MADS-CLAUSEN-INSTITUTE, SDU
Christian Wolff

01/03/2023
contact: cwo@mci.sdu.dk

Computational Physics: Problem Set 4

Discussion: 08/03/2023

1 Particle-in-cell method for gas and plasma physics

One approach to problems especially in fluid dynamics (liquids, gases, but mostly plasmas) is called “particle-in-cell”. The idea is to treat the fluid as a swarm of point-particles described by the Newtonian equations of motion. The j-th particle satisfies the equations

$$\begin{aligned}\partial_t \mathbf{r}_j(t) =& \frac{1}{m}\mathbf{p}_j(t),\\ \partial_t \mathbf{p}_j(t) =& \mathbf{F}_j,\end{aligned}$$

where $\mathbf{F}_j$ is the sum of external forces (e.g. the container) and the forces the particles exert on each other. Hard-wall boundaries cause difficulties, so as a “container” we will use a cubic restoring force $\mathbf{F}_j^{(\mathrm{trap})} = -D|\mathbf{r}_j - \mathbf{R}|^2(\mathbf{r}_j - \mathbf{R})$, where $\mathbf{R}$ is the center of the “container”. Position, momentum and force vectors have x, y and z components, position and momentum components should be initialized with random values between -0.5 and 0.5 on program start.

a) Using your Kepler-problem code as inspiration, write a “Kepler-code” that can handle N particles simultaneously, where N is a constant parameter of the program. Use the simple parameters $m = D = 1$ and include only the cubic restoring force with $\mathbf{R} = (0; 0; 0)^T$, i.e. implement the equations

$$\begin{aligned}\partial_t \mathbf{r}_j(t) =& \mathbf{p}_j(t),\\ \partial_t \mathbf{p}_j(t) =& -|\mathbf{r}(t) - \mathbf{R}|^2(\mathbf{r}(t) - \mathbf{R}),\end{aligned}$$

using the leap-frog method for time-evolution. Plot the “top view” of the particle cloud in regular time intervals, e.g.
`plot(rx(:), ry(:), '.');`
assuming that the variables `rx, ry, rz` are vectors of length N that store the particle coordinates. Run the program over the time interval $0 \le t \le 50$ with time step $\Delta t = 0.01$ and $N = 300$ particles (try also larger N, it looks cool). Verify that the cloud size remains more or less the same.
The fact that there is no interaction between the particles makes this a model for an ideal gas.

b) Repeat the same calculation, but this time let the trap center jump to $\mathbf{R} = (1; 0; 0)^T$ when the time variable reaches $t = 25$. How would you interpret what happens in a macroscopic picture? Why is the cloud diameter at $t = 50$ larger than at $t = 24$? How is it possible that the oscillation rings down although there is no loss in the model?

c) Add the following repulsive electrostatic Coulomb-interaction between the particles: each particle (here the j-th one) experiences a force from all other particles according to

$$\mathbf{F}_j^{(\mathrm{Coulomb})} = \sum_{i \neq j} \frac{\alpha(\mathbf{r}_j - \mathbf{r}_i)}{|\mathbf{r}_j - \mathbf{r}_i|^3},$$

with $\alpha = 1/N$. The sum runs over all indices from 1 to N, but skips j, because a particle does not interact with itself (there would a division by zero). Rerun the problem from part b); what is

different in the physical behavior in each stage of the simulation and why? Why is the program so much slower, especially for large N?
This is a simple model for a charged plasma (e.g. in a fusion reactor).

Students are encouraged to solve these problems in groups. They should be able to informally present their solution in the problem class on 08/03/2023 at the white board (no PowerPoint slides to be prepared).