PHY407 Formal Lab Report 4

Natalia Tabja

November 2024

1 Question 1: Electrostatics and Laplace's Equation

Part (a): Potential Without Over-Relaxation

The system was modeled using a 100×100 grid with boundary conditions as described:

- Walls of the box at $0\,\mathrm{V}$.
- Two flat plates set to $+1\,\mathrm{V}$ and $-1\,\mathrm{V}$.
- Convergence criterion: Precision of $10^{-6}\,\mathrm{V}$ at each grid point.

The Gauss-Seidel method was implemented, and the resulting potential is shown in Figure 1.

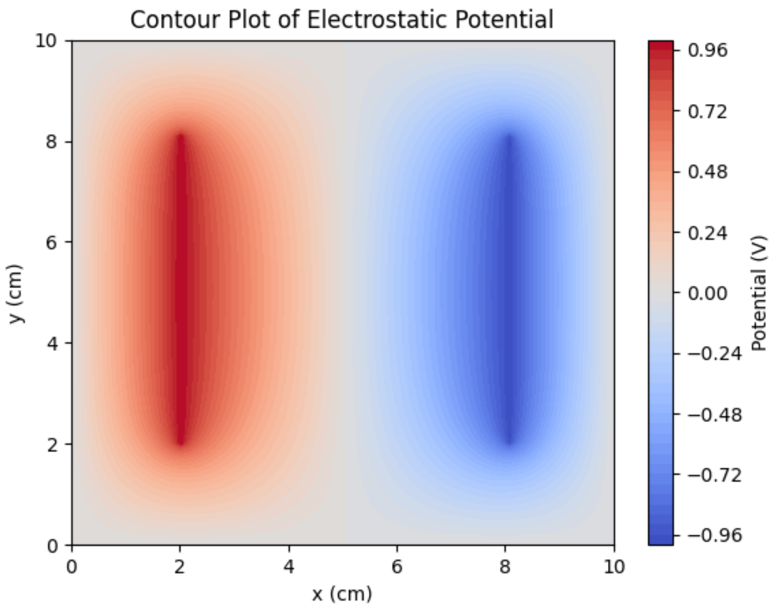

Figure 1: Contour plot of electrostatic potential without over-relaxation.

Part (b): Potential With Over-Relaxation

The Gauss-Seidel method was repeated with over-relaxation, using $\omega = 0.1$ and $\omega = 0.5$. The potential plots for both cases are shown in Figure 2. The system converged in:

- 1931 iterations for $\omega = 0.1$.
- 1112 iterations for $\omega = 0.5$.

The visual results are nearly identical for both cases. However, as expected, increasing ω significantly reduced the number of iterations required for convergence. This demonstrates that over-relaxation accelerates convergence by improving the iterative updates.

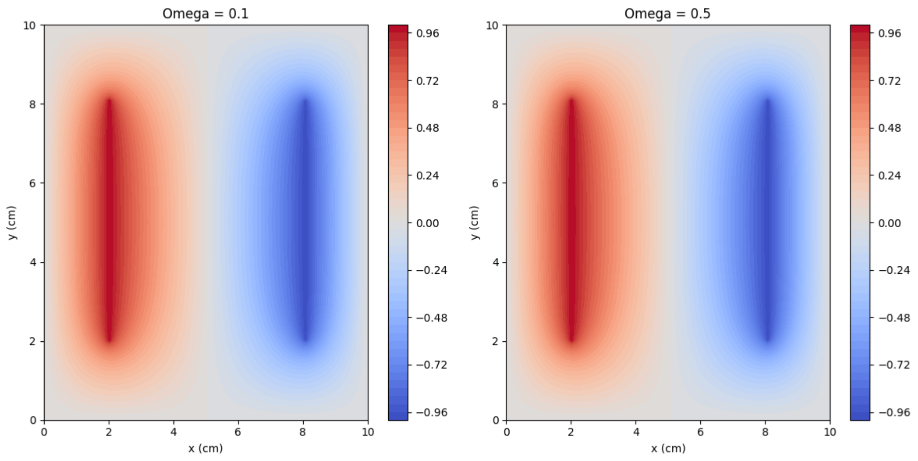

Figure 2: Contour plots of potential for $\omega = 0.1$ (left) and $\omega = 0.5$ (right).

Part (c): Electric Field Lines

The electric field was computed as $\vec{E} = -\nabla V$. Figure 3 shows a stream plot of the electric field lines colour-coded by the electric potential. The field lines originate from the positive plate and terminate at the negative plate, remaining relatively straight between the plates and fringing towards the edges, as expected. Moreover, the density of the field lines is highest near the plates, indicating that the field is strongest in those regions, which is consistent with what one would expect due to the concentration of charge on either plate. Overall, the magnitude and direction of the electric field are consistent with the boundary conditions of the system.

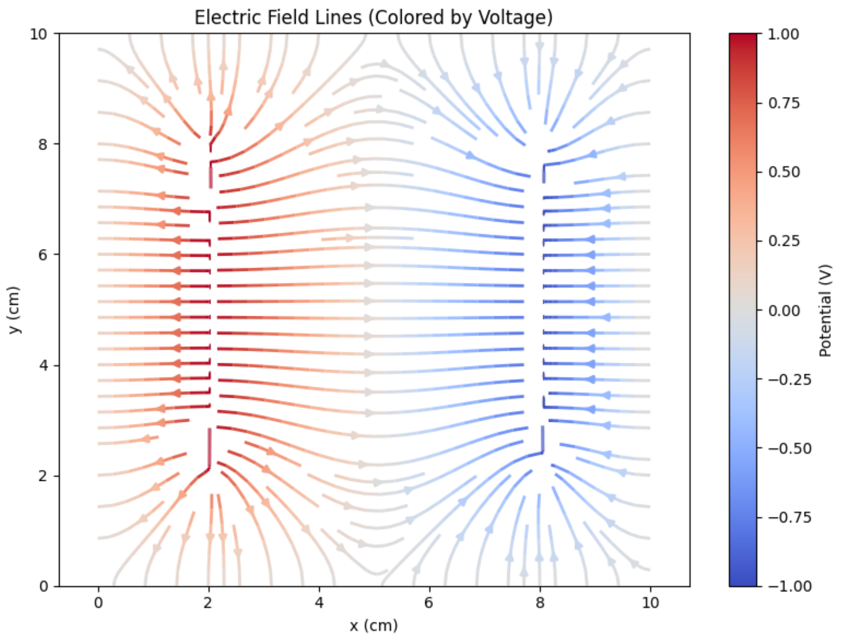

Figure 3: Electric field lines with potential color-coding.

2 Question 2: Shallow Water System

2.1 Part (a): Derivation of the Flux-Conservative Form and Discretization

We begin with the 1D shallow water equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \eta}{\partial x}, \tag{1}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}(uh) = 0, \tag{2}$$

where $h = \eta - \eta_b$ is the height of the water column above the bottom surface.

1. The term $u\frac{\partial u}{\partial x}$ can be rewritten as $\frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right)$.
2. Substituting this into the first equation gives:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = -g\frac{\partial \eta}{\partial x}.$$

Rearranging:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2 + g\eta\right) = 0.$$

3. The second equation becomes:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\left[u(\eta - \eta_b)\right] = 0.$$

4. Combining these results, we define:

$$\tilde{u} = \begin{bmatrix} u \\ \eta \end{bmatrix}, \quad \mathbf{F}(\tilde{u}) = \begin{bmatrix} \frac{1}{2}u^2 + g\eta \\ u(\eta - \eta_b) \end{bmatrix}.$$

The equations can now be written in flux-conservative form:

$$\frac{\partial \tilde{u}}{\partial t} + \frac{\partial \mathbf{F}(\tilde{u})}{\partial x} = 0,$$

or explicitly:

$$\frac{\partial}{\partial t}\begin{bmatrix} u \\ \eta \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \frac{1}{2}u^2 + g\eta \\ u(\eta - \eta_b) \end{bmatrix} = 0.$$

Discretization

To solve numerically, we can discretize the system using the Forward-Time Centered-Space (FTCS) scheme. The discrete equations for u and η are:

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}\left(F_{0,j+1}^n - F_{0,j-1}^n\right), \tag{3}$$

$$\eta_j^{n+1} = \eta_j^n - \frac{\Delta t}{2\Delta x}\left(F_{1,j+1}^n - F_{1,j-1}^n\right), \tag{4}$$

where:

$$F_{0,j}^n = \frac{1}{2}(u_j^n)^2 + g\eta_j^n, \quad F_{1,j}^n = (\eta_j^n - \eta_b)u_j^n.$$

The boundary conditions are the following:

$$u(0,t) = u(L,t) = 0, \quad \eta(0,t) = \eta(L,t).$$

2.2 Part (b): Implementation and Results

The system was implemented in Python using the parameters:

- Domain: $x = [0, 1]$, with $L = 1$.
- Grid points: $J = 50$, $\Delta x = 0.02$.
- Time step: $\Delta t = 0.01$.
- Gravity: $g = 9.81\,\text{m/s}^2$.
- Initial conditions:

$$u(x,0) = 0, \quad \eta(x,0) = H + Ae^{-(x-\mu)^2/\sigma^2} - \langle Ae^{-(x-\mu)^2/\sigma^2} \rangle.$$

The following plot shows $\eta(x,t)$ at $t = 0\,\text{s}, t = 1\,\text{s}, \text{and}\, t = 4\,\text{s}$:

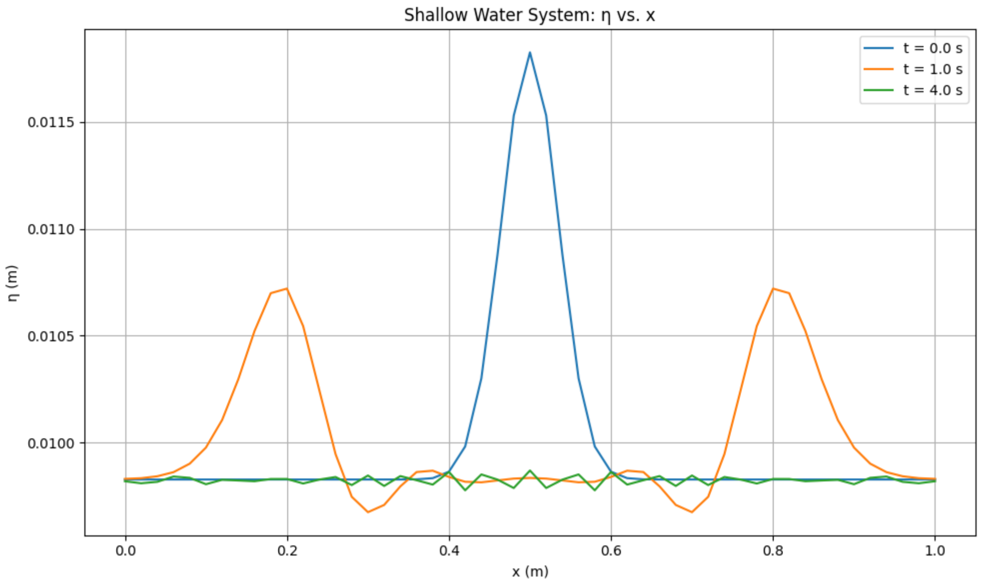

Figure 4: Plot of $\eta(x,t)$ at different time steps.

The plot of $\eta(x,t)$ above (Figure 4) demonstrates the propagation of waves in the shallow water system. Over time, the wave disperses, and the solution remains consistent with the boundary conditions. The Gaussian perturbation at $t = 0$ evolves and spreads out, as expected in this system.

A potential reason for why instability was not observed, despite the expectation noted in the question, could be the relatively small time step ($\Delta t = 0.01\,\text{s}$) and spatial step ($\Delta x = 0.02\,\text{m}$), which may have been enough to satisfy the numerical stability conditions of the FTCS scheme for this specific problem setup. Additionally, the boundary conditions and the smooth initial condition of $\eta(x,t)$ might have contributed to minimizing numerical errors that usually amplify error over time.

3 Question 3: Waves with Burger's Equation

3.1 Part (a)

To solve the wave equation using the leapfrog-based method, an iterative numerical approach was implemented in Python. The leapfrog method updates the solution u at each spatial and temporal point based on the discretized form of the wave equation (Equation (13) from the assignment handout).

First, the spatial grid is initialized, along with the time step and parameters such as ϵ, Δx, Δt, L_x, and T_f to allow flexibility in testing different values for these parameters. The initial condition $u(x, t = 0)$ is defined as a sine function ($\sin(x)$), and the boundary conditions ($u(0, t) = 0$ and $u(L_x, t) = 0$) are enforced at every iteration.

The leapfrog scheme begins with a forward Euler step to calculate the values at the first time step, providing the necessary "starter" values for subsequent iterations. For later time steps, the leapfrog update formula computes u_i^{j+1} at each spatial point i using the values of u from the current and previous time steps. This approach ensures second-order accuracy in both space and time.

To visualize the wave propagation, snapshots of $u(x, t)$ at specified times (e.g., $t = 0.5\,\mathrm{s}, 1.0\,\mathrm{s}, 1.5\,\mathrm{s}$) are recorded and plotted (as seen in part Figure 5). The loop iterates until the simulation reaches the specified final time T_f. The stability of the system is determined by the parameter $\beta = \frac{\epsilon \Delta t}{\Delta x}$, which must remain below a critical threshold. If β becomes too large, divergence may be observed.

3.2 Part (b)

The simulation was run using the provided parameter values: $\epsilon = 1$, $\Delta x = 0.02\,m$, $\Delta t = 0.005\,s$, $L_x = 2\pi\,m$, and $T_f = 2\,s$. The initial condition $u(x, t = 0) = \sin(x)$ and boundary conditions $u(0, t) = 0$ and $u(L_x, t) = 0$ were applied. The following plots show $u(x, t)$ at $t = 0, 0.5, 1.0, 1.5\,s$:

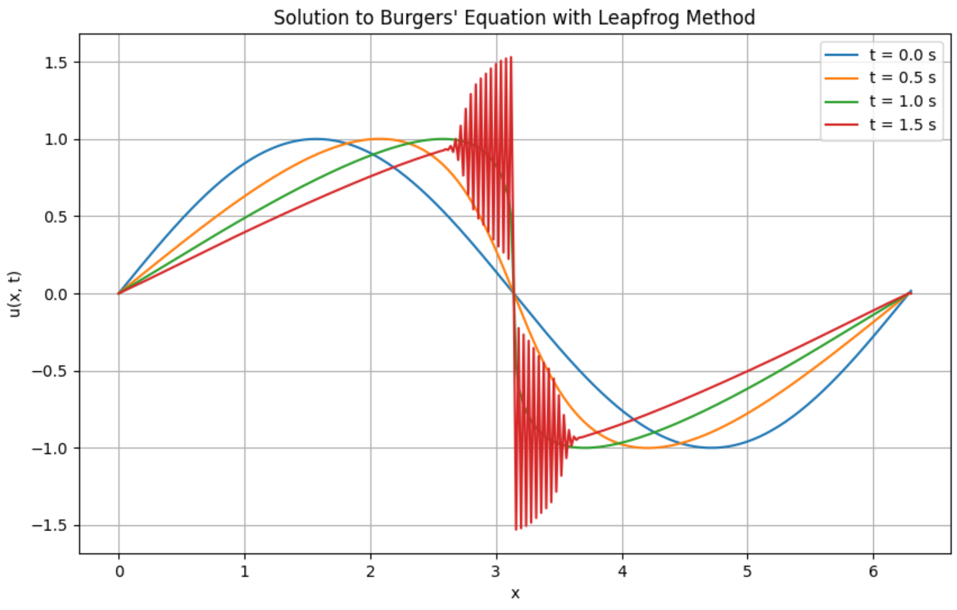

Figure 5: Solution to Burger's Equation at different time steps.

3.3 Analysis of Results

Figure 5 illustrates the evolution of the wave disturbance over time. Initially, the wave starts as a perfectly sinusoidal wave. However, over time, the nonlinear term causes the wave to steepen and form sharper gradients. The snapshots for times $t = 0.0\,s$, $t = 0.5\,s$, and $t = 1.0\,s$, show equal amplitudes and relatively similar sinusoidal shapes, but beyond that point, the behaviour of the system deteriorates. By $t = 1.5\,s$, high-frequency oscillations appear near the $x = 3.0$ mark due to the leapfrog scheme's dispersive nature and the lack of sufficient dissipation. This behaviour aligns with expectations for this problem and reveals the method's limitations in handling steep gradients without more stabilization techniques. The solution, while realistic for small time values, diverges from expected behaviour as time increases due to the numerical instability that is inherent in the leapfrog method without viscosity or stabilization.