TRINITY COLLEGE DUBLIN, SCHOOL OF MATHEMATICS

Assignment #3, Module: MA5634

Gustavo Ramirez

January 23, 2017

General explanations to understand the structure and implementations of this solution to the assignment are given here.

1 METROPOLIS CORE CODE

A specific and somewhat clear explanation of the Metropolis algorithm can be found in the file ./part_a/metropolis_description.pdf.
An implementation of part a of this assignment, based on the explanation in that pdf file, can be found in the file ./part_a/program.py.
Also, in the same directory (i.e. ./part_a/) a .png file plotting the samples can be found, to check that the obtained distribution is the appropriate.

2 SIMULATIONS FOR THREE VALUES OF Δ

The results for this part are more diverse.
In the directory ./part_b/ can be found a set of image files whose names start with "HISTOGRAM_MC_history_D_", which are the histograms for the six different samples generated for the six different integrations carried out (three different values of Δ were used for each function: $\cos(x)$ and x^2).
On the other hand, the image files starting with "MC_history_D_" correspond to the plots of MC history, for each of the six cases.
A particular execution and results for the integrals, in the case of using a number of states equal to 50000 is:

DELTA = 0.6
****
– function integrated: cosine
result: 0.79270421652
– function integrated: x^2
result: 0.464075256692
DELTA = 0.7
****
– function integrated: cosine
result: 0.802398788268
– function integrated: x^2
result: 0.466724006355

DELTA = 0.8

****

– function integrated: cosine

result: 0.792037725042

– function integrated: x^2

result: 0.466929212051

As can be seen in the MC histories plotted, the data is so dense that the data can be visualized almost like a single stripe. And it's particularly interesting that, the bigger the value of Δ, the wider that stripe, which can be seen as an indirect and informal measurement of the variance.

3 "RIGHT" CHOICE FOR Δ

To be able to perform more simulations, in this case the number of sampled states was changed to 5000.

The file ./part_c/program.py was modified with respect to the previous two parts, because of the innecessary code from those previous integrations.

In image ./part_c/accept_ratio_vs_delta.png can be found a plot of acceptance ratio vs. the selected value of Δ for that specific sampling; as can be seen there, the smaller the value of Δ, the better the acceptance of states in the Metropolis algorithm. But what is "good" for an acceptance ratio also depends on the specific problem under evaluation.

Here, a value of $\Delta = 0.2$ will be chosen, which leads to an acceptance rate of approximately 90%.

4 TRUE VARIANCE

The number of states sampled in this case, were $N = 50000$.

The selected value of $\Delta = 0.2$ is an informal measure for the variance. Now, the true variance of the data, for that value of Δ, will be measured.

The binning of data is taking a value b_s, the bin size, and averaging in chunks size b_s (from the whole data set of sampled values of total size N), and in the end obtaining a binned array of size N/b_s.

After binning the data, a variance can be calculated for that data, for the specific value of the function $f(x)$ to be used as integrand, obtaining with this a value of the value for the binned data of: $V^{b_s}(\bar{f})$; if the sampled array is not binned, then the obtained value for the variance is the naive calculation: $V^0(\bar{f})$. The integrated autocorrelation time is calculated as:

$$\tau_{int} = \lim_{b_s \to \infty} \frac{V^0(\bar{f})}{V^{b_s}(\bar{f})} \tag{4.1}$$

Results for integrated autocorrelation times and variances, in these two cases of integration, can be found in the images within the directory ./part_e/.

5 Statistical "independence"

For a group of n random variables:

$$var(X_1 + X_2 + ... + X_n) = \sigma^2_{total} = n\sigma^2 \tag{5.1}$$

where each independent distribution has variance σ^2.

Here, a bin size of $b_s = 30$ is selected, to illustrate the numerical calculations; for this bin size, the autocorrelation time is of approximately 1.273 in the case of the function x^2, and of approximately 1.2 in the case of the function $\cos(x)$. For this choice of bin size, it can be noticed, from the results of the previous part of this assignment, that the corresponding variances are: $V(\cos(x)) \approx 0.065$ and $V(x^2) \approx 0.4$.

When integrating with a set of n independent random variables, the variance and the expectation value are calculated as:

$$V(f) = \frac{\sum_{i=1}^{n} V_i(f)}{n} \tag{5.2}$$

$$E(f) = \frac{\sum_{i=1}^{n} E_i(f)}{n} \tag{5.3}$$

and, on the other hand, if done with just one set of data instead of n sets of sampled data:

$$\mu = E(f) = \frac{\sum_{i=1}^{N} f(x_i)}{N} \tag{5.4}$$

$$V(f) = \frac{\sum_{i=1}^{N} (f(x_i) - \mu)^2}{n} \tag{5.5}$$

with: $N = 2\tau_{int} \cdot n$ being the number of sampled states in this single Markov chain.
In this specific implementation, the value $N = 50000$ was selected, from which, for the function x^2 the value $n = N/(2\tau_{int}) \approx 19\,639$ is obtained, and for the funcion $\cos(x)$ the value $n \approx 20\,833$ is instead used.
When executing ./part_e/program.py, the results of one execution are (with two different values of n, one for each of the two integrands):

DELTA = 0.2

****

– function integrated: cosine

result: 0.777896870441

error: 0.000903912659038

– function integrated: x^2

result: 0.407638436078

error: 0.0923615639 22

Integration through n statistically independent Gaussian:

– result for cosine: 0.779399270054

– error for cosine: 0.000598486954

– result for x^2: 0.508391392359

– error for x^2: 0.00839139235865