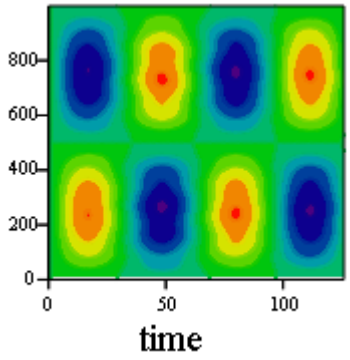
Basic Particle-In-Cell Code for Cold Plasma Oscillations

This particle-in-cell (PIC) code solves for the motion of plasma electronss in an electric field that is determined using the particle densities. The equation of motion for each particle is solved by iteration, and at each time step the electric field is found from a solution of Poisson's equation. In this exercise, we will write a "bare bones" PIC code with a minimum amount of effort. We will follow the motion of electrons and assume that the ions are a uniform neutralizing background. We will assume that the electrons are cold (no thermal velocity) so that they do not move outside the simulation domain.



For an electrons, the equation of motion is: $$\frac{d^2x}{dt^2} = \frac{-qE}{m}$$

The electric field is found from the particle density: $$\frac{dE}{dx} = \frac{q}{\varepsilon_0}\left(n_i - n_e\right)$$

We will assume that the ion density n_i is a constant n_0. The mass and charge of the electron are m and -q, respectively.

Dimensionless units

We will rewrite the equations in dimensionless form by dividing x by a characteristic distance and dividing t by a characteristic time.

For the characteristic time, we use the inverse of the plasma frequency: $$\hat{t} = \sqrt{\frac{\varepsilon_0 m}{n_0 q^2}}$$

Since our plasma is cold, there is no Debye length to use for a characteristic distance. Instead, we will assume plasma oscillations with amplitude Φ_0. The potential energy $q\Phi_0$ will replace the thermal energy kT in the Debye length formula that we will use to define a basic length scale: $$\hat{x} = \sqrt{\frac{\varepsilon_0 \phi_0}{n_0 q}}$$

We divide t by the characteristic time and x by the characteristic distance and obtain the dimensionless forms of our equations:

$$\frac{d^2\bar{x}}{d\bar{t}^2} = -\bar{E} \quad \text{and} \quad \frac{d\bar{E}}{d\bar{x}} = \frac{\left(n_i - n_e\right)}{n_0} = \bar{n}_i - \bar{n}_e \quad \text{with} \quad \bar{x} = x/\hat{x} \quad \text{etc.}$$

Note that the dimensionless density appears in the final math and is obtained by dividing n_e by n_0. We assume that the ion density is fixed and equal to n_0, then the net charge density is 1 - n_e/n_0.

Equations in finite-difference form

The finite difference form of the equation for E is: $\overline{E}_{k+1} = \overline{E}_k + \Delta\overline{x}\left(1-\overline{n}_e\right)$

Below, we will omit the bar over the dimensionless quantities for convenience.

E_k is the value of E at grid point k. For second order accuracy, the number density n_e should be evaluated half way between the grid points. We will take a simpler approach and evaluate the density at the grid points, and find the density half way between points by averaging the values at the points on either side. We could count the number of electrons in each cell to find the number density. However, this density would make discontinuous jumps as an electron moved from one cell to another. A less noisy method is to use a linear weighting function. If the electron is 10% of the distance from a first point to a second, then 10% of the charge is assgned to the second point and 90% to the first.

We will use x for the grid point locations and s for the electron locations. We will use the subscript i to indicate the values at successive times. Our program will output values of $M_{k,i}$ which is a matrix of values of the locations for the k^{th} particle at the i^{th} time step.

The finite difference form of the equation of motion is: $s_{i+1} - 2s_i + s_{i-1} = -\left(\Delta t\right)^2 E$
where the left side is d^2s/dt^2.

This equation can be rearranged to give: $s_{i+1} = 2s_i - s_{i-1} - \left(\Delta t\right)^2 E$

The positions s of particles at successive times will be placed in the columns of a matrix M and the calculations will be done using the columns of the matrix. We will define a column matrix $E\Delta t\Delta t$ that has the electric field at particle locations scaled by Δt^2.

If there is no electric field, the particle drifts at constant speed and the new location s_{i+1} is obtained by adding the distance moved in a time step, $s_i - s_{i-1}$, to the location s_i at the last time step to obtain $2s_i-s_{i-1}$.

Note that we have omitted finding the velocity in order to save computation time. The velocity can be found by dividing the distance moved between time steps by the size of the time step:

$$v_i = \frac{s_{i+1} - s_i}{\Delta t}$$

Grid points

We will define a grid with jmax points separated by distance Δx. jmax := 25

$\Delta x := 1$ $j := 0 .. jmax$ $x_j := \Delta x \cdot j$ $xmax := x_{jmax}$ xmax = 25

For this exercise, we have let $\Delta x = 1$ and have 26 grid points numbered zero to 25.
The computation domain has length 25 in our dimensionless units system.

Initial conditions for a standing wave

We will create a standing wave with electric field varying as sin(kx) sin(ωt). This definition has no electric field at t = 0 and creates no acceleration at the boundaries at any time that would move electrons to positions outside the boundaries. Integrating the acceleration with time, we find that the initial velocity varies as -sin(kx) cos(ωt) hence the velocites are not zero at t = 0 and must be assigned to be consistent with the assumed wave. Integrating with time again, we find that the positions vary as -sin(kx) sin(ωt) and hence there is no particle displacement at t = 0. If the particles are not displaced from uniform spacing at t = 0, then the electric field at t = 0 is zero. In the equation for E, the constant of integration is then zero.

Initial electron positions

Define kmax particle locations s_k that are evenly spaced. Use the variable name s1 for the locations at the first time step. Let the number of electrons be 1000.

$kmax := 1000 - 1 \qquad k := 0 .. kmax$

$s1_k := \left(\frac{k + 0.5}{kmax + 1}\right) \cdot xmax$

This definition puts the particles on evenly spaced points with the two boundaries x = 0 and x = xmax omitted. In the numerator, k + 0.5 is used rather than k in order to move the first particles away from the left boundary at x = 0. the right boundary is at xmax = x_{jmax}.

Time step and number of iterations

In our dimensionless system, we will let the time step be 0.1 so that there are many time steps in a wave period:

$\Delta t := 0.1$

Two oscillations of the wave will be t = 4π which will require 126 iterations:

$iters := 126$

Initial electron velocities

Recall that the electron velocities must vary as -sin(kx) at t = 0. Assume that the wave is of small amplitude with 0.05 being the maximum velocity. We will make the electrons move initially by assigning a different set of positions at the time step previous to the first one being considered. We will call the particle positions at this time step s0 and use the definition s0 = s1 + V Δt sin(2πs/xmax) with V = 0.05 being the amplitude of the velocity oscillation. The equation has a plus sign because the velocity varies as -sin(kx) and we are backing up one time step which introduces another minus sign.

Amplitude of the velocity oscillations:

$V := 0.05$

$s0_k := s1_k - V \cdot \Delta t \cdot \sin\left(\frac{2 \cdot \pi \cdot s1_k}{x_{jmax}}\right)$

$s0_k$ is the particle location at the "zeroth" time step previous to the first time step.

Charge density calculation

To calculate the charge density, will need to know the grid points that are nearest to a particle. The grid index j for a grid point is simply $s_k/\Delta x$. For a value that is not a grid point, the function **floor**($s_k/\Delta x$) will return the index of the grid point immediately to the left of s_k. This index will be called jlower and the index of the grid point to the right of the particle is jlower +1. As discussed above, if the electron is 10% of the distance from one point to the next, then 10% of the charge is assgned to the second point and 90% to the first. The fraction assigned to the second point is $(s_k - x_{jlower})/\Delta x$ and the fraction assigned to the first point is $(x_{jlower+1} - s_k)/\Delta x$. The program loop below finds the number density at each grid point by summing the contributions from each particle using the weightings we have just defined. There is density modulation in the set of locations $s0_k$ but the locations $s1_k$ are uniform.

We define N_j as the number of particles associated with grid point j. In the program loop below we "look" at each particle and decide what percentage of the charge on the particle to associate with the grid point to the left and right of the particle. The program loop begins by initializing the "counter" N_j to zero. At the end of the loop, the number of particles are divided by the volume Δx between points to obtain the number density.

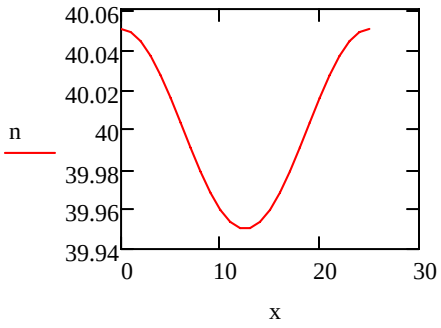
$$
\begin{aligned}
n := \;& N_{jmax} \leftarrow 0 \\
& \text{for } k \in 0 \,..\, kmax \\
& \quad jlower \leftarrow \text{floor}\left(\frac{s0_k}{\Delta x}\right) \\
& \quad N_{jlower+1} \leftarrow \left(\frac{s0_k - x_{jlower}}{\Delta x}\right) + N_{jlower+1} \\
& \quad N_{jlower} \leftarrow \left(\frac{x_{jlower+1} - s0_k}{\Delta x}\right) + N_{jlower} \\
& N_0 \leftarrow 2 \cdot N_0 \\
& N_{jmax} \leftarrow 2 \cdot N_{jmax} \\
& n \leftarrow \frac{N}{\Delta x}
\end{aligned}
$$

Average charge density:

$$n0 := \frac{kmax + 1}{xmax}$$

$$n0 = 40$$

Charge density plot



The last two lines of the loop multiply the number of particles in the first and last grid points by two, because these grid points have electrons only on one side, whereas all the other points have electrons both on the left and on the right. The effective volume contributing to the number of electrons is then smaller by a factor of two. Alternatively, we could invoke periodic boundary conditions and then note that the last grid point and the zeroth grid point are the same point. Hence the charge associated with x_0 should be assigned to x_{jmax}, and vice versa.

Notice in the plot that the charge density is varying cosinusoidally about a mean value n0 that is the total number of particles divided by the width of the domain xmax.

Electric field at grid points

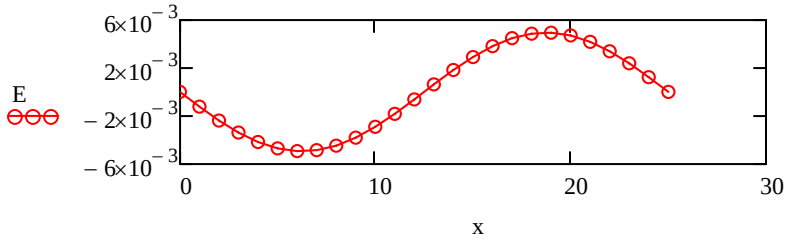
We calculate the electric field at grid points j by integrating the equation for E using finite differences.

$\overline{E}_{j+1} = \overline{E}_j + \Delta\bar{x}\left(1 - \overline{n}_e\right)$ where E_k is the electric field at grid point k.

$$E := \left| \begin{array}{l} E_{jmax} \leftarrow 0 \\ \text{for } j \in 1 .. jmax \\ \quad E_j \leftarrow E_{j-1} + \left[1 - 0.5 \cdot \left(\frac{n_j + n_{j-1}}{n0}\right)\right] \cdot \Delta x \\ E \end{array} \right.$$

This program loop implements the integration, starting at x = 0. The density is a cell is the average of the density at the nearest grid points.

Plot of E at grid points



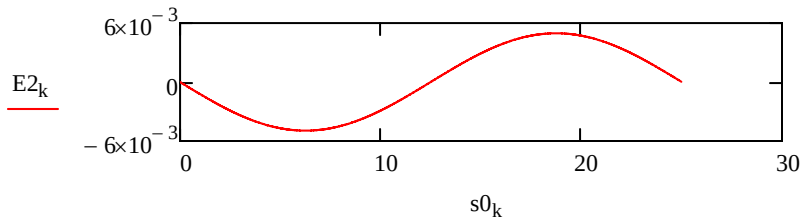
Electric field at particle locations

In order to find the electric field at the particle locations, we will need to interpolate using the values at the grid points. We will use the Mathcad interpolation function **interp** which fits a second order polynomial locally. The coefficients for the polynomial are stored in a variable vs which is obtained by calling the function **pspline**.

$vs := pspline(x, E)$ This function finds the vector array of coefficients vs.

$E2_k := interp\left(vs, x, E, s0_k\right)$ This interpolation routine finds the electric field at the particle locations.

E at particle locations



The program loop

There will be an answer matrix M with the particles locations in each column. The columns will be successive iterations.

$iters := ceil\left(\frac{4\cdot\pi}{\Delta t}\right)$ $\quad iters = 126$ Recall that the number of iterations corresponds to two oscillations of the wave.

$Mblank_{kmax,\,iters-2} := 0$ Mblank is a blank matrix with iters-2 columns and kmax rows that will eventually contain the particle positions at each iteration.

$Mstart := augment(s0, s1, Mblank)$ This line fills in the first two columns of M with the positions at the zeroth and first iterations calculated above.

$M :=$

$E_{jmax} \leftarrow 0$

$M \leftarrow Mstart$

for $i \in 2 .. iters$

for $j \in 0 .. jmax$

$N_j \leftarrow 0$

for $k \in 0 .. kmax$

$jlower \leftarrow floor\left(\frac{M_{k,i-1}}{\Delta x}\right)$

$N_{jlower+1} \leftarrow \left(\frac{M_{k,i-1} - x_{jlower}}{\Delta x}\right) + N_{jlower+1}$

$N_{jlower} \leftarrow \left(\frac{x_{jlower+1} - M_{k,i-1}}{\Delta x}\right) + N_{jlower}$

Recall that:
i is the number of iterations
j is the grid point number
k is the particle number

$n \leftarrow \frac{N}{\Delta x}$

$n_0 \leftarrow 2\cdot n_0$

$n_{jmax} \leftarrow 2\cdot n_{jmax}$

for $j \in 1 .. jmax$

$E_j \leftarrow E_{j-1} + \left[1.0 - 0.5\cdot\left(\frac{n_j + n_{j-1}}{n0}\right)\right]\cdot\Delta x$

$vs \leftarrow pspline(x, E)$

$E\Delta t\Delta t \leftarrow \Delta t^2\cdot interp\left(vs, x, E, M^{\langle i-1\rangle}\right)$

$M^{\langle i\rangle} \leftarrow 2\cdot M^{\langle i-1\rangle} - M^{\langle i-2\rangle} - E\Delta t\Delta t$

$E\Delta t\Delta t$ is a vector containing E at particle locations multiplied by Δt^2.

M

The program loop calculates n, E, and the new positions repetetively.

Particle displacements as a function of time

To "see" the wave we will plot the particle displacements as a function of time. We will find the displacements by subtracting the original locations $s1_k$ from the locations as a function of time.

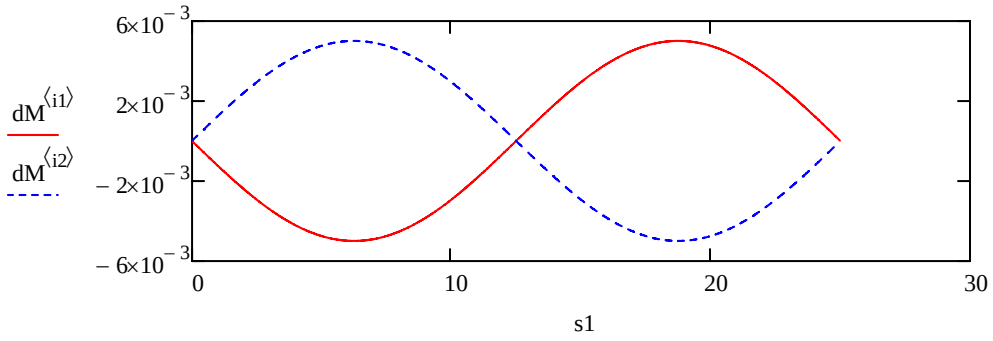
$$dM := \left| \begin{array}{l} dM_{kmax,\,iters} \leftarrow 0 \\ \text{for } i \in 0 .. iters \\ \quad dM^{\langle i \rangle} \leftarrow M^{\langle i \rangle} - s1 \\ dM \end{array} \right.$$

This program loop finds the dispacements and puts them in the matrix dM.

Find the times (iteration numbers) where the oscillation is 1/4 and 3/4 of a period.

$i1 := \text{floor}\left(\frac{\pi}{4}\right)$ $i2 := \text{floor}\left(\frac{3\cdot\pi}{4}\right)$ The plot will be made at these times.

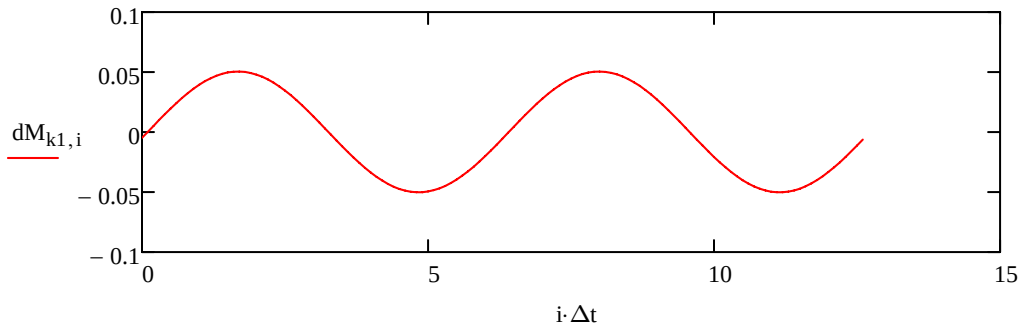
Plot of the particle displacements as a function of distance



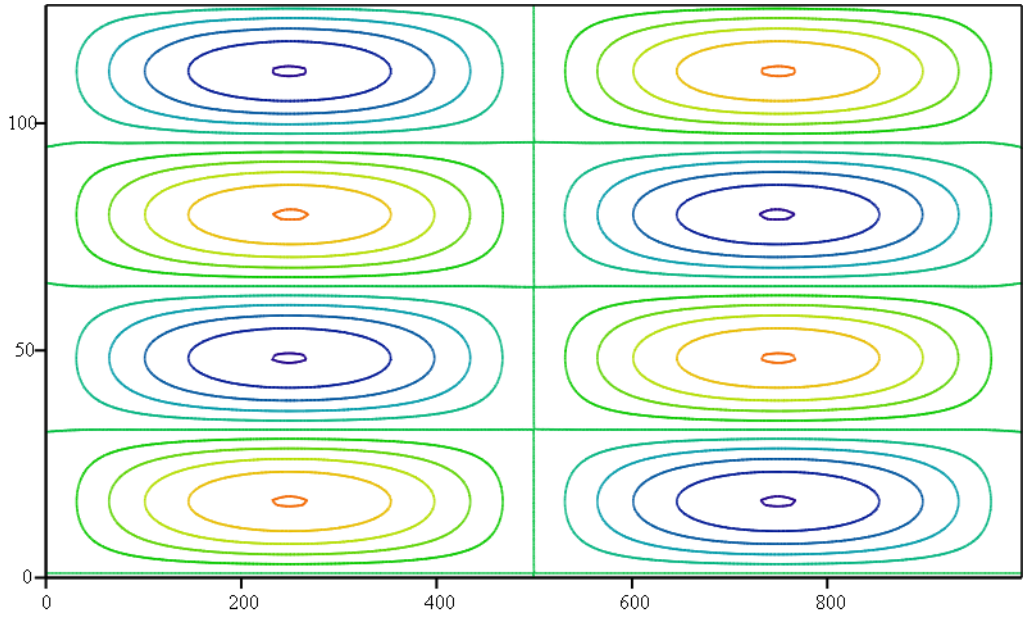
Plot of particle displacements as a function of time

$k1 := \text{ceil}\left(\frac{kmax}{4}\right)$ Plot displacement at the position k1 of a maximum in the wave.

$i := 0 .. iters$ Index for interation number.



Surface plot of particle displacement



dM

In this plot, distance is the horizontal scale and time is the vertical scale. The color contours indicate the particle displacement. There is one wavelength of the wave in the domain and the oscillations are followed for 2 periods.

Reference :

"Plasma Physics via Computer Simulation" by C.K. Birdsall and A.B Langdon, Institute of Physics, Bristol and Philadelphia, 1991.