Solving Classical Systems using Monte-Carlo Simulation Techniques

Diptarko Choudhury, 2011061

School of Physical Sciences, NISER, Bhubaneswar

December 1, 2022

Problem Statement

Often when solving complex physical systems in the classical domain, a bunch of constraints have to be imposed on the system, which includes Newton's laws of motions, and corresponding Conservation laws, characterisation of the system, and it's domain. Generally, solutions to these equations are not trivial, and we need to settle for results obtained through approximation methods. In this project, one of the most famous approximation method, often used by physicists, which Monte Carlo importance sampling and simulation, which is done to understand the system [1].
Monte Carlo simulation is a statistical method, which works on the principles of approximating an expectation by using a probability distribution of similar kind, through random sampling [2].
In Monte Carlo importance sampling, the distribution from which random sampples are drawn, happen to be of more importance. Hence, it is necessary for a practitioner to choose a proper probability distribution, which is suitable for the given problem.
In this project, I am trying to find the position and momentum of a simple harmonic oscialltor, using Monte Carlo Simulation technique. This school of thought can be expanded and applied on a large number of problems, whose initial conditions and constraints are known, but are particularly difficult to solve. For example, plasma simulation, Kinetic model of gases, etc.

Theoretical Framework

Monte Carlo methods are not only restricted to evaluation of integrals, but can also be used for probabilistic modelling which are stochastic in nature. In Monte-Carlo integration techniques, or importance sampling integration techniques, we use a weighted sum technique to evaluate our integrals. The nature of the weigth function should be similar to the nature of the integrand. In our case, we are using the weight function as a way to determine stochastically, various parameters of the system under study.
Considering the example of a one dimensional simple harmonic oscillator, in the usual way. The spring is considerd to be massless and with associated spring constant (k), with a point mass (m), attached to it, as described in the figure below.

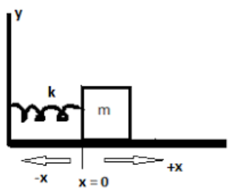

Figure 1: Diagrammatic representation of the System

This system is a very simple system, whose solutions can be analytically obtained exactly. But the sake of demonstration and for availability of a control, I am trying to solve this system.
In this case, we know the equations of motion that governs this system:

$$m\frac{d^2x}{dt^2} = \sum_i F_i \tag{1}$$

For the system, that we have picked up, the equations stand as:

$$m\frac{d^2x}{dt^2} = -kx \tag{2}$$

The analytical solution for this equation, with the origin set at the static equilibrium of the system, we obtain:

$$x = A\cos(\omega t) \tag{3}$$

where, A is the amplitude of of the oscillation.
We proceed to solve this equations using Monte Carlo methods. We first linearize the differential equatins as follows:

$$m\frac{dv}{dt} = -kx \tag{4}$$

$$m\left(\frac{v_{n+1} - v_n}{t_{n+1} - t_n}\right) = -kx_{n+1} \tag{5}$$

We take $t_{n+1} - t_n = \Delta t$, where each time step is Δt. We define χ_n to be a quantity that gives us the physical state of the system at some time t_n. Now, this can be approximated into the following equations, if we consider the simluation steps, the equation (4), linearizes into (5). To solve equation (5), we need to determine, both v_{n+1} and x_{n+1}. Hence, for every degree of freedom, we will have to determine 2n number of parameters, to completely specify the system. In the case of a three dimensional system, we need to determine x_n, y_n, z_n and the corresponding velocities $(v_x)_n, (v_y)_n, (v_z)_n$. In our case Δt is a user defined parameter, which helps in fine tuning our solutions and is related to the granularity of the same.
If we take Δt to be sufficiently small, then we can approximate equation (5), to be:

$$m\left(\frac{v_{n+1} - v_n}{t_{n+1} - t_n}\right) = -kx_n \tag{6}$$

This reduces our problem of determining two variables into one variable.
Suppose we draw x from a given probability distribution $p(x)$.

$$p(|x - \hat{x}| \leq \epsilon)p(|v - \hat{v}| \leq \epsilon) \tag{7}$$

Where, $|x - \hat{x}| < \epsilon$, is our convergence criteria, $\hat{x}$, being the predicted value, and x, being the theoretical value. This shows that the complexity of the problem increases exponentially if the number of independent parameters, which are sampled from the probability distribution are large. This is the reason, why we made approximation in equation (6). For other systems like, the two dimensional Keplerian problem can be reduced to a single independent parameter, using approximation and conservation of angular momentum. This reduces the complexity of the problem by fourth square root of the previous complexity.

Algorithm

In the previous sections, we have made it clear that we will be only sampling a single varibale, which in this case is v, for making our simulations, and use similar approximations to find rest of the parameters.
From basic intuition and empirical observation, we can see that the oscillator spends some of it's time at very low velocities (at extremum positions), also the oscialltor accelerates rapidly, when it comes towrads the mean position.
If we use a gaussian distribution, a large number of values, will be present within the range $[-\sigma, +\sigma]$, and if proper care is not taken in scaling the distribution using the velocity profile, we might never find the required velocity to match convergence criteria of the solutions, who get proper convergence on the high velocity regime. We scale the gaussian, with $\sigma = \frac{v_{max}}{2}$.
The next problem which arises, is when v_{max} is large. The standard deviation will be high, and gaussian will spread more over space, which in turn will make sampling from low velocity do-

main, difficult. To compensate for this, we will superimpose, a log uniform distribution with the gaussian distribution.
clearly from figure 3, we can see that, for a very high v_{max}, which in this case is 200, the probability distribution is more or less uniform when seen in a small window $[-10, 10]$. Which shows, that getting values close to zero, becomes imporbable in this distribution, whereas in figure 4, when we see the probabilty density near zero, we find the peak which makes it more probable for our sampler to find velocities close to zero. More details is provided in the jupyter notebook, attached.

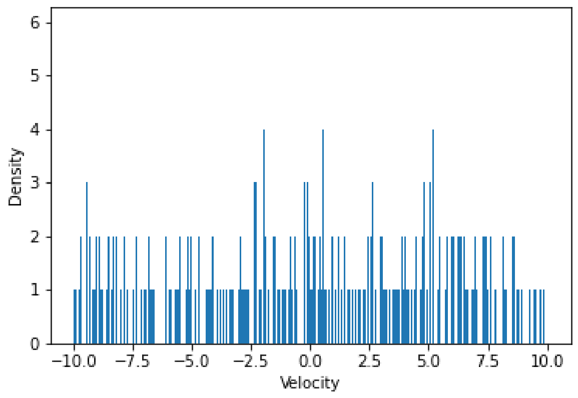

Figure 3: Gaussian low velocity regime

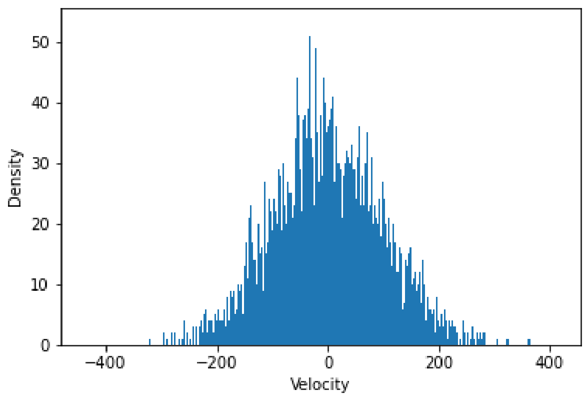

Figure 2: Gaussian Distribution

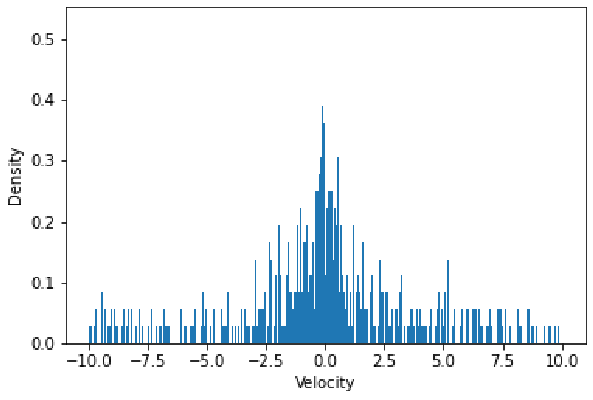

Figure 4: Superimpose Gaussian low velocity regime

The algorithm is thus:

```
1. Sample $v_{n+1}$ from the given probability distribution
2. Choose the $v_{n+1}$, which is closest to the convergence criteria
3. if the convergence criteria is achieved
   $x_{n+1} = x_n + v_{n+1}(t_{n+1} - t_n)$
4. else reduce the convergence criteria and return to step 1
```

From this algorithm starting with v_0, x_0, after n iterations we will reach v_n, x_n. For this particular case the convergence criteria is:

$$\left| \frac{kx_n - m\frac{v_{n+1} - v_n}{t_{n+1} - t_n}}{kx_n} \right| \leq \epsilon \tag{8}$$

where ϵ is the user defined precision.

Results and Conclusion

Below we have provided the sample solution that we have obtained by solving the one dimensional simple harmonic oscialltor, by using initial condition as $x = 1$ and $v = 20$, $m = 2$, $k = 20$.

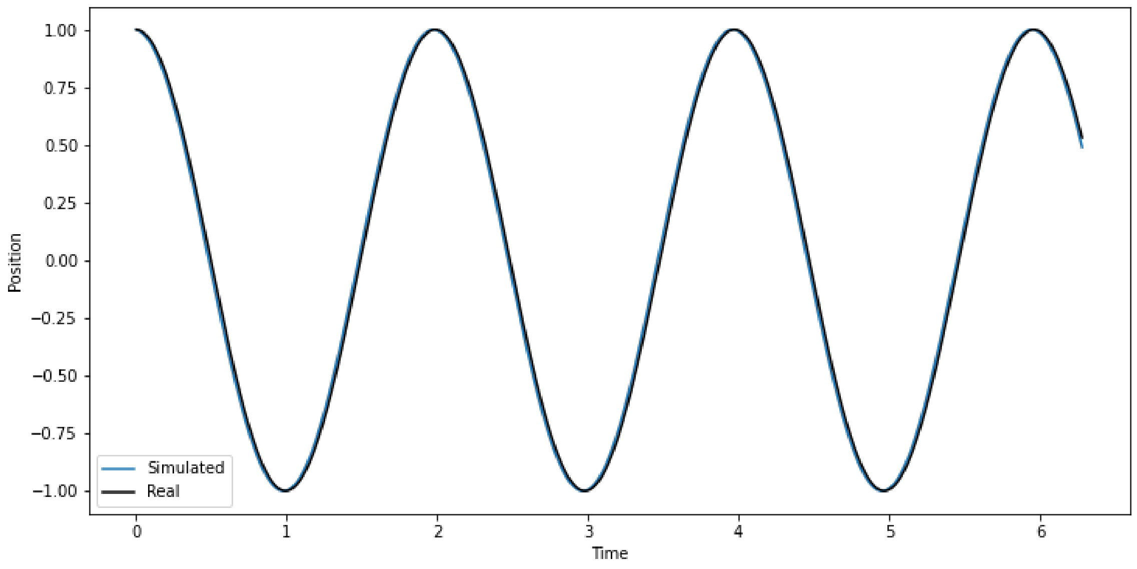

Figure 5: Solution from simulation

We have used Numpy and CuPy extensively throughout our program due to the fact that Monte-Carlo simualation is extremely time consuming and is difficult to vectorize with using the LCG code written in the class. CuPy is specifically used to add CUDA capabilities to our code so that we can massively parallelise the process, the benchmarks, are given in the attached jupyter notebook..
All codes and simulations were run on Perlmutter Cluster (Nersc-9).

References

[1] Importance sampling, November 2022. Page Version ID: 1124796758.

[2] T. Kloek and H. K. van Dijk. Bayesian estimates of equation system parameters: An application of integration by monte carlo. *Econometrica*, 46(1):1–19, 1978.