Integrated simulation solutions for the plasma and transversal physics in electric propulsion systems

Adam Obrousňík¹, Martin Kubečka², Petr Zikán¹, Zdeněk Bonaventura²

¹PlasmaSolve Ltd., Brno, Czech Republic
²Faculty of Science, Masaryk University, Brno, Czechia

Motivation: from plasma coatings to electric propulsion

Plasma simulations computationally difficult (multiphysics, numerically stiff systems, timescales), possibilities?

a) In-house codes from scratch – powerful but difficult to maintain, generalize to other problems or couple to other solvers
b) Commercial platforms (COMSOL, TechX USim) – often expensive, closed-source but usually good user support
c) Open-source solutions – not common in plasma physics. Is it feasible to build plasma models using open-source frameworks?

Workflow & Tools

We create a common framework (mostly Python scripts) based on open-source libraries:

1. Geometry pre-processing
   - SAlOMEs Platform
   - Python API – programmatic geometry generation
2. Mesh generation
   - NETGEN (tetrahedral) or snappyHexMesh (hex) algorithms
3. Physics solvers
   - Elmer FEM (finite element method)
   - OpenFOAM FVM (finite volume method)
4. Post-processing
   - unified .vtu format readable by Paraview

Hydrodynamic plasma models

Laboratory problem
- Plasma discharge propagation in complex discharge chambers
- Vacuum arc, HiPIMS magnetron

Use in electric propulsion
- Simulation of secondary plasma jets in vacuum arc thrusters.
- Interaction of plasma jet with grids and spacecraft

Easily modifiable model equations

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0
\]

\[
\frac{\partial (\frac{1}{2} \rho \mathbf{u} \cdot \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla \cdot (\rho \mathbf{F}) - \nabla P + \rho \mathbf{E}
\]

\[
\rho \mathbf{u} \cdot \mathbf{u} = \rho \mathbf{u} - \rho_k
\]

\[
\frac{\partial P}{\partial t} + \nabla \cdot (P \mathbf{u}) = \frac{1}{T_e} \left( \rho \mathbf{E} \cdot \mathbf{E} \right)
\]

Advanced thermal balance models

Laboratory problem:
- Thermal balance in ion sources, plasma lamps and other plasma-based devices.

Use in electric propulsion
- Thermal balance in thrusters and ion engines

Thermal balance model
- Radiation through optically thin media (vacuum) and thick media (plasma).
- Coupling to plasma simulation interfaces.
- CAD-to-simulation approach (STEP files as input), 3D simulations easy.

Global plasma models for alternative fuels

Laboratory problem:
- Predicting plasma kinetics in various mixtures – N₂, O₂, H₂O, Ar, Xe, Kr, Xe...
- Global model with user-friendly editing of model equations and coefficients

Use in electric propulsion
- Global modeling of electric propulsion ionizers with alternative fuels, including air-breathing thruster or CO2 thruster kinetics simulation

Workflow & Tools

Key enabling technology for complex geometries and large studies: Solvers integration with Amazon cloud computing centers.

Cutting edge IT makes our software truly multi-platform – supporting Windows 10, Linux and Mac systems.

Bottomline

What are the pros and cons of building a plasma model code on open-source?

Superior performance and scaling
- Provide a framework for implementing any PDE or Lagrangian solver
- No node/core license locking
- Possible to couple different codes seamlessly

Very poor documentation - the source code is your only reference
- Steeper learning curve

Contact info:

Masaryk University, Brno CZ
adam.obrousnik@gmail.com
https://physics.muni.cz/kfe
+420 776 185 170

PlasmaSolve company, Brno CZ
obrusnik@plasmasolve.com
https://plasmasolve.com

Postitive and negative ions in N₂/O₂ RF plasma at 0.05 Pa

Contact info:

Masaryk University, Brno CZ
adam.obrousnik@gmail.com
https://physics.muni.cz/kfe
+420 776 185 170

PlasmaSolve company, Brno CZ
obrusnik@plasmasolve.com
https://plasmasolve.com