High-Order Numerical Methods for KEEN Wave Vlasov-Poisson Simulations

Institut de Recherche Mathématique Avancée

UNIVERSITÉ DE STRASBOURG

Michel Mehrenberger [a] **Nicolas Crouseilles** [b] **Eric Sonnendrücker** [c] **Bedros Afeyan** [d]

[a] IRMA, Université de Strasbourg, France [b] INRIA-Rennes, France [c] Max-Planck-Institut für Plasmaphysik, Garching, Germany [d] Polymath Research, Pleasanton, USA

mehrenbe@math.unistra.fr, crouseil@inria.fr, sonnen@ipp.mpg.de, bafeyan@gmail.com

Abstract

KEEN waves [1] provided a challenging test case for Vlasov Poisson numerical solvers since they involve highly non stationary, multiple-harmonic self-organized kinetic states. They require high resolution in the phase space region around the phase velocity of the drive wave. Different interpolation strategies are discussed and compared to classical cubic splines.

1. KEEN wave test case

We solve Vlasov-Poisson equation

$$\partial_t f + v\partial_x f + (E - E_{\rm app})\partial_v f = 0,\ \partial_x E = \int_{\mathbb{R}} f dv - 1,$$

where $E_{\rm app}(t,x)$ is of the form

$$E_{\rm app}(t,x) = E_{\max} k a(t)\sin(kx - \omega t),$$

where

$$a(t) = \frac{0.5(\tanh(\frac{t-t_L}{t_{wL}}) - \tanh(\frac{t-t_R}{t_{wR}})) - \epsilon}{1-\epsilon},$$

$$\epsilon = 0.5(\tanh(\frac{t_0 - t_L}{t_{wL}}) - \tanh(\frac{t_0 - t_R}{t_{wR}}))$$

is the amplitude, $t_0 = 0,\ t_L = 69,\ t_R = 307,\ t_{wL} = t_{wR} = 20,\ k = 0.26,\ \omega = 0.37$ and $E_{\max} = 0.2$. The initial condition is

$$f_0(x,v) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{v^2}{2}\right),\quad (x,v) \in [0, 2\pi/k] \times [-6, 6].$$

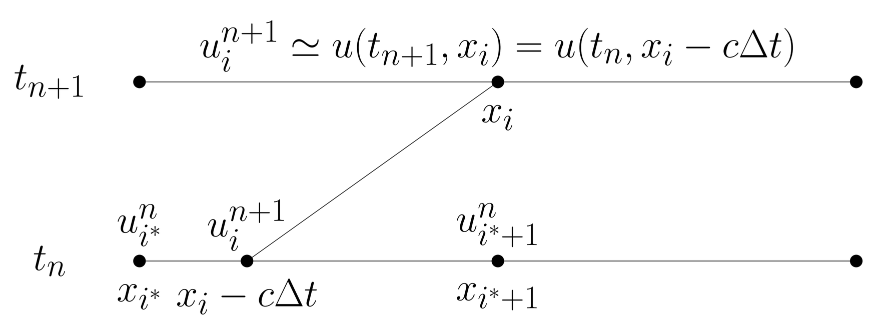
2. Implementation on uniform mesh

Strang (or higher order) splitting leads to

$$\partial_t f + v\partial_x f = 0,\ \partial_t f + (E - E_{\rm app})\partial_v f = 0,$$

i.e. $N \in \{N_x,\ N_v\}$ $1D$ constant advection equations

$$\partial_t u + c\partial_x u = 0,\ u = u(t,x)$$



We can write $u^{n+1} = Au^n$ with circulant matrix

$$A := \begin{pmatrix} a_0 & a_1 & \cdots & \cdots & a_{N-1} \\ a_{N-1} & a_0 & a_1 & \cdots & a_{N-2} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ a_1 & \cdots & \cdots & a_{N-1} & a_0 \end{pmatrix}$$

- Use of FFT for efficient matrix/vector product
- Same form for different interpolations:
  - **LAG(2d+1)**: symmetric Lagrange of order $p = 2d+1$
  - **SPL(p)**: B-splines of order p
  - SPL3 corresponds to classical cubic splines
- Almost same complexity independently of k

3. Acceleration on GPU

Use of existing GPU kernels

- Transposition
- `cufft` for complex FFT
- Adaption of `ScalarProd` for charge density:

$$\sum f_i = \langle f, 1\rangle$$

Kernel for computing A:

- analytical formula for LAG-(2d+1)
- switch complexity from $O(Nd)$ to $O(Nd^2)$

Dealing with single precision issues:

- δ-f method

$$f(x,v) = \delta f(x,v) + f_{\rm eq}(v),\quad f_{\rm eq}(v) = \frac{1}{\sqrt{2\pi}}\exp(-v^2/2).$$

→ Modification of v-advection

$$\delta f(x, v - \Delta t E^n(x)) + f_{\rm eq}(v - \Delta t E^n(x)) - f_{\rm eq}(v)$$

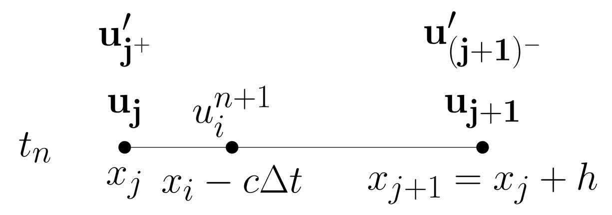
- Zero mean condition for charge density $\rho = \int f dv$

$$\tilde\rho_k = \rho_k - M,\ M = \frac{1}{N}\sum_{k=0}^{N-1}\rho_k,$$

instead of $\tilde\rho_k = \rho_k - 1$, whose RHS is only approximatively of zero mean.

4. Cubic Hermite formulation on uniform mesh

Classical cubic splines method is reinterpreted as cubic Hermite interpolation with particular choice of **derivative reconstruction** leading to a tridiagonal system for $u'_{j^+} = u'_{j^-}$



⇒ In this framework, we can choose other reconstructions for the derivatives

FD(p): compact finite difference of order p

$$u'_{j^+},\ u'_{(j+1)^-}: \text{formula with stencil } j - \lfloor\frac{p}{2}\rfloor, j + \lfloor\frac{p+1}{2}\rfloor$$

- The order p is for point derivatives not for interpolation which remains third order
- Formulae are explicit, which permits easy change of parameter p in the code
- Interpolation becomes local and remains third order
  → generalizations not prohibitive w.r.t cost
- For p even, we get a C^1 reconstruction with $u'_{j^+} = u'_{j^-}$
- For p odd, upwinding effect; better for small Δt
  → no dispersion effect as for p even or SPL3
- FD3=LAG3
- FD(2d+1) $\simeq$ LAG(2d+1), $d \geq 2$
  - equality for limit $\Delta t \to 0$
  - schemes remain different, as FD(2d+1) is third order
- FD6 $\simeq$ SPL3

5. Description of a simple non uniform mesh

Higher resolution needed for velocity around ω/k

⇒ We have to adapt the methods to deal with a **uniform mesh with a refined zone**

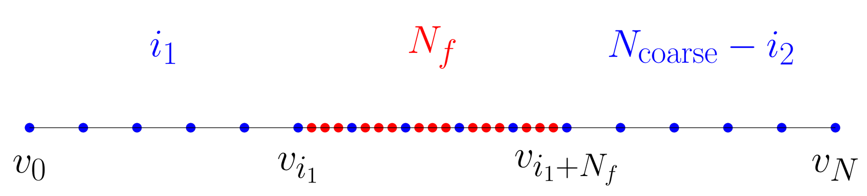
Mesh spacing on coarse/fine grids are

$$\Delta v_{\rm coarse} = \frac{v_{\max} - v_{\min}}{N_{\rm coarse}},\ \Delta v_{\rm fine} = \frac{v_{\max} - v_{\min}}{N_{\rm fine}}$$

and $N_{\rm fine}$ is an integer multiple of $N_{\rm coarse}$.

The refined zone is chosen with $0 \leq i_1 < i_2 \leq N_{\rm coarse}$ and the total number of cells is

$$N = i_1 + N_f + N_{\rm coarse} - i_2,\ N_f = \frac{N_{\rm fine}}{N_{\rm coarse}}(i_2 - i_1)$$



6. Hermite formulation on such mesh

Classical **non uniform cubic splines** can be used and are again reinterpreted as cubic Hermite interpolation with a particular choice of derivative reconstructions

- Again, tridiagonal solver for derivatives
- Works for arbitrary non uniform mesh (not only uniform refined mesh)

For using the specificity of this uniform refined mesh, we can use a two-grid reconstruction for the derivatives:

Two-grid cubic splines:

- Compute derivatives on coarse grid points to get it at points

$$v_j,\ j \in \{0, \ldots, i_1\} \cup \{i_1 + N_f, \ldots, N\}.$$

- Compute it on fine grid points

$$v_j,\ j \in \{i_1, \ldots, i_1 + N_f\},$$

using boundary conditions at points $v_{i_1}, v_{i_1+N_f}$

As in the case of uniform grid, we can adapt the reconstruction of derivatives in the FD(p) case.

Two-grid FD(p):

- Compute derivatives using FD formula on coarse grid
- Compute function values on some boundary fine grid points in $[v_0, v_{i_1}] \cup [v_{i_1+N_f}, v_N]$, that are needed for next step, using interpolation on coarse grid
- Compute derivatives using FD formula on fine grid

7. Conservative version

Previous version has to be changed on non uniform grids in order to be mass conservative.

- Unknowns are $u_{j+1/2} = \frac{1}{v_{j+1} - v_j}\int_{v_j}^{v_{j+1}} u(v)dv$
- Use of previous Hermite interpolation on primitive data

$$U(v_j) = \int_{v_0}^{v_j} u(y)dy,\ v_j,\ j = 0, \ldots, N$$

- Choose adhoc integration constant for dealing with a primitive that is also periodic

8. Numerical results

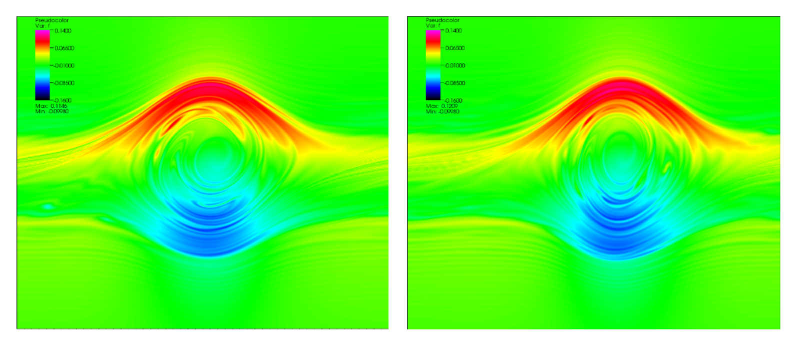


Figure 1: $f(1000,x,v) - f_0(x,v),\ \Delta t = 0.05$. *Comparison of SPL3, with $N_x = N_v = 4096$ on CPU (left) and LAG17 $N_x = N_v = 2048$ on GPU double precision (right).*

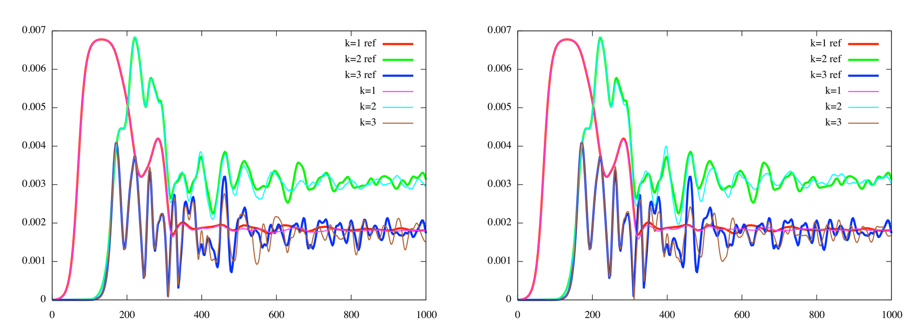


Figure 2: *Absolute values of the first three Fourier modes of ρ vs time. Reference solution with LAG17 $N_x = N_v = 2048$ on GPU double precision (red, green and blue) compared to solution on uniform mesh in space (LAG17, $N_x = 256$) and uniform refined mesh in velocity with $N_v = 374$ ($N_{\rm coarse} = 64$, $N_{\rm fine} = 2048$, $i_1 = 34$, $i_2 = 44$) (left): conservative non uniform cubic splines (right): conservative two-grid FD5*

9. Conclusion

- Efficient GPU $1D \times 1D$ uniform Vlasov-Poisson solver
  - double precision: 2048×2048, 30 Gflops
  - single precision: 4096×4096, 100 Gflops
- Non uniform $1D \times 1D$ Vlasov-Poisson solver
  - number of points are reduced
  - encouraging results for future $2D \times 2D$ simulations

References

[1] B. Afeyan, K. Won, V. Sachenko et al., *Kinetic Electrostatic Electron Nonlinear (KEEN) waves and their interactions driven by the ponderomotive force of crossing laser beams,* IFSA Proceedings 2003 and ArXiv:1210.8105

[2] M. Mehrenberger, C. Steiner, L. Marradi, N. Crouseilles, E. Sonnendrücker, B. Afeyan, *Vlasov on GPU*, submitted to ESAIM Proceedings, 2013.