aligned} \tag{5}$$

in other words the estimator is unbiased.

2.2 Sampling distributions

Now we need a method for generating random samples from the distribution $\mathbb{p}(x)$. Obtaining uniform (canonical) random variables is simple and efficient in every programming language, by built-in functions. The goal is to transform the random variable with uniform distribution into an arbitrary distribution.

A general technique is called the Inverse Cumulative Method, where the uniform random variable is set equal to the cdf (cumulative distribution function) of an arbitrary distribution. Let r follow a uniform distribution on $[0, 1]$ i.e. $r \sim (0, 1)$, with the pdf $\mathbb{u}(r)$. Let us find a $x = g(r)$ function that transforms r to a variable x with pdf $\mathbb{p}(x)$. From the definition of $g(r)$ follows: $r = g^{-1}(x)$ and the expression for the transformed pdf reads:

$$\mathbb{p}(x) = \mathbb{u}(g^{-1}(x)) \frac{\mathrm{d}}{\mathrm{d}x} g^{-1}(x), \tag{6}$$

where $\mathbb{u}$ is flat 1 while $g^{-1}(x)$ falls into $[0, 1]$, which is satisfied while x is in the domain of the pdf $\mathbb{p}(x)$. By integrating both sides:

$$\int_{-\infty}^{x} \mathbb{p}(x')\,\mathrm{d}x' = \mathbb{P}(x) = g^{-1}(x) = r \tag{7}$$

we can see that to obtain the desired transformation we need to invert the cdf:

$$r = g(x) = \mathbb{P}^{-1}(x) \tag{8}$$

That is where the name Inverse Cumulative Method comes from. Often the cdf can not be analytically inverted, in that case the inversion must be done numerically, for example by the Newton-Raphson method. Even more often the cdf can not be analytically calculated, in that case the pdf should be integrated numerically to obtain the cdf, and finding the inverse becomes a table look up method.

Another frequently used technique for sampling distributions is called rejection sampling. Let $\mathrm{maj}(\mathbb{p}(x))$ be a majorant value to the pdf $\mathbb{p}(x)$, after that an uniform random number is sampled on the domain of $\mathbb{p}(x)$ using a canonical random number. Next, a new canonical random number r is sampled, the sample x is accepted if the condition:

$$\frac{\mathbb{p}(x)}{\mathrm{maj}(\mathbb{p}(x))} < r \tag{9}$$

is satisfied, otherwise both x and r are rejected (discarded) and new ones are drawn until a sample is accepted. This method is ineffective in case of very sharp distributions, but it can be improved by using different majorants on different sections of the distributions, in that case first a section must be sampled.

2.2.1 Sampling Beer-Lambert law

One of the most frequently sampled distribution in particle transport Monte Carlo simulations is the exponential distribution. The reason for this is that the free path of particles between two interactions follows exponential distribution, this is called Beer-Lambert law. The pdf of the free path in homogeneous media with constant Σ (total) cross section is the following:

$$\mathbb{p}(x) = \Sigma e^{-\Sigma x} \tag{10}$$

Using the formulas from equations 7 and 8, we obtain:

$$1 - e^{-\Sigma x} = r \;\longrightarrow\; x = -\frac{1}{\Sigma}\ln(1 - r) \tag{11}$$

As $r \sim U(0,1)$, r and $1-r$ has the same pdf, thus $1-r$ in the argument is often swapped out by r resulting:

$$x = -\frac{1}{\Sigma}\ln(r) \tag{12}$$

In this later case smaller r means larger x and vice versa.

2.2.2 Sampling free path in piecewise constant media

In inhomogeneous media the (total) cross section depends on the position: $\Sigma(x)$. In general the pdf of free path in inhomogeneous media:

$$\mathbb{p}(x) = \Sigma(x) e^{-\int_0^x \Sigma(x')\mathrm{d}x'} \tag{13}$$

This yields the cdf:

$$\mathbb{P}(x) = 1 - e^{-\int_0^x \Sigma(x')\mathrm{d}x'} = r \tag{14}$$

By reordering the terms we obtain:

$$-\ln(1-r) = \int_0^x \Sigma(x')\,\mathrm{d}x' \tag{15}$$

This equation holds in any inhomogeneous media, but often inhomogeneous media can be split into piecewise constant cross section regions. In that case the integral on the right hand side can be swapped out by a summation:

$$-\ln(1-r) = \sum_{i=1}^{N} \Sigma_i l_i + \Sigma_{N+1}\Delta x \tag{16}$$

where Σ_i are the constant cross sections of the regions the particle travels through along its free path, l_i are the distances travelled in each of these sections. The $-\ln(1-r)$ term is called optical distance and it is decreased by $\Sigma_i l_i$ $(1, 2, ..., N)$ until it becomes 0 while passing through the (N+1)-th region. This way the sampled free path becomes:

$$x = \sum_{i=1}^{N} l_i + \Delta x \tag{17}$$

2.2.3 Woodcock tracking

Sometimes the cross section changes continuously, e.g. due to gradient material density or temperature change, thus the integral in equation 15 has to be evaluated

numerically.

Another approach to sampling free path in inhomogeneous media is called Woodcock or delta-tracking. The idea behind the (analog) Woodcock tracking is that instead of using the local cross section, an arbitrary cross section is used to sample free path. The chosen sampling cross section must be majoring the cross section at any position in the system and is called majorant cross section:

$$\Sigma_{maj} \geq \max_{V} \Sigma(x) \tag{18}$$

where V denotes the volume of the system. The cross section can also depend on the energy of the particles passing trough the material, in that case the majorant must be taken in energy as well. A sampled collision point is accepted if a random number drawn from an uniform distribution is smaller than the ratio of the local cross section (at the sampled collision point) to the majorant cross section e.g. the probability of accepting is: $p = \Sigma(x)/\Sigma_{maj} \leq 1$. The probability of rejecting a sampled collision point is $1 - p$. When a collision point is rejected a virtual (delta) scatter occurs and the particle continues its flight without changing direction or energy. The sampling continues until a sampled collision point is accepted.

Proof in cite...

2.3 Sampling multidimensional probability density functions

In multiple dimensions, Monte Carlo integrals, detector functions and the definition of expected value are essentially the same as in section 2.1, the only difference is that the scalar variable x has to be swapped out for a (usually 3D or 2D) vector $\underline{r}$.

On the other hand sampling multidimensional pdf-s is not as trivial as in the 1D case. As an example in 3D the process is the following: let $\mathbb{p}(x, y, z)$ be the joint pdf of a 3D distribution. This pdf can be factored into conditional pdf-s using Bayes' theorem:

$$\mathbb{p}(x, y, z) = \mathbb{p}(y, z|x)\mathbb{p}(x) = \mathbb{p}(z|x, y)\mathbb{p}(y|x)\mathbb{p}(x) \tag{19}$$

where the marginal pdf-s are the following:

$$\mathbb{p}(y, z|x) = \frac{\mathbb{p}(x, y, z)}{\mathbb{p}(x)} = \frac{\mathbb{p}(x, y, z)}{\iint \mathbb{p}(x, y, z)\,\mathrm{d}y\,\mathrm{d}z} \tag{20}$$

and

$$\mathbb{p}(z|x, y) = \frac{\mathbb{p}(y, z|x)}{\mathbb{p}(y|x)} = \frac{\mathbb{p}(y, z|x)}{\int \mathbb{p}(y, z|x)\,\mathrm{d}z} = \frac{\mathbb{p}(x, y, z)}{\iint \mathbb{p}(x, y, z)\,\mathrm{d}y\,\mathrm{d}z \int \mathbb{p}(y, z|x)\,\mathrm{d}z} \tag{21}$$

Now the three (1D) pdf-s from equation 19 have to be sampled one-by-one (e.g. by

using inverse cumulative or any other method). First x_i is sampled from $\mathbb{p}(x)$, than y_i is sampled from $\mathbb{p}(y|x_i)$, lastly z_i is sampled from $\mathbb{p}(z|x_i, y_i)$, thus the desired sample from $\mathbb{p}(x, y, z)$ is (x_i, y_i, z_i).

2.3.1 Sampling uniformly on the surface of a unit sphere

One of the most often sampled 3D distributions in Monte Carl transport simulations is the sampling of isotropic direction or in other words sampling uniformly on the surface of a unit sphere. It is mostly used in isotropic scatter and source events.

One of the most efficient and most often used method for sampling on the surface of a sphere was developed by Marsaglia CITE. The method does not rely on the technique proposed in section 2.3, instead uses a clever trick to generate three independent variables from two uniform random variables, using rejection.

The method goes like this: x_1, x_2 uniform random variables are sampled until $s = x_1^2 + x_2^2 \leq 1$, otherwise rejected. From the remaining x_1, x_2

$$(x, y, z) = (x_1\sqrt{1-s}, x_2\sqrt{1-s}, 1-2s) \tag{22}$$

is uniformly distributed on the surface of an unit sphere.

The disadvantage of this technique is that it uses rejection which is disadvantageous for the parallel calculations on the GPU.

2.4 Variance estimation

In most cases the best measure for the reliability of a Monte Carlo simulation is variance. The variance of the estimate of a Monte Carlo integral in equation 3 is the following:

$$\mathbb{D}^2[\hat{I}] = \mathbb{D}^2\left[\frac{1}{N}\sum_{i=1}^{N} c_i\right] = \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{D}^2[c_i] = \frac{1}{N}\mathbb{D}^2[c_i] = \frac{1}{N}\mathbb{D}^2[c] \tag{23}$$

The last two equality are valid because each c_i are independent and have the same variance: $\mathbb{D}^2[c]$, as they are sampled from the same distribution. To estimate $\mathbb{D}^2[c]$, the definition of variance is needed to be used:

$$\mathbb{D}^2[c] = \mathbb{E}[c^2] - \mathbb{E}^2[c] \approx \frac{1}{N}\sum_{i=1}^{N} c_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} c_i\right)^2 \tag{24}$$

The expected value of the obtained estimation for the variance has to be taken to see if it is unbiased:

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}c_i^2-\left(\frac{1}{N}\sum_{i=1}^{N}c_i\right)^2\right]=\frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}c_i^2\right]-\frac{1}{N^2}\mathbb{E}\left[\sum_{i=1}^{N}c_i\right]^2=$$
$$=\mathbb{E}[c^2]-\frac{1}{N^2}\left(N\mathbb{E}[c^2]+N(N-1)\mathbb{E}^2[c]\right)=\frac{N-1}{N}(\mathbb{E}[c^2]-\mathbb{E}^2[c])=\frac{N-1}{N}\mathbb{D}^2[c] \tag{25}$$

Thus the unbiased form of the estimation of the variance is:

$$\mathbb{D}^2[\hat{I}]\approx\frac{1}{N-1}\left(\frac{1}{N}\sum_{i=1}^{N}c_i^2-\left(\frac{1}{N}\sum_{i=1}^{N}c_i\right)^2\right)\approx\frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}c_i^2-\left(\frac{1}{N}\sum_{i=1}^{N}c_i\right)^2\right) \tag{26}$$

Despite being biased, in most cases the $1/(N-1)$ term is swapped out by $1/N$ especially for large N values, where the difference is insignificant.

It is worth to be noted that the c_i contribution is meant to be the total contribution delivered by the whole sampling process of a single particle. The reason for this is that the partial contributions from the same starting particle are correlated and it is really hard to estimate the correlation. By adding the partial contributions to get the total c_i, this problem is eliminated.

2.4.1 Relative variance

Often the relative variance: $\mathbb{r}^2[\hat{I}]$ carries much more information regarding the efficiency of the sampling than the (absolute) variance:

$$\mathbb{r}^2[\hat{I}]=\frac{\mathbb{D}^2[\hat{I}]}{\mathbb{E}^2[\hat{I}]}\approx\left(\sum_{i=1}^{N}c_i^2\right)\Big/\left(\sum_{i=1}^{N}c_i\right)^2-\frac{1}{N} \tag{27}$$

From equation 23, it can be seen that the variance is inversely proportional to the number of samples: $\mathbb{D}^2[\hat{I}]\sim 1/N$. From the definition of the relative variance the same dependency can be seen: $\mathbb{r}^2[\hat{I}]\sim 1/N$. This means that, to reduce relative variance of an integral by a factor of n, the number of samples has to be multiplied by n^2.

Even more often the square root of the variances, the (absolute) standard deviation $\mathbb{D}$, and the relative standard deviation $\mathbb{r}$ are used as they can be compared to the estimation directly. Naturally the standard deviations of Monte Carlo simulations is inversely proportional to the square root of the number of samples: $\mathbb{D}\sim 1/\sqrt{N}$ and $\mathbb{r}\sim 1/\sqrt{N}$.

2.4.2 FoM

An other important measure of the efficiency of a Monte Carlo simulation is the FoM (Figure of Merit). It is defined in a way that, if the same simulation runs with different numbers of samples i.e. for different amounts of time T, than the FoM should be roughly constant:

$$FoM = \frac{1}{T\mathbb{r}^2} \tag{28}$$

This is of course the direct consequence of the fact that: $\mathbb{r}^2 \sim 1/N$ and $T \sim N$. The higher the FoM the more efficient the simulation is, the goal of variance reduction techniques is to increase FoM.

2.5 Non-analog sampling

Now, that the variance and other measures of efficiency are defined, methods to decrease the the variance are needed. The vast majority of Monte Carlo variance reduction techniques are based on the same principle, by performing non-nature mimicking, so called non-analog sampling. Non-analog sampling must be unbiased and should be able to reduce the variance, but if not used correctly the variance can increase as well. This technique is also useful in case of a pdf that is difficult to sample.

Let $\mathbb{p}'(x)$ be the new pdf used for sampling. Without changing the expected value, $\mathbb{p}'(x)$ is plug into equation 1, thus the integral becomes:

$$I = \int_{-\infty}^{\infty} g(x)\frac{f(x)}{\mathbb{p}'(x)}\mathbb{p}'(x)\,\mathrm{d}x \tag{29}$$

A new detector function $g'(x)$ can be defined:

$$g'(x) = g(x)\frac{f(x)}{\mathbb{p}'(x)} \tag{30}$$

and with

$$w(x) = \frac{f(x)}{\mathbb{p}'(x)} \tag{31}$$

$$c_i = g(x_i)w(x_i) = g(x_i)w_i \tag{32}$$

2.6 Variance reduction techniques

2.6.1 FoM

2.6.2 Russian roulette

2.6.3 SIRS

3 Zero Variance Transport Monte Carlo

exp tr

4 A

Let I be equal to the following 1D Monte Carlo integral:

$$I = \int_{-\infty}^{\infty} f(x)g(x)\,\mathrm{d}x \tag{33}$$

where $g(x)$ is called the pay-off or the detector function of the integral. Let $g(x)$

References

[1] `http://awing.reak.bme.hu/GUARDYAN`

[2] Molnar B., *A GPU based direct Monte Carlo Simulation*, 2018.

[3] T. Viitanen, *Development of a stochastic temperature treatment technique for Monte Carlo neutron tracking*, 2015.

[4] E. Woodcock, et al., *Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry*, 1965.

[5] J. F. Briesmeister, et al., *MCNP-A general Monte Carlo N-particle transport code*, 2000.

[6] Tolnai G., *GUARDYAN fejlesztési segédlet*, 45-46, 2018.

[7] Everett, C.J., and Cashwell, E.D., *Third Monte Carlo sampler. Revision and extension of samplers I and II.*, 1983.

[8] `https://www.nndc.bnl.gov/endf/b7.1/acefiles.html`