Exponential methods for solving hyperbolic problems with application to kinetic equations

N. Crouseilles [1,2] L. Einkemmer [3] J. Massot [2,1]

[1]Inria Rennes – Bretagne Atlantique

[2]IRMAR, Université de Rennes

[3]University of Innsbruck

October 11, 2019

Table Of Contents

1. Vlasov-Poisson equations
2. Linear analysis
   - exponential Runge-Kutta method
   - Lawson Runge-Kutta method
3. Numerical simulation: Vlasov-Poisson equations
4. Numerical simulation: drift-kinetic equations
5. Conclusion

Table Of Contents

1. Vlasov-Poisson equations
2. Linear analysis
   - exponential Runge-Kutta method
   - Lawson Runge-Kutta method
3. Numerical simulation: Vlasov-Poisson equations
4. Numerical simulation: drift-kinetic equations
5. Conclusion

Vlasov-Poisson equations 1D×1D

Our model: a non-linear transport in $(x, v) \in \Omega \times \mathbb{R}$ of a density distribution $f = f(t, x, v)$:

$$\begin{cases} \partial_t f + v\partial_x f + E\partial_v f = 0 \\ \partial_x E = \int_{\mathbb{R}} f \,\mathrm{d}v \end{cases}$$

- Filamentation $\rightarrow$ high order methods are needed in phase space (x, v)
- We want extension to multi-dimensional Vlasov-Poisson.
  - Easier with FV/FD than splitting strategy

Fourier transform in x direction

$$\begin{cases} \partial_t \hat{f} + ikv\hat{f} + \widehat{E\partial_v f} = 0 \\ \hat{E} = -\frac{i}{k}\int \hat{f}\,\mathrm{d}v - 1 \end{cases}$$

Duhamel formula: no CFL in x of the form $\Delta t \leq \sigma \frac{\Delta x}{v_{\max}}$ with $[-v_{\max}, v_{\max}] \equiv \mathbb{R}$

Toy model (same difficulties than Vlasov equation):

$$\dot{u} = Au + F(u)$$

$$u(t_n + \Delta t) = \exp(\Delta t A)u(t_n) + \underbrace{\int_0^{\Delta t} \exp((\Delta t - s)A)\, F(u(t_n + s))\,\mathrm{d}s}_{\text{needs approximation}}$$

with $\Delta t > 0$, $t_n = n\Delta t$ with $n \in \mathbb{N}$.

Idea of *exponential integrators*

2 strategies:

exponential Runge-Kutta: solve exactly what we can and interpolate the other. For example first order exponantial Euler method:

$$u(t_n + \Delta t) \approx u^{n+1} = \exp(\Delta t A)u^n + \Delta t \varphi_1(\Delta t A)F(u^n)$$

where $\varphi_1(z) = \frac{(e^z - 1)}{z}$

Lawson: Change of variable : $v(t) := \exp(-tA)u(t)$ just have to solve

$$\dot{v}(t) = \tilde{F}(t, v) = e^{-tA}F(e^{tA}v(t))$$

For example, Lawson Euler method:

$$v(t_n + \Delta t) \approx v^{n+1} = v^n + \Delta t \exp(-t_n A)F(\exp(t_n A)v^n)$$

or as an expression of u :
$u^{n+1} = \exp(\Delta t A)u^n + \Delta t \exp(\Delta t A)F(u^n)$.

Table Of Contents

1 Vlasov-Poisson equations

2 Linear analysis
- exponential Runge-Kutta method
- Lawson Runge-Kutta method

3 Numerical simulation: Vlasov-Poisson equations

4 Numerical simulation: drift-kinetic equations

5 Conclusion

Approximation of continuous operator:

$$\partial f(x_j) \approx (\mathrm{D}f)_j = \mu f$$

$$f_{j+k} \mapsto e^{ik\varphi}$$

μ: function of φ: Fourier symbol.
We can only compute Fourier symbol for linear scheme.
Example for CD2:

$$(\partial_x f)_j \approx \frac{f_{j+1} - f_{j-1}}{2\Delta x} \mapsto \frac{e^{i\varphi} - e^{-i\varphi}}{2\Delta x} = \frac{i\sin(\varphi)}{\Delta x}$$

For an explicit Runge-Kutta method $RK(s,p)$:

- p: order
- s: stages

we can compute $\dot{u} = \lambda u$ with:

$$u^{n+1} = p(\lambda \Delta t) u^n$$

where p (*stability function*) is (for eRK) first terms of exponential series of order p plus some terms to the degree s:

$$p(z) = \sum_{k=0}^{p} \frac{z^k}{k!} + \sum_{k=p+1}^{s} \alpha_k z^k$$

Stability domain is define as:

$$\mathcal{D}_{(s,p)} = \{ z \in \mathbb{C}, |p(z)| \leq 1 \}$$

Reminder on analysis of stability

Geometric interpretation of CFL

It is possible to interpret CFL number between time integrator and space method as the biggest homothety ratio that wedges all the amplification factor curve into the stability domain of considered Runge-Kutta methods.

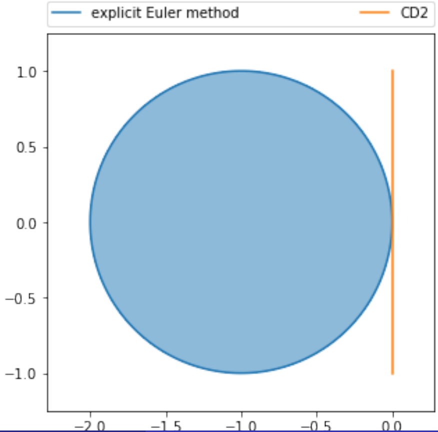
For example:

- explicit Euler method, stability function:

$$p(z) = z + 1$$

- centered difference scheme of order 2 (CD2), Fourier symbol:

$$\mu = i \sin(\varphi)$$



Phase discretization

In v direction we use a DF method :

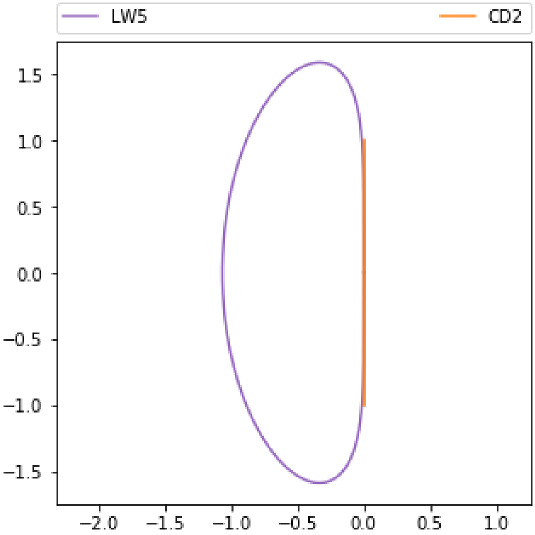
- CD2 (centred difference of order 2): $(\partial_v \hat{f}_k)(v_j) \approx \dfrac{\hat{f}_{k,j+1} - \hat{f}_{k,j-1}}{2\Delta v}$
- WENO5 (weighted essentially non-oscillatory of order 5):

$$(\partial_x f)_j = \frac{f^+_{j+1/2} - f^+_{j-1/2}}{\Delta x} + \frac{f^-_{j+1/2} - f^-_{j-1/2}}{\Delta x}$$

  - WENO5: non linear scheme. . . ✗
  - LW5 (linearized WENO5): linear scheme ✓

$$(\partial_v \hat{f}_k)(v_j) \approx \frac{1}{\Delta v}\left(-\frac{1}{30}\hat{f}_{k,j-3}+\frac{1}{4}\hat{f}_{k,j-2}-\hat{f}_{k,j-1}+\frac{1}{3}\hat{f}_{k,j}+\frac{1}{2}\hat{f}_{k,j+1}-\frac{1}{20}\hat{f}_{k,j+2}\right)$$

Phase discretization



Stability function for ExpRK(2,2):

$$
\begin{aligned}
p(z) = & z^2 \left(-\frac{e^{\frac{3}{2} ia\Delta t}}{a\Delta t^2} + \frac{e^{\frac{1}{2} ia\Delta t}}{a\Delta t^2} + \frac{e^{ia\Delta t}}{a\Delta t^2} - \frac{1}{a\Delta t^2} \right) \\
& + z \left(-\frac{ie^{\frac{3}{2} ia\Delta t}}{a\Delta t} + \frac{ie^{\frac{1}{2} ia\Delta t}}{a\Delta t} \right) + e^{ia\Delta t}
\end{aligned}
$$

Stability domain depends on a !

ExpRK

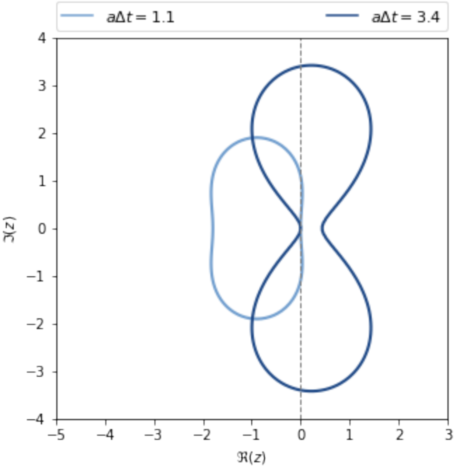


Figure: Stability domain of ExpRK(2,2) for $a\Delta t \in \{1.1, 3.4\}$

ExpRK

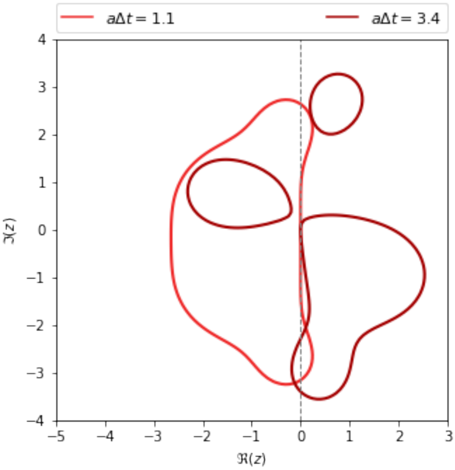


Figure: Stability domain of Cox-Matthews for $a\Delta t \in \{1.1, 3.4\}$

ExpRK

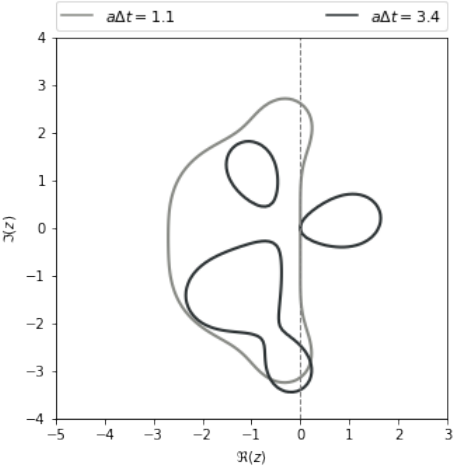


Figure: Stability domain of Krogstad for $a\Delta t \in \{1.1, 3.4\}$

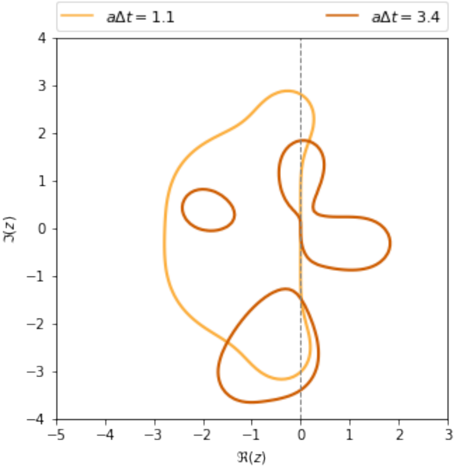


Figure: Stability domain of Hochbruck–Ostermann for $a\Delta t \in \{1.1, 3.4\}$

Stability domain conclusion

Fourier symbol must fit in the stability domain of ExpRK method **for each** values of $a\Delta t \in \mathbb{R}$.

✗ Impossible with WENO5 (LW5 Fourier symbol)

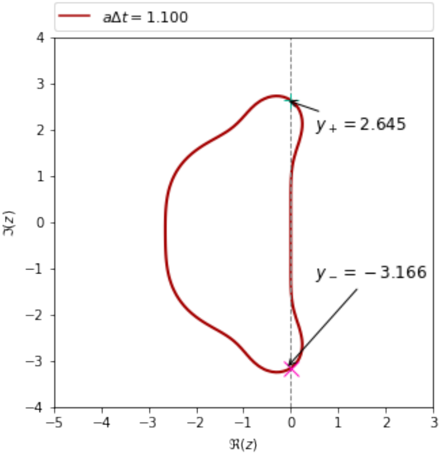
$\rightarrow$ Numerical test: error diverges in very short time

✓ Singleton $\{0\}$ is alway in stability domain of ExpRK method for each values of $a\Delta t$

$\rightarrow$ We can try to stabilize it

✗ SPOILER: CFL is equal to zero

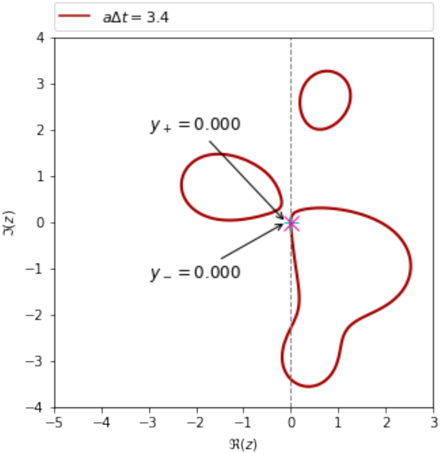
Find $y_{\max}^{exp}(a\Delta t)$ for each value of $a\Delta t$:



The CFL of an ExpRK method with CD2 is $y_{\max} = \min_{a\Delta t} y_{\max}^{exp}(a\Delta t)$

ExpRK – CD2

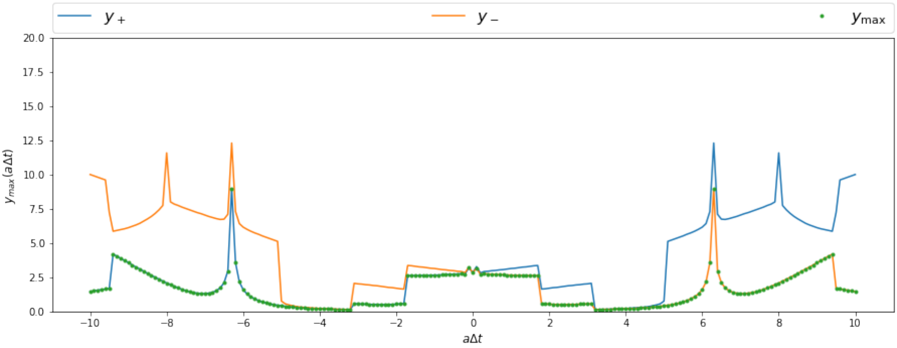
Find $y^{exp}_{\max}(a\Delta t)$ for each value of $a\Delta t$:



The CFL of an ExpRK method with CD2 is $y_{\max} = \min_{a\Delta t} y^{exp}_{\max}(a\Delta t)$

ExpRK – CD2

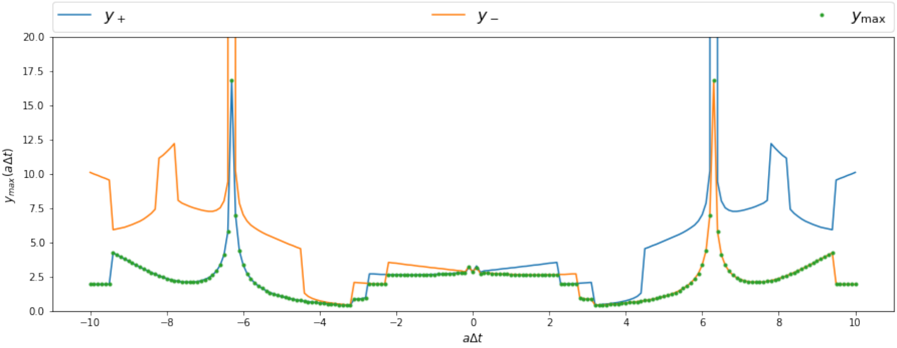
CFL estimation



CFL is still equal to 0

ExpRK – CD2

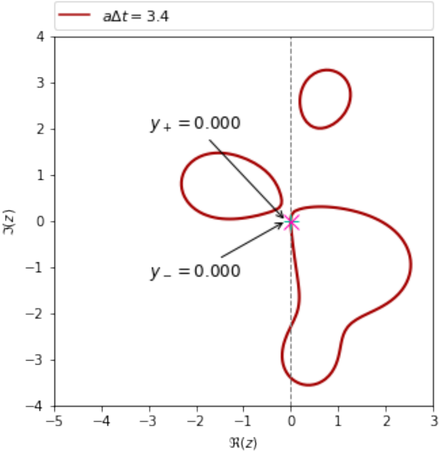
CFL estimation



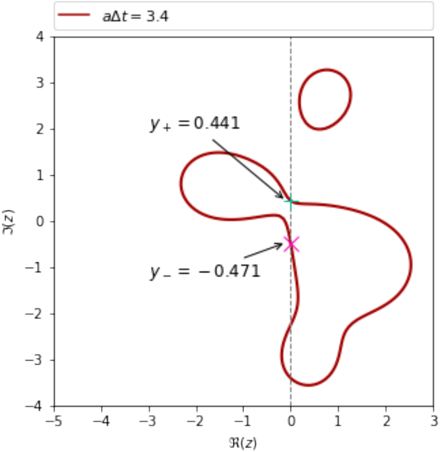
Need to relax CFL condition: $\mathcal{D}_\varepsilon = \{z \in \mathbb{C}, |p(z)| \leq 1 + \varepsilon\}$

ExpRK – CD2

CFL relaxation



Cox-Matthews stability domain, relaxation $\varepsilon = 0$



Cox-Matthews stability domain, relaxation $\varepsilon = 10^{-2}$

Methods	ExpRK22	Krogstad	Cox–Matthews	Hochbruck –Ostermann
$y_{\max}(\varepsilon = 10^{-3})$	0.300	0.100	0.150	0.250
$y_{\max}(\varepsilon = 10^{-2})$	0.551	0.200	0.450	0.501
$y_{\max}(\varepsilon = 10^{-1})$	1.001	0.601	1.351	1.702

Table: CFL number, assuming the relaxed stability constraint, for some exponential integrators.

Lawson

For a problem like:

$$\dot{u} = Au + F(u)$$

Stability function of $Lawson(RK(s,n))$ method is:

$$p_{Lawson(RK(s,n))}(z) = e^{\Delta t A} p_{RK(s,n)}(z)$$

BUT: in our case: $A = ia \in i\mathbb{R}$ so:

Stability domain of $Lawson(RK(s,n))$ is the same than $RK(s,n)$

We just loot at $u_t + u_x = 0$ problem.

Lawson – CD2

With CD2, we work only on the imaginary axis, solve:

$$|p(iy)| = 1, \quad y \in \mathbb{R}$$

We can find analytic value while $s < 5$ (polynomial roots).

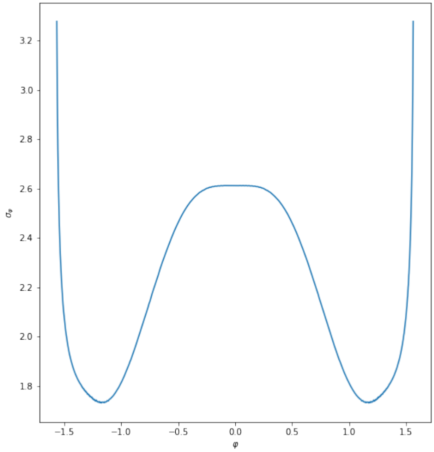
Methods	Lawson($RK(3,2)$ $best$)	Lawson($RK(3,3)$)	Lawson($RK(4,4)$)
y_{max}	2	$\sqrt{3}$	$2\sqrt{2}$

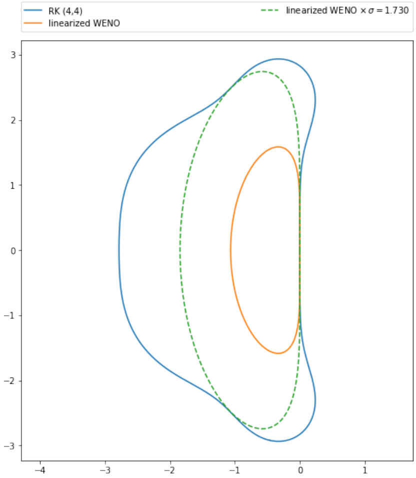
Table: CFL number for some Lawson schemes

With WENO we need to numerical estimation

1. Evaluate Fourier symbol of LW5 with a fine discrete angular grid $\mu_k = \mu(\theta_k)$ with $\{\theta_k\} \subset [0, 2\pi]$. We note $\varphi_k = \arg(\mu_k)$ ($\varphi_k \neq \theta_k$)
2. A discretized version of the boundary of the stability domain of the underlying Runge-Kutta method is computed.
3. For each discretized μ_k, we look for the closest boundary point of the Runge-Kutta stability domain. This enables us to compute the associated stretching factor $\sigma(\varphi_k)$.
4. Taking the minimum over all the discretized eigenvalues yields $\sigma := \min_k \sigma(\varphi_k)$.

With WENO we need to numerical estimation





Lawson – WENO5

With WENO we need to numerical estimation

Methods	Lawson($RK(3,2)$ *best*)	Lawson($RK(3,3)$)	Lawson($RK(4,4)$)
σ	1.344	1.433	1.73

Table: CFL number for some Lawson schemes.

Table Of Contents

1. Vlasov-Poisson equations
2. Linear analysis
   - exponential Runge-Kutta method
   - Lawson Runge-Kutta method
3. Numerical simulation: Vlasov-Poisson equations
4. Numerical simulation: drift-kinetic equations
5. Conclusion

Vlasov-Poisson equation

$$\begin{cases}\partial_t f + v\partial_x f + E\partial_v f = 0\\ \partial_x E = \int f \,\mathrm{d}v\end{cases}$$

Solve with FFT in x direction, WENO5 or CD2 in v direction, and $Lawson(RK(s,n))$ or ExpRK method in time t

Landau damping

$$f(t=0,x,v) = f_0(x,v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} (1 + 0.001\cos(0.5x)),$$

$x \in [0, 4\pi]$, $v \in [-8, 8]$, $N_x = 81$, $N_v = 128$

Landau damping

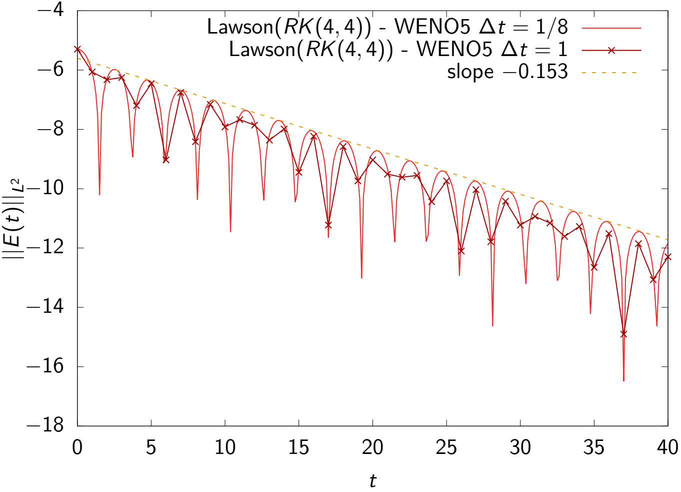


Figure: Landau damping test: time history of $\|E(t)\|_{L^2}$ (semi-log scale) obtained with Lawson($RK(4,4)$) and WENO5 with $\Delta t = 1/8$ and $\Delta t = 1$.

Landau damping

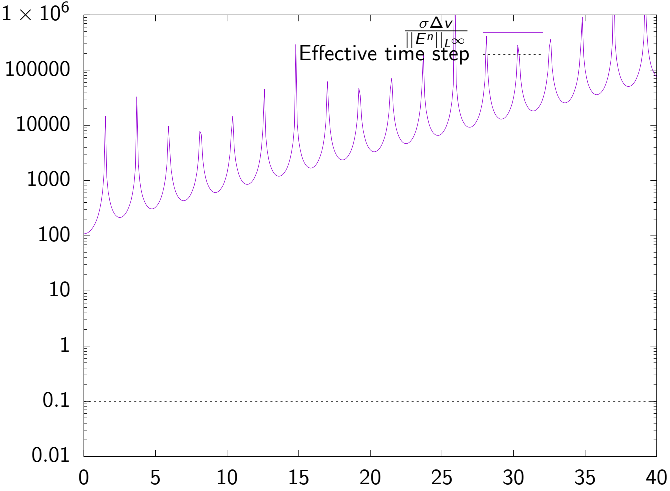


Figure: Landau damping test: time history of the CFL condition (semi-log scale).

Bump on Tail (BoT)

$$f(t=0,x,v) = f_0(x,v) = \left[\frac{0.9}{\sqrt{2\pi}}e^{-\frac{v^2}{2}} + \frac{0.2}{\sqrt{2\pi}}e^{-2(v-4.5)^2}\right](1+0.001\cos(0.5x$$

$x \in [0, 20\pi]$, $v \in [-8, 8]$, $N_x = 135$, $N_v = 256$
A first simulation with small Δt to know: $E_{\text{max}} \approx 0.6$.

$$\Delta t = \frac{C\Delta v}{E_{\text{max}}},\ C = y_{\text{max}} \text{ or } C = \sigma$$

Bump on Tail (BoT)

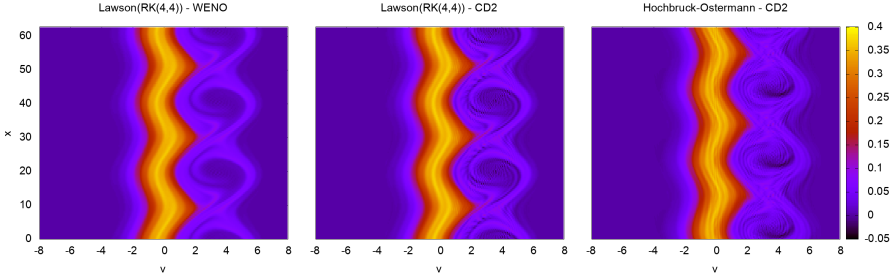


Figure: Distribution function at time $t = 40$ as a function of x and v for Lawson($RK(4,4)$) + WENO5 (left), Lawson($RK(4,4)$) + centered scheme (center), Hochbruck–Ostermann + centered scheme (right).

Bump on Tail (BoT)

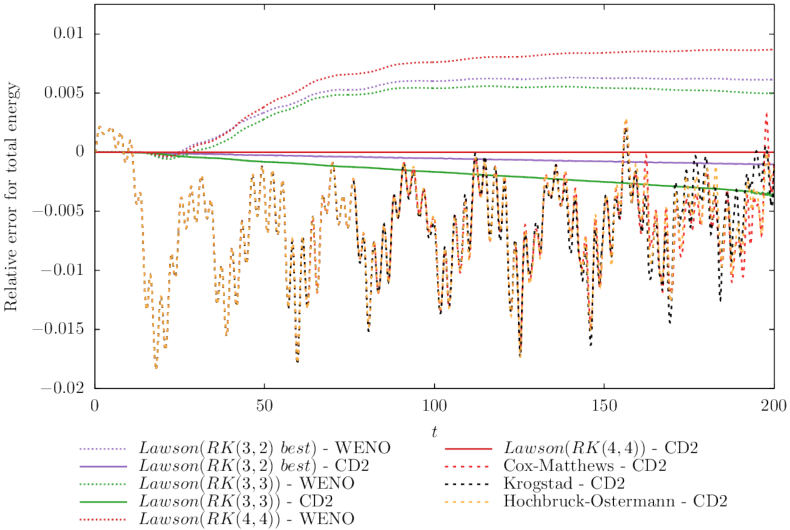
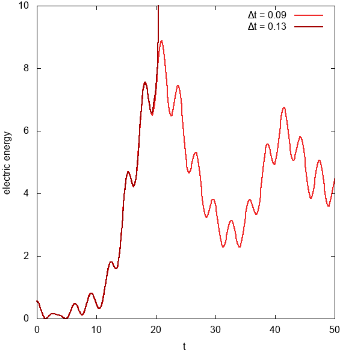


Figure: Time evolution of the relative error of the total energy for the different

Bump on Tail (BoT)



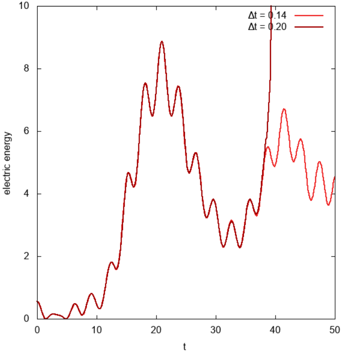


Figure: Illustration of the accuracy of the CFL estimate obtained from the linear theory. History of electric energy with Lawson($RK(4,4)$) + WENO5 (left), Lawson($RK(4,4)$) + centered scheme (middle)

Bump on Tail (BoT)

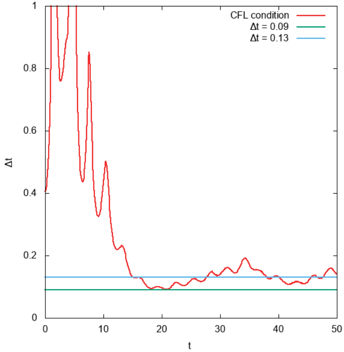


Figure: History of CFL condition for Lawson($RK(4,4)$) + WENO5 case (right)

Adaptive time step size (easy to use)

To capture correctly the phenomena involved in the bump on tail test, we take the following time step size:

$$\Delta t_n = \min\left(0.1, \frac{C\Delta v}{||E^n||_{L^\infty}}\right)$$

with $C = y_{\max}$ or σ
$\rightarrow$ Good estimate in practice for Lawson methods.

Table Of Contents

1 Vlasov-Poisson equations

2 Linear analysis
- exponential Runge-Kutta method
- Lawson Runge-Kutta method

3 Numerical simulation: Vlasov-Poisson equations

4 Numerical simulation: drift-kinetic equations

5 Conclusion

Adaptive time step size (error estimate)

For adaptive time step size with any time integrator φ:

$$f^{n+1} = \varphi_{\Delta t_n}(f^n) \qquad ; \qquad \tilde{f}^{n+1} = \varphi_{\Delta t_n/2} \circ \varphi_{\Delta t_n/2}(f^n)$$

Richardson extrapolated numerical solution of the method of order p:

$$f_R^{n+1} = \frac{2^{p+1}\tilde{f}^{n+1} - f^{n+1}}{2^{p+1} + 1}$$

estimate of the local error:

$$e_{n+1} = ||f_R^{n+1} - f^{n+1}||_{L^\infty} + \mathcal{O}(\Delta t_n^{p+2})$$

If $e_{n+1} > \text{tol}$: we reject the step and start again form time t_n. Else we determine the new time step size:

$$\Delta t_{new} = s\Delta t_n \left(\frac{\text{tol}}{e_{n+1}}\right)^{1/(p+1)}$$

$s = 0.8$ is safety factor.

Drift-kinetic equations

$f = f(t, r, \theta, z, v)$

$$\begin{cases} \partial_t f - \frac{\partial_\theta \phi}{r}\partial_r f + \frac{\partial_r \phi}{r}\partial_\theta f + v\partial_z f - \partial_z \phi \partial_v f = 0 \\ -\left[\partial_r^2 \phi + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r \phi + \frac{1}{r^2}\partial_\theta^2 \phi\right] + \frac{1}{T_e(r)}(\phi - \langle\phi\rangle) = \frac{1}{n_0(r)}\int_{\mathbb{R}} f \,\mathrm{d}v - 1 \end{cases}$$

$(r, \theta, z, v) \in [0.1, 14.5] \times [0, 2\pi] \times [0, L] \times \mathbb{R}$,
Solve with FFT in z direction, CD2 in others

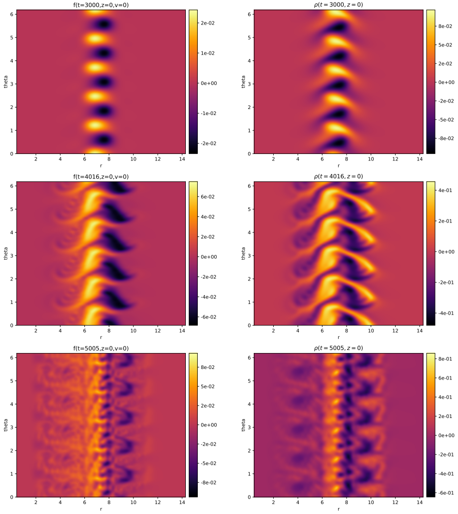
Numerical result

$$f(t=0,r,\theta,z,v) = f_{\text{eq}}(r,v)\left[1+\epsilon\exp\left(-\frac{(r-r_p)^2}{\delta r}\right)\cos\left(\frac{2\pi n}{L}z+m\theta\right)\right],$$

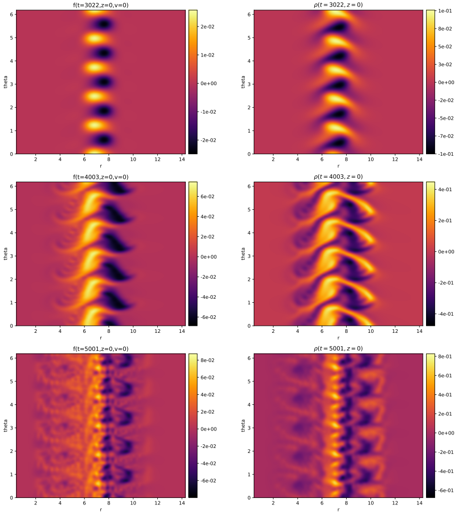
where the equilibrium distribution is given by:

$$f_{\text{eq}}(r,v) = \frac{n_0(r)\exp\left(-\frac{v^2}{2T_i(r)}\right)}{(2\pi T_i(r))^{1/2}}$$

Numerical result



Lawson(RK(4,4)) – CD2



Cox-Matthews – CD2

Numerical result

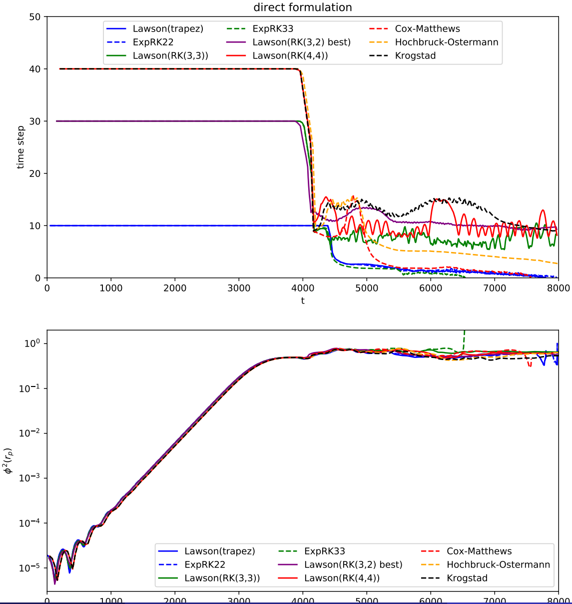


Table Of Contents

1 Vlasov-Poisson equations

2 Linear analysis
- exponential Runge-Kutta method
- Lawson Runge-Kutta method

3 Numerical simulation: Vlasov-Poisson equations

4 Numerical simulation: drift-kinetic equations

5 Conclusion

Conclusion

- Better understand on stability of Lawson or ExpRK methods
- Automatic script for CFL estimation between Lawson – CD2 or Lawson – WENO or ExpRK – CD2
- Adaptive time step size with minimal cost or with an estimate of the local error

Tank you for your attention

Backup