FIRST MIDTERM

Scientific Computing II
Fundación Universitaria Konrad Lorenz
March 16, 2024

- This is an **individual exam**. Therefore, do not provide or receive help from anyone for the completion of this midterm.
- The exam will be available from March 16th, 2023, 8.15am. From that time on you will have 2 hours and 15 minutes to complete the midterm exam.
- The solution must be turned in only through the Aula virtual. Midterms sent through email will not be graded.
- Full score will only be given to correct and completely justified answers. Miraculous, obtuse and unnecessarily complex solutions will receive partial or null score.
- You can ask any question of this assignment during class or through email: julian.jimenezc@konradlorenz.edu.co.

Finite Differences Method

The Taylor series expansion of a univariate function f, centered at x_0, is given by

$$f(x_0+h)=\sum_{n=0}^{\infty}\frac{f^{(n)}(x_0)}{n!}h^n=f(x_0)+\frac{f'(x_0)}{1!}h+\frac{f''(x_0)}{2!}h^2+\cdots, \tag{1}$$

so that, at first-order with respect to h, it follows that

$$f(x_0+h)=f(x_0)+f'(x_0)h+O(h^2),$$

from where we get that, if h is sufficiently small, then

$$f'(x_0)\approx\frac{f(x_0+h)-f(x_0)}{h}. \tag{2}$$

If we do a similar procedure for the first-order derivative of f, we get that

$$f''(x_0)\approx\frac{f(x_0+h)-2f(x_0)+f(x_0-h)}{h^2}. \tag{3}$$

Wave Equation

This approach can be extended to solve partial differential equations. For example, consider the unidimensional wave equation, describing the elongation $U(x,t)$ of a string at a point x and time t, of a rod of length L, whose ends are fixed.

$$\frac{\partial^2 U}{\partial t^2}=c^2\frac{\partial^2 U}{\partial x^2}. \tag{4}$$

In equation (4), c is the propagation velocity of waves in the string. From the problem's statement, the elongation U is subject to the following initial and boundary conditions:

$$U(0,t) = U(L,t) = 0 \text{ (boundary conditions)}, \tag{5}$$

$$U(x,0) = 0, \quad U_t(x,0) = \sin x \text{ (initial conditions)}. \tag{6}$$

Our objective will be to numerically solve (4) subject to (5)-(6). To do this, we discretize both time and space by a finite number of mesh points evenly distributed:

$$0 = t_0 < t_1 < \cdots < t_{N_t} = T, \quad 0 = x_0 < x_1 < \cdots < x_{N_x} = L. \tag{7}$$

Under this discretization, we define the double sequence u_j^n to be

$$u_j^n = U(x_j, t_n), \tag{8}$$

where x_j and t_n are taken from the discretization (7). Therefore, (4) can be approximated using (3) to become

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\Delta t^2} = c^2 \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}, \tag{9}$$

with Δx and Δt being the distance between elements of the discretization (7). Finally, the initial condition for U_t can be approximated to be

$$U_t(x_j, 0) \approx \frac{u_j^1 - u_j^{-1}}{2\Delta t}. \tag{10}$$

u_j^{-1} is the position at time $-\Delta t$ and position x_j. **Do not confuse it with a power.**

1. (0/30) Use equations (9)-(10) to numerically integrate the PDE (4) subject to conditions (5)-(6). Graph the solution as a 3D surface or as a colormap (or as an animation, if you are fancy), where the dynamics of the string as a function of time can be appreciated. You may take $c = 1$ and $L = 1$. Remember to:

- Modularize and comment your code.
- Optimize your code looking for possible bottlenecks and using the optimization strategies seen in class.
- Profile your code if required.
- Some variables were not explicitly defined, like $\Delta t, \Delta x, T$. Choose a reasonable selection for them to get a smooth numerical integration. *Hint:* $C = c\Delta t/\Delta x$ **must** be smaller than 1.

Random Walk

This project aims to deepen your understanding of probability distributions, particularly the emergence of the normal distribution from seemingly simple stochastic movements.

Your task is to implement a program in Python to simulate a random walk in one dimension. A random walk is a mathematical model where an object moves randomly step by step. In this case, we will consider a one-dimensional scenario where the object moves either left or right with equal probability.

2. (0/20) Take $N = 10^6$ objects of this kind, starting all at $x = 0$, and evolve them $T = 1000$ iterations. Calculate and plot the probability distribution of the final positions of the objects after a large number of simulations. You should observe the emergence of the normal distribution. Remember to:

- Modularize and comment your code.
- Optimize your code looking for possible bottlenecks and using the optimization strategies seen in class.
- Profile your code if required.