Title: One Dimensional Benchmarks for a Particle Based Kinetic Model (Report on 1D Test Cases and Proxy-App)
Authors: Christopher Ridgers, Adam Dearling, University of York
Date: 30th March 2023
Report code: 2067270-TN-04-01

1 Equations and benchmark simulations setup

In this section we will summarise the equations to be solved. Full details can be found in Ref. [1] where the system is denoted 'System 2-4'. We will also summarise the methodology by which we will produce the benchmark solutions to these equations. In particular we will outline the code used, i.e. the EPOCH particle-in-cell (PIC) code [2], as well as the setup of the benchmark problems.

1.1 Recap of system of equations

We will solve a system of equations (system 2-4) consisting of the Vlasov equation for the electrons and ions in the case where the magnetic field is zero.

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \frac{\partial f_\alpha}{\partial x} + \frac{q\mathbf{E}}{m} \cdot \frac{\partial f_\alpha}{\partial \mathbf{v}} = 0 \tag{1}$$

Where f is the distribution function for electrons (when $\alpha = e$) or ions (when $\alpha = i$) defined in the usual phase space coordinates $(\mathbf{x}, \mathbf{v})$. The electric field is given by the Ampere-Maxwell equation, given the current $\mathbf{j}$.

$$\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} = -\mu_0 \mathbf{j} \tag{2}$$

1.2 Simulation code – EPOCH

Equations (1) and (2) are solved by the particle-in-cell method commonly employed in plasma simulation. Here we will use the PIC code EPOCH described in detail in Ref. [2]. The standard PIC loop, used by EPOCH, is broken down into two steps as follows. First the particle pusher solves the Lorentz force law for the charged particles under the action of the local electromagnetic fields. The current density is then determined by interpolating particle properties onto the grid and solving Maxwell's equations. The particle pusher uses the standard Boris algorithm. The field solver uses the FDTD approach on a staggered Yee grid. There are multiple options for interpolation functions to obtain the current, using different particle shape functions. In the simulations that follow we use third order splines.

1.3 Benchmark problems

The following two benchmark problems are chosen.

1. **Benchmark problem 1 – small amplitude density modulation.** The plasma is initialised with number density $n = n_0 - n_1 \tanh(kx)$ for both electrons and ions. The temperature of

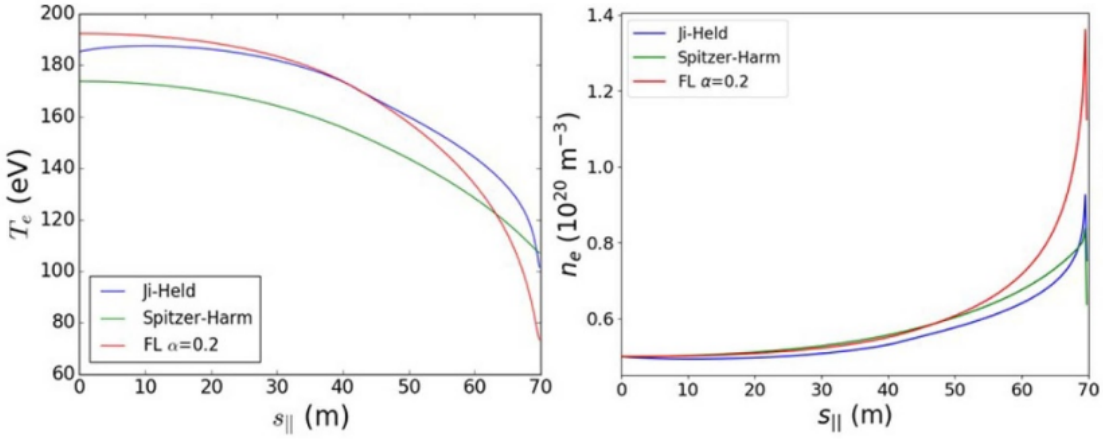

Figure 1: Example simulations of the ITER-SOL in steady state. These electron temperature (left) and number density (right) profiles were produced using the fluid code SD1D [3]. Different thermal conduction models ('Ji-Held', 'Spitzer-Harm' and flux-limited Spitzer-Harm 'FL' with flux-limiter of 0.2) produce similar results, motivating a simple choice of parameters for suitable benchmark problems.

both species is T_0. The perturbation is small in the sense that $n_1 \ll n_0$.

2. **Benchmark problem 2 – large amplitude temperature modulation** The temperature of both electrons and ions is initially set to $T = T_0 - T_1 \tanh(kx)$. The perturbation in this case is large as $T_1 \sim T_0$.

EPOCH simulations of these benchmark problems were set up as follows. In both benchmark problems $k = 1/30\,\mathrm{mm}$. The spatial domain is represented by 200 spatial grid cells. The timescale for the simulations is set to be $120/kv_{Te}$, for benchmark problem 1 and $9/kv_{Te}$ for benchmark problem 2. v_{Te} is the electron thermal speed with the timestep adaptive and set by the CFL condition. The timescales are set this way to ensure time for substantial particle and energy transport to occur.

The plasma parameters are chosen to be reasonably indicative of the ITER SOL. An example simulation is shown in figure 1. This simulations shows an attached case so has a relatively high temperature at the target but nevertheless provides motivation for the following choice of parameters. $n_0 = 10^{19}\,\mathrm{m}^{-3}$, $n_1 = 0.05 \times 10^{19}\,\mathrm{m}^{-3}$, $T_0 = 30\,\mathrm{eV}$, $T_1 = 20\,\mathrm{eV}$. Figure 1 shows that the steepest gradients in density and temperature are in the last few metres before the target in the SOL. This motivates the choice of k.

The plasma is represented by 50,000 macroparticles per cell equally divided between electrons and ions. The EPOCH simulations are 1D-3V, i.e. while macroparticles can have velocities in all three dimensions, spatial gradients are limited to the x-direction only.

2 EPOCH simulation results

2.1 Transport quantities

The transport properties are those of most interest in the aforementioned benchmark cases. Namely the transport of particles and energy. The former is determined by the particle flux $\mathbf{\Gamma}$ while the

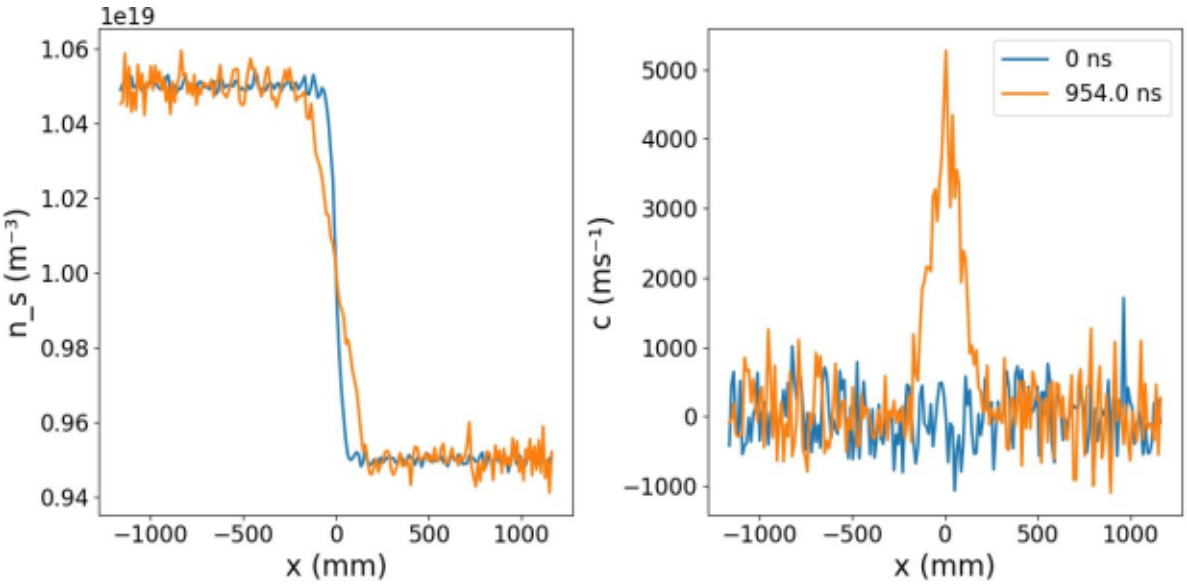

Figure 2: Left: Initial ion number density (n_s) profile (blue) in EPOCH simulations of benchmark problem 1 (blue). Also shown is the ion number density profile at the end of the simulation (orange), i.e. after 0.954 μs. Right: average ion velocity in the x-direction $c = \langle v_x \rangle_i$ initially (blue) and at the end of the EPOCH simulation (orange).

latter is determined by the heat flux $\mathbf{q}$. These are defined in terms of velocity-space averages of the distribution function.

$$\mathbf{\Gamma}_\alpha = n_\alpha \langle \mathbf{v} \rangle_\alpha \tag{3}$$

$$\mathbf{q}_\alpha = n_\alpha \left\langle \frac{1}{2} m v^2 \mathbf{v} \right\rangle_\alpha \tag{4}$$

Here n_α is the particle number density for the species $\alpha = i, e$ (i for ions and e for electrons). The angled brackets denote averages of the distribution function over velocity space as follows.

$$\langle \Phi(\mathbf{v}) \rangle_\alpha = \frac{1}{n_\alpha} \int f_\alpha \Phi(\mathbf{v}) \mathbf{d^3 v}. \tag{5}$$

We will consider both electron and ion transport.

2.2 EPOCH simulation results for benchmark problem 1

Ion inertia dominates the particle flux. Figure 2 shows the initial and final ion number density in the EPOCH simulations of benchmark problem 1. It also shows the average ion velocity that develops in the x-direction as a result of the imposed density modulation. This average velocity is given by $\langle v_x \rangle_i$. As EPOCH discretises the distribution functions as discrete macroparticles this velocity space average is computed by simply taking the mean of v_x for the macroparticles in a given spatial cell.

2.3 EPOCH simulation results for benchmark problem 2

As the temperature of the electrons and ions is equal, the electrons move much faster and thus dominate the heat flux. Figure 3 shows the initial and final electron temperature in the EPOCH

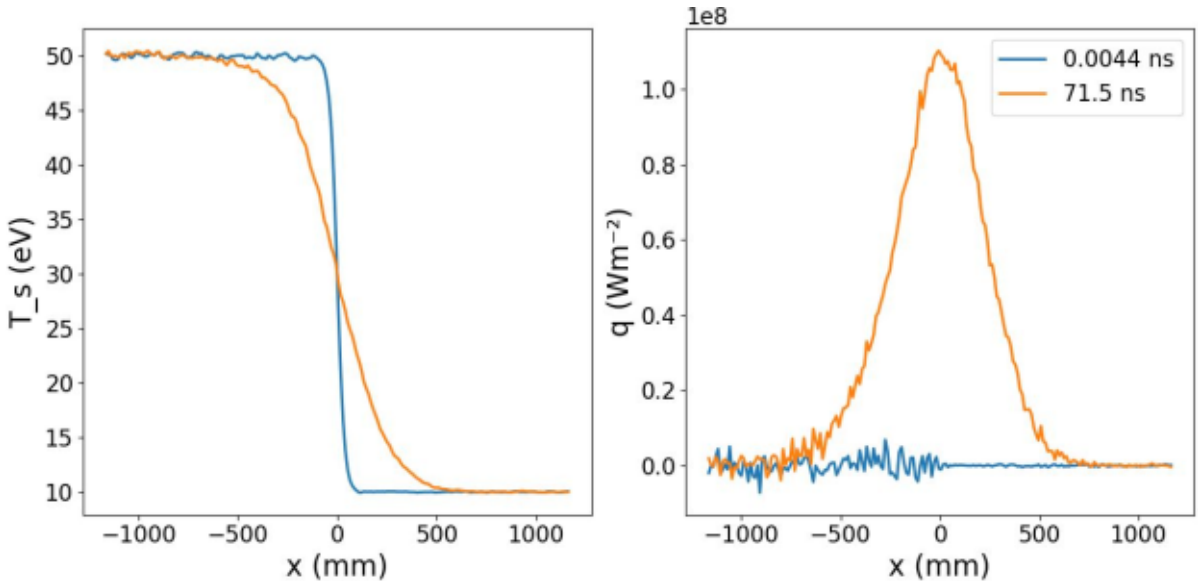

Figure 3: Left: Initial electron temperature (T_s) in eV in EPOCH simulations of benchmark problem 2 (blue). Also shown is the electron temperature profile at the end of the simulation (orange), i.e. after 0.0715 μs. Right: heat flux in the x-direction initially (blue) and at the end of the EPOCH simulation (orange).

simulations of benchmark problem 2 as well as the heat flux. The heat flux is given by $\langle m_e v^2 v_x/2 \rangle_e$. Again this is computed by simply taking the mean for the macroparticles in a given spatial cell.

2.4 Numerical convergence

2.4.1 Convergence with number of particles and gridding

Transport phenomena such as particle and heat flux are challenging to simulate accurately with PIC codes. This is because the transport arises from the anisotropic part of the distribution function, which is often small. Resolving this anisotropic part therefore requires many macroparticles. In addition the steep temperature and density gradients must be adequately resolved with sufficient spatial grid cells. To determine whether this was the case we halved both the number of grid cells and number of macroparticles per cell. Figure 4 shows the peak electron heat flux and average ion velocity as a function of time for benchmark problems 2 & 1 respectively. It is clear that in both cases the results are not changed significantly by reducing the resolution and thus we conclude that numerical convergence has been achieved. We note that the convergence is better for benchmark problem 2 than for problem 1. This is because for the larger modulation in problem 2 the anisotropy in the distribution function driving the transport is bigger and so easier to resolve with macoroparticles.

2.4.2 Minimising numerical heating

Numerical heating is another well known problem when running under-resolved PIC simulations. For low-order particle shape functions, if the grid cell size is not smaller than the Debye length then the plasma spontaneously heats up. It was shown that the use of high-order shape functions suppresses numerical heating in EPOCH simulations, as shown in figure 5. In benchmark problem 2 the grid cell size is approximately 850 Debye lengths (for the 50 eV plasma). This is substantially

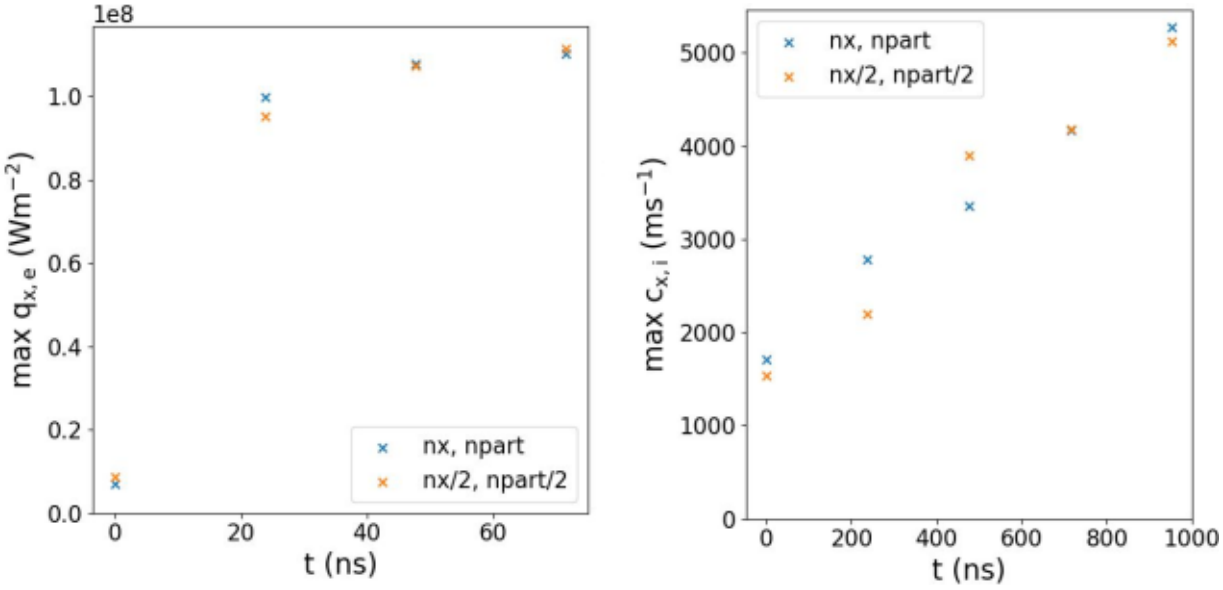

Figure 4: Left: Peak electron heat flux in EPOCH simulations of benchmark problem 2 (blue) compared to the simulations of the same problem with half the number of particles per cell 'npart' and number of grid cells 'nx' (orange) Right: Peak average ion velocity in EPOCH simulations of benchmark problem 1 (blue) compared to the simulations of the same problem with half the number of particles per cell and number of grid cells (orange).

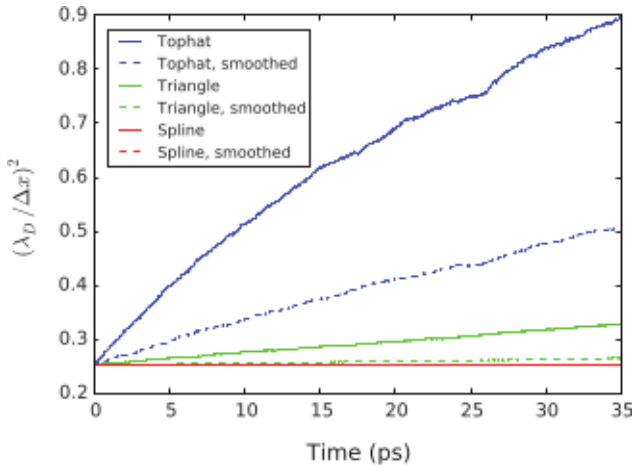

Figure 5: Demonstration of the suppression of numerical heating in EPOCH by using higher order particle shape functions. The plasma is initially uniform and the Debye length (λ_D) – initially 0.5 times the grid cell size Δx – increases as the plasma gets hotter due to numerical heating. Taken from [2].

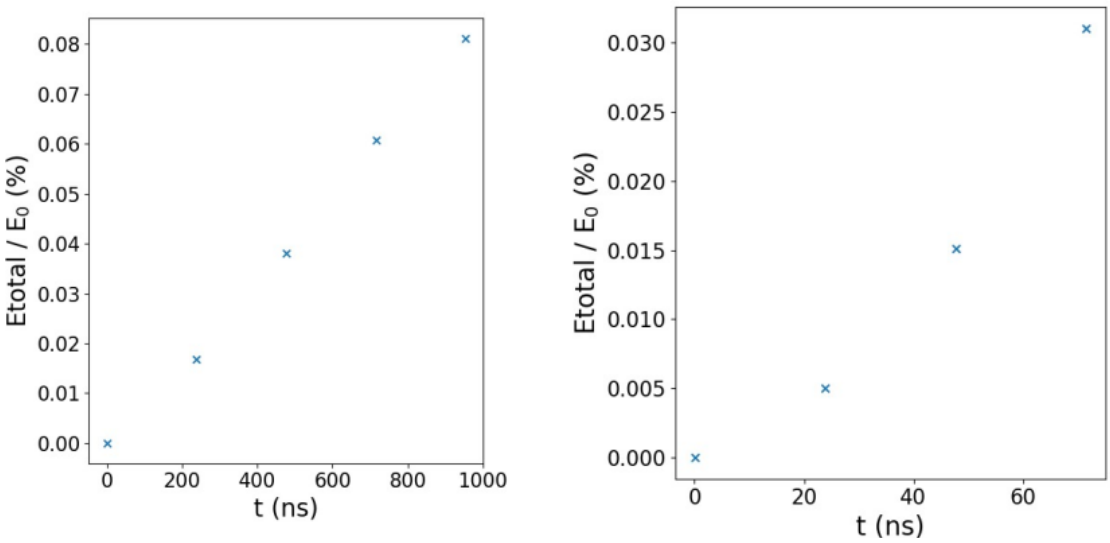

Figure 6: Left: Total energy in the simulation box in EPOCH simulations of benchmark problem 1 as a function of time in the simulation. Right: The same for EPOCH simulations of benchmark problem 2.

larger than in the test cases in figure 5. It is therefore necessary to check that numerical heating is small for our benchmark problems. Figure 6 shows the total energy in the simulations for benchmark problems 1 and 2. It shows that the energy grows by a negligible amount over the course of the simulations of both problems. Numerical heating can thus be safely neglected. It should be noted that in the test cases in figure 5 collisions were included which makes numerical heating worse. This could explain why numerical heating is so small in our (collisionless) simulations, even with very large grid cell sizes.

3 Conclusions

- EPOCH PIC simulations have been used to benchmark system 2-4 (Vlasov and Ampere-Maxwell equations for both electrons and ions). Two benchmark problems were defined to investigate particle and energy transport in scenarios relevant to typical SOL conditions.

- Energy transport was dominated by electron heat flux and bulk transport by ion velocity. The mass difference of these species makes finding a problem where transport from both is important challenging.

- Numerical convergence was achieved for both problems but a large number of particles per cell was required (50,000), This is necessary to resolve small scale anisotropies in the electron and ion distribution which drive the transport. Simulations with large modulations in the driving thermodynamic variable s(temperature , density) converge more easily as the anisotropic part of the distribution function is larger.

- To simulate large enough spatial domains the grid cell must be very large compared to the Debye length of the plasma (850 times larger). Numerical heating can be suppressed in this case by using a high order shape function for the particles.

References

[1] W. Arter, R. Akers, 'Equations for ExCALIBUR/NEPTUNE Proxyapps', Version 1.20, p19-20 (2021).

[2] T.D. Arber et al., Plasma Phys. Control. Fusion, 57, 113001 (2015).

[3] M.R.K. Wigram et al. Nucl. Fusion, 60, 076008 (2020).