HAP708P "Modelization and simulation in physics", University of Montpellier, 2022

# Exercise sheet 4

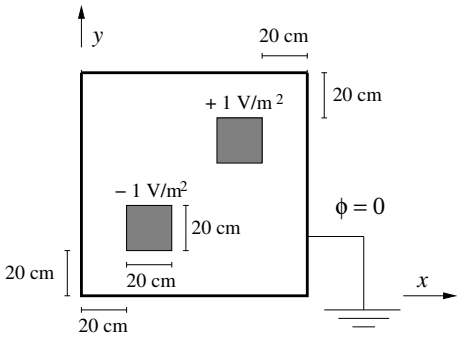
## Exercise 1: Successive overrelaxation

1. Write a program to solve the electrostatics problem of the lecture, using successive overrelaxation. Experiment with several values of the overrelaxation parameter $\omega$ and study the effect on the speed of convergence. Plot the solution.

2. Write a program to solve the two-dimensional Poisson equation

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\phi(x,y) = -\frac{\rho(x,y)}{\epsilon_0}$$

where $\rho$ is a charge density and $\epsilon_0$ is the permittivity. The solution region is a 1 m $\times$ 1 m square whose boundaries are at ground potential $V = 0$. In the interior, $\rho$ is zero everywhere except for two homogeneously charged squares of 20 cm $\times$ 20 cm, where $\rho/\epsilon_0 = \pm 1$ V m$^{-2}$. Use successive overrelaxation.




## Exercise 2: Minimal surface

In Cartesian coordinates, a *minimal surface* $(x, y, z(x, y))$ satisfies the non-linear PDE

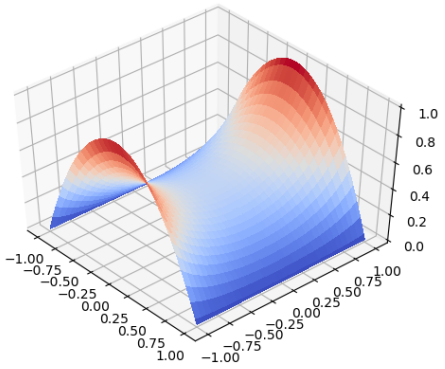
$$\left(1 + (\partial_x z)^2\right)\partial_y^2 z + \left(1 + (\partial_y z)^2\right)\partial_x^2 z - 2\,(\partial_x z)(\partial_y z)(\partial_x \partial_y z) = 0$$

where we have abbreviated $\partial_x = \frac{\partial}{\partial x}$, $\partial_x^2 = \frac{\partial^2}{\partial x^2}$ etc.

1. By a pen-and-paper calculation, discretize the minimal-surface equation on a regular $(N+1)\times(N+1)$ lattice in the $(x, y)$ plane: $z(x, y) \to z_{ij}$. Thus, find a finite-distance expression for $z_{ij}$ as a function of the neighbouring values $z_{i\pm1,j\pm1}$.

2. A tent cloth is suspended between two parabolic arcs at $x = -1$ and at $x = 1$, and fixed to the ground at $y = -1$ and at $y = 1$:

$$z(-1, y) = 1 - y^2\,, \quad z(1, y) = 1 - y^2\,, \quad z(x, -1) = 0\,, \quad z(x, 1) = 0\,.$$




Find the shape of the tent numerically (assuming the tent cloth to be perfectly deformable, so that it forms a minimal surface). Plot $z$ as a function of $y$ for $x = 0$, $x = 0.4$, $x = 0.8$.

*Instructions:* Use a grid of $100 \times 100$ squares in the $(x, y)$ plane, i.e. $101 \times 101$ points. Use successive overrelaxation or (better) the vectorized Jacobi method to speed up convergence.

### Exercise 3: Temperature profile of the Earth's crust

The surface temperature of the Earth is susceptible to seasonal variation; however, at a depth of 20 m, the temperature is approximately constant (and higher than the average surface temperature because of heating from the Earth's radioactive interior). We take $D = 0.1\,\mathrm{m}^2/\mathrm{day}$ as the thermal diffusion coefficient, and set $T_0(t) = A + B\sin\frac{2\pi t}{\tau}$ for the daily mean temperature at the surface with $A = 10°\mathrm{C}$, $B = 12°\mathrm{C}$, and $\tau = 365$ days, as well as $T_1 = 11°\mathrm{C}$ at a depth of 20 m.

Write a program calculating the temperature profile between 0 et 20 m depth, by solving the heat equation with the FTCS method. Run it over a sufficiently long time interval (a few years) to eliminate all dependence on the initial conditions, until a stable seasonal cycle is established. Then plot the solutions at 4 dates with a 3-month separation between them, in order to see how the temperature depends on the depth in the different seasons.

### Exercise 4: Schrödinger equation

In this exercise you will numerically solve the time-dependent Schrödinger equation for a particle in a one-dimensional box of length $L$. The potential is zero for $0 < x < L$ and otherwise infinite, and the Schrödinger equation reads

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x,t) \qquad (0 < x < L)\,.$$

1. Write down the Crank-Nicolson equation for this system (i.e. the equation for $\psi_n(t+h)$ given by the mean of the explicit and implicit Euler schemes).

2. Putting all the $\psi_n(t)$ into a vector $\boldsymbol{\psi}(t)$ such that the C-N equations become

$$\mathbf{A}\cdot\boldsymbol{\psi}(t+h) = \mathbf{B}\cdot\boldsymbol{\psi}(t)\,,$$

find explicit expressions for the matrices $\mathbf{A}$ and $\mathbf{B}$.

3. Write a program to compute the time evolution of a wave function in this potential. Use $m = 9.109 \cdot 10^{-31}$ kg (the electron mass), $\hbar = 1.0545718 \cdot 10^{-34}$ m$^2$ kg s$^{-1}$ and $L = 10^{-8}$ m. At $t = 0$, the wave function profile in the interior of the box is Gaussian:

$$\psi(x, 0) = A \exp\left(-\frac{(x - L/2)^2}{2\sigma^2}\right) \exp(ikx) \ .$$

Here $\sigma = 10^{-10}$ m, $k = 2 \cdot 10^{10}$ m$^{-1}$ and $A$ is an unimportant normalization (you can take $A = 1$). The boundary conditions are $\psi(0, t) = \psi(L, t) = 0$. Plot the real part and the absolute value of the solution computed with $N = 400$ intermediate points after $t = (1, 5, 10, 20) \cdot 10^{-16}$ s have evolved.
*Hints:* To set up a `numpy.ndarray` with complex entries, use the `dtype` argument, e.g. `psi = numpy.zeros([N+1], dtype=complex)`. The real part is then accessed with `psi.real` and the absolute value with `numpy.abs(psi)`. To solve the C-N equations, use the pre-defined function `numpy.linalg.solve()`, even if it is inefficient on sparse matrices (see below).

4. *Bonus*: If you are familiar with Gaussian elimination, optimize your code such that the solution of the C-N equation is computed in $T(N) \in \Theta(N)$ time. This can be done thanks to the sparse structure of the $\mathbf{A}$ and $\mathbf{B}$ matrices; note that the only nonzero elements are on the main diagonal or next to it (for comparison, for a generic $N \times N$ matrix, Gaussian elimination takes $\Theta(N^3)$ time!)