

PISAM

Plasma Interacting Super Atoms and Molecules

*- A direct approach for
modeling neutrals in
fusion plasma, coupled
to the 2D edge turbulence
code HESEL*

Presenting a review of my master's thesis

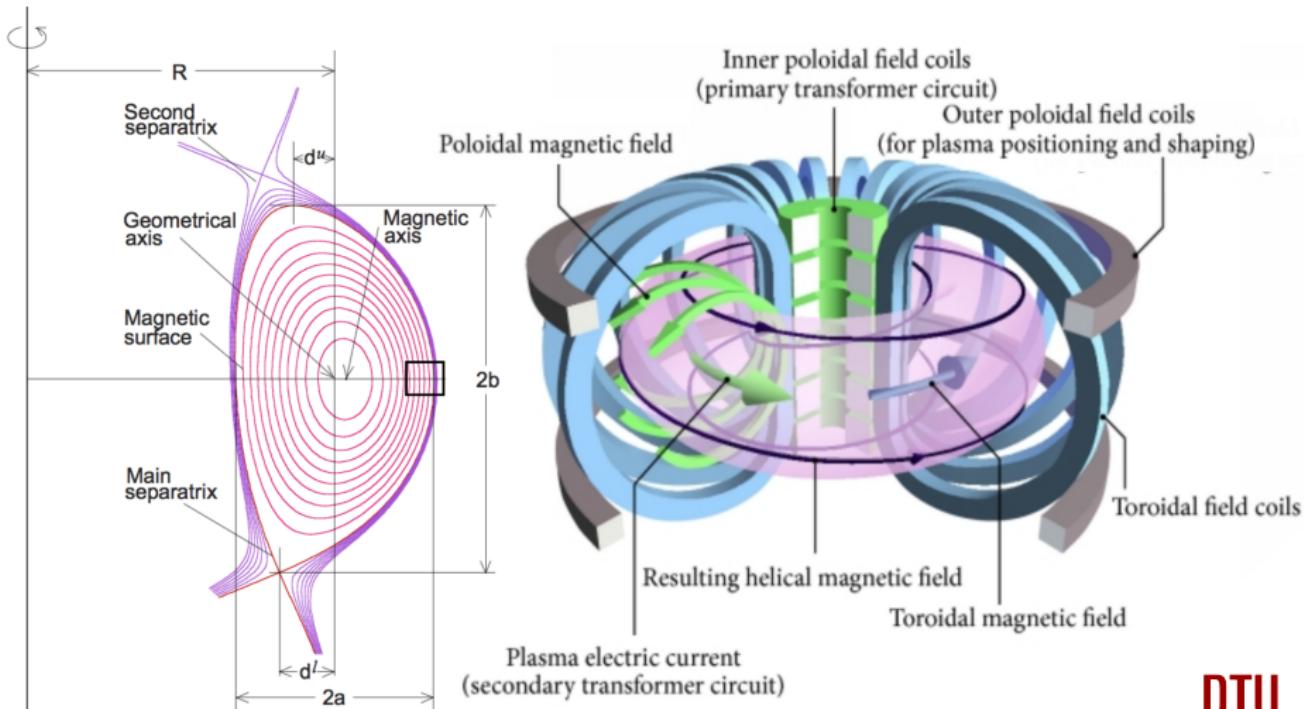
The code and thesis is available at:

<https://github.com/kristofferkvist/PISAM-HESEL>

Contact: kristofferkvist1994@gmail.com



Tokamak



Motivations for Developing PISAM-HESEL.

Adding Neutrals to HESEL

- Fueling efficiency
- Influence of neutrals on the energy containment time:
 - CX-mediated energy transport
 - Altered conduction losses
 - Effects on anomalous transport

Direct Kinetic Approach

- Kinetic approach for comparison with fluid model
- Intuitive, simplistic and robust

nHESEL Equations

$$\frac{d}{dt}n + n\mathcal{K}(\phi) - \mathcal{K}(p_e) = \Lambda_n + \Sigma_n$$

$$\nabla \cdot \left(\frac{d^0}{dt} \nabla_{\perp} \phi^* \right) - \mathcal{K}(p_e + p_i) = \Lambda_w + \Sigma_w$$

$$\frac{3}{2} \frac{d}{dt} p_e + \frac{5}{2} p_e \mathcal{K}(\phi) - \frac{5}{2} \mathcal{K} \left(\frac{p_e^2}{n} \right) = \Lambda_{p_e} + \Sigma_{pe}$$

$$\frac{3}{2} \frac{d}{dt} p_i + \frac{5}{2} p_i \mathcal{K}(\phi) + \frac{5}{2} \mathcal{K} \left(\frac{p_i^2}{n} \right) - p_i \mathcal{K}(p_e + p_i) = \Lambda_{p_i} + \Sigma_{pi}$$

Where $\mathcal{K}(\psi) = - \left(\nabla \psi \times \hat{\mathbf{b}} \right) \cdot \nabla \left(\frac{1}{B} \right) = - \frac{1}{B_0(R+a)} \frac{\partial}{\partial y} \psi.$

Neutral Terms

$$\Sigma_n = I^{(1)},$$

$$\Sigma_w = \nabla \cdot (n \mathbf{u}_{I,i}),$$

$$\Sigma_{pe} = I_e^{(\frac{1}{2}mv^2)}$$

$$\Sigma_{pi} = I_i^{(\frac{1}{2}mv^2)} - \mathbf{u}_{i\perp,0} \cdot I_i^{(mv)}$$

$$+ \frac{1}{2} u_{i\perp,0}^2 I^{(1)} + p_i \nabla \cdot (n \mathbf{u}_{I,i})$$

$$- p_i \nabla \cdot (\mathbf{u}_{I,i}) - \frac{3}{2} \nabla \cdot (p_i \mathbf{u}_{I,i}),$$

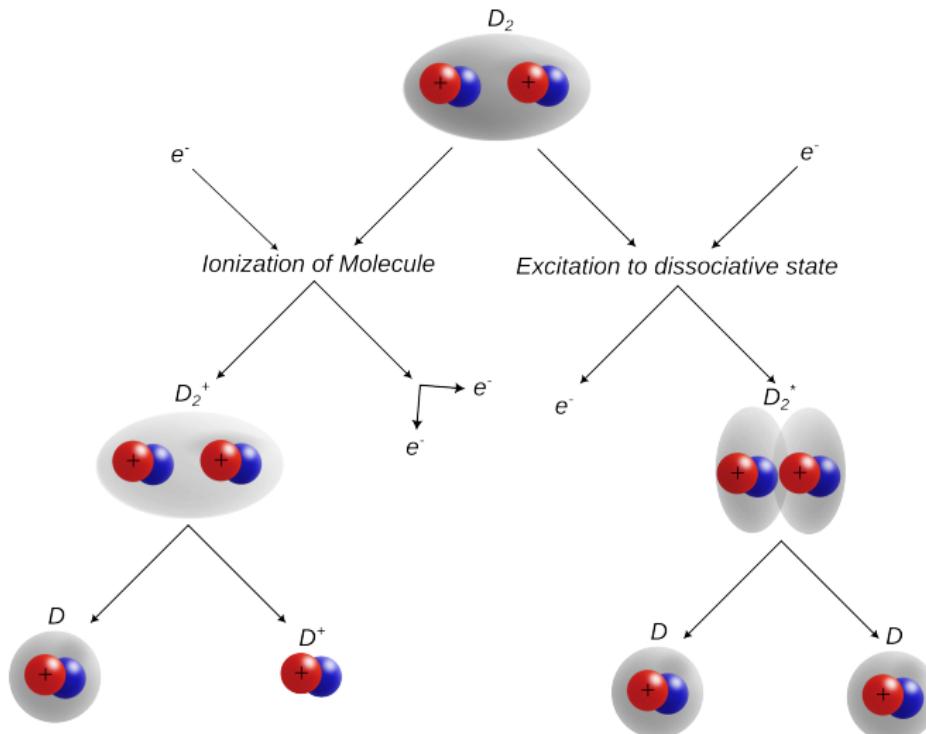
where:

$$I_s^\phi = \int \phi \mathcal{I}_s d\mathbf{v}_s,$$

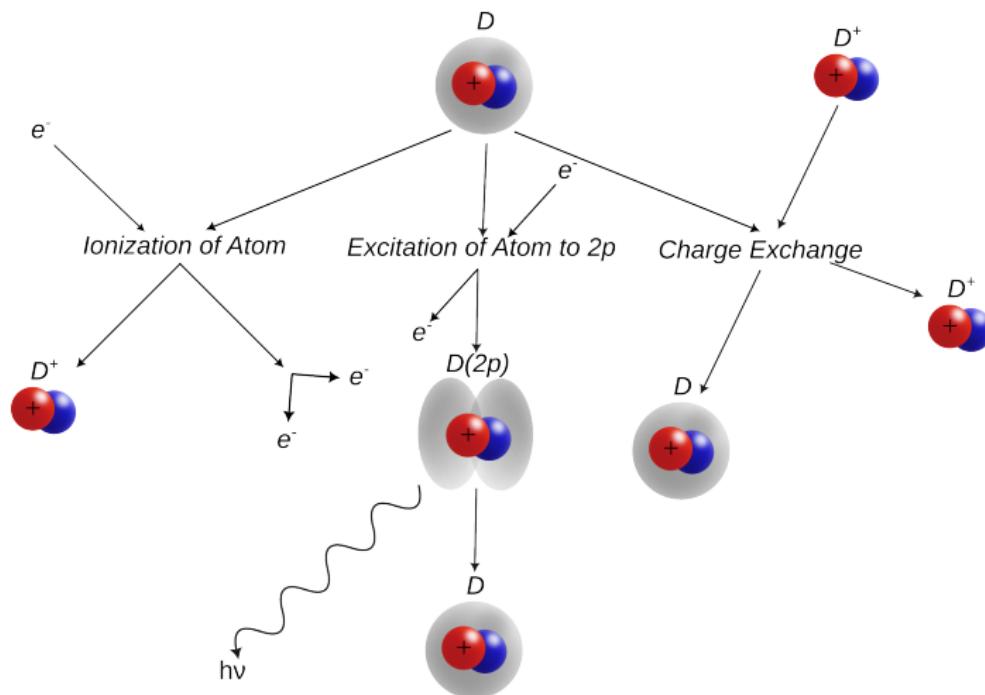
and:

$$\mathbf{u}_{I,i} = -\frac{\hat{\mathbf{b}} \times I_i^{(m_iv)}}{n} + \frac{\hat{\mathbf{b}} \times \mathbf{u} I^{(1)}}{n}.$$

Interactions between Plasma and Deuterium Molecules



Interactions between Plasma and Deuterium Atoms



Time Step of PISAM

- Inject n_{new} supermolecules with weight w_{init} according to

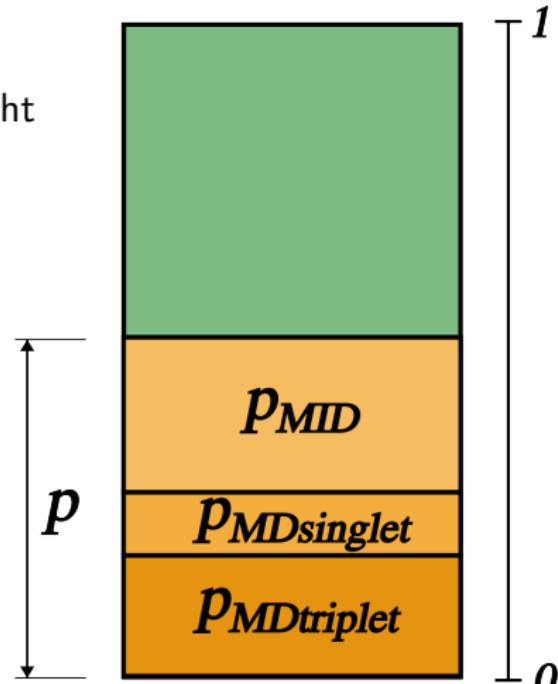
$$F_s L_{pol} L_{tor} \Delta t = n_{new} w_{init}$$

- Calculate the collision frequency of each individual reaction for each superparticle

$$\Gamma_i = n \langle \sigma_i v \rangle, \quad \Gamma = \sum_i \Gamma_i$$

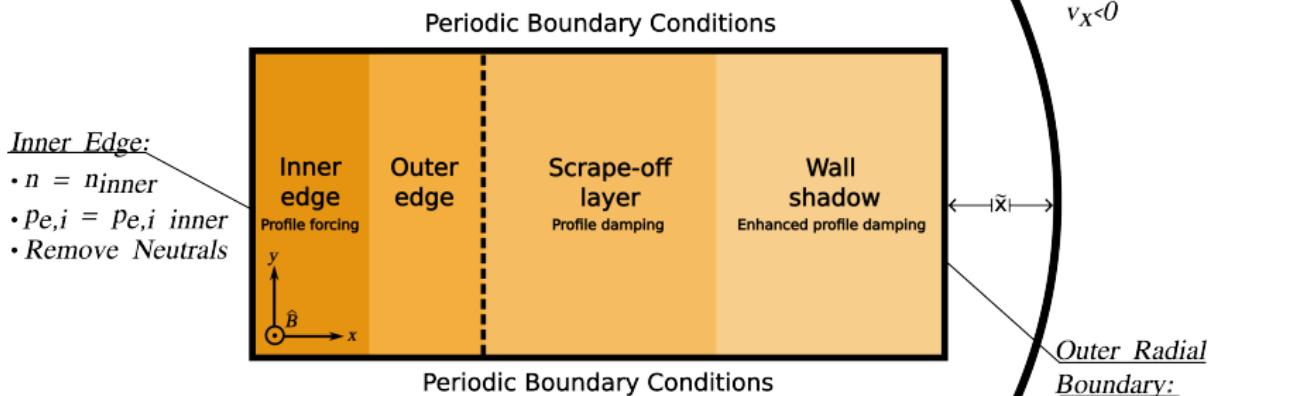
The collision probabilities are thus

$$P_i = \frac{\Gamma_i}{\Gamma} (1 - e^{-\Gamma \Delta t})$$



- Sample and resolve collisions
- Move the particles, $\mathbf{r}_i^{t+\Delta t} = \mathbf{r}_i^t + \Delta t \mathbf{v}_i$

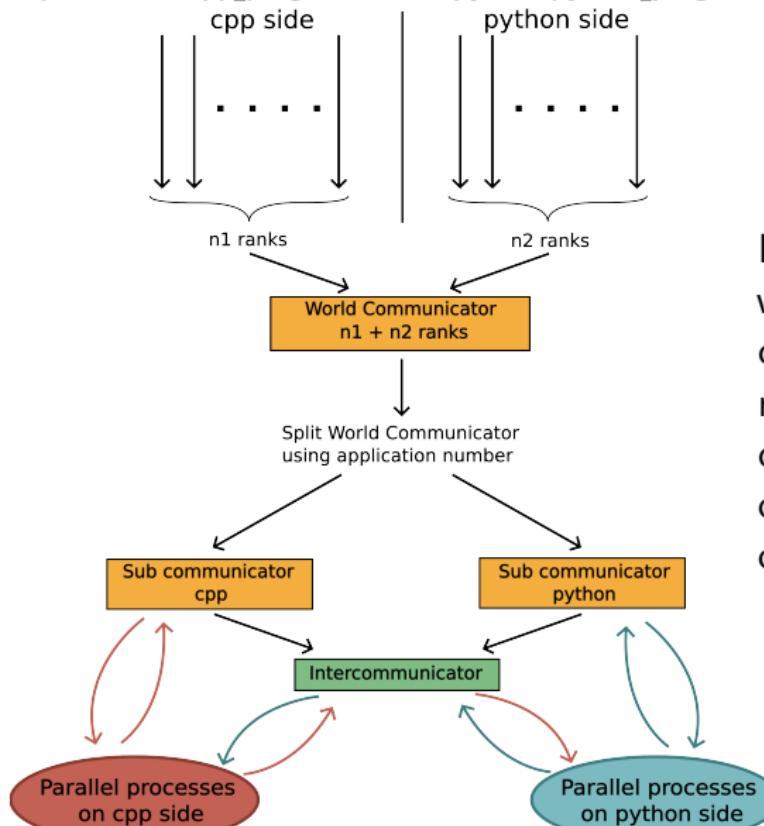
Coupling of PISAM and HESEL



Time step of the coupled model:

- 1) Communicate the plasma fields to PISAM
- 2) Run PISAM to obtain neutral source terms
- 3) Smooth the source terms and communicate them to HESEL
- 4) Develop the fields of HESEL.

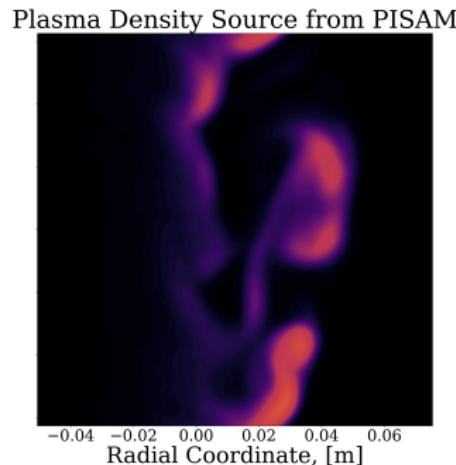
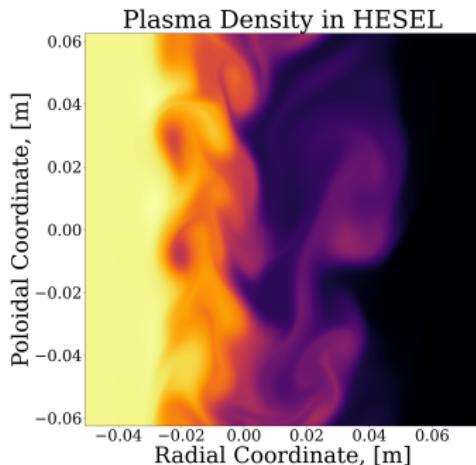
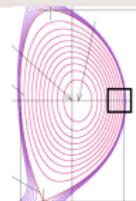
```
mpirun -n n1 cpp_program : -n n2 python python_program
```



Implemented on Marconi, with HESEL running 64 cores and PISAM running 128 cores in current simulations, to obtain equal wall times of the codes.

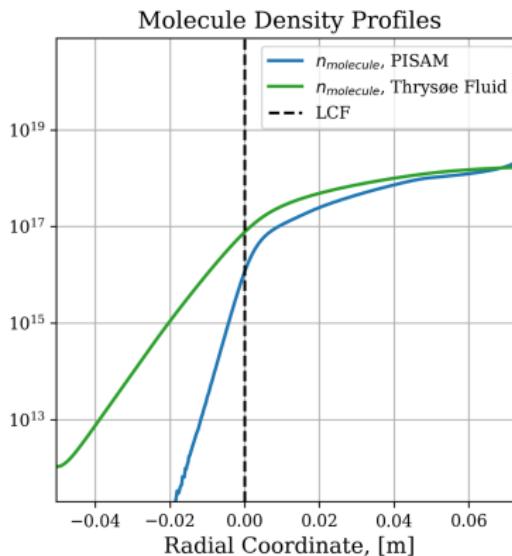
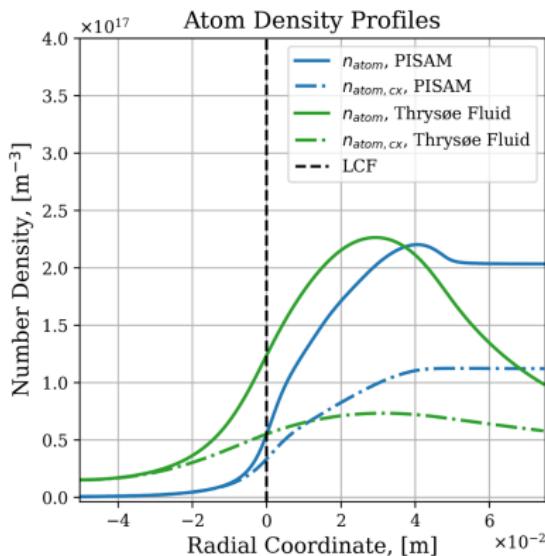
Dynamic PISAM-HESEL simulation

Time = 0 μ s

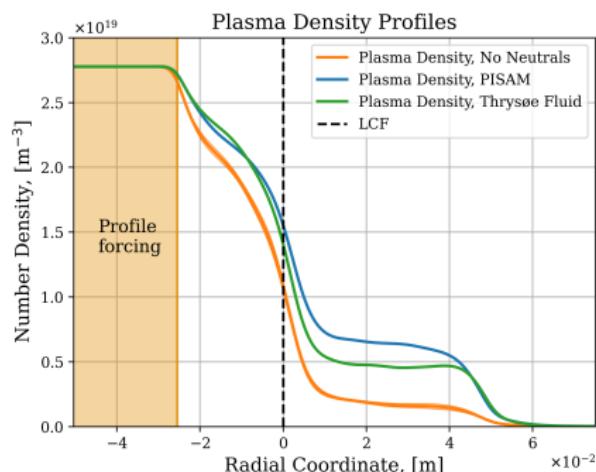
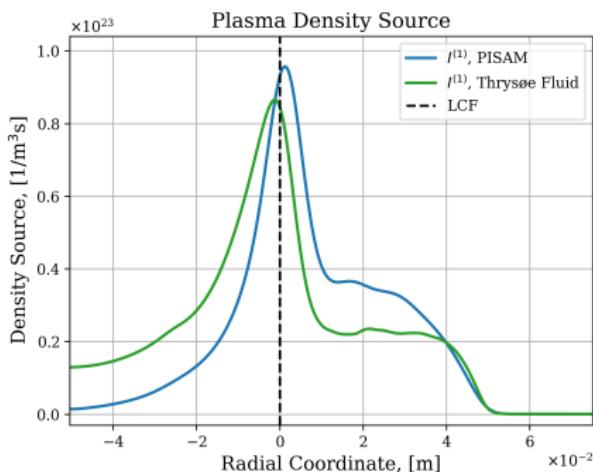


Simulation Results, Neutral Densities

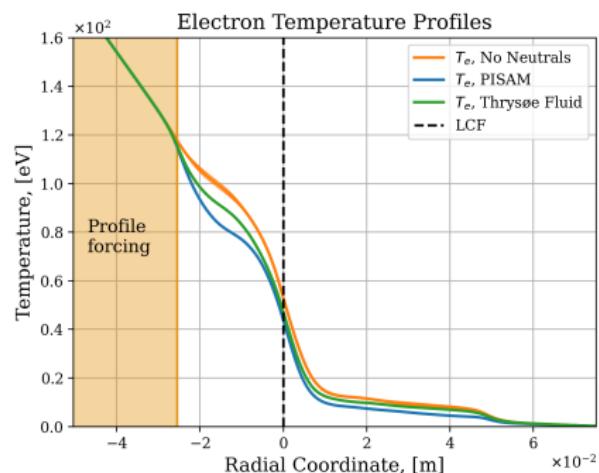
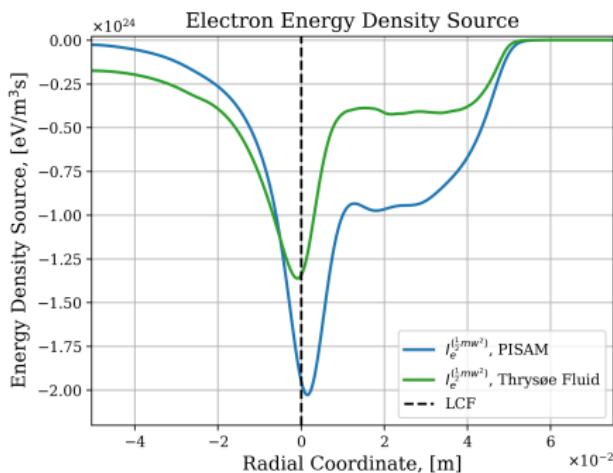
$$F_{\text{fluid}}^{\text{rad}} = n \left(V_{\text{adv}} - D_n \frac{\partial n}{\partial r} \right)$$



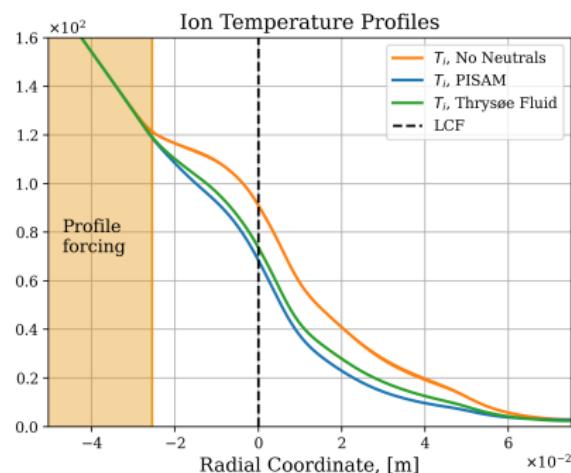
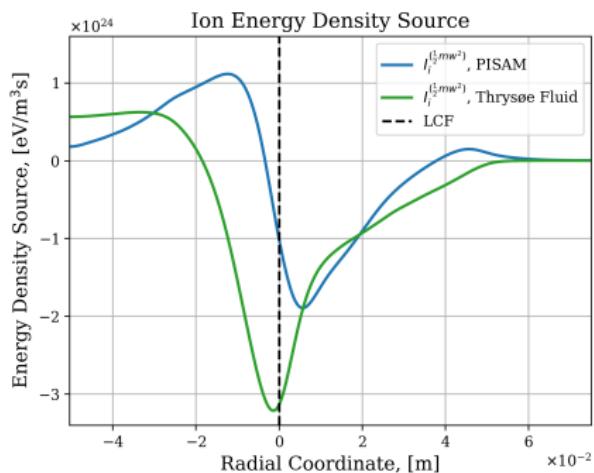
Simulation Results, Density



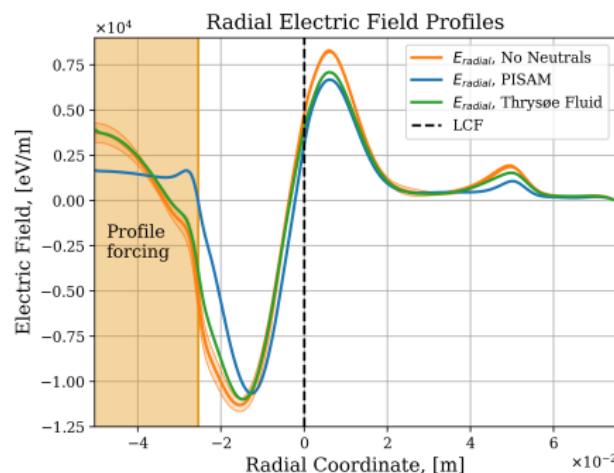
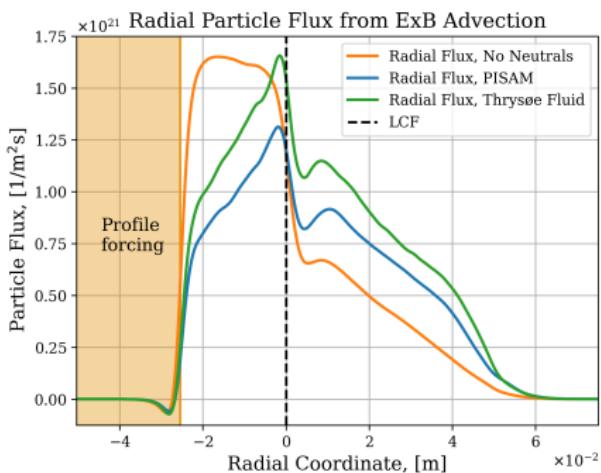
Simulation Results, Electron Temperature



Simulation Results, Ion Temperature



Simulation Results, Electric Fields



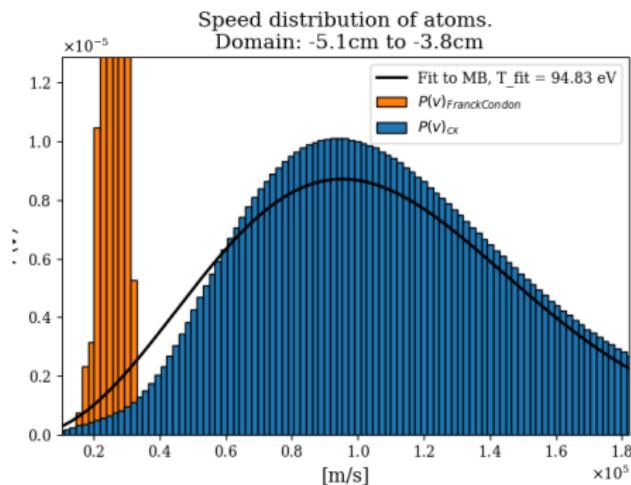
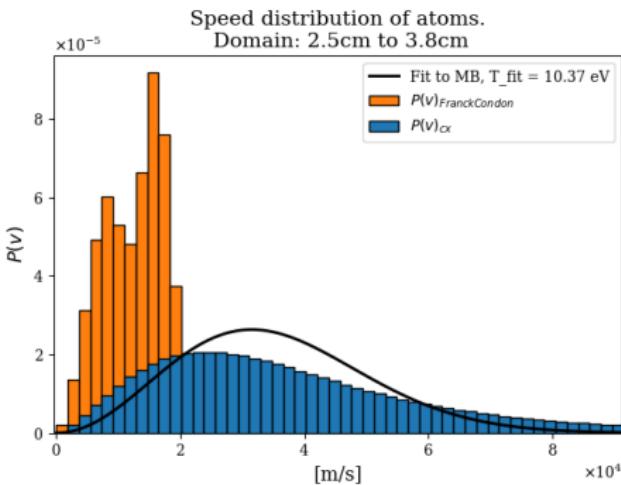
Concluding Remarks

- PISAM represents a novel approach for closing a fluid model with a discrete particle model.
- Major advantages:
 - Intuitive approach implemented in Python → Easy to add new particles and reactions.
 - Exploits the embarrassingly parallel nature of the problem → Easy to balance accuracy/computational resources/performance.
- MPI parallel Python and C++ codes can be coupled efficiently on existing HPC systems.

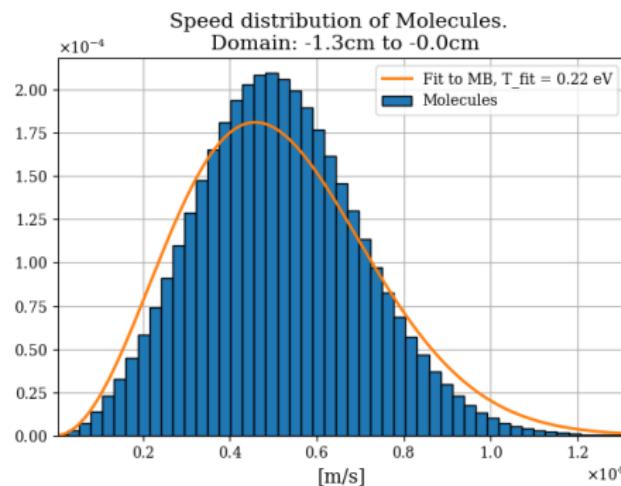
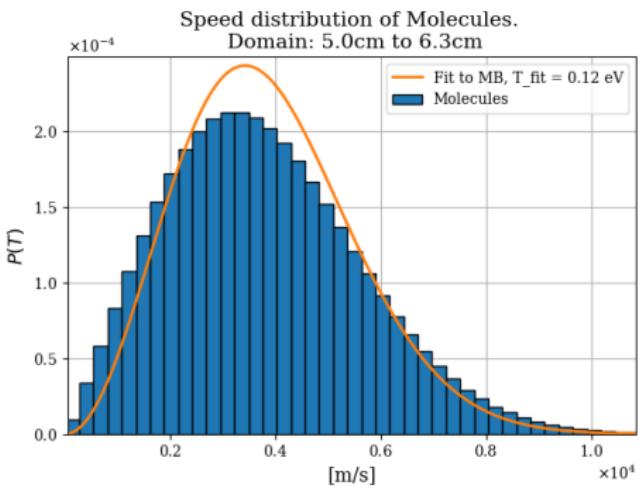
Questions?

Kristoffer Kvist, contact: kristofferkvist1994@gmail.com

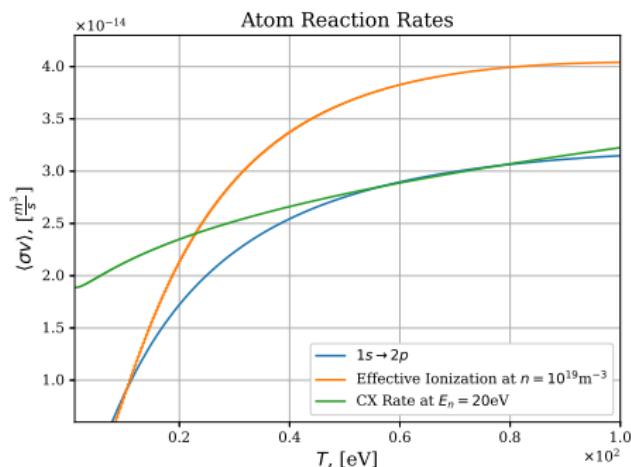
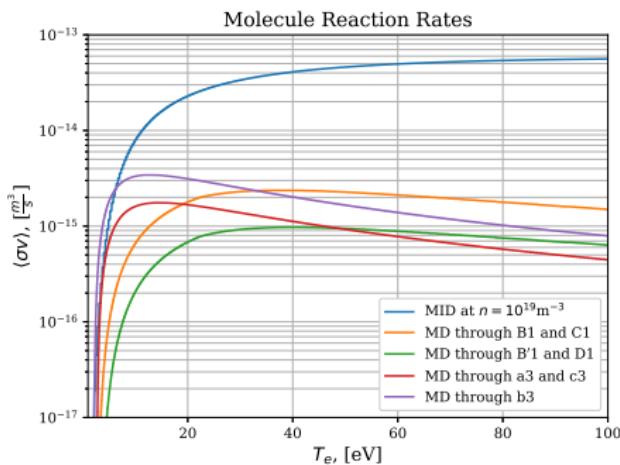
Simulation Results, Speed Distribution of Atoms



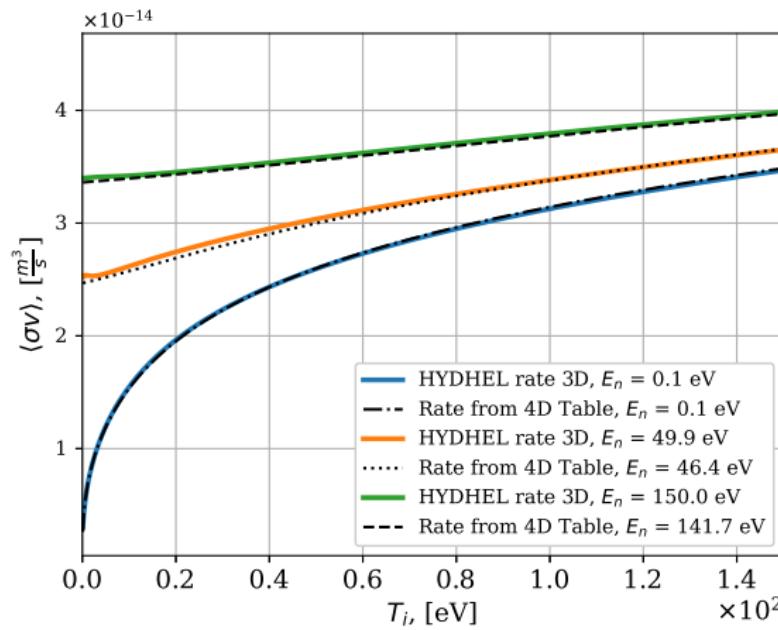
Simulation Results, Speed Distribution of Molecules



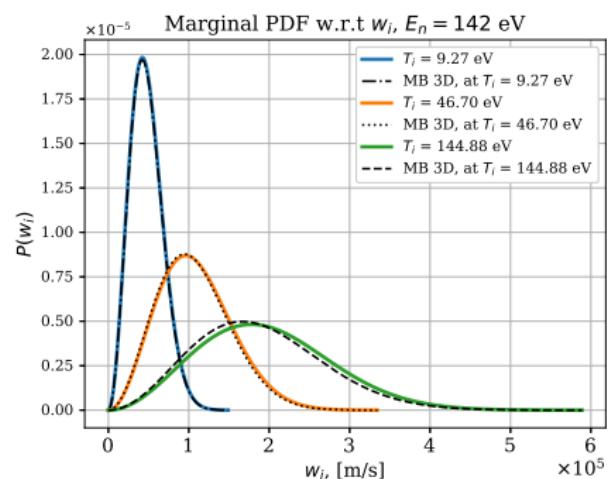
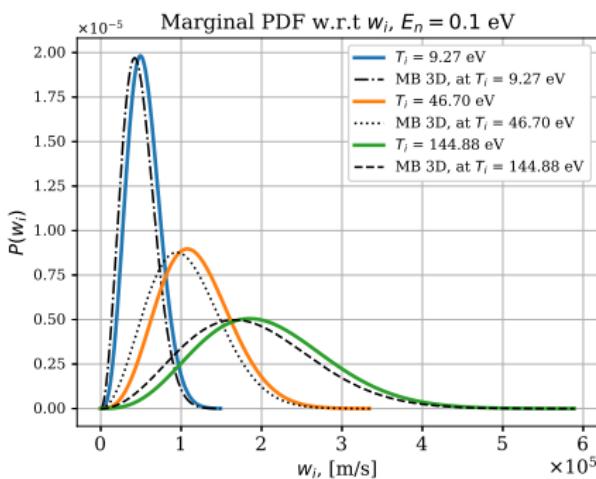
PISAM Reaction Rates



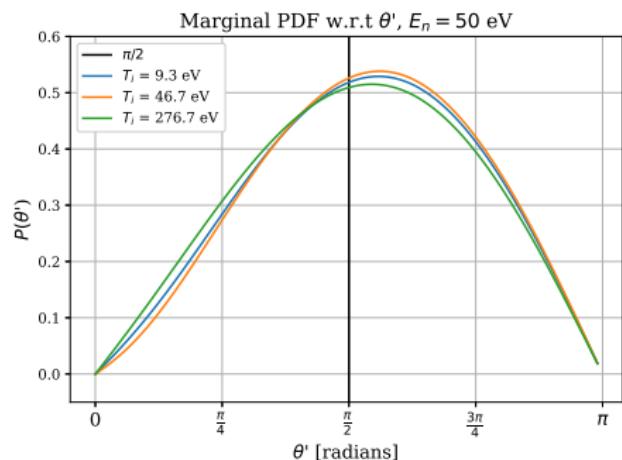
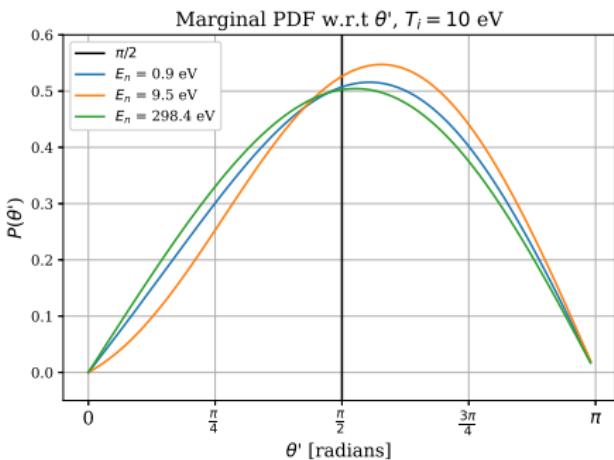
Charge-Exchange Rate



Velocity Distribution of CX Ions

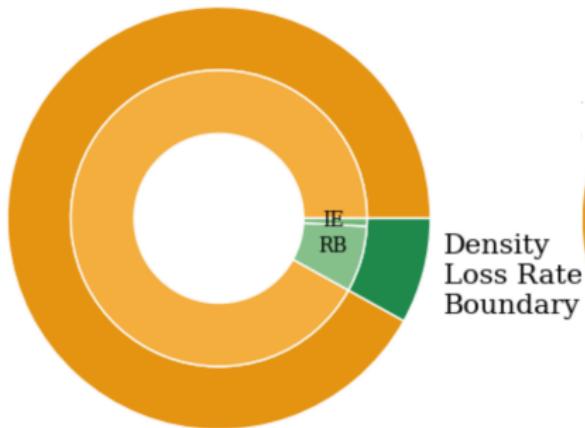


Angular Distribution of CX Ions

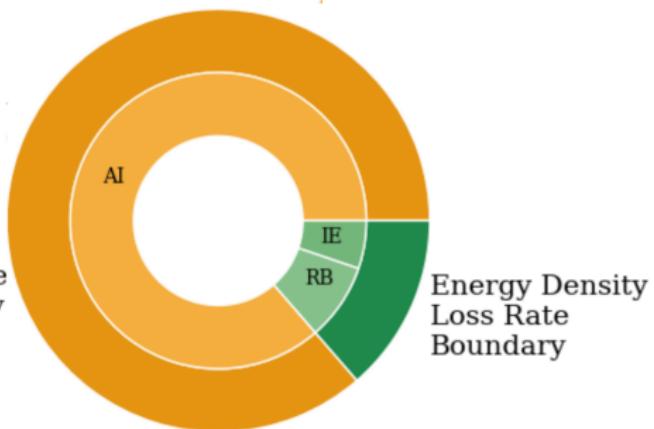


Neutral Loss Distributions

Density Source



Ion Energy Density Source



- In general:

Collision Frequency

$$\Gamma_{np}(\mathbf{r}, \mathbf{v}_n, t) = \int \sigma_{np}(g) g f_p(\mathbf{r}, \mathbf{v}_p, t) d\mathbf{v}_p = n_p \langle \sigma_{np} g \rangle_p,$$

where $g = |\mathbf{v}_n - \mathbf{v}_p|$.

- Electron-neutral collisions assuming a Maxwellian f_e :

$$\Gamma_{ne} = n_e \sqrt{\frac{2}{\pi}} \left(\frac{m_e}{T_e} \right)^{3/2} \int_0^{\infty} \exp\left(-\frac{m_e w_e^2}{2 T_e}\right) \sigma_{ne}(w_e) w_e^3 dw_e.$$

- Charge-Exchange reactions assuming a Maxwellian f_i :

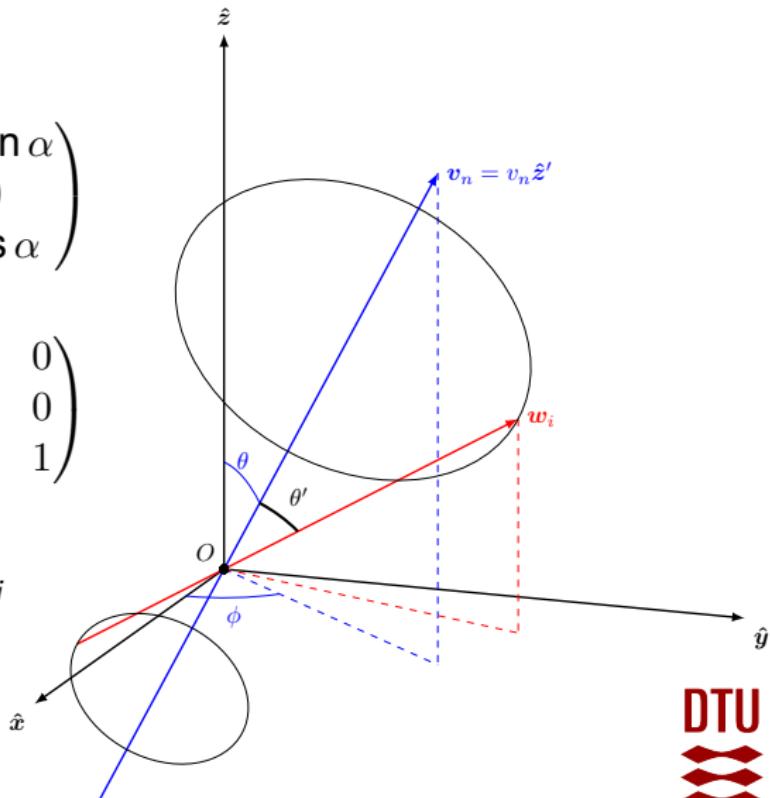
$$\Gamma_{cx} = \frac{n_i}{\sqrt{2\pi}} \left(\frac{m_i}{T_i} \right)^{3/2} \int_0^{\pi} \int_0^{\infty} \exp\left(-\frac{m_i w_i^2}{2 T_i}\right) \sigma_{cx}(g) g w_i^2 \sin(\theta') dw_i d\theta'.$$

Axial Symmetry in Charge-Exchange Reactions

$$\mathbf{R}_y(\alpha) = \begin{pmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{pmatrix}$$

$$\mathbf{R}_z(\alpha) = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{v}_i = \mathbf{R}_z(-\phi) \mathbf{R}_y(-\theta) \mathbf{w}'_i + \mathbf{u}_i$$



Plasma Fluid Equations

- Continuity equation:

$$\frac{d_s n}{dt} + n_s \nabla \cdot \mathbf{u}_s = I_s^{(1)}.$$

- Momentum equation:

$$\begin{aligned} m_s n_s \frac{d_s \mathbf{u}_s}{dt} + \nabla p_s + \nabla \cdot \bar{\pi}_s - n_s q_s (\mathbf{E} + \mathbf{u}_s \times \mathbf{B}) \\ = \mathbf{E}_s^{(mv)} + \mathbf{I}_s^{(mv)} - m_s \mathbf{u}_s I_s^{(1)}. \end{aligned}$$

- Pressure equation:

$$\begin{aligned} \frac{3}{2} \frac{d_s p_s}{dt} + \frac{5}{2} p_s \nabla \cdot \mathbf{u}_s + \bar{\pi}_s : \nabla \mathbf{u}_s + \nabla \cdot \mathbf{q}_s \\ = E_s^{(\frac{1}{2}mv^2)} - \mathbf{u}_s \cdot \mathbf{E}_s^{(mv)} + I_s^{(\frac{1}{2}mv^2)} - \mathbf{u}_s \cdot \mathbf{I}_s^{(mv)} + \frac{1}{2} m_s u_s^2 I_s^{(1)}. \end{aligned}$$