

# (Approximate) Exponential methods for solving hyperbolic problems for electron hybrid model

A. Crestetto <sup>1</sup>   N. Crouseilles <sup>2,3</sup>   Y. Li <sup>4</sup>   J. Massot <sup>3,2</sup>

<sup>1</sup>LMJL, Université de Nantes

<sup>2</sup>Inria Rennes – Bretagne Atlantique

<sup>3</sup>IRMAR, Université de Rennes

<sup>4</sup>Max Planck Institute for Plasma Physics, Garching, Germany

October 21, 2021

# Outline

- 1 Introduction
- 2 Numerical method and approximation
- 3 Numerical tests
- 4 Numerical results
- 5 Conclusion

# Outline

- 1 Introduction
- 2 Numerical method and approximation
- 3 Numerical tests
- 4 Numerical results
- 5 Conclusion

# Introduction

A nonlinear transport in  $(z, v_x, v_y, v_z) \in \Omega \times \mathbb{R}^3$  of a cold (fluid) electron density distribution (reconstruction from current variable  $\mathbf{j}_c$ ) and a hot (kinetic) electron density distribution  $f_h$ :



Holderied et al. 2020, *Journal of Computational Physics*

$$\begin{cases} \partial_t \mathbf{j}_c = \Omega_{pe}^2 \mathbf{E} - J \mathbf{j}_c B_0 \\ \partial_t \mathbf{B} = J \partial_z \mathbf{E} \\ \partial_t \mathbf{E} = -J \partial_z \mathbf{B} - \mathbf{j}_c + \int_{\mathbb{R}^3} v_{\perp} f_h d\mathbf{v} \\ \partial_t f_h + v_z \partial_z f_h - (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h = 0 \end{cases}$$

with:

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

**We want:**

- High order time integrator and space integrator (FFT + WENO)
- Efficient adaptive time step method

# Outline

- 1 Introduction
- 2 Numerical method and approximation
- 3 Numerical tests
- 4 Numerical results
- 5 Conclusion

# Lawson method

We would like to solve

$$\partial_t u = Lu + N(t, u)$$

Change of variable:  $v = e^{-tL}u$ , we obtain:

$$\begin{aligned}\dot{v}(t) &= -Le^{-tL}u(t) + e^{-tL} \underbrace{(Lu(t) + N(t, u))}_{\dot{u}(t)} \\ &= e^{-tL}N(t, e^{tL}v)\end{aligned}$$

which can be solve with a **Runge-Kutta method** in  $v$ , that can be rewritten in  $u$ , for example with Euler method:

$$v(t^n + \Delta t) \approx v^{n+1} = v^n + \Delta t e^{-t^n L} N(t^n, e^{t^n L} v^n)$$

or as an expression of  $u$ :

$$u^{n+1} = e^{\Delta t L} u^n + \Delta t e^{\Delta t L} N(t^n, u^n)$$



Lawson 1967, *SIAM Journal on Numerical Analysis*



Hochbruck, Leibold, and Ostermann 2020, *Numerische Mathematik*

# Lawson method

## Pros & Cons

- ✓ Numerically efficient (order increases linearly-ish with the number of stages)
- ✓ Literature on Runge-Kutta method (embedded-RK, IMEX methods, low storage methods, ...)
- ✓ Linear part is solved exactly
- ✗ Stability constraint (not from the linear part ✓)
- ✗ Behavior in long time
- ~ Needs to compute (efficiently)  $e^{\tau L}$  for any  $\tau = c_j \Delta t$  and  $L$

# How to compute efficiently $e^{\tau L}$ ?

## Why this could be complicated ?

- ✗ We would like a formal form depending on time parameter  $\tau$  and all hidden parameters of matrix  $L$

## Solutions:

Taylor series: simplest and first method studying

Padé approximant: defined as the best rational approximation of a function

...: Some other methods don't explore yet



Moler and Van Loan 2003, *SIAM Review*

## Simplification:

- Suppose  $L$  diagonalizable and all its eigenvalues are pure imaginary  $\text{sp}(L) \subset i\mathbb{R} \implies \text{sp}(e^{tL}) \subset \mathcal{C}(0, 1)$ .



Simplest approximation:

$$T_p(\tau L) = \sum_{k=0}^p \frac{\tau^k}{k!} L^k = e^{\tau L} + \mathcal{O}(\tau^{p+1})$$

## Theorem

$sp(L) \subset i\mathbb{R} \setminus i[-1, 1]$  implies eigenvalues diverge

*Proof:* compute Taylor series outside of its convergence radius

## Conclusion:

- ✗ Bad behavior of eigenvalues
- ✗ Numerical instability in scheme

# Eigenvalues of Taylor series

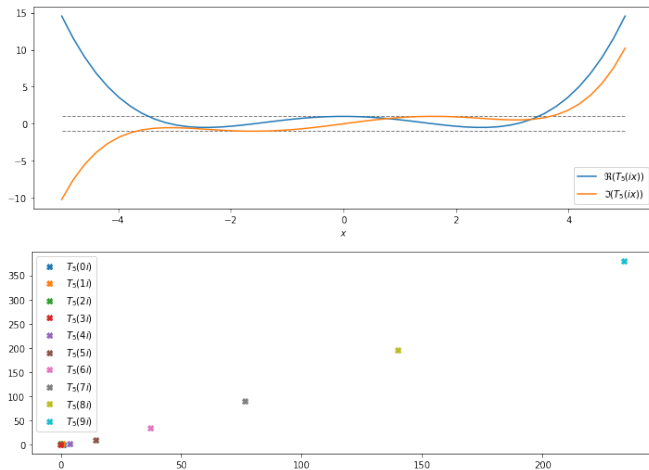


Figure:  $T_5(ix)$ ,  $x \in [0, 9]$

# Padé approximant

Best rational approximation of exponential function.

Defined (for order  $(p, q)$ ) as:

$$h_{p,q}(M) = \sum_{i=0}^p \frac{\frac{p!}{(p-i)!}}{\frac{(p+q)!}{(p+q-i)!}} \frac{M^i}{i!} \quad , \quad k_{p,q}(M) = \sum_{j=0}^q (-1)^j \frac{\frac{q!}{(q-j)!}}{\frac{(p+q)!}{(p+q-j)!}} \frac{M^j}{j!}$$

Finally Padé approximant is:

$$P_{p,q}(\tau L) = h_{p,q}(\tau L) (k_{p,q}(\tau L))^{-1} = e^{\tau L} + \mathcal{O}(\tau^{p+q+1})$$

## Theorem

$$sp(L) \subset i\mathbb{R} \implies sp(P_{p,p}(tL)) \subset \mathcal{C}(0, 1)$$

## Conclusion:

✗ Needs matrix inversion, or some tricks:



Li, Zhu, and Gu 2011, *Applied Mathematics*

✓ Best approximation for this numerical cost

✓ Preserve eigenvalues

# Proof

$L$  diagonalizable and  $\text{sp}(L) \subset i\mathbb{R} \implies L = Q^{-1}DQ$

$$\begin{aligned} P_{p,p}(L) &= \left( \sum_{k=0}^p \frac{1}{k!} Q^{-1} D^k Q \right) \cdot \left( \sum_{\ell=0}^p (-1)^\ell \frac{1}{\ell!} Q^{-1} D^\ell Q \right)^{-1} \\ &= Q^{-1} \left( \sum_{k=0}^p \frac{1}{k!} D^k \right) \cdot \left( \sum_{\ell=0}^p (-1)^\ell \frac{1}{\ell!} D^\ell \right)^{-1} Q \end{aligned}$$

with  $D = \text{diag}(\{i\alpha_j, j = 1, \dots, d\})$

$$\begin{aligned} \sum_{k=0}^p \frac{1}{k!} D^k &= \text{diag} \left( \left\{ \sum_{k=0}^{\lfloor \frac{p}{2} \rfloor} (-1)^k \frac{\alpha_j^{2k}}{2k!} + i \sum_{k=0}^{\lfloor \frac{p}{2} \rfloor - 1} (-1)^k \frac{\alpha_j^{2k+1}}{(2k+1)!}, j \in \llbracket 0, d \rrbracket \right\} \right) \\ \sum_{\ell=0}^p (-1)^\ell \frac{1}{\ell!} D^\ell &= \text{diag} \left( \left\{ \sum_{\ell=0}^{\lfloor \frac{p}{2} \rfloor} (-1)^\ell \frac{\alpha_j^{2\ell}}{2\ell!} - i \sum_{\ell=0}^{\lfloor \frac{p}{2} \rfloor - 1} (-1)^\ell \frac{\alpha_j^{2\ell+1}}{(2\ell+1)!}, j \in \llbracket 0, d \rrbracket \right\} \right) \end{aligned}$$

$$\lambda^- = \overline{\lambda^+} \text{ so } \left| \frac{\lambda^+}{\lambda^-} \right| = 1.$$

# Eigenvalues of Padé approximant

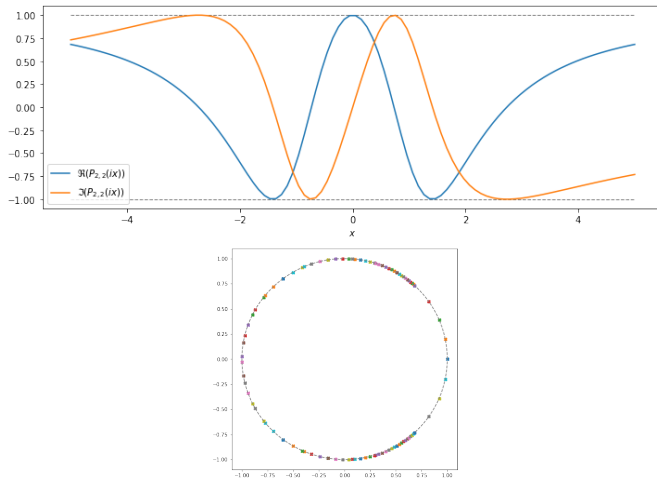


Figure:  $P_{2,2}(ix)$ ,  $x \in [-5, 5]$

# Eigenvalues of assymetric Padé approximants

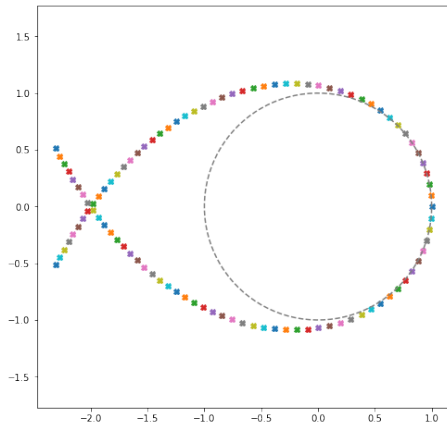


Figure:  $P_{2,1}(ix)$   $x \in [-5, 5]$

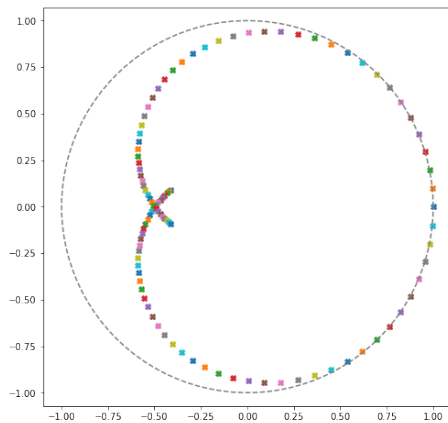


Figure:  $P_{1,2}(ix)$   $x \in [-5, 5]$

# Error on approximate Lawson method

We note  $\exp(L) = P_{p,q}(L)$  or  $T_p(L)$ , generic approximation.

$$\exp(L) = e^L + \mathcal{O}(L^{r+1})$$

Lawson RK(3,3) method:

$$\begin{aligned}u^{(1)} &= e^{\Delta t L} u^n + \Delta t e^{\Delta t L} N(t^n, u^n) \\u^{(2)} &= \frac{3}{4} e^{\frac{\Delta t}{2} L} u^n + \frac{1}{4} e^{-\frac{\Delta t}{2} L} u^{(1)} + \frac{\Delta t}{4} e^{-\frac{\Delta t}{2} L} N(t^n + \Delta t, u^{(1)}) \\u^{n+1} &= \frac{1}{3} e^{\Delta t L} u^n + \frac{2}{3} e^{\frac{\Delta t}{2} L} u^{(2)} + \frac{2}{3} \Delta t e^{\frac{\Delta t}{2} L} N(t^n + \frac{\Delta t}{2}, u^{(2)})\end{aligned}$$

**If**  $L$  and  $N$  commute:  $u^{n+1} = e^{\Delta t L} \left( I + N + \frac{N^2}{2} + \frac{N^3}{6} \right) u^n$ , stability is same as RK(3,3). **Else**...

# Error on approximate Lawson method

If  $L$  and  $N$  don't commute:

$$\begin{aligned} u^{n+1} &= \left[ e^{\Delta t L} + \Delta t \left( \frac{2}{3} e^{\frac{\Delta t}{2} L} N e^{\frac{\Delta t}{2} L} + \frac{1}{6} e^{\Delta t L} N + \frac{1}{6} N e^{\Delta t L} \right) \right] u^n \\ &\quad + \frac{\Delta t^2}{2} \left( \frac{1}{3} N e^{\Delta t L} N + \frac{1}{3} e^{\frac{\Delta t}{2} L} N e^{\frac{\Delta t}{2} L} N + \frac{1}{3} e^{\frac{\Delta t}{2} L} N e^{-\frac{\Delta t}{2} L} N e^{\Delta t L} \right) u^n \\ &\quad + \frac{\Delta t^3}{6} e^{\frac{\Delta t}{2} L} N e^{-\frac{\Delta t}{2} L} N e^{\Delta t L} N u^n \\ &\rightsquigarrow e^{\Delta t L} \left[ \frac{(\Delta t N)^2}{2} + \frac{(\Delta t N)^3}{6} \right] u^n \end{aligned}$$

Same results if  $e^{\Delta t L} \mapsto \exp(\Delta t L) = e^{\Delta t L} + \mathcal{O}(\Delta t^{r+1})$

## Lemma

Error for Lawson RK( $m, s$ ) is always in  $\mathcal{O}(\Delta t^{m+1}) + \mathcal{O}(\Delta t^{r+1})$



# Outline

- 1 Introduction
- 2 Numerical method and approximation
- 3 Numerical tests**
- 4 Numerical results
- 5 Conclusion

# Test 1

## 2D translation test case

$$\partial_t u + a \partial_x u + b \partial_y u = 0$$

After a Fourier transform in  $y$

$$\partial_t \hat{u} + \underbrace{ibk \hat{u}}_{L\hat{u}} + \underbrace{\widehat{a \partial_x u}}_{N(\hat{u})} = 0$$

First test with:

- Lawson RK(3,3) method
- Lawson RK(3,3) method with Taylor series  $T_p$ ,  $p \in \llbracket 1, 4 \rrbracket$
- Lawson RK(3,3) method with Padé approximant  $P_{p,q}$ ,  $p, q \in \llbracket 1, 2 \rrbracket$

**Mesure order:**  $x, y \in [-2, 2]$ ,  $N_x = N_y = 243$ ,  $a = 1.0$ ,  $b = 0.75$ ,  
 $T_f = 0.07111$ ,  $\Delta t \in [0.00158, 0.02370]$ .

# Test 1

2D translation test case: mesure of order

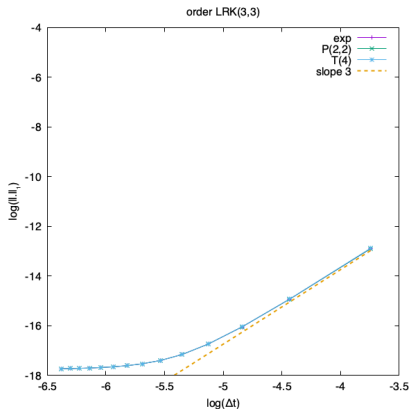


Figure: Order of Lawson RK(3,3) method, and Lawson RK(3,3),  $P_{2,2}$  approximant method and Lawson RK(3,3)  $T_4$  series method.

# Test 1

2D translation test case: mesure of order

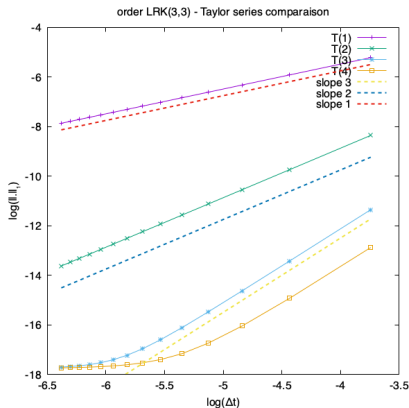


Figure: Order of Lawson RK(3,3)  $T_p$  series method,  $p = 1, \dots, 4$ .

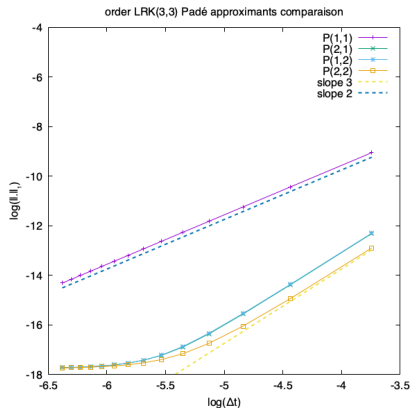


Figure: Order of Lawson RK(3,3)  $P_{p,q}$  approximant,  $p = 1, 2, q = 1, 2$

# Test 2

## 2D rotation test case

$$\partial_t u - y \partial_x u + x \partial_y u = 0$$

After a Fourier transform in  $y$

$$\partial_t \hat{u} + \underbrace{ixk\hat{u}}_{L\hat{u}} + \underbrace{-y\partial_x u}_{N(\hat{u})} = 0$$

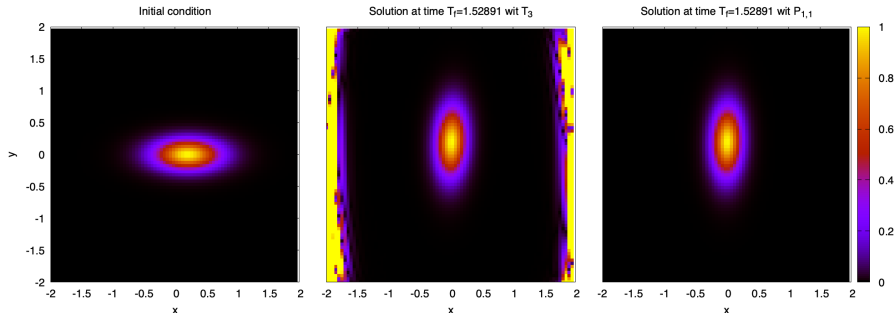
Second test with same methods:

- Lawson RK(3,3) method
- Lawson RK(3,3) method with Taylor series  $T_3$ ,  $p = 3$
- Lawson RK(3,3) method with Padé approximant  $P_{1,1}$ ,  $p = q1$

**Test Taylor instability:**  $x, y \in [-2, 2]$ ,  $N_x = N_y = 81$ ,  $T_f = 1.52891$ ,  $\Delta t = 0.020944$ .

# Test 2

## 2D rotation test case: test Taylor series instability



**Figure:** Initial condition (left), solution with Lawson RK(3,3)  $T_3$  series (middle) and Lawson RK(3,3)  $P_{1,1}$  approximant (right)

# Outline

- 1 Introduction
- 2 Numerical method and approximation
- 3 Numerical tests
- 4 Numerical results**
- 5 Conclusion

# 4D hybrid fluid-kinetic plasma model

Vlasov-Maxwell  $1dz - 3dv$  hybrid model

- Cold electrons  $\rightarrow$  fluid: cold current  $(j_{c,x}(t, z), j_{c,y}(t, z))$
- Hot electrons  $\rightarrow$  kinetic: density in phase space  $f_h(z, v_x, v_y, v_z)$

$$\partial_t U = LU + N(t, U)$$

with  $U = (j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y, f_h)^\top$  and

$$L = \begin{pmatrix} 0 & -B_0 & 0 & 0 & \Omega_{pe}^2 & 0 & 0 \\ B_0 & 0 & 0 & 0 & 0 & \Omega_{pe}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \partial_z & 0 \\ 0 & 0 & 0 & 0 & -\partial_z & 0 & 0 \\ -1 & 0 & 0 & -\partial_z & 0 & 0 & 0 \\ 0 & -1 & \partial_z & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -v_z \partial_z \end{pmatrix}, \quad N: t, U \mapsto \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \int v_x f_h d\mathbf{v} \\ \int v_y f_h d\mathbf{v} \\ (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_h \end{pmatrix}$$

✗ can't compute  $e^{tL}$  formally

✓  $P_{p,p}(tL)$  Padé approximant!



We compare on this model:

- Strang *classical* method, based on Hamiltonian splitting: order 2
- Lawson RK(4,4) classical (but some terms added in nonlinear term: Maxwell equations): order 4
- Lawson RK(4,4) with Padé  $P_{2,2}$  approximant: order 4
- Lawson DP4(3) with Padé  $P_{2,2}$  approximant: adaptive time step method

## Implementation details:

Problem with 7 variables, Padé approximant implies huge rational functions (with invert of matrix), high order Lawson methods have a lot of coefficients... Code generation ✓

# Main idea of adaptive time step methods (error estimate)

for a generic ODE  $\dot{u} = f(t, u)$ , adaptive time step method need 2 numerical estimations of solution  $u(t^{n+1})$  of different order,  $p$  and  $p + 1$ :

$$u_{[p]}^{n+1} = u(t^{n+1}) + \mathcal{O}(\Delta t^{p+1}), \quad u_{[p+1]}^{n+1} = u(t^{n+1}) + \mathcal{O}(\Delta t^{p+2})$$

Estimate of local error:

$$L_{[p]}^{n+1} = \left| u_{[p+1]}^{n+1} - u_{[p]}^{n+1} \right|$$

**If**  $L_{[p]}^{n+1} > \text{tol}$ : we reject the step and start again from time  $t^n$ . **Else** we accept the step. **In both cases**, the optimal new time step is:

$$\Delta t_{\text{opt}} = \sqrt[p]{\frac{\text{tol}}{L_{[p]}^{n+1}}} \Delta t^n$$

In practice  $u_{[p]}^{n+1}$  is computed from sub-steps of  $u_{[p+1]}^{n+1}$ .



Dormand and Prince 1978, *Celestial mechanics* (for RK method)

# Numerical results

$$N_z \times N_{v_x} \times N_{v_y} \times N_{v_z} = 27 \times 32 \times 32 \times 41$$

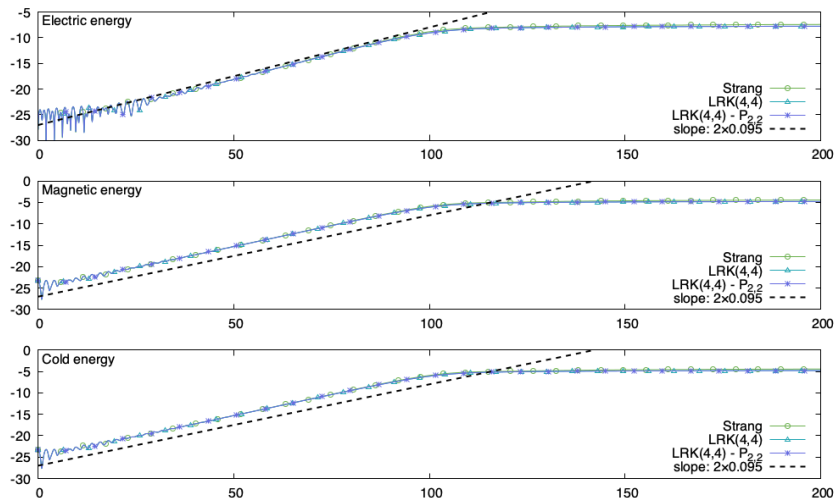
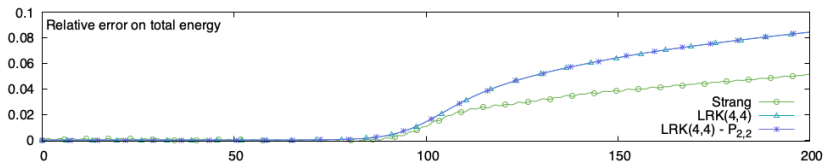


Figure: Energies evolution for Strang, LRK(4,4) and LRK(4,4)- $P_{2,2}$  methods

# Numerical results

$$N_z \times N_{v_x} \times N_{v_y} \times N_{v_z} = 27 \times 32 \times 32 \times 41$$



**Figure:** Relative error on total energy for Strang, LRK(4,4) and LRK(4,4)- $P_{2,2}$  methods

# Numerical results

$$N_z \times N_{v_x} \times N_{v_y} \times N_{v_z} = 27 \times 32 \times 32 \times 41$$

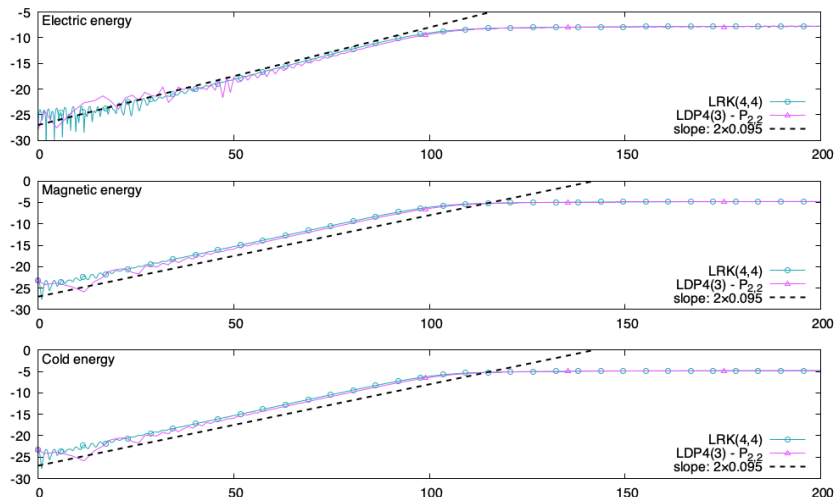


Figure: Energies evolution for LDP4(3)- $P_{2,2}$  method

# Numerical results

$$N_z \times N_{v_x} \times N_{v_y} \times N_{v_z} = 27 \times 32 \times 32 \times 41$$

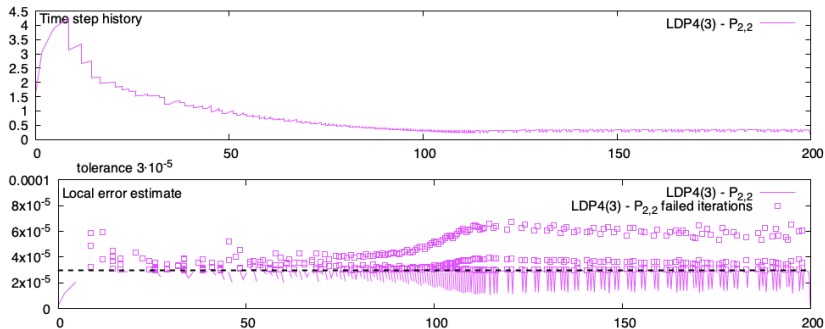


Figure: Time step and estimate of local error evolution for LDP4(3)-P<sub>2,2</sub> method

# Simulation time

time integrator	simulation time
Strang splitting	17 h 09 min 54 s
LRK(4,4)	14 h 06 min 15 s
LRK(4,4) - $P_{2,2}$	13 h 59 min 59 s
LDP4(3) - $P_{2,2}$	04 h 09 min 44 s

**Table:** Simulation time for some simulation, on mesh

$N_z \times N_{v_x} \times N_{v_y} \times N_{v_z} = 27 \times 32 \times 32 \times 41$  and time step  $\Delta t = 0.05$  (initial time step for adaptive time step strategy).

# Outline

- 1 Introduction
- 2 Numerical method and approximation
- 3 Numerical tests
- 4 Numerical results
- 5 Conclusion



# Conclusion

## Conclusion:

- ✗ Numerical cost of Hamiltonian splitting methods (not bad in  $1dx - 1dv$  but very bad in  $1dz - 3dv$ , must be very very very bad in  $3dx - 3dv$  case)
- ✓ Numerical cost of Lawson methods
- ~ Behavior of total energy of Lawson method (but we can use high order method easily)
- ✓ Error of approximation with Padé approximant can be lower than time integrator
- ✓ Very efficient adaptive time step method with computation of  $e^{\tau L}$  with Padé approximant (more thinks can be in linear part)

## Future works

- Add  $\int \mathbf{v} f_h d\mathbf{v}$  in linear part (for  $1dx - 1dv$  Vlasov-Ampère model)

Thank you for your attention