RUHR UNIVERSITÄT BOCHUM RUB

Institute for Theoretical Physics I, Faculty of Physics and Astronomy
Numerics Group, Faculty of Mathematics

Multi-Physics Simulations in Space Plasma Physics

Conceptualization and Implementation with the `muphyII` framework

Simon Lautenbach

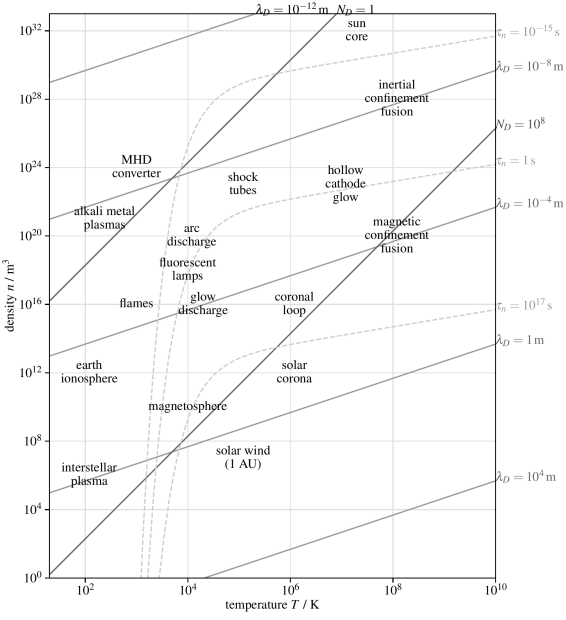
15th International Symposium for Space Simulations *August 3, 2024*

Session Outline

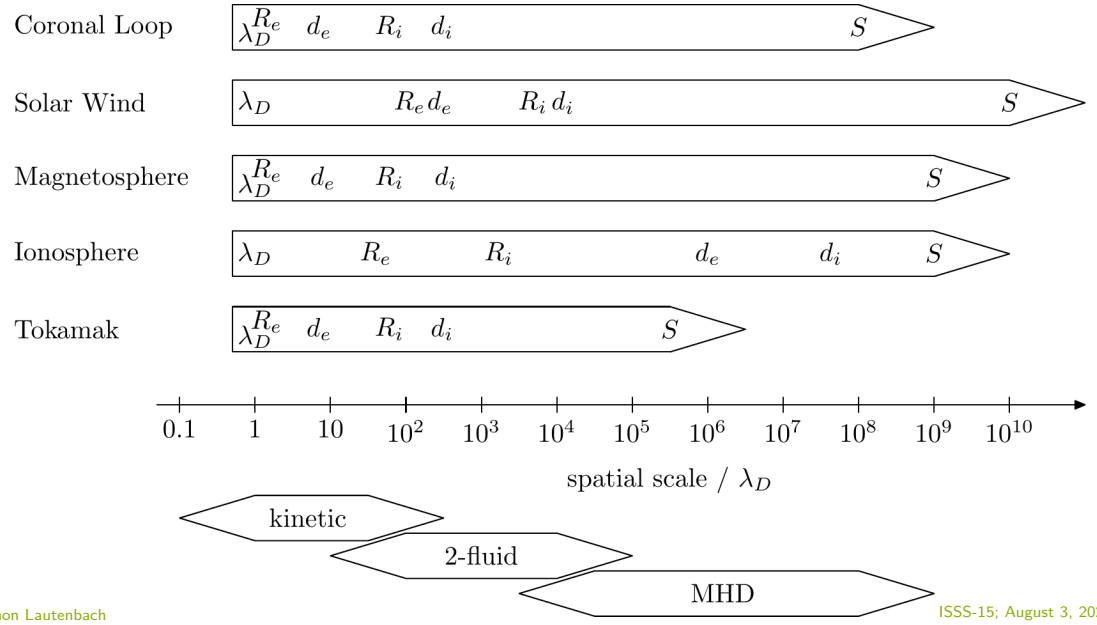
- Multi-Physics Plasma Modelling
  - Motivation
  - Plasma Physics Models
  - Model Implementations
  - Model Coupling
- Hands-on: Multi-Physics Plasma Modelling with the muphy2 framework
  - muphy2 Framework Implementation
  - Simple Examples

Multi-Physics Plasma Modelling

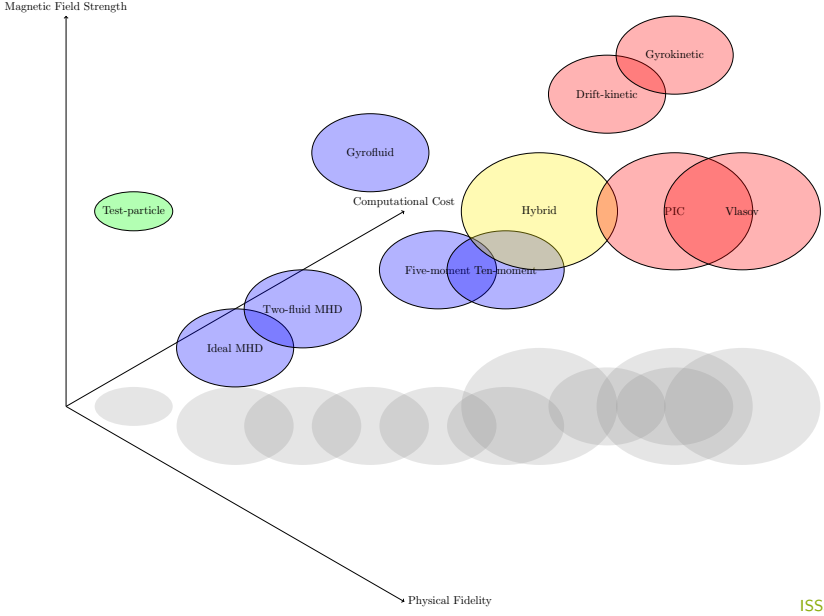
Space Plasma Parameters



Separation of Scales



The Range of Plasma Models

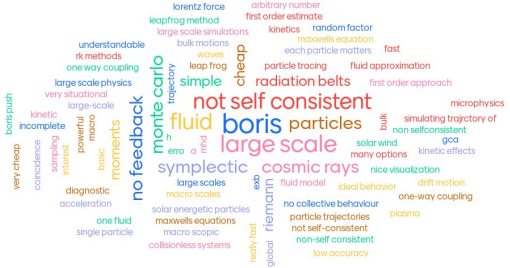


RUHR UNIVERSITÄT BOCHUM RUB

Mentimeter

What springs to your mind about test particle methods?

114 responses



Test-particle Methods

Evolve particle trajectories in prescribed fields

Advantages:

- Extremely computationally efficient
- Useful for studying single-particle dynamics
- Can provide insights into particle behavior in complex field geometries

Disadvantages:

- No self-consistency (fields don't evolve with particles)
- Misses collective effects
- Cannot capture plasma self-organization or instabilities

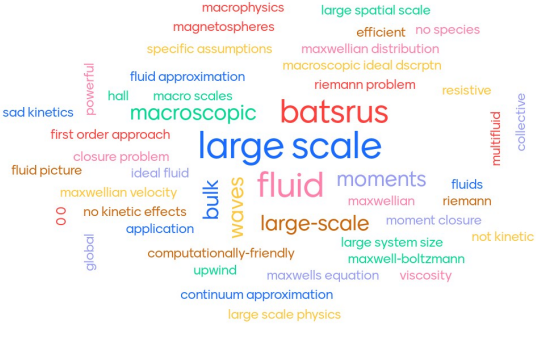
Common Improvements:

- Incorporate simple field solvers for partial self-consistency

Mentimeter

What springs to your mind about MHD?

64 responses



30

Magnetohydrodynamics

Single-fluid model treating plasma as a perfectly conducting fluid

Advantages:

- Computationally efficient for large-scale phenomena
- Captures basic plasma dynamics and equilibria
- Well-suited for studying macroscopic instabilities

Disadvantages:

- Ignores kinetic effects and separate species dynamics
- Assumes infinite conductivity
- Cannot resolve small-scale phenomena or high-frequency waves

Common Improvements:

- Include resistivity for finite conductivity effects
- Add Hall terms for better small-scale physics
- Incorporate two-fluid effects for improved species separation

RUHR UNIVERSITÄT BOCHUM RUB

Mentimeter

What springs to your mind about multi-moment fluid models?

47 responses

26

Multi-moment two-fluid

Evolves density, momentum, and energy for each species

Advantages:

- Captures separate evolution of electron and ion fluids
- Includes basic kinetic effects through heat flux
- More complete than standard two-fluid MHD

Disadvantages:

- Limited description of pressure tensor anisotropy
- Misses higher-order kinetic effects
- Closure problem for heat flux

Common Improvements:

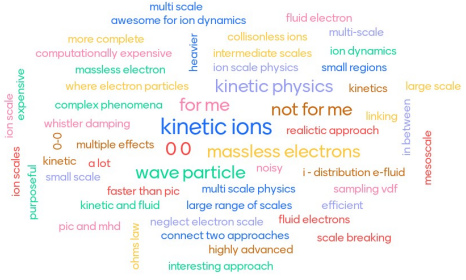
- Add higher-order moments for improved energy transport modeling
- Develop improved closure schemes based on kinetic theory
- Couple with reduced kinetic models for specific particle populations

RUHR UNIVERSITÄT BOCHUM RUB

Mentimeter

What springs to your mind about hybrid models?

62 responses



34

Hybrid

Kinetic ions with fluid electrons

Kinetic ions with fluid electrons using full Maxwell equations

Advantages:

- Captures ion kinetic effects
- More efficient than full kinetic
- Good for ion-scale phenomena

Disadvantages:

- Misses electron kinetic effects
- Challenging to implement efficiently
- Limited by fluid electron description

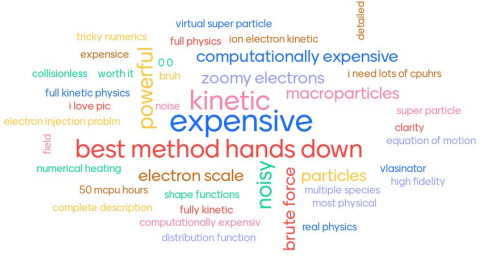
Common Improvements:

- Implement delta-f method for electrons to capture some kinetic effects
- Couple with local fully kinetic solvers in regions of interest
- Develop improved electron fluid models

Mentimeter

What springs to your mind about PIC models?

65 responses



Particle-In-Cell

Fully kinetic particle-in-cell method coupled with Maxwell's equations

Advantages:

- Captures full range of kinetic phenomena
- Intuitive particle representation
- Flexible for complex geometries

Disadvantages:

- Computationally expensive
- Suffers from numerical noise
- Challenging for high dynamic range problems

Common Improvements:

- Implement adaptive particle weighting to reduce noise
- Develop implicit methods for improved efficiency
- Couple with reduced models for multi-scale simulations

RUHR UNIVERSITÄT BOCHUM RUB

Mentimeter

What springs to your mind about continuous Vlasov models?

47 responses

30

Vlasov

Solves Vlasov equation for distribution function coupled with Maxwell's equations

Advantages:

- Fully kinetic with low numerical noise
- Captures full range of plasma phenomena
- Excellent conservation properties

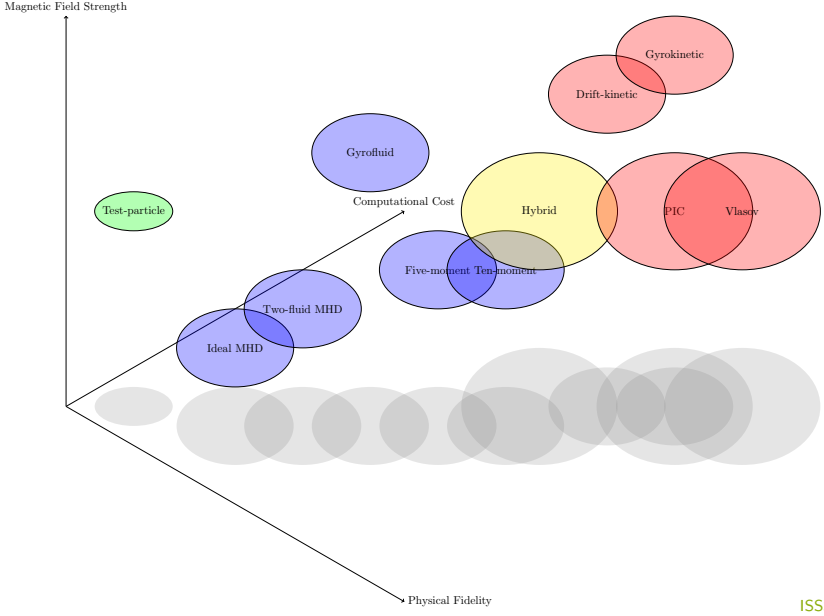
Disadvantages:

- Extremely computationally expensive
- Challenging to implement efficiently
- Suffers from curse of dimensionality

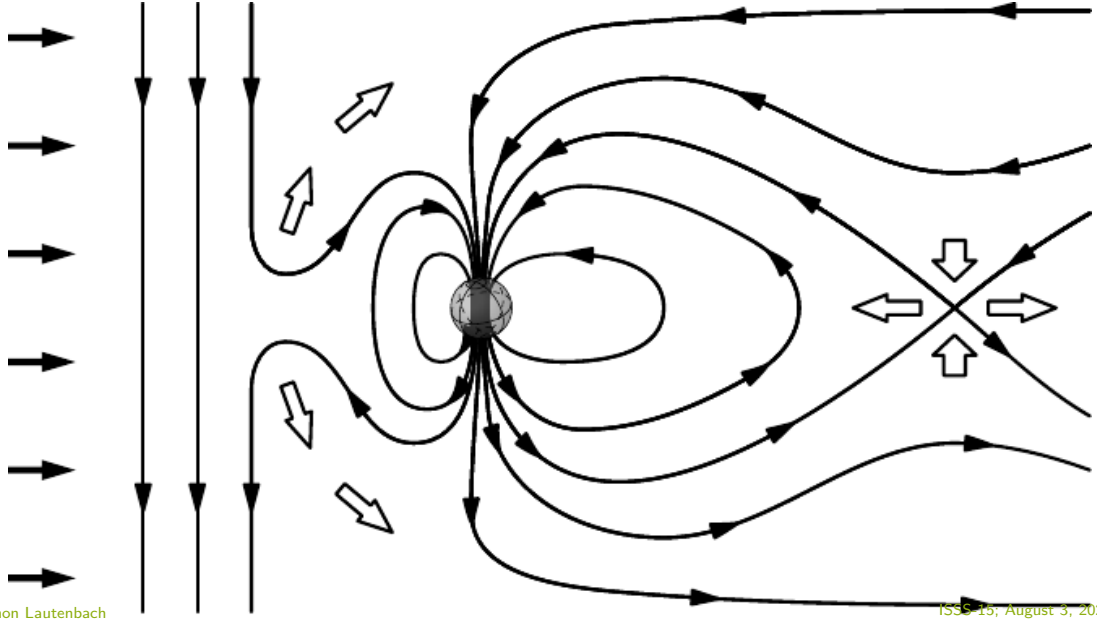
Common Improvements:

- Implement adaptive mesh refinement in phase space
- Develop semi-Lagrangian methods for improved efficiency
- Couple with reduced models for multi-scale simulations

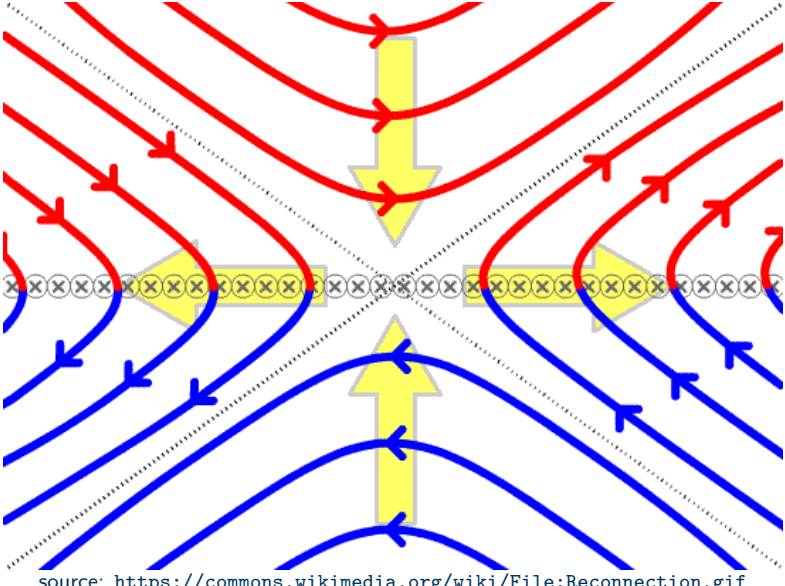
The Range of Plasma Models



Localization of 'Activity'

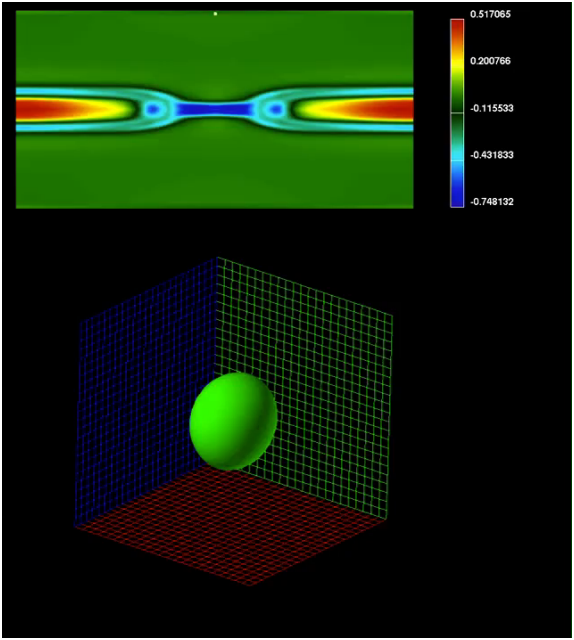


Magnetic Reconnection

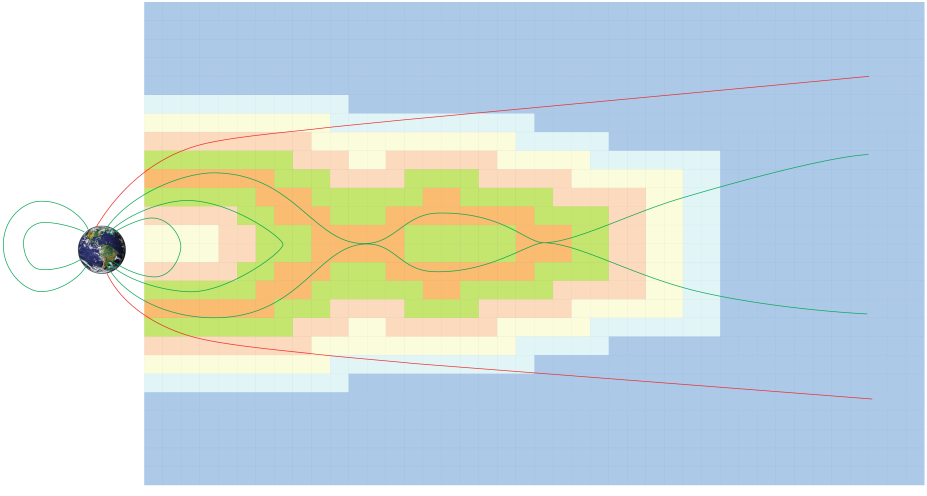


source: https://commons.wikimedia.org/wiki/File:Reconnection.gif
author: ChamouJacoN

Magnetotail Reconnection VDF



Multi-Physics Approach



Kinetic-Fluid Coupling: Histroy

Early efforts to couple kinetic Boltzmann descriptions to fluid models include

- Patrick Le Tallec and François Mallinger (1997). "Coupling Boltzmann and Navier–Stokes Equations by Half Fluxes". In: *Journal of Computational Physics* 136.1, pp. 51–67
- S. Tiwari and A. Klar (1998). "An adaptive domain decomposition procedure for Boltzmann and Euler equations". In: *Journal of Computational and Applied Mathematics* 90.2, pp. 223–237
- Stéphane Dellacherie (2003). "Kinetic-Fluid Coupling in the Field of the Atomic Vapor Laser Isotopic Separation: Numerical Results in the Case of a Monospecies Perfect Gas". In: *AIP Conference Proceedings* 663.1, pp. 947–956
- Pierre Degond, Giacomo Dimarco, and Luc Mieussens (2010). "A multiscale kinetic–fluid solver with dynamic localization of kinetic effects". In: *Journal of Computational Physics* 229.13, pp. 4907–4933
- Thierry Goudon et al. (2013). "Asymptotic-preserving schemes for kinetic-fluid modeling of disperse two-phase flows". In: *Journal of Computational Physics* 246, pp. 145–164
- Sudarshan Tiwari et al. (2013). "Coupled solution of the Boltzmann and Navier–Stokes equations in gas–liquid two phase flow". In: *Computers & Fluids* 71, pp. 283–296

Kinetic-Fluid Coupling: Histroy

In the context of space plasma physics: Static kinetic regions:

- Fixed-region 1D MHD-PIC coupling:
  Tooru Sugiyama and Kanya Kusano (2007). "Multi-scale plasma simulation by the interlocking of magnetohydrodynamic model and particle-in-cell kinetic model". In: *Journal of Computational Physics* 227.2, pp. 1340–1352
- Two-way HMHD-PIC interface coupling using iPIC3D and BATS-R-US:
  Lars K. S. Daldorff et al. (2014). "Two-way coupling of a global Hall magnetohydrodynamics model with a local implicit particle-in-cell model". In: *Journal of Computational Physics* 268, pp. 236–254
- Two-way MHD-PIC interface coupling using iPIC3D and MPI-AMRVAC:
  K. D. Makwana, R. Keppens, and G. Lapenta (2017). "Two-way coupling of magnetohydrodynamic simulations with embedded particle-in-cell simulations". In: *Computer Physics Communications* 221, pp. 81–94
- One-way MHD-iPIC3D coupling in global simulations:
  Raymond J. Walker et al. (2019). "Embedding particle-in-cell simulations in global magnetohydrodynamic simulations of the magnetosphere". In: *Journal of Plasma Physics* 85.1, p. 905850109
  Giovanni Lapenta et al. (2020). "Multiscale MHD-Kinetic PIC Study of Energy Fluxes Caused by Reconnection". In: *Journal of Geophysical Research: Space Physics* 125.3, e2019JA027276

Kinetic-Fluid Coupling: Histroy

In the contect of space plasma physics: Adaptive kinetic regions:

- Adaptive two-way MHD-PIC (AMPS, BATS-R-US)
  Yinsi Shou et al. (2021). "Magnetohydrodynamic with Adaptively Embedded Particle-in-Cell model: MHD-AEPIC". In: *Journal of Computational Physics* 446, p. 110656
- Adaptive two-way MHD-PIC (FLEKS, BATS-R-US) global simulations
  Xiantong Wang, Yuxi Chen, and Gábor Tóth (2022a). "Global Magnetohydrodynamic Magnetosphere Simulation With an Adaptively Embedded Particle-In-Cell Model". In: *Journal of Geophysical Research: Space Physics* 127.8, e2021JA030091
  Xiantong Wang, Yuxi Chen, and Gábor Tóth (2022b). "Simulation of Magnetospheric Sawtooth Oscillations: The Role of Kinetic Reconnection in the Magnetotail". In: *Geophysical Research Letters* 49.15, e2022GL099638

Kinetic-Fluid Coupling: Histroy

RUHR UNIVERSITÄT BOCHUM RUB

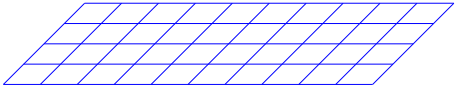
Two-fluid/Maxwell and kinetic coupling :

- One-way bulk coupling two-fluid/Maxwell with PIC:
  Stefano Markidis et al. (2014). "The Fluid-Kinetic Particle-in-Cell method for plasma simulations". In: *Journal of Computational Physics*. Frontiers in Computational Physics 271, pp. 415–429
- Two-way coupling two-fluid/Maxwell with Vlasov M. Rieke, T. Trost, and R. Grauer (2015). "Coupled Vlasov and two-fluid codes on GPUs". In: *Journal of Computational Physics* 283, pp. 436–452
- Adaptive two-way coupling two-fluid/Maxwell with Vlasov
  Simon Lautenbach and Rainer Grauer (2018). "Multiphysics Simulations of Collisionless Plasmas". In: *Frontiers in Physics* 6
- Application to mid-size problems
  F. Allmann-Rahn et al. (2024). "The *muphy*II code: Multiphysics plasma simulation on large HPC systems". In: *Computer Physics Communications* 296, p. 109064

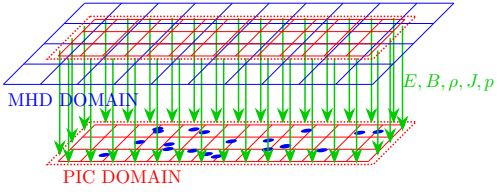
Two-Way MHD-PIC Coupling

RUHR UNIVERSITÄT BOCHUM RUB

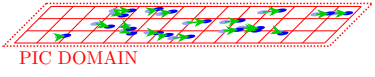
1. ADVANCE MHD SIMULATION $t^n \rightarrow t^{n+1}$



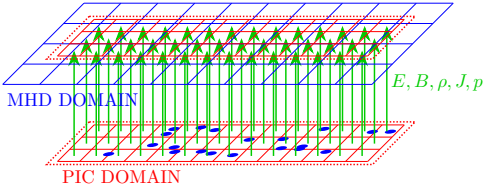
2. PROVIDE BOUNDARY CONDITIONS TO PIC REGION



3. ADVANCE PIC SIMULATION $t^n \rightarrow t^{n+1}$



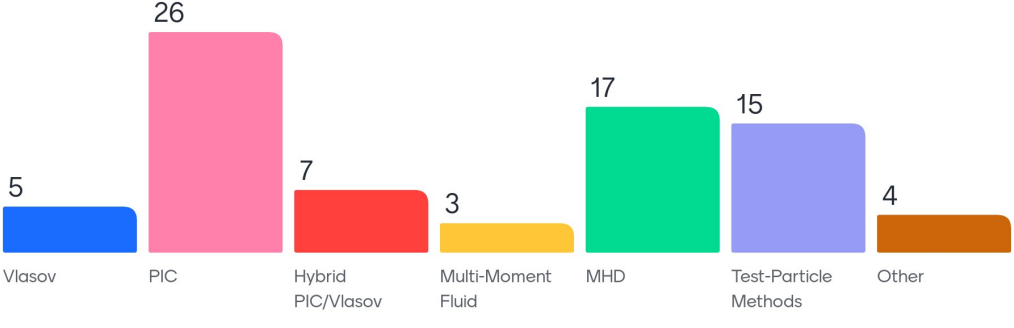
4. CORECT MHD SIMULATION WITH PIC REGION VALUES



Plasma Model Equations

Mentimeter

Which models do you use in your research?



49

Kinetic Model (Vlasov-Maxwell)

The *Vlasov equation* describes the evolution of a distribution $f_s(\mathbf{x}, \mathbf{v}, t)$ of interacting charged particles:

$$\partial_t f_s = -\mathbf{v}_s \cdot \nabla_{\mathbf{x}} f_s - \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_s$$

The electromagnetic fields are governed by *Maxwell's equations*:

$$\partial_t \mathbf{E} = c^2(\nabla \times \mathbf{B} - \mu_0 \mathbf{J}) \qquad \nabla \cdot \mathbf{E} = \frac{\rho_c}{\varepsilon_0}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E} \qquad \nabla \cdot \mathbf{B} = 0$$

Fluid Model ("10-moment" Fluid-Maxwell)

The moments $\int v^{(k)} \langle \cdot \rangle \mathrm{d}v$ of the Vlasov equation constitute the *fluid moment hierarchy*

$$\partial_t \mu_s^{(k)} = -\nabla \cdot \mu_s^{(k+1)} + \frac{q_s}{m_s} \operatorname{sym}\left(\mu_s^{(k-1)} \mathbf{E} + \mu_s^{(k)} \times \mathbf{B} \right)$$

where the electromagnetic fields are governed by *Maxwell's equations*.

The first three moments are given by the following ten equations:

$$\begin{aligned} \partial_t n_s &= -\nabla \cdot (n_s \mathbf{u}_s) \\ m_s \partial_t (n_s \mathbf{u}_s) &= n_s q_s (\mathbf{E} + \mathbf{u}_s \times \mathbf{B}) - \nabla \cdot \mathbb{P}_s \\ \partial_t \mathbb{P}_s &= q_s \left(n_s \operatorname{sym}[\mathbf{u}_s \otimes \mathbf{E}] + \frac{1}{m_s} \operatorname{sym}[\mathbb{P}_s \times \mathbf{B}] \right) - \nabla \cdot \mathbb{Q}_s \end{aligned}$$

To close the hierarchy, an expression for the heat flux divergence $\nabla \cdot \mathbb{Q}_s$ has to be found.

10-moment Fluid Closure

- Hammet-Perkins 1990:

$$\mathbb{Q}_s = -n_{0,s}\frac{2}{\sqrt{\pi}}v_{\mathrm{th},s}\mathcal{H}\mathbb{T}_s$$

- Snyder-Hammett-Dorland 1997:

$$\mathbb{Q}_{\parallel,s} = -n_{0,s}\frac{2}{\sqrt{\pi}}v_{\mathrm{th},\parallel,s}\mathcal{H}\mathbb{T}_{\parallel,s}$$

$$\mathbb{Q}_{\perp,s} = -n_{0,s}\frac{1}{\sqrt{\pi}}v_{\mathrm{th},\parallel,s}\mathcal{H}\left(T_{\perp,s} - T_{\perp,0,s}\left(1 - \frac{T_{\perp,0,s}}{T_{\parallel,0,s}}\right)\frac{|\mathbf{B}|}{B_0}\right)$$

- Sulem-Passot 2015

$$\mathbb{Q}_{\parallel,s} = -n_{0,s}\frac{2}{\sqrt{\pi}}v_{\mathrm{th},\parallel,s}\mathcal{H}\mathbb{T}_{\parallel,s}$$

$$\mathbb{Q}_{\perp,s} = -n_{0,s}\left(1 - \frac{T_{\parallel,0,s}}{T_{\perp,0,s}}\right)\frac{\hat{\mathbf{b}}}{\Omega_e m_e}\cdot\left(\nabla\times\frac{\mathbf{B}}{B_0}\right)$$
$$- n_{0,s}\frac{1}{\sqrt{\pi}}v_{\mathrm{th},\parallel,s}\mathcal{H}\left(T_{\perp,s} - T_{\perp,0,s}\left(1 - \frac{T_{\perp,0,s}}{T_{\parallel,0,s}}\right)\frac{|\mathbf{B}|}{B_0}\right)$$

10-moment Fluid Closure

To approach the Hilbert transform $\mathcal{H} \sim ik_\parallel/|k_\parallel|$, it is (conceptually) decomposed into $ik_\parallel \sim \partial_\parallel$ and $1/|k_\parallel|$. For the latter, approximations include:

- Sharma-Hammett-Quataert-Stone 2006: $1/|k_\parallel| := 1/k_L = \text{const.}$
- Passot-Henri-Laveder-Sulem 2014: $1/|k_\parallel| := \mathcal{F}^{-1}\left(1/\sqrt{\mathbf{k}\cdot\langle\hat{\mathbf{b}}\hat{\mathbf{b}}\rangle\cdot\mathbf{k}}\right)$

Another caveat lies in the generalization of the parallel derivative.
Options include a Laplacian diffusion characteristic:

$$\nabla\cdot\mathbb{Q}_s = -\frac{\chi}{k_{L,s}} n_s v_{\text{th},s}\nabla^2\mathbb{T}_s$$

or a symetrization of the temperature gradient:

$$\nabla\cdot\mathbb{Q}_s = -\frac{\chi}{k_{L,s}} n_s v_{\text{th},s}\nabla\cdot\text{sym}\,\nabla\mathbb{T}_s$$

Hybrid Model (Vlasov-Fluid-Maxwell)

Combine kinetic ion physics with fluid electron modeling and Maxwell's equations:

$$\partial_t f_i = -\mathbf{v}_i \cdot \nabla_{\mathbf{x}} f_i - \frac{q_i}{m_i} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_i$$

$$\partial_t n_e = -\nabla \cdot (n_s \mathbf{u}_e)$$

$$m_e \partial_t (n_e \mathbf{u}_e) = n_e q_e (\mathbf{E} + \mathbf{u}_e \times \mathbf{B}) - \nabla \cdot \mathbb{P}_e$$

$$\partial_t \mathbb{P}_e = q_e \left(n_e \, \mathrm{sym}[\mathbf{u}_e \otimes \mathbf{E}] + \frac{1}{m_e} \mathrm{sym}[\mathbb{P}_e \times \mathbf{B}] \right) - \nabla \cdot \mathbb{Q}_e$$

$$\nabla \cdot \mathbb{Q}_e = -\frac{\chi}{k_{L,e}} n_e v_{\mathrm{th},e} \nabla^2 \mathbb{T}_e$$

$$\partial_t \mathbf{E} = c^2 (\nabla \times \mathbf{B} - \mu_0 \mathbf{J}) \qquad \nabla \cdot \mathbf{E} = \frac{\rho_c}{\varepsilon_0}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E} \qquad \nabla \cdot \mathbf{B} = 0$$

Isotropic Fluid ("5-moment" Fluid-Maxwell)

In the fluid equations, assume isotropy and vanishing heat flux

$$\begin{aligned}
\partial_t n_s &= -\nabla \cdot (n_s \mathbf{u}_s) \\
m_s \partial_t (n_s \mathbf{u}_s) &= n_s q_s (\mathbf{E} + \mathbf{u}_s \times \mathbf{B}) - \nabla \cdot \mathcal{P}_s \\
\partial_t \mathcal{E}_s &= q_s n_s \mathbf{u}_s \cdot \mathbf{E} - \frac{1}{N} \nabla \cdot \left(\mathbf{u}_s ((N+2)\mathcal{E}_s - m_s n_s u_s^2) \right)
\end{aligned}$$

$$\begin{aligned}
\partial_t \mathbf{E} &= c^2 (\nabla \times \mathbf{B} - \mu_0 \mathbf{J}) & \nabla \cdot \mathbf{E} &= \frac{\rho_c}{\varepsilon_0} \\
\partial_t \mathbf{B} &= -\nabla \times \mathbf{E} & \nabla \cdot \mathbf{B} &= 0
\end{aligned}$$

Magnetohydrodynamic Fluid (MHD-Ohm)

$\partial_t \mathbf{E}/c^2 \ll \mu_0 \mathbf{J}$ $\Rightarrow$ Maxwell's equations become

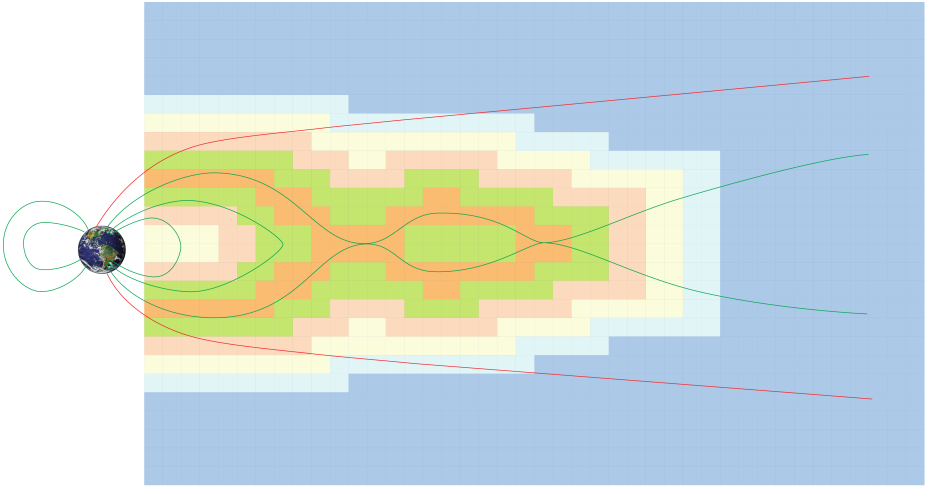
$$\mathbf{J} = \frac{1}{\mu_0} \nabla \times \mathbf{B}$$
$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Need generalized Ohm's law

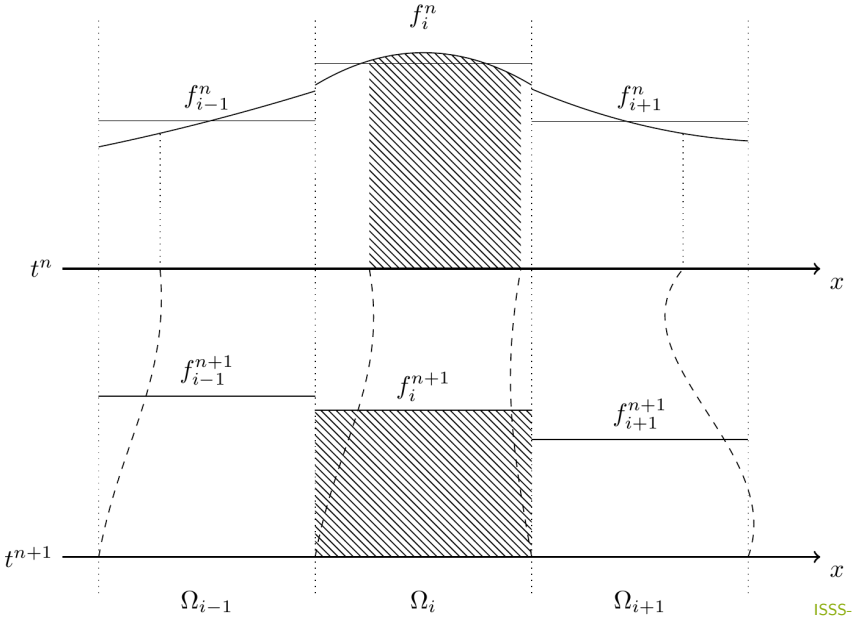
$$\mathbf{E} = -\mathbf{u} \times \mathbf{B} + \frac{1}{q_0 n} \mathbf{J} \times \mathbf{B} - \frac{1}{q_0 n} \nabla \cdot \mathbb{P}_e - \frac{m_e}{q_0} \mathrm{d}_t \mathbf{u}_e$$

Couling Plasma Solvers

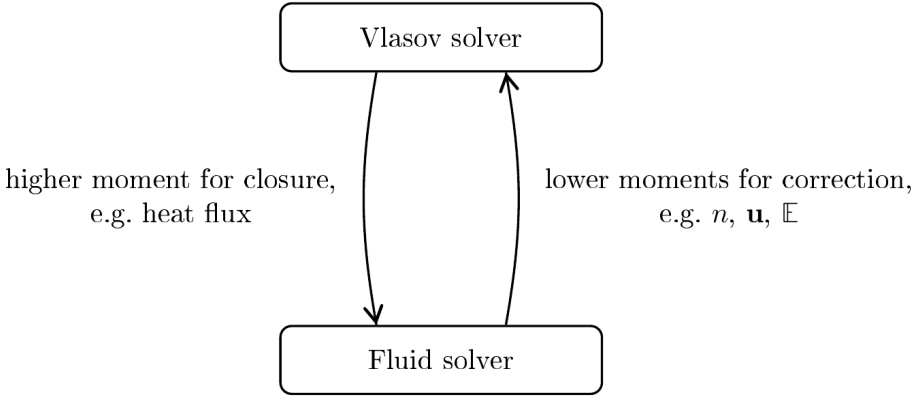
Multi-Physics Approach

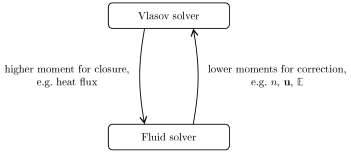


PFC (Vlasov)


f_i^n
f_{i-1}^n
f_{i+1}^n
t^n
x
f_{i-1}^{n+1}
f_i^{n+1}
f_{i+1}^{n+1}
t^{n+1}
x
Ω_{i-1}
Ω_i
Ω_{i+1}

Moment-Matched PFC





Algorithm 1: Time stepping of the moment fitting Vlasov-Maxwell solver as it is implemented

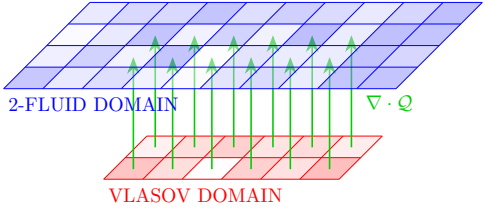
Initialize with a half step of the Maxwell solver
Time Step
  Calculate third moment $\mathcal{Q}^t$ from f^t
  Full Vlasov leapfrog step to advance to f^{t+1}
  Calculate $\mathcal{Q}^{t+1}$ from f^{t+1}
  Interpolate to get $\mathcal{Q}^{t+1/2}$
  Full Runge-Kutta fluid step (input $\mathcal{Q}$ at appropriate times)
  Moment Fitting
    Calculate moments n_V, $\mathbf{u}_\mathrm{V}$, $\mathcal{P}_\mathrm{V}$ from f
    Calculate ten-moment Maxwellian $f_\mathrm{M,V}$ from n_V, $\mathbf{u}_\mathrm{V}$, $\mathcal{P}_\mathrm{V}$
    Multiply the fluid solver's moments n_F, $\mathbf{u}_\mathrm{F}$, $\mathcal{P}_\mathrm{F}$ by $n_\mathrm{V}/n_\mathrm{F}$ to ensure conservation of f
    Calculate ten-moment Maxwellian $f_\mathrm{M,F}$ from the fluid solver's moments
    Exchange ten-moment Maxwellians: $f = f - f_\mathrm{M,V} + f_\mathrm{M,F}$
    Limit f
  end
  Full step of the Maxwell solver

Conservative Vlasov-Fluid Coupling

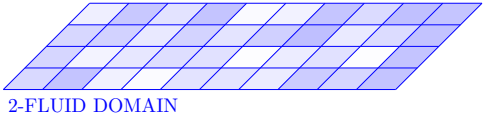
RUHR UNIVERSITÄT BOCHUM RUB

1. ADVANCE VLASOV SIMULATIONS $t^n \to t^{n+1}$

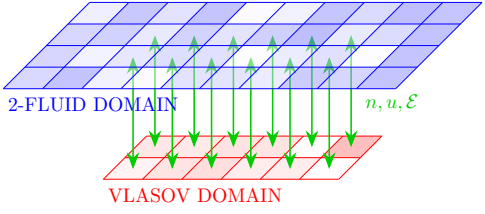
2. PROVIDE KINETIC FLUID CLOSURE FROM VLASOV DISTRIBUTION FUNCTION



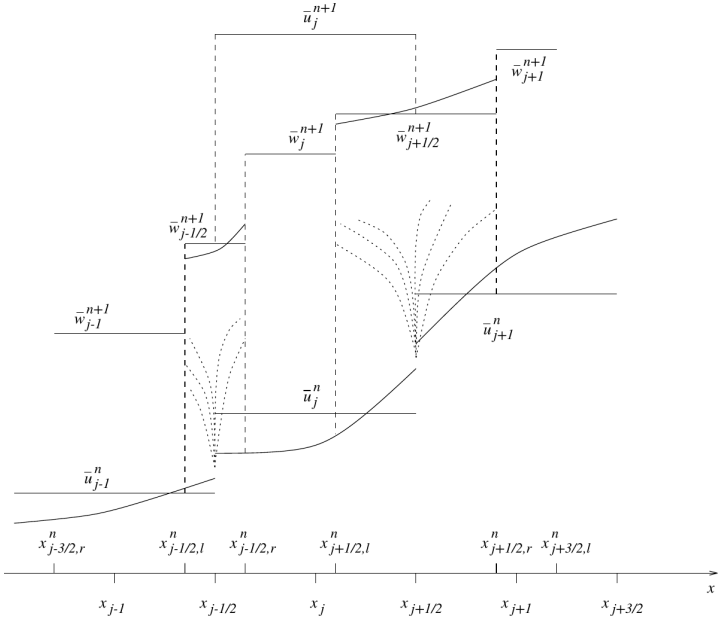
3. ADVANCE 2-FLUID SIMULATIONS $t^n \to t^{n+1}$



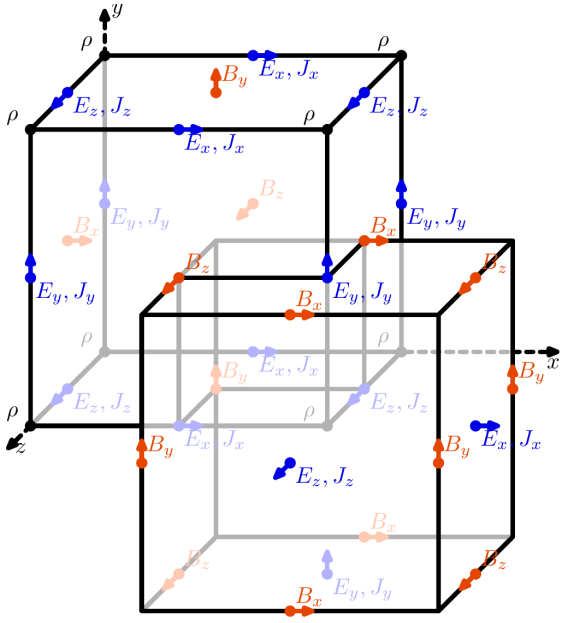
4. MATCH CONSERVATIONAL PROPERTIES



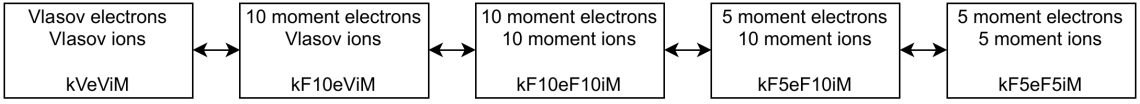
CWENO (Fluid)


$\bar{u}_j^{n+1}$
$\bar{w}_{j+1}^{n+1}$
$\bar{w}_j^{n+1}$
$\bar{w}_{j+1/2}^{n+1}$
$\bar{w}_{j-1/2}^{n+1}$
$\bar{w}_{j-1}^{n+1}$
$\bar{u}_{j+1}^{n}$
$\bar{u}_j^{n}$
$\bar{u}_{j-1}^{n}$
$x_{j-3/2,r}^{n}$
$x_{j-1/2,l}^{n}$
$x_{j-1/2,r}^{n}$
$x_{j+1/2,l}^{n}$
$x_{j+1/2,r}^{n}$
$x_{j+3/2,l}^{n}$
x
x_{j-1}
$x_{j-1/2}$
x_j
$x_{j+1/2}$
x_{j+1}
$x_{j+3/2}$

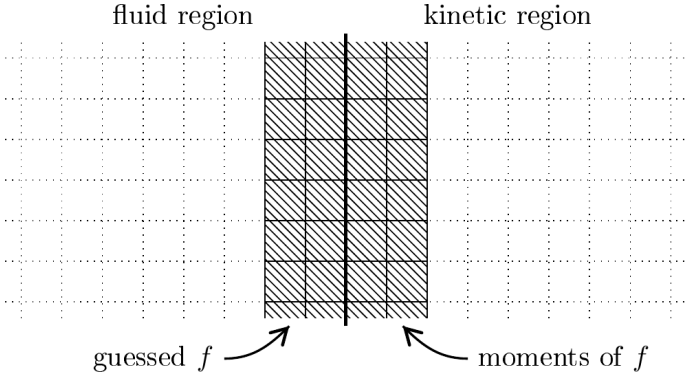
FDTD (Maxwell)



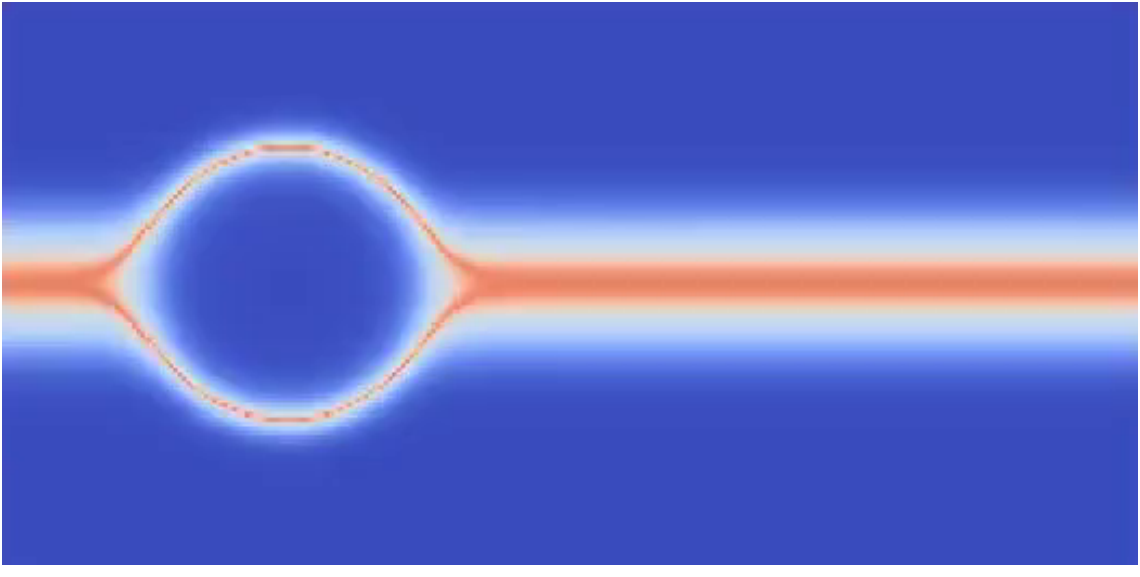
Model Coupling



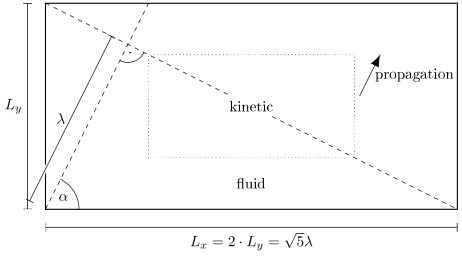
Vlasov-Fluid Interface Coupling

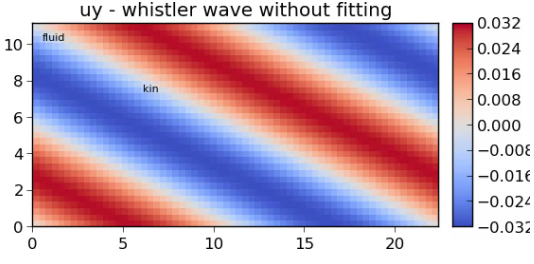


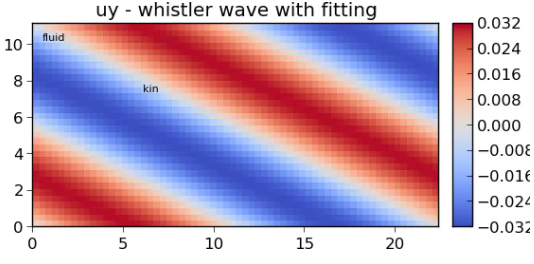
Vlasov-Fluid Interface Coupling: Static Coupling



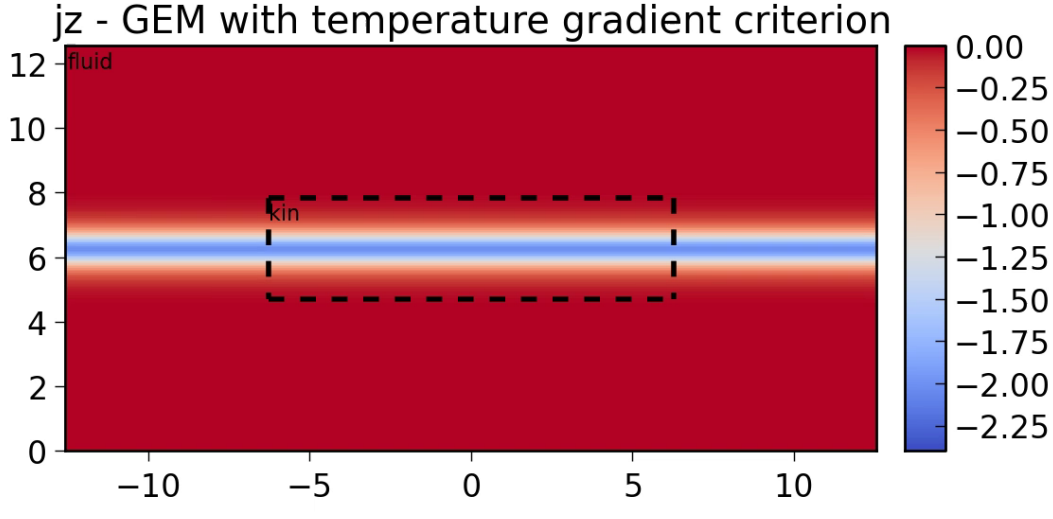
Vlasov-Fluid Interface Coupling: Dissipation Characteristics



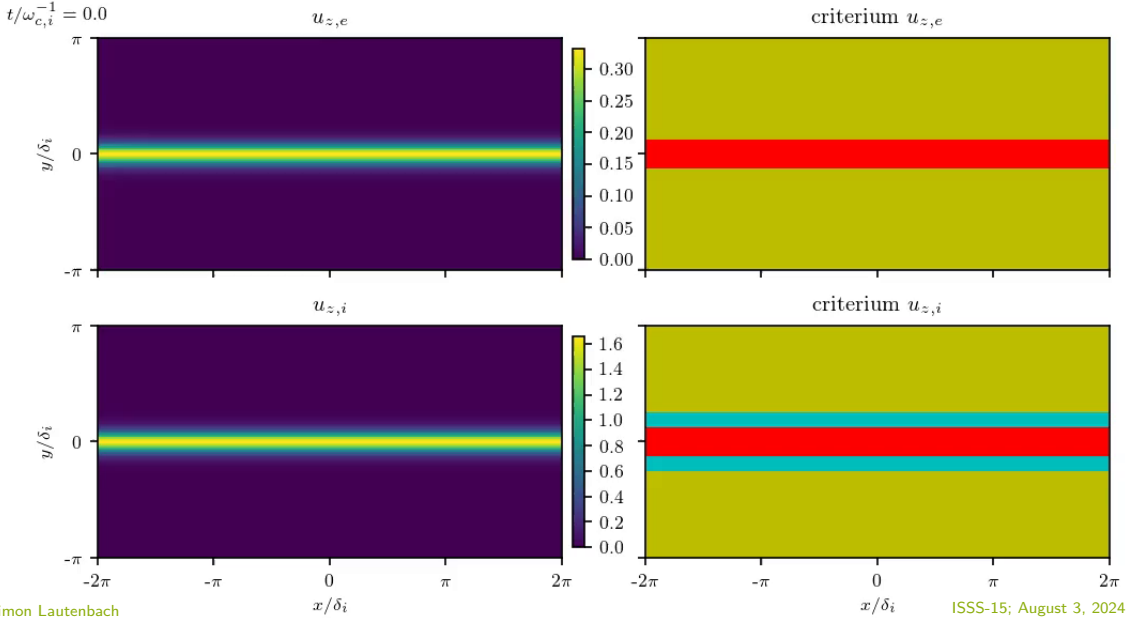




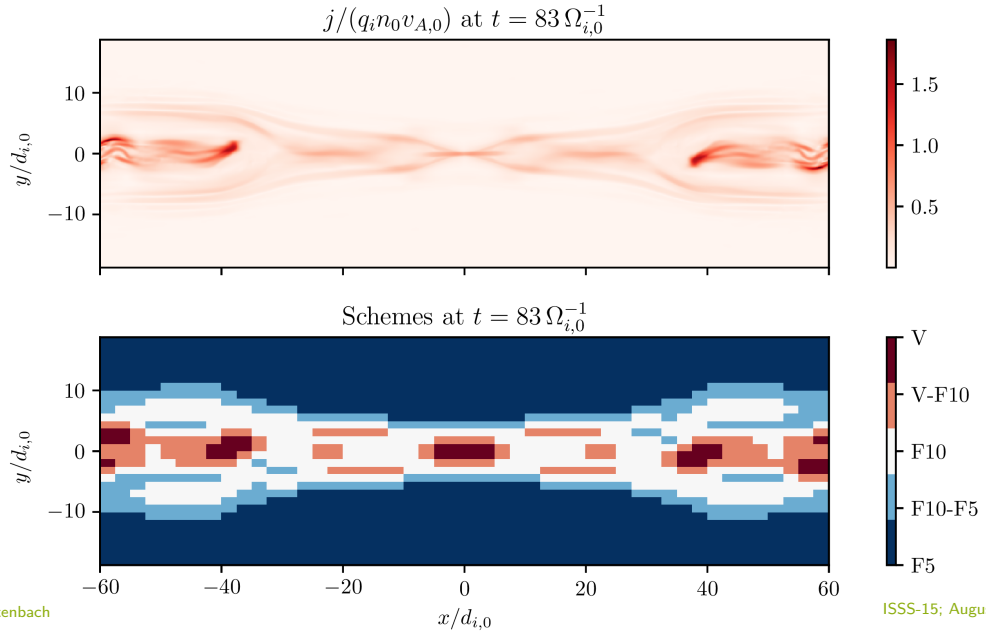
Vlasov-Fluid Interface Coupling: Dynamic Coupling



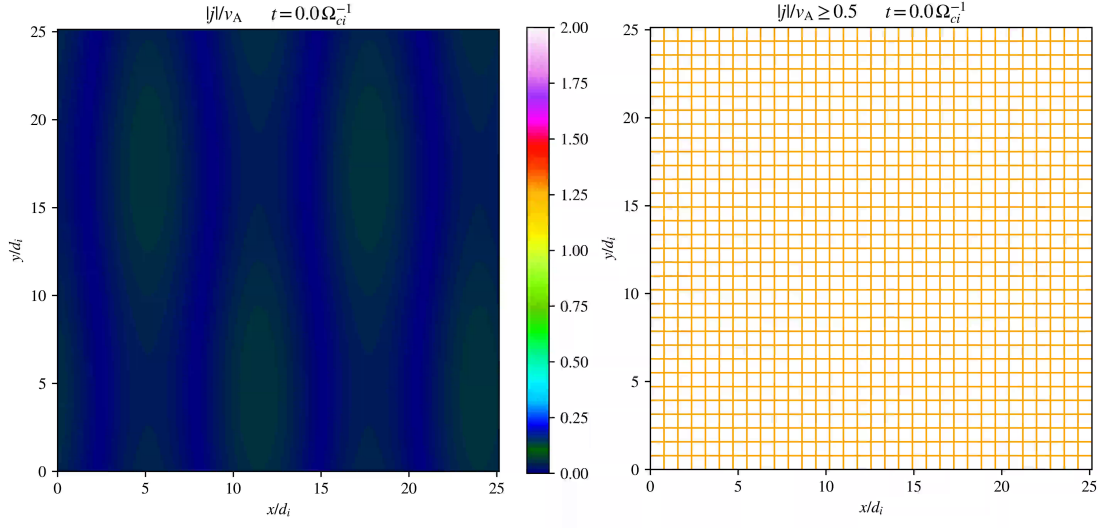
Refinement Criteria



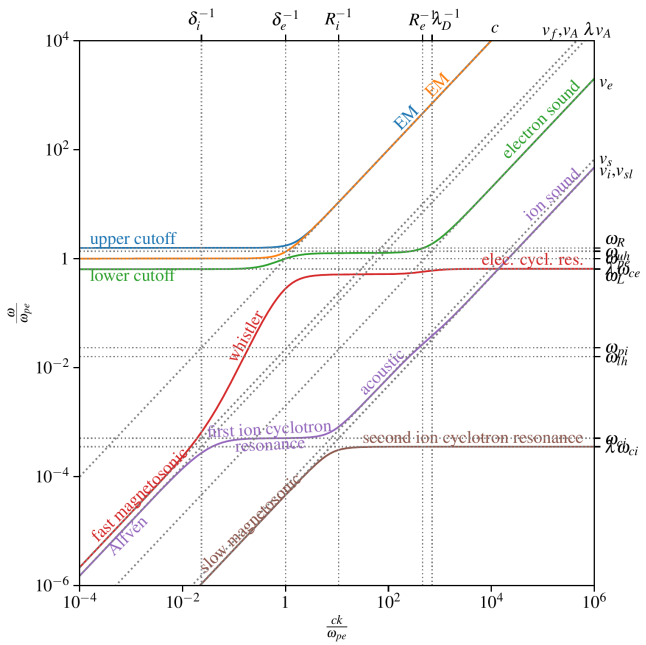
Coupling Complexity



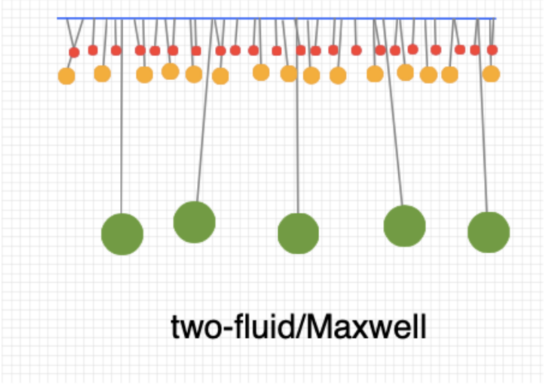
Domain Decomposition in Turbulent Environments

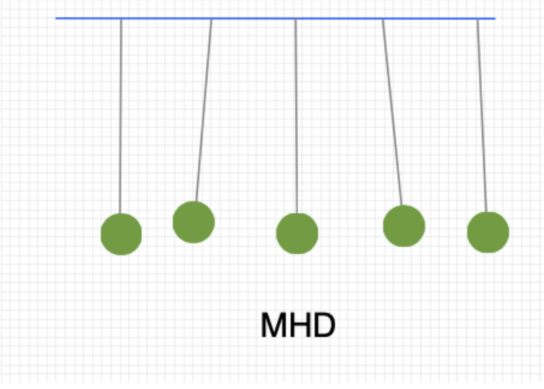


Maxwell-Ohm Coupling: EM Waves



Asymptotic-Preserving Fluid-Maxwell / MHD-Ohm Coupling





Asymptotic-Preserving Fluid-Maxwell / MHD-Ohm Coupling

RUHR UNIVERSITÄT BOCHUM RUB

5-moment two fluid

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot \mathbf{p}_s = 0$$

$$\frac{\partial \mathbf{p}_s}{\partial t} + \nabla \cdot \left(\frac{\mathbf{p}_s \otimes \mathbf{p}_s}{\rho_s} + P_s \overleftrightarrow{\mathbf{I}} \right) = \left(\frac{L}{\delta_p} \right) \left(\frac{Z_s}{A_s} \right) (\rho_s \mathbf{E} + \mathbf{p}_s \times \mathbf{B})$$

$$\frac{\partial e_s}{\partial t} + \nabla \cdot \left((e_s + P_s) \frac{\mathbf{p}_s}{\rho_s} \right) = \left(\frac{L}{\delta_p} \right) \left(\frac{Z_s}{A_s} \right) \mathbf{p}_s \cdot \mathbf{E}$$

two temperature MHD

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{p} = 0$$

$$\frac{\partial \mathbf{p}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{p} \otimes \mathbf{p}}{\rho} - \mathbf{B} \otimes \mathbf{B} + \left(P + \frac{\mathbf{B} \cdot \mathbf{B}}{2} \right) \overleftrightarrow{\mathbf{I}} \right) = 0$$

$$\frac{\partial e_s}{\partial t} + \nabla \cdot \left((e_s + P_s) \frac{\mathbf{p}_s}{\rho_s} \right) = \left(\frac{L}{\delta_p} \right) \left(\frac{Z_s}{A_s} \right) \mathbf{p}_s \cdot \mathbf{E}$$

$$\nabla \times \mathbf{B} = \mathbf{J} \text{ initially} \qquad \frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0 \qquad \nabla \cdot \mathbf{B} = 0$$

Maxwell

$$-\frac{1}{(\omega_p \tau)^2} \left(\frac{L}{\delta_p} \right)^2 \frac{\partial \mathbf{E}}{\partial t} + \nabla \times \mathbf{B} = \left(\frac{L}{\delta_p} \right) \mathbf{j}$$

$$\frac{1}{(\omega_p \tau)^2} \frac{L}{\delta_p} \nabla \cdot \mathbf{E} = \rho_c$$

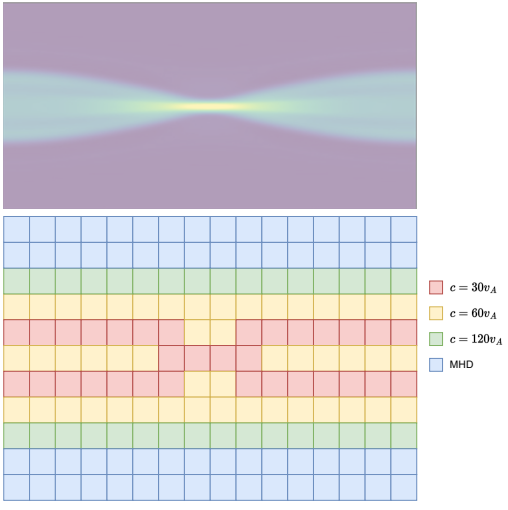
Ohm

$$\rho \mathbf{E} + \mathbf{p} \times \mathbf{B} = \left(\frac{A_i}{Z_i} \right) \left(\left(\frac{\delta_p}{L} \right) \nabla P_e + \mathbf{j} \times \mathbf{B} \right)$$

Transition Strategies

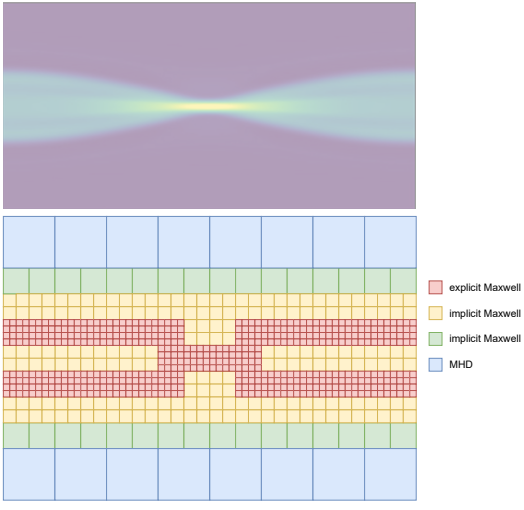
RUHR UNIVERSITÄT BOCHUM RUB

speed of light $c \to \infty$



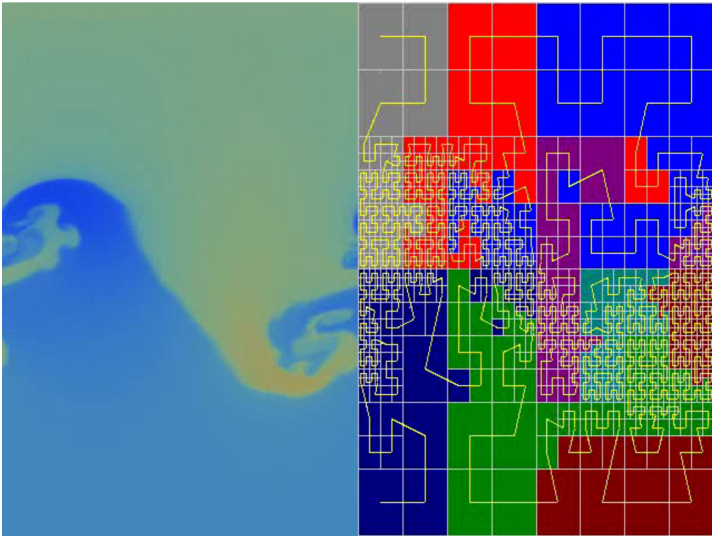
single-scale approach

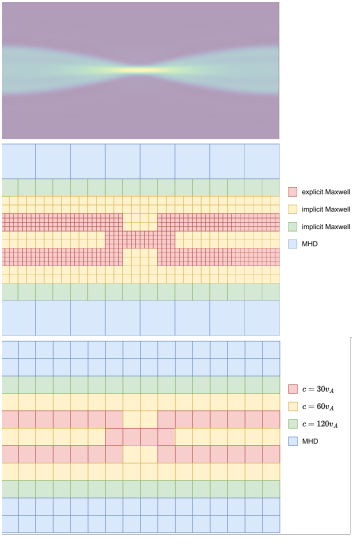
system scale $\to \infty$



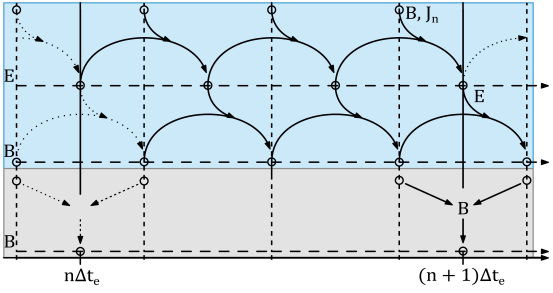
multi-scale approach

Transition Strategies

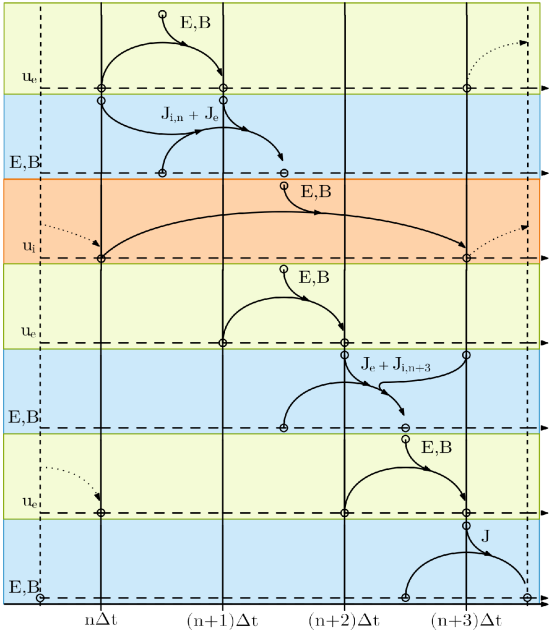




Maxwell Subcycling



Electron Subcycling



RUHR UNIVERSITÄT BOCHUM RUB

Mentimeter

Model coupling takeaway impressions

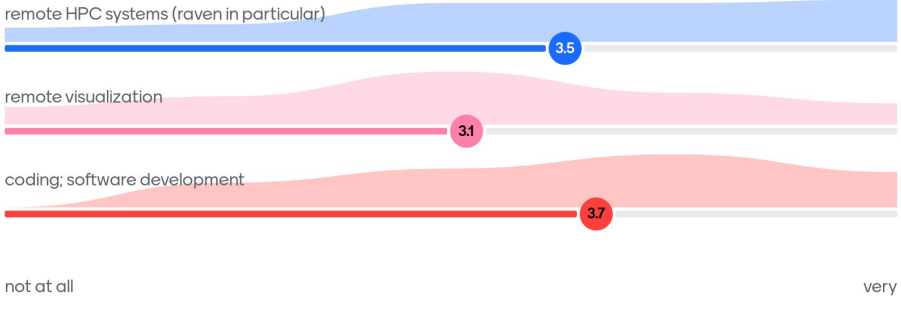
55 responses

33

The muphy2 framework

Mentimeter

How comfortable are you with using ...



43

Block-centric Design

RUHR UNIVERSITÄT BOCHUM RUB

class Block

+ struct Block_id
+ struct Parameter
+ class *Scheme
+ class Boundary

loop_cycle
update_scheme
load_balancing

struct Block_id

+ my_id
+ neighbour_ids[]
+ my_coords
+ my_scheme_id
+ neighbour_scheme_ids{}

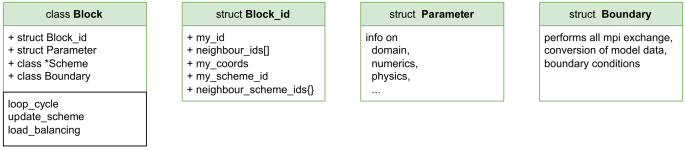
struct Parameter

info on
domain,
numerics,
physics,
...

struct Boundary

performs all mpi exchange,
conversion of model data,
boundary conditions

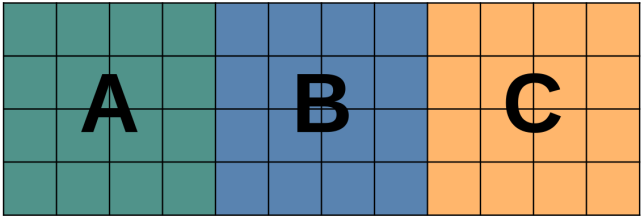
Block-centric Design

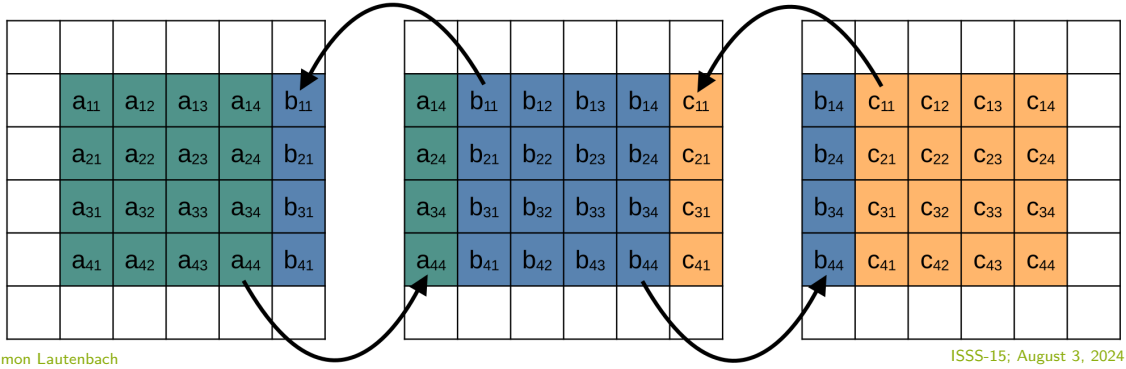



```
1  class Block {
2    public:
3    Block();
4    ~Block();
5
6    void loop_cycle(real &t);
7
8    real get_t_begin();
9    real get_t_end();
10
11   private:
12   void update_scheme();
13   void load_balancing();
14
15   Block_id block_id_;
16   Parameter parameter_;
17   Mpi_boundary boundary_;
18   scheme::Scheme *scheme_;
19
20   MPI_Comm comm3d_;
21
22   ...
23 }
```

```
1  struct Block_id {
2    int my_id,
3    neighbour_ids[6],
4    my_coords[3],
5    my_scheme_id,
6    neighbour_scheme_ids[6],
7    compute_node = -1,
8    selected_device = -1;
9  };
```

Distributed Memory Parallelism

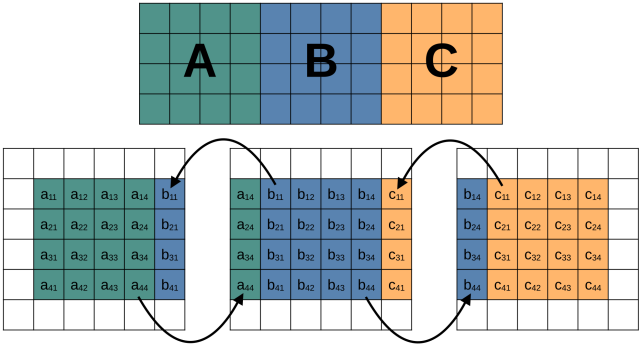




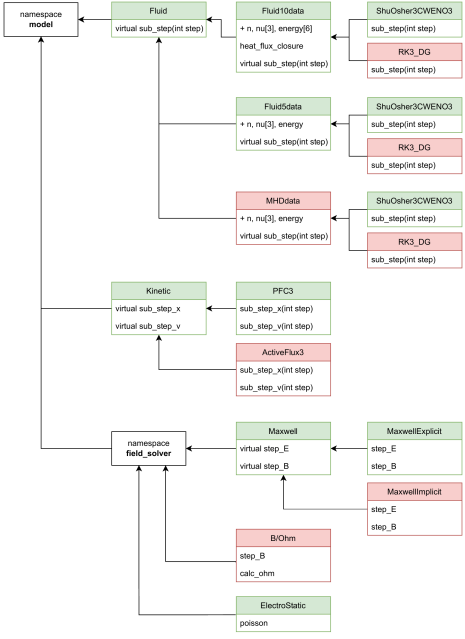
Distributed Memory Parallelism

RUHR UNIVERSITÄT BOCHUM RUB

```
1  class Mpi_boundary {
2    public:
3    Mpi_boundary(const MPI_Comm& comm3d, const int* neighbour_ranks,
4    const int* my_coords, const Parameter& parameter);
5    ~Mpi_boundary();
6
7    real mpi_min(real local_value);
8    int mpi_min(int local_value);
9    real mpi_max(real local_value);
10   int mpi_max(int local_value);
11   real mpi_sum(real local_value);
12   int mpi_sum(int local_value);
13
14   // ... (other methods)
15
16   private:
17   const MPI_Comm& comm3d_;
18   const int* neighbour_ranks_,
19   * my_coords_;
20   const Parameter& p_;
21
22   int tags_[6] = {0,1,2,3,4,5};
23   MPI_Status status_[12];
24   MPI_Request requests_[12];
25 };
```



Model Layout



Example Model: Vlasov

RUHR UNIVERSITÄT BOCHUM RUB

```
1  namespace model {
2
3    class Vlasov : public Model {
4      public:
5      Vlasov(const Parameter& parameter, const Species& species);
6      ~Vlasov();
7
8      int get_model_id();
9      real get_max_dt();
10
11     // wrappers for fortran functions
12     void step_x(const real dt);
13     void step_y(const real dt);
14     void step_z(const real dt);
15     void step_vx(const real dt, const real& E, const real& B, bool include_boundary_cells=false);
16     void step_vy(const real dt, const real& E, const real& B, bool include_boundary_cells=false);
17     void step_vz(const real dt, const real& E, const real& B, bool include_boundary_cells=false);
18
19     real* f;
20
21     private:
22     const Parameter& parameter_;
23     const Species& species_;
24   };
25
26 } // namespace model
```

Language Paradigm

RUHR UNIVERSITÄT BOCHUM RUB

C++

- high-level
- general purpose language
- great for complex and dynamic data structures
- gives numerous chances to ruin performance
- it takes a lot of programming experience to write "good" C++ programs

→ **use for framework**

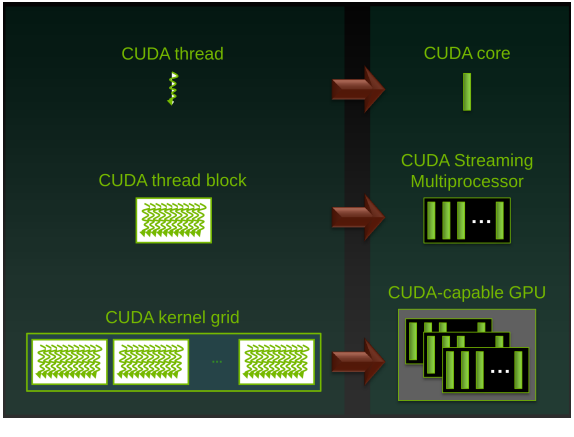
Fortran90

- low-level
- domain-specific language
- excels at array processing
- lacks flexibility
- it takes only a little programming experience to write "fast" Fortran90 programs

→ **use for numeric kernels**

OpenACC

More Science, Less Programming



Shared Memory Parallelism

RUHR UNIVERSITÄT BOCHUM RUB

```
1   subroutine step_x_vlasov_pfc(f,dimX,dimV,BD,dx,dv,v_b,dt,dimensionality_x)
2     ...
3     ! cell update
4
5     !$acc data present(f)
6     !$acc parallel
7     !$acc loop gang collapse(3)
8     do vz = 1,dimV(3)
9       do vy = 1,dimV(2)
10        do vx = 1,dimV(1)
11          !$acc loop vector collapse(2)
12          do z = -BDZ,dimX(3)+BDZ
13            do y = -BDY,dimX(2)+BDY
14              ...
15
16              !$acc loop seq
17              do x = 0,dimX(1)+1
18                ...
19              enddo
20              f(dimX(1),y,z,vx,vy,vz) = f(dimX(1),y,z,vx,vy,vz) + flux_minus_1 - flux_current
21            enddo
22          enddo
23        enddo
24      enddo
25    enddo
26    !$acc end parallel
27    !$acc end data
28  end subroutine step_x_vlasov_pfc
```

Combining Models: Example Scheme

RUHR UNIVERSITÄT BOCHUM RUB

```
1   namespace scheme {
2     // ten moments fluid electrons, Vlasov fluid ions, Maxwell
3
4     class F10eViM : public Scheme {
5       public:
6       F10eViM(const Block_id &block_id, Mpi_boundary &mpi_boundary, const Parameter &parameter);
7       ~F10eViM();
8
9       int get_scheme_id();
10      void init();
11      void step();
12      real get_dt(int species = 0);
13      void set_dt(real dt) { dt_ = dt; };
14      void output(real t, int output_number, bool output_vtk);
15      int evaluate_criterion();
16      void replace_dealloc_old_models(model::Model* plasma_model[],
17      model::Model* electromagnetic_model);
18      void calc_alloc_converted_model(model::Model*& model_out,
19      int species, int target_scheme_id);
20      void send_model_data(real* out_buffer, int receiver_id);
21      void receive_model_data(real* in_buffer, int sender_id);
22      int get_data_size();
23
24      // for electron subcycling
25      int get_substeps() { return electron_substeps_; };
26      int get_substep_counter() { return electron_substep_counter_; };
27      void set_substeps(int substeps) { electron_substeps_ = substeps; };
28      void set_substep_counter(int substep_counter) { electron_substep_counter_ = substep_counter; };
29
30      private:
31      void heat_flux_closure(model::Fluid10 *fluid, const real *ten_moments_tmp, real *source);
32      void exchange_plasma_solvers(real* data_e, real* data_i, bool runge_kutta_only=false);
```

RUHR UNIVERSITÄT BOCHUM RUB

Directory Structure:

```
.
|-- bin
|   |-- machinefiles
|   `-- Makefile
|-- framework
|-- physics
|   `-- plasma
|       |-- converters
|       |-- criteria
|       |-- models
|       |-- output
|       |-- schemes
|       |-- scripts
|       `-- setup
|           `-- initial_conditions
`-- README.md
```

Key Components:

- `bin`: Contains machine-specific files and Makefile
- `framework`: Core framework code
- `physics`: Physics-specific modules
- `README.md`: Documentation

Problem Setup


```
1       subroutine setup_whistler_wave(n_e,n_i,u_e,u_i,E,B,dimX,BD,dx,xb_loc)
2         use definitions
3         implicit none
4
5         integer, intent(in) :: dimX(3), BD(3)
6         real(kind=PRC), intent(in) :: dx(3), xb_loc(3)
7         real(kind=PRC), intent(inout),
    ↪   dimension(-BD(1):dimX(1)+BD(1),-BD(2):dimX(2)+BD(2),-BD(3):dimX(3)+BD(3)) :: n_e, n_i
8         real(kind=PRC), intent(inout),
    ↪   dimension(-BD(1):dimX(1)+BD(1),-BD(2):dimX(2)+BD(2),-BD(3):dimX(3)+BD(3),3) :: u_e, u_i, E, B
9
10        integer :: x, y, z
11        real(kind=PRC) :: xVal, yVal, zVal
12        real(kind=PRC) :: kw, cw, db, du, cosa, sina, beta
13
14        kw = 0.62831853_PRC
15        cw = 1.169_PRC
16        db = 0.05_PRC
17        du = 0.05845_PRC
18        cosa = 0.44721_PRC
19        sina = 0.89443_PRC
20        beta = 0.9978_PRC
21
22        n_e = 1._PRC
23        n_i = 1._PRC
24
25        do z = -BD(3),dimX(3)+BD(3)
26          zVal = xb_loc(3) + (z+0.5_PRC)*dx(3)
27          do y = -BD(2),dimX(2)+BD(2)
28            yVal = xb_loc(2) + (y+0.5_PRC)*dx(2)
29            do x = -BD(1),dimX(1)+BD(1)
30              xVal = xb_loc(1) + (x+0.5_PRC)*dx(1)
```

Initial Conditions Configuration File

RUHR UNIVERSITÄT BOCHUM RUB

```
1   void Parameter::init() {
2     // GEM
3     default_scheme = kF10eF10iM;
4     output_directory = "YOUR_OUTPUT_PATH/";
5     noutputs_vtk = 40;
6     noutputs_csv = 40;
7
8     // time
9     t_end = 40.;
10
11    // physical parameters
12    nspecies = 2;
13    species = new Species[nspecies];
14    species[kElectron].q = -1.;
15    species[kIon].q = 1.;
16    species[kElectron].m = 0.04;
17    species[kIon].m = 1.;
18    species[kElectron].T0 = 1. / 12.;
19    species[kIon].T0 = 5. / 12.;
20    c0 = 20.;
21    mu0 = 1.;
22    eps0 = 1./(c0*c0);
23
24    // setup specific
25    setup = "harris_sheet";
26    setup_var["n_bg"] = 0.2;
27    setup_var["T_bg_e"] = species[kElectron].T0;
28    setup_var["T_bg_i"] = species[kIon].T0;
29    setup_var_bool["drifting_background"] = true;
30    setup_var_bool["sine_perturbation"] = true;
31    setup_var["lambda"] = 0.5;
32    setup_var["psi"] = 0.1;
```

Hands-on