

Self-Consistent Simulations of Beam and Plasma Systems

Homework 2

S. M. Lund, J.-L. Vay, R. Lehe, and D. Winklehner

Tuesday, June 14th, 2016

Problem 1 - Discrete numerical operations

a) Derive the uniform mesh formulas and error estimates covered in class (3 point centered):

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

$$\left. \frac{\partial^2 f}{\partial x^2} \right|_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{(\Delta x)^2} + \mathcal{O}(\Delta x^2)$$

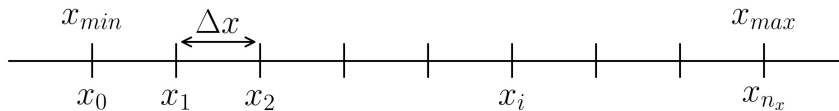
b) Derive the quadrature formulas and error estimates covered in class:

Trapezoidal Rule: $\int_{x_{i-1}}^{x_{i+1}} dx f(x) = \frac{f_{i-1} + 2f_i + f_{i+1}}{2} \Delta x + \mathcal{O}(\Delta x^3)$

Simpson's Rule: $\int_{x_{i-1}}^{x_{i+1}} dx f(x) = \frac{f_{i-1} + 4f_i + f_{i+1}}{3} \Delta x + \mathcal{O}(\Delta x^5)$

Hint: start with a forward approximation for $f(x)$ starting at f_{i-1} and integrate over the half-interval $x_{i-1} - x_i$. The other half interval is analogous.

c) Show that the formulas in b) are consistent with:

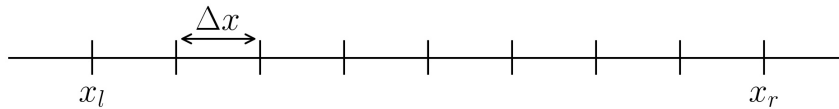


Trapezoidal Rule: $\int_{x_{min}}^{x_{max}} dx f(x) = \left\{ \frac{1}{2}f_0 + \sum_{i=1}^{n_x-1} f_i + \frac{1}{2}f_{n_x} \right\} \Delta x$

Simpson's Rule (n_x even): $\int_{x_{min}}^{x_{max}} dx f(x) = \left\{ \frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{2}{3}f_2 + \frac{4}{3}f_3 + \frac{2}{3}f_4 + \dots + \frac{2}{3}f_{n_x-2} + \frac{4}{3}f_{n_x-1} + \frac{1}{3}f_{n_x} \right\} \Delta x$

Problem 2 - Numerical field solutions 1

In 1D electrostatics, on a uniform grid



the Poisson equation is

$$\frac{\partial^2}{\partial x^2} \phi = \frac{-\rho}{\epsilon_0},$$

where

$$\begin{aligned} \phi &= \phi(x) \\ \rho &= \rho(x) \end{aligned}$$

and

$$\begin{aligned} x_j &= x_l + \Delta x * j, & \Delta x &= \frac{x_r - x_l}{n_x} \\ j &= 0, 1, 2, \dots, n_x, & n_x + 1 &= \text{number of grid points.} \end{aligned}$$

In class we defined a discrete Fourier transform for the 1D potential:

$$\tilde{\phi}_n = \Delta x \sum_{j=0}^{n_x} e^{\frac{i2\pi nj}{n_x+1}} \phi_j, \quad n = 0, 1, 2, \dots, n_x$$

a) Show that the transform can be inverted exactly with

$$\phi_j = \frac{1}{(n_x + 1)\Delta x} \sum_{n=0}^{n_x} e^{\frac{-i2\pi nj}{n_x+1}} \tilde{\phi}_n$$

b) Define a gridded electric field

$$E_{xj} = \frac{-(\phi_{j+1} - \phi_{j-1})}{2\Delta x}$$

and show that

$$\tilde{E}_{xn} = i\hat{k}_n \tilde{\phi}_n$$

where

$$\hat{k}_n = k_n \left[\frac{\sin(k_n \Delta x)}{k_n \Delta x} \right]$$

and

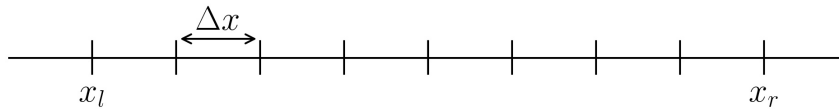
$$k_n = \frac{2\pi n}{(n_x + 1)\Delta x}.$$

Here

$$\tilde{E}_{xn} = \Delta x \sum_{j=0}^{n_x} e^{\frac{i2\pi nj}{n_x+1}} E_{xj}.$$

Problem 3 - Numerical field solutions 2

In 1D electrostatics, on a uniform grid



the Poisson equation is

$$\frac{\partial^2}{\partial x^2} \phi = \frac{-\rho}{\epsilon_0},$$

where

$$\begin{aligned} \phi &= \phi(x) \\ \rho &= \rho(x) \end{aligned}$$

and

$$\begin{aligned} x_j &= x_l + \Delta x * j, & \Delta x &= \frac{x_r - x_l}{n_x} \\ j &= 0, 1, 2, \dots, n_x, & n_x + 1 &= \text{number of grid points.} \end{aligned}$$

- a) In class we setup an explicit matrix discretization of this problem for Dirichlet boundary conditions with $\phi(x_l) = V_l$ and $\phi(x_r) = V_r$ where source terms were isolated on the right-hand side of the resulting matrix equation. Repeat this construction using

$$\left. \frac{\partial^2 f(x)}{\partial x^2} \right|_j = \frac{f_{j+1} - 2f_j + f_{j-1}}{\delta x^2} + \mathcal{O}(\Delta x^2)$$

for Neumann boundary conditions with

$$\left. \frac{\partial \phi}{\partial x} \right|_{x_l} = G_l.$$

- b) Could we also specify

$$\left. \frac{\partial \phi}{\partial x} \right|_{x_r} = G_r$$

and have a solution? Why?

Problem 4 - Python script for 1D EM solver in vacuum

In the 1D case and in vacuum ($\vec{j} = \vec{0}$), the discretized Maxwell equations for E_x and B_y are:

$$\frac{B_{y_{k+1/2}}^{n+1/2} - B_{y_{k+1/2}}^{n-1/2}}{\Delta t} = - \left(\frac{E_{x_{k+1}}^n - E_{x_k}^n}{\Delta z} \right)$$
$$\frac{E_{x_k}^{n+1} - E_{x_k}^n}{\Delta t} = -c^2 \left(\frac{B_{y_{k+1/2}}^{n+1/2} - B_{y_{k-1/2}}^{n+1/2}}{\Delta z} \right)$$

Download the file `em_pic_1d.py` from:

http://raw.githubusercontent.com/RemiLehe/uspas_exercise/master/em_pic_1d.py

It is an incomplete implementation of a field solver, based on these equations (with periodic boundaries)

- a) Read the code and try to understand it (you are encouraged to run the code to see what it produces). Answer the following questions:
 - What does `np.zeros` do? What type of object are `self.Ex` and `self.By`?
 - In the method `plot_fields`, explain why we use:
 - `self.z` in the line `plt.plot(self.z, self.Ex[1:-1], 'o-')`
 - but `self.z+0.5*self.dz` in the line `plt.plot(self.z+0.5*self.dz, self.By[1:-1], 'o-')`.
 - With the current version of the code, does the initial pulse propagate with time? Why?
- b) Rewrite the above Maxwell equations in the form $B_{y_{k+1/2}}^{n+1/2} = \dots$ and $E_{x_k}^{n+1} = \dots$. These are the update equations that need to be implemented in `em_pic_1d.py`.

In the code, find the lines that are tagged with the text `ASSIGNEMENT` and write the correct update equations, using Python syntax. Note that, as explained in the text of the code, at a given iteration n , the element `Ex[k]` represents $E_{x_k}^n$ and the element `By[k]` represents $B_{y_{k+1/2}}^{n-1/2}$ (thus be mindful of using the correct indices for the array `Ex` and `By`).
- c) In order to validate your implementation, check that the pulse propagates to the right. At what speed should it propagate physically? From the images, can you evaluate at which speed it propagated?
- d) (Extra Credit) Rewrite the `for` loops by using `numpy` vector syntax.

Problem 5 - Warp: parallel plates and Child-Langmuir law

Let us assume two parallel conducting plates separated by the distance d , and with a constant potential difference V between the two plates.

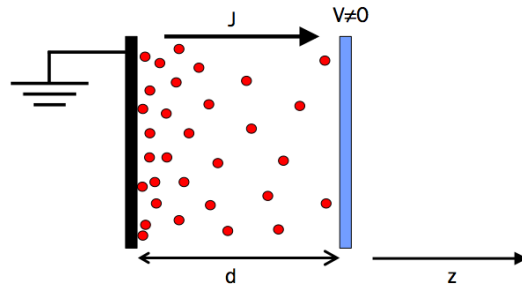


Figure 1: Two parallel conducting plates separated by the distance d and powered by a voltage difference V will, for a proper value of V , accelerate charged particle initially at rest at one of the plates.

Charged particles injected with initial velocity $v = 0$ will be accelerated toward the opposite plate when the sign of the voltage difference is set appropriately.

- a) Run a parallel plate example in warp:
 - Open the input script “Parallel_plate_injection.py”.
 - Run the script: “python -i Parallel_plates_injection.py”.
 - Observe the injection of particles in window 0, and of the current, charge density, electric field and potential in window 1. You may run additional steps by using the command “step(n)” where n is the number of time steps to run.
 - In the same terminal, after exiting the run, or in another terminal window, open the output file created by window(0) with the command “gist Parallel_plates_injection.000.cgm”. At the gist prompt, type ‘help’ for the help menu. Select the graphic window and type “f” multiple times, followed by “g” multiple times. Refer to the help menu for additional options. You may also explore the graphics produced in window(1) by opening the file with “gist current.cgm”.
 - Look at the input script line-by-line until the end, trying to understand the meaning of each command.
 - Search for the line ”diode_current = 1.”. Augment the diode_current up to 6A, by increment of 1, rerunning the script each time. What do you observe?
- b) Let us examine why the emitted current does not seem to exceed a certain limit in steady-state mode.

Assuming a steady state flow, the current will be constant ($\mathbf{J} = \rho\mathbf{v}$) and the energy of a particle at a given position is given by $1/2mv^2 = -q\phi(z)$ where q , m , v and z are respectively the charge, mass, velocity and position of the particle. Using Poisson’s equation ($d^2\phi/dz^2 = -\rho/\epsilon_0$), and posing $\Phi = -q\phi$ show that

$$\frac{d^2\Phi}{dz^2} = \frac{q\mathbf{J}\sqrt{m/2}}{\epsilon_0}\Phi^{-1/2}. \quad (1)$$

Multiply each side by $d\Phi/dz$ and integrate to find

$$\left(\frac{d\Phi}{dz}\right)^2 = \frac{4q\mathbf{J}\sqrt{m/2}}{\epsilon_0}\Phi^{1/2} + C. \quad (2)$$

Considering now the special case $d\Phi/dz = 0$ at $z = 0$, we get $C = 0$, integrate again to find

$$\frac{4}{3}\Phi^{3/4} = 2\sqrt{\frac{q\mathbf{J}}{\epsilon_0}}\left(\frac{m}{2}\right)^{1/4}z. \quad (3)$$

Using $\phi(d) = V$, find that

$$\mathbf{J} = \frac{4}{9}\epsilon_0\sqrt{\frac{2}{m}}\frac{(-qV)^{3/2}}{qd^2}, \quad (4)$$

and

$$\phi(z) = V\left(\frac{z}{d}\right)^{4/3}. \quad (5)$$

The expression

$$\mathbf{J} = \frac{4}{9}\epsilon_0\sqrt{\frac{2|q|}{m}}\frac{|V|^{3/2}}{d^2}, \quad (6)$$

is known as the *Child's Law* or *Child-Langmuir Law* and gives the maximum current that can be extracted for a given voltage and plate separation.

- c) Compare the theoretical Child-Langmuir limit with the steady state values in warp.
- Find the command “plzprofiles(LCL=False)” and replace by “plzprofiles(LCL=True)”.
 - The input script will now fail at execution. Fix the input script by replacing the expressions for Child-Langmuir Law and the potential dependency at the appropriate locations.
 - At the end of the simulation, the simulation printed ”Maximum current from Child-Langmuir law = ***”. Replace the input value for the current by this value, restart the simulation and verify that the emitted current, charge density, electric field and potential profiles converge when steady-state is reached, toward the ones predicted at the Child-Langmuir limit.