Assignment 5, PX390, 2025/2026: beams in a potential

January 15, 2026

This assignment considers the behaviour of a particle beam, injected at both ends of a beam line, subject to accellaration by an external force, as well as collisions with a background gas. The distribution function f is evolved in time.

The equations to be solved are

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}\left[\dot{x} f\right] + \frac{\partial}{\partial v}\left[\dot{v} f\right] - C\frac{\partial^2 f}{\partial v^2} = 0 \tag{1}$$

with

$$\dot{x} \equiv v \tag{2}$$

and

$$\dot{v} \equiv F(x). \tag{3}$$

The equations are solved on a rectangular domain in (x, v) space, with $x \in [0, L]$ and $v \in [-v_m, v_m]$. Boundary conditions for $f(x, v, t)$ are $f(x, v_m, t) = f(x, -v_m, t) = 0$, $f(0, v, t) = \exp(-v^2)$ for $v \geq 0$, and $f(L, v, t) = \exp(-v^2)$ for $v \leq 0$. The initial condition is $f(x, v, 0) = 0$.

$F(x)$ is a force (per unit mass) specified in an input file.

Ideally, your time-evolution code should implement an implicit solver, to allow long timesteps. Timesteps chosen should be shorter than the Courant condition implied by the advective terms (2nd and 3rd terms on the LHS of the equation), but the use of an implicit scheme should allow the timestep to become independent of the choice of the value C.

1 Grids

The grid for output have N_x and N_v points in the x and v direction respectively. They are evenly spaced with $x_i = i/(N_x - 1)$ and $v_j = 2v_m j/(N_v - 1) - v_m$. You may assume $N_x, N_v > 2$.

2 Input

The input file, ‘input.txt’ will contain the following values, one on each line:

- N_x: number of x grid points.
- N_v: number of v grid points.
- t_f: final time.
- L: length of x domain.
- v_m: half-width of v domain.
- C: coefficient of diffusive term.
- I_{min}: minimum number of timesteps (of equal length) to take.

The coefficients file, ‘coefficients.txt’ will have N_x lines containing values of F for a particular grid point x_i, in order of increasing i.

The grid points for output are indexed in the order $L = IN_y + J$, where I is the x grid point index running from $[0, N_x - 1]$, and J is the y grid point indices running from $[0, N_y - 1]$. That is, the first line in the output file contains values corresonding to grid point $L = 0$,and subsequent lines are in order of increasing values of L.

3 Compiling

You can (and probably should) use the LAPACK library to invert matrices. This library is installed on vonneumann. Some instructions are given for installing it on a linux system on the course website; if you aren’t used to compiling/installing libraries on your own system, it might be quicker just to log in to vonneumann. The code must compile and generate no warnings when compiled using the options

```
-Wall -std=c99
```

on the vonneumann machine.

Please call the LAPACKE C interface and not LAPACK directly (because these are FORTRAN functions and the interface is not guaranteed to work the same way when compiled on different machines).

4 Test cases

Useful test cases include the trivial cases where the force is zero. Late-time solutions may be found from the method of characteristics for cases with $C = 0$. Simplifying the boundary conditions for testing purposes may be useful for checking aginst other analytical results.

5 Output

The code should the temperature $F(x, v)$ to a file 'output.txt' in single column format (ie one value per line) for all the grid points (x_i, v_j), for the final state at $t = t_f$. The order of output is in order of increasing values of $i + jN_x$; in other words, first (x_0, v_0), then (x_1, v_0), etc.