Structure preserving low-rank algorithms for plasma simulations: Exercises

Lukas Einkemmer
University of Innsbruck

Structure-Preserving Scientific Computing and Machine Learning Summer School and Hackathon, UW Seattle, 2025

Link to slides: `http://www.einkemmer.net/training.html`

Setup

All the templates can be found at:
`https://github.com/leinkemmer/seattle2025-exercises`

For exercise 1 and 2 you need: Python3+Numpy+Matplotlib

For exercise 3 you need a C++ environment and Ensign.

- See exercise3/Readme.md for instructions
- The dependencies are downloaded/build automatically, but this takes some time.
- On Mac this does not work with the default compiler (no OpenMP support)

You can also use our Virtualbox image.

- Download VirtualBox from `https://www.virtualbox.org/`
- Download the image `https://tinyurl.com/SeattleVM`
- Open Virtualbox and select File → Import Appliance and select the .ova file.

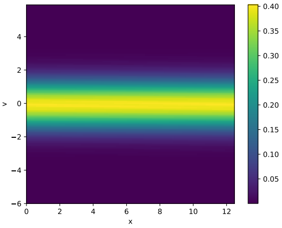
This includes all the required software ready for use (username: user, password: user).

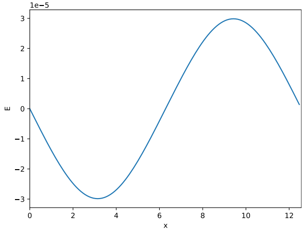
Lab Period I

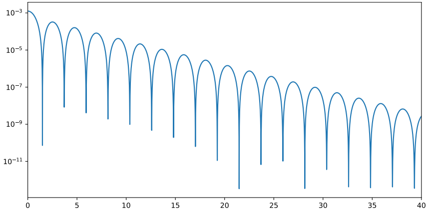
Exercise 1.1

Based on the template `exercise1/lr_template.py` develop a projector splitting based dynamical low-rank algorithm for the Vlasov–Poisson equation.

You should get the following results (Landau damping).







Exercise 1.2

Try a more challenging problem, e.g. the two-stream instability given by

$$f(0,x,v) = \frac{1}{2}(1+10^{-3}\cos(0.2x))\left(\mu(v-2.4)+\mu(v+2.4)\right), \qquad \mu(v) = \frac{\exp(-v^2/2)}{\sqrt{2\pi}}$$

on $\Omega = [0, 10\pi]$.

What rank r do you need to get good results? Compare this to the Landau damping problem in exercise 1.

Lab Period II

Exercises

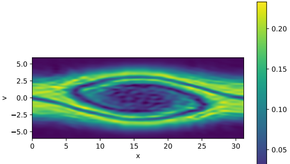
I would suggest you make a choice between

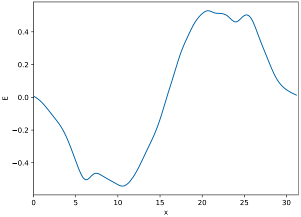
- **Exercise 2** focus on algorithmic aspects and conservation.
- **Exercise 3** focuses on efficient implementation using our low-rank framework Ensign.

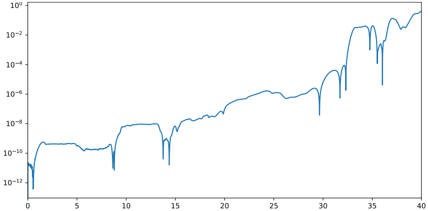
Exercise 2.1

Based on the template `exercise2/lr-conservative_template.py` implement the augmented BUG integrator in the function `time_step_augBUG`.

Check how well mass is conserved for the two-stream instability with $r = 20$.





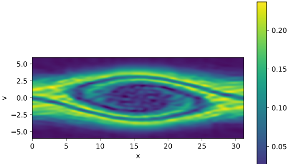


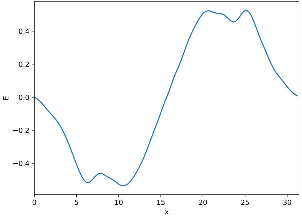
Mass error becomes large!

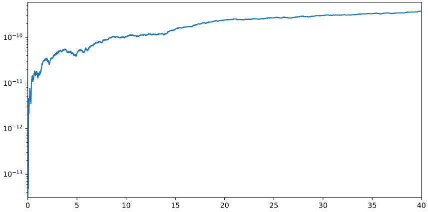
Exercise 2.2

Based on the template `exercise2/lr-conservative_template.py` implement the conservative BUG integrator in the function `time_step_consBUG`.

Check how well mass is conserved for the two-stream instability with $r = 20$.







Mass error is close to machine precision!

Exercise 3

Based on the template `exercise3/main_template.cpp` develop a projector splitting based dynamical low-rank algorithm for the Vlasov–Poisson equation.

The structure of the code is similar to the Python code in exercise 1.

If you use the Virtualbox everything is ready to go.

- How to compile is explained in exercise3/Readme-virtualbox.md