UCSB, Physics 129AL, Computational Physics: Section Worksheet, Week 9A

Zihang Wang (UCSB), zihangwang@ucsb.edu

March 4, 2025

Section Participation and Submission Guidelines

Section attendance is required, but you do not need to complete all the work during the section. At each section, the TA will answer any questions that you might have, and you are encouraged to work with others and look for online resources during the section and outside of sections. Unless otherwise stated, the work will be due one week from the time of assignment. The TA will give you 1 point for each task completed. You can see your grades on Canvas.

We will use GitHub for section worksheet submissions. By the due date, you should have a single public repository on GitHub containing all the work you have done for the section work. Finally, upload a screenshot or a .txt file to Canvas with your GitHub username and repository name so the TA knows who you are and which repository you are using for the section.

Remember: talk to your fellow students, work together, and use GPTs. You will find it much easier than working alone. Good luck! All work should be done in the Docker container, and don't forget to commit it to Git!

Task 1: Black Body Radiation

The total rate at which energy is radiated by a black body per unit area over all frequencies is,

$$W = \frac{2\pi k_B^4 T^4}{c^2 h^3} \int_0^\infty \frac{x^3}{e^x - 1}\, dx \tag{1}$$

A)

Write a function to to evaluate the integral in this expression. You will need to change the variables to go from an infinite range to a finite range. What is the change of variable and new functional form? The variable to go from range 0 to ∞ to a finite range of is

$$z = \frac{x}{1+x}$$

or equivalently

$$x = \frac{z}{1-z}$$

Here are some constants and the result.

$$\begin{aligned}
k &= 1.38064852 \times 10^{-23}\,\mathrm{J/K} \\
h &= 6.626 \times 10^{-34}\,\mathrm{J\cdot s} \\
\pi &= \pi \\
c &= 3 \times 10^{8}\,\mathrm{m/s} \\
\hbar &= \frac{h}{2\pi} \\
\text{prefactor} &= \frac{k^4}{c^2 \hbar^3 4\pi^2} \\
\text{True value} &= 5.670367 \times 10^{-8}\,\mathrm{W/m^2K^4}
\end{aligned} \tag{2}$$

B)

According to Stefan's law, the total energy given off by a black body per unit area per second is given by

$$W = \sigma T^4.$$

Use the integral to calculate the value of the Stefan-Boltzmann constant, σ. Use the `fixed_quad` function to perform the integral.

C)

Inbuilt 'quad' function can support an infinite range for integration. Write another function to do the integration from 0 to ∞ and compare your answer.

Task2: Orbits

Consider the following two-body problem, wherein a single planet orbits around a large star. Stellar mass is much larger than planetary mass, so we choose the star as the center of our coordinate system. Now, consider the planet's two-dimensional elliptical orbit around the star. The position of the planet is given by the coordinates $q = (q_1, q_2)$, with the planet's velocity given by $p = \dot{q}$.

Newton's laws, with a suitable normalization, yield the following ordinary differential equations:

$$\ddot{q}_1 = -\frac{q_1}{(q_1^2 + q_2^2)^{3/2}}, \quad \ddot{q}_2 = -\frac{q_2}{(q_1^2 + q_2^2)^{3/2}}.$$

This is equivalent to a Hamiltonian system with the Hamiltonian:

$$H(p,q) = \frac{1}{2}(p_1^2 + p_2^2) - \frac{1}{\sqrt{q_1^2 + q_2^2}},$$

where $p_i = \dot{q_i}$.

We will consider the initial position and velocity of the planet to be:

$$q_1(0) = 1 - e, \quad q_2(0) = 0, \quad \dot{q_1}(0) = 0, \quad \dot{q_2}(0) = \sqrt{\frac{1+e}{1-e}}.$$

Now determine q as a function of time t.

0.1 A)

Using 100000 steps, use the explicit Euler method (Let $f(q) = \frac{dq}{dt}$. Then, $q(t + \Delta t) = \Delta t \cdot f(q)$ for small Δt) and plot the orbit of the planet. Assume $e = 0.6$ and integrate to a final time of $T_f = 200$.

$$q_{n+1} = q_n + \Delta t \cdot \dot{q_n}$$
$$\dot{q}_{n+1} = p_{n+1} = p_n + \Delta t \cdot \dot{p_n}$$

0.2 B)

Using 400000 steps, use the symplectic Euler method.

$$p_{n+1} = p_n - \Delta t H_q(p_{n+1}, q_n)$$
$$q_{n+1} = q_n + \Delta t H_p(p_{n+1}, q_n)$$

or

$$q_{n+1} = q_n + \Delta t H_p(p_n, q_{n+1})$$
$$p_{n+1} = p_n - \Delta t H_p(p_n, q_{n+1})$$

where H_p and H_q denote the column vectors of partial derivatives of the Hamiltonian with respect to p and q, respectively. i.e. $H_{p_1} = p_1, H_{q_1} = \frac{q_1}{(q_1^2+q_2^2)^{3/2}}, H_{p_2} = p_2, H_{q_2} = \frac{q_2}{(q_1^2+q_2^2)^{3/2}}$. Plot the orbit of the planet. Compare your results in A) and B) by plotting both solutions in the same figure.

Task3:Gradient descent, Metropolis–Hastings algorithm, Simulated Annealing

Gradient descent is a deterministic method for optimization. It requires the function to be differentiable and convex.

Let $H(\theta)$ be the cost or loss function, where θ represents the model parameters. The goal is to find the optimal θ that minimizes $H(\theta)$.

The process of gradient descent involves starting with an initial guess for θ_i, then iteratively applying the update rule until convergence. Convergence is typically determined by monitoring the change in the cost function or the norm of the gradient.

The update rule for gradient descent is as follows:

$$\theta_{i+1} = \theta_i - \alpha_i \cdot \nabla H(\theta_i),$$

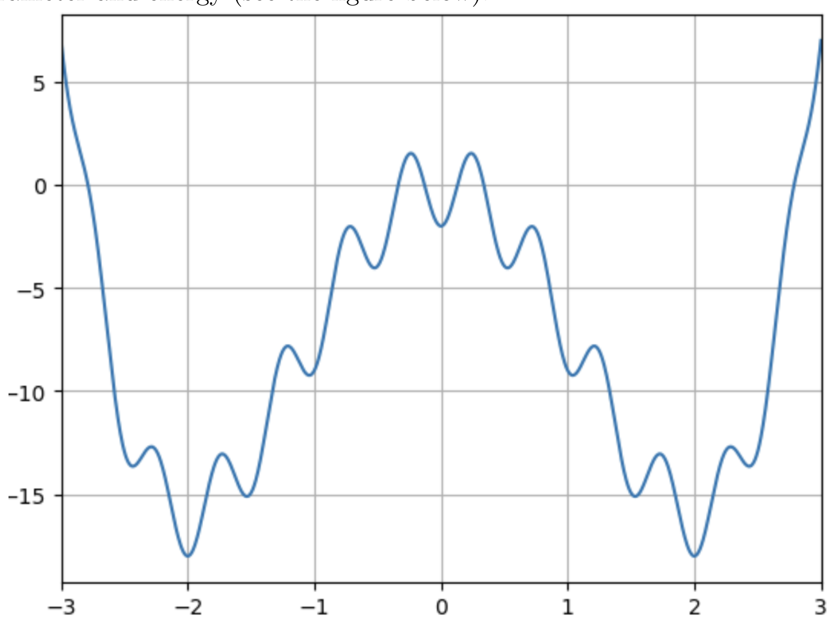
where α_i is the learning rate, which is a hyperparameter that determines the step size of each update. The choice of the learning rate (α) is crucial in gradient descent, as it can affect the algorithm's convergence and stability. It often requires experimentation to find an appropriate learning rate for a specific problem. The gradient (∇H) is a vector that contains the partial derivatives of the cost function with respect to each parameter in θ. It represents the rate of change of the cost function with respect to each parameter and guides the updates.

$\phi 4$ theory in 1D

Let's say you have a noisy $\phi 4$ theory in 1D, given by:

$$H = \theta^4 - 8\theta^2 - 2\cos(4\pi\theta),$$

where θ is an order parameter. You want to find the ground state order parameter and energy (see the figure below).



A)

Using the gradient descent method, locate the global minimum starting with three initial guesses $\theta_0 = -1, 0.5, 3$. For each descent step, plot a **red dot** on the above plot and save it locally until it converges. Make a **video** by processing your saved figures (you may want to look into "cv2.VideoWriter"). You should tune the learning parameter at each step! Do you get consistent results?

B)

As we discussed in class, the Metropolis–Hastings algorithm is a Monte Carlo method that is used for optimization. In here, we will look at some basics. In Bayesian inference, the posterior can be expressed as Boltzmann factors:

$$P(\theta) = \frac{e^{-\beta H(\theta)}}{Z},$$

where H is the Hamiltonian, $\beta = \frac{1}{kT}$, and Z is the partition function. Let's start with an initial parameter guess θ_0. Let's randomly move from $\theta_1 \to \theta_0 + \Delta\theta$, where the step follows a Gaussian distribution $\Delta\theta \sim \mathcal{N}(0, \sigma)$ (Markov process). Note, it must be symmetric. Then, the ratio is:

$$r = \frac{e^{-\beta H(\theta^*)}}{e^{-\beta H(\theta)}} = e^{-\beta H(\theta^*) + \beta H(\theta)} = e^{-\beta \Delta H(\theta^*, \theta)}.$$

If $r > 1$, we accept it and set $\theta_1 \to \theta_0$. On the other hand, if $r < 1$, we accept it with probability r and set $\theta_1 \to \theta_0$. If rejected, keep $\theta_0 = \theta_0$. This is an elementary demonstration of the Hamiltonian Monte Carlo (very special with no auxiliary momentum and kinetic energy).

Use the Metropolis–Hastings algorithm above to estimate the minimum of the noisy $\phi 4$ with initial guesses $\theta_0 = -1, 0.5, 3$. You should try different β.

C)

Simulated Annealing is a probabilistic optimization algorithm inspired by the annealing process in metal. The cooling schedule is usually defined as the following:

$$\beta_{i+1} = \beta_i + \delta_i,$$

where we update the inverse temperature at each step. This update will change the Metropolis criterion:

$$r_i = e^{-\beta_i \Delta H(\theta^*, \theta)} > u_i,$$

where $u_i \sim U(0, 1)$. The cooling schedule is a critical aspect of Simulated Annealing. It determines the rate at which the temperature decreases.

Add a cooling schedule to the Metropolis–Hastings algorithm above. Then, estimate the minimum of the noisy $\phi 4$ with initial guesses $\theta_0 = -1, 0.5, 3$. You should try different cooling schedule e.g. δ_i. Make a graphical comparison of the convergence steps with cooling and without cooling.