

A new finite-element solver for plasma edge studies using Sparselizard C++ library

Rahul K Nagaraja
Quanscient Oy

Aaro E Järvinen
VTT

27 May 2024

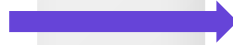
QUANSCIENT
ADVANCED COMPUTING HUB,
University of Helsinki

**BUSINESS
FINLAND**

Contents

Background

- State of the art
- Current challenges
- Scope of the project

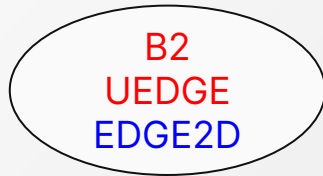


Implementation and results

- Isothermal fluid model
- Self-consistent temperature
- Convection in heat transfer mechanism
- Neutral diffusion and atomic physics

Plasma edge codes: STATE OF THE ART

fluid codes



- Finite volume based: B2, UEDGE [1,2].
- Finite difference based: EDGE2D [3].
- Solution in the 2-dimensional poloidal plane.
- Fluid equations for Ion and electron.
- Diffusive/fluid equations for neutrals.

kinetic Monte Carlo neutral code

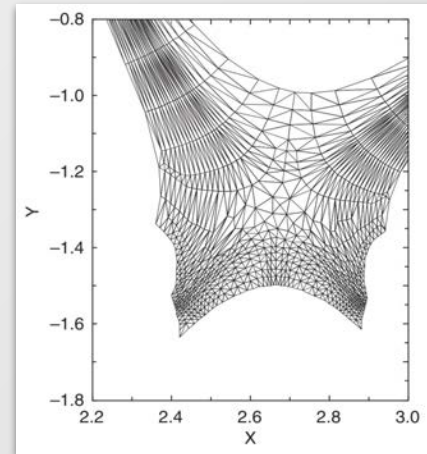
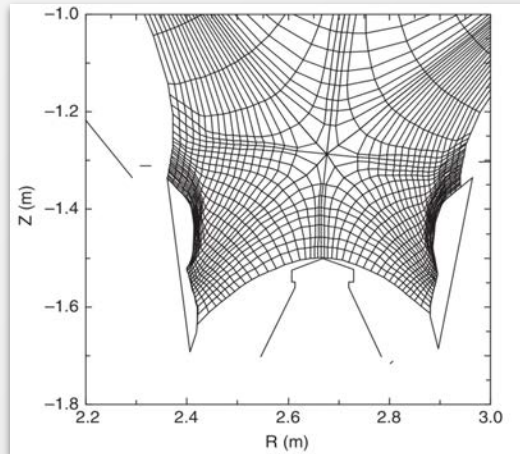


- For accurate description of neutrals [1].
- B2 + EIRENE → SOLPS-ITER today [4]. Widely used by many research groups.
- Sophisticated physics model.

Check out Oskar Lappi's poster on "Parallel algorithms for Monte Carlo neutral particle transport solver"

What about FEM codes for edge plasma?

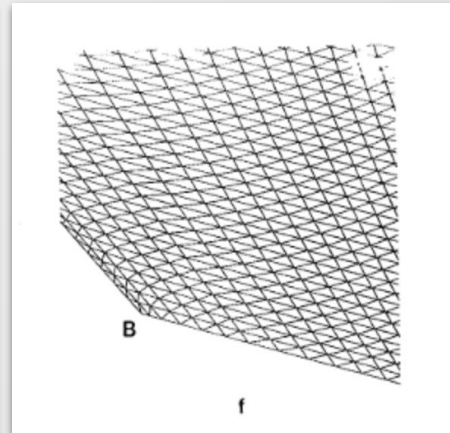
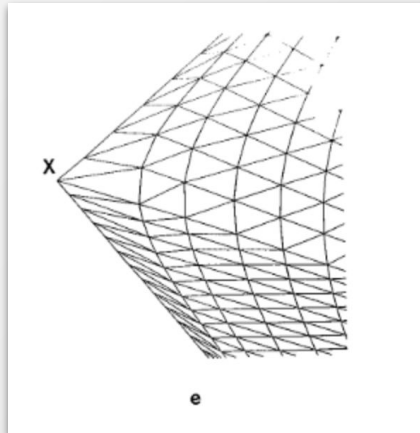
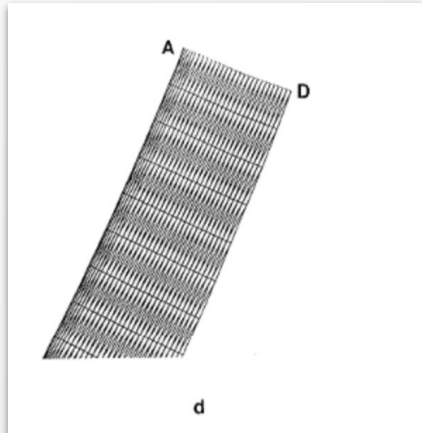
- Development started in the late 1980s. Utilize benefits of finite element method:
 - Structured orthogonal quadrilateral in FVM, FDM → difficult to capture wall geometry.
 - unstructured grids → realistic wall geometry representation → wall boundary conditions [8].
 - easily apply sheath physics at the wall boundary - natural derivative type boundary conditions.



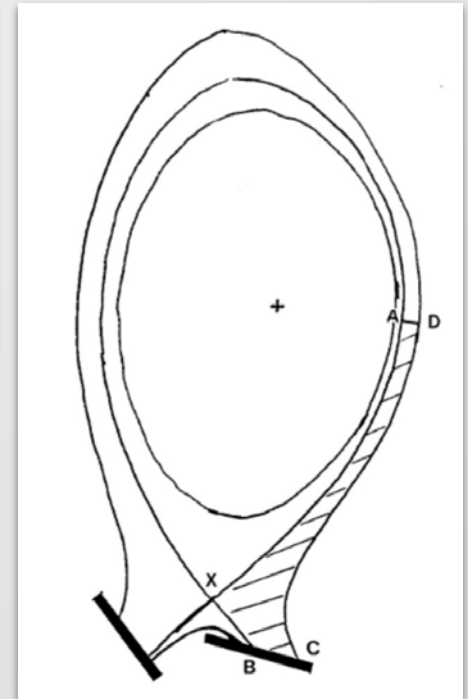
Structured quadrilateral and unstructured triangular mesh computed for JET in the divertor region (Fig. 1,2 [FEM7]).

Did unstructured triangular mesh benefit edge plasma studies ?

- There are strict restrictions on the grid generation for edge plasma simulation due to strong anisotropy between the parallel and perpendicular transport.
- The grid must be aligned along the magnetic flux surfaces [6,7,8]
→ could be triangular and non-orthogonal, but structured.



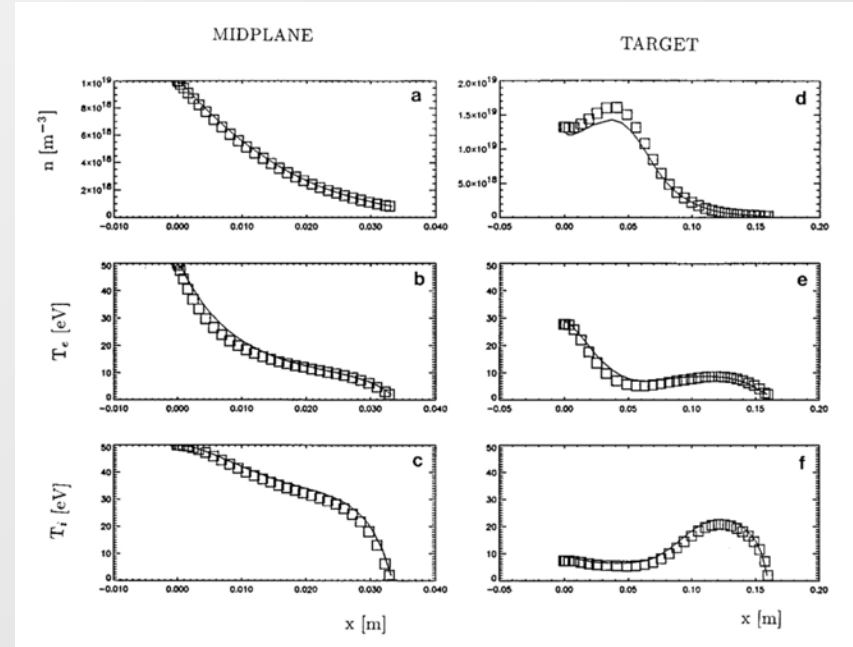
Typical structured (d–f) finite element meshes (Fig. 2 [6])



Poloidal cut view of the full axisymmetrical plasma edge region in a poloidal divertor tokamak (Fig.1 [6])

How does FEM compare to FVM, FDM discretization ?

- Finite element codes have also been coupled with kinetic Monte Carlo neutral model (FELS + EIRENE) [6,9].
- Good agreement have been found between the two schemes in edge plasma studies.



Comparison of steady state results across radial distance between FELS (solidlines) and B2 (opensquares) (Fig. 3 [6])

Current challenge is the computational time

- Irrespective of the discretization scheme, one of the current challenges of plasma edge simulation like in any high fidelity physics is the high computational time. To improve computational speeds:
 - UEDGE parallelization attempts [10].
 - In SOLPS-ITER, EIRENE is parallelized (but not B2.5).
 - BOUT++: a framework for parallel plasma fluid simulations. Finite difference based [11].
- In recent times, focus of numerical codes have shifted towards not just parallelization but also on modularity and symbolic implementation frameworks.
- This makes it easier to work with multiphysics models: implementation, extension, modification.
- Sparselizard library is one such framework written in C++. It is based on finite element method and is intended for high performance computing in multiphysics simulations [12].

Scope of the current project

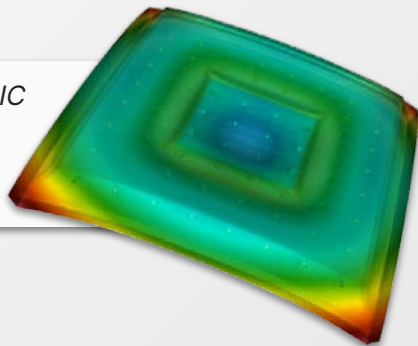
- Build a new finite element solver for edge plasma using Sparselizard library.
- Exploit the library's scalable domain decomposition method for parallel running with the intention of reducing computational time.

Thermal expansion of an IC

~10 millions DoFs

60 CPUs

~10 minutes runtime



Modeling non-planar coils in a full-scale stellarator [13]

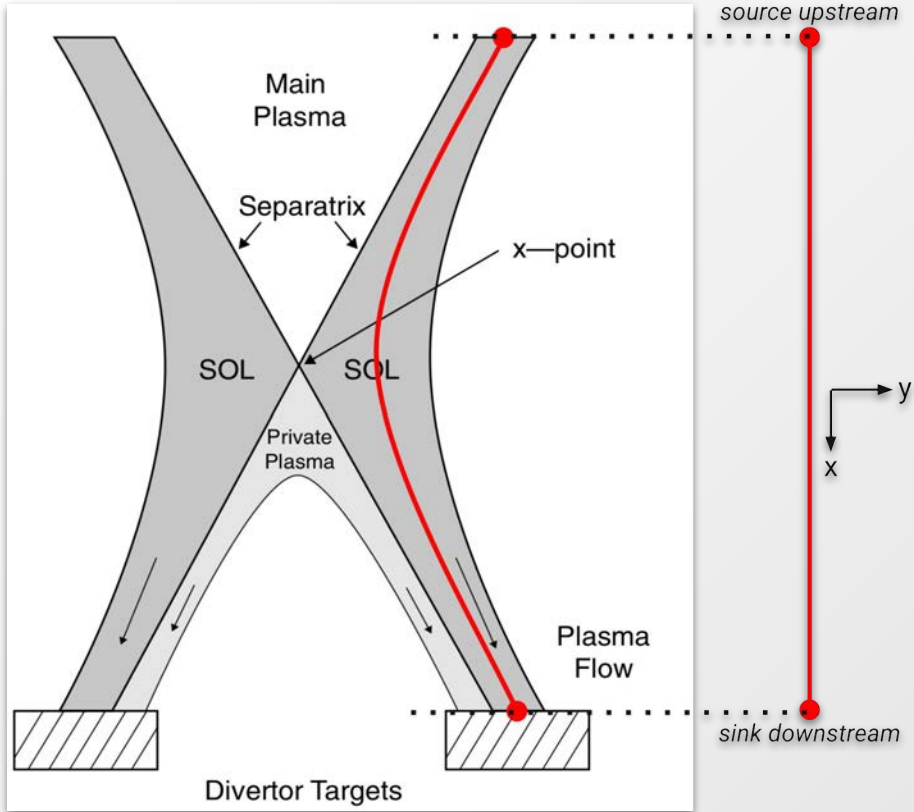
~39.5 million DoFs

50 CPUs

~5 minutes runtime



Scrape of layer (SOL)



Assumptions:

- one-dimensional
- $T_{\text{electron}} = T_{\text{ion}}$
- Quasineutrality: $n_{\text{electron}} = n_{\text{ion}}$
- Ambipolar outflow: $(nv)_{\text{electron}} = (nv)_{\text{ion}} \rightarrow v_{\text{electron}} = v_{\text{ion}}$

Results in a SINGLE SPECIES fluid equations.

Finite element framework

Step 1:
Normalize the plasma parameters.

Step 2:
The terms are expanded to change the conservative form of equations to non-conservative form.

Step 3:
The residual of the strong form is transformed into weak formulation.

- Particle conservation for Plasma:

$$\frac{d}{dx}(nv) = S_p(x)$$

- Normalization: divide by $n_0 c_s$ on both sides

$$\frac{1}{n_0 c_s} * \left(\frac{d}{dx}(nv) = S_p \right)$$

$$\frac{d}{dx}(n/n_0 * v/c_s) = \frac{S_p}{n_0 c_s}$$

- Defining normalized quantities:

$$N = \frac{n}{n_0} \quad M = \frac{v}{c_s} \quad S_{pn} = \frac{S_p}{n_0 c_s}$$

- The resulting equation with normalized quantities:

$$\frac{d}{dx}(N*M) = S_{pn}$$

- Conservative → Non-conservative form:

$$N \frac{dM}{dx} + M \frac{dN}{dx} = S_{pn}$$

- Transform strong form to Weak formulation for FEM:

$$\text{Residual : } \mathcal{R}_{particle} = N \nabla M + M \nabla N - S_{pn} = 0$$

$$\text{Galerkin FEM : } \int_L \{ \mathcal{R}_{particle} \} * N' dX = 0$$

where,

N' : Test function for plasma density

Any non-linear terms are Newton linearized in the implementation.

1D isothermal fluid model

- Particle conservation:

$$\frac{d}{dx}(nv) = S_p \rightarrow \nabla \cdot (nv) = S_p$$

$$\rightarrow \nabla \cdot (N*M) = S_{pn}$$

- Momentum conservation:

$$nv \frac{dv}{dx} = -c_s^2 \frac{dn}{dx} - vS_p \rightarrow nv \cdot \nabla v = -c_s^2 \nabla n - vS_p$$

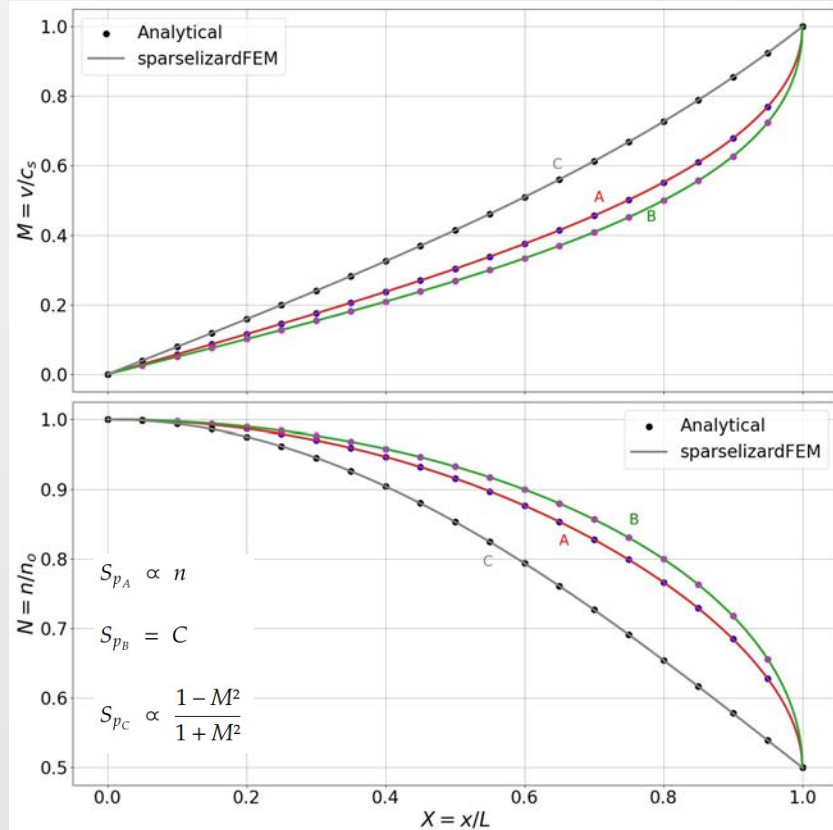
$$\rightarrow N*M \cdot \nabla M = -\nabla N - M*S_{pn}$$

- Normalized entities are defined as

$$N = \frac{n}{n_0} \quad M = \frac{v}{c_s} \quad S_{pn} = \frac{S_p}{n_0 c_s / L}$$

where,

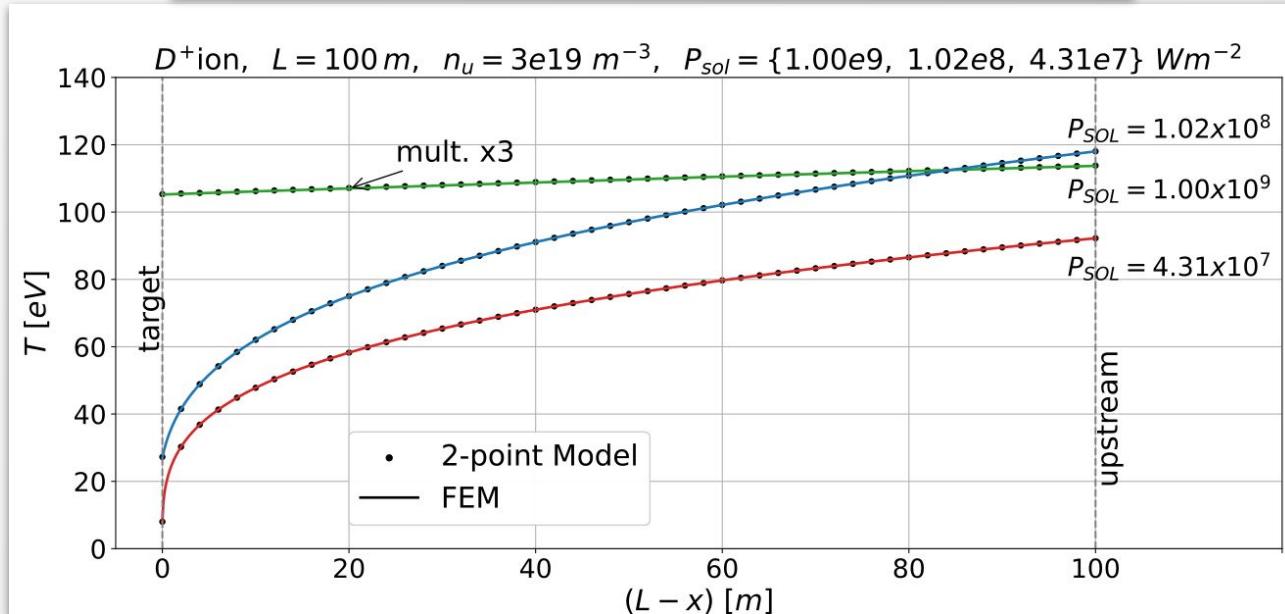
n	plasma density	particles / m ³
v	plasma velocity	ms ⁻¹
S_p	particle source	particles / m ³ s
c_s	isothermal acoustic speed	ms ⁻¹



Self-consistent temperature

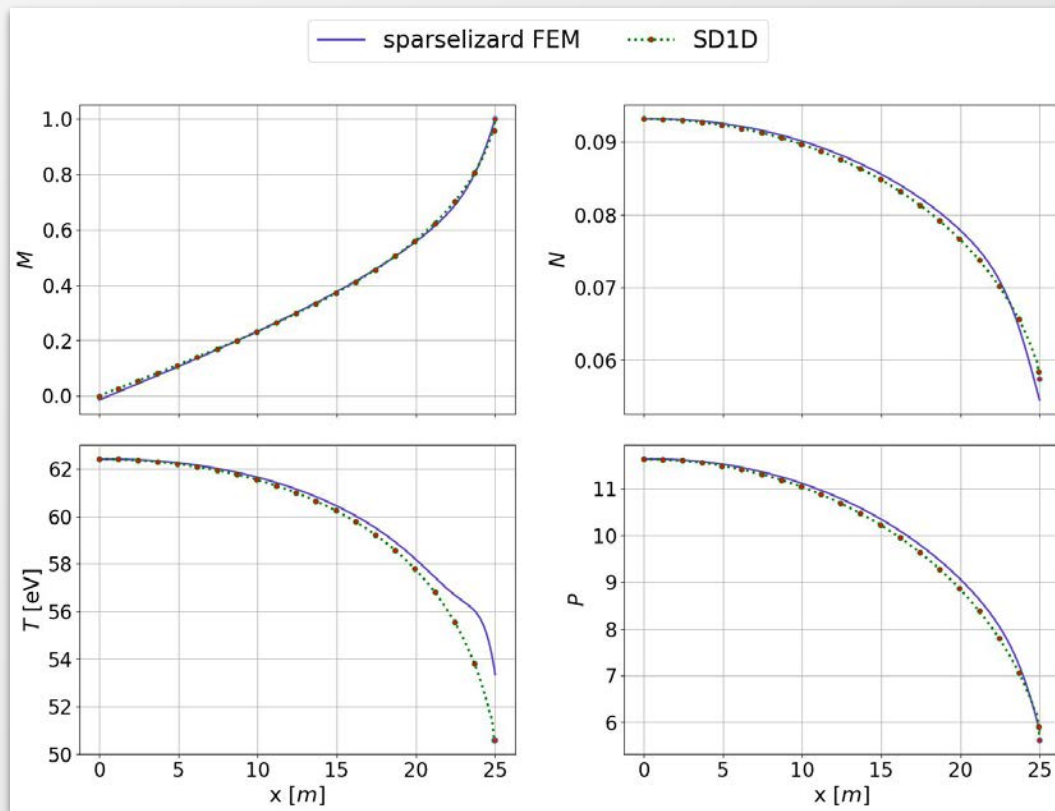
Spitzer heat conduction

$$\frac{d}{dx} \left(-k_{0_e} T_e^{5/2} \frac{dT_e}{dx} \right) = 0 \quad \rightarrow \quad \nabla \cdot \left(-k_{0_e} T_e^{5/2} \nabla T_e \right) = 0$$



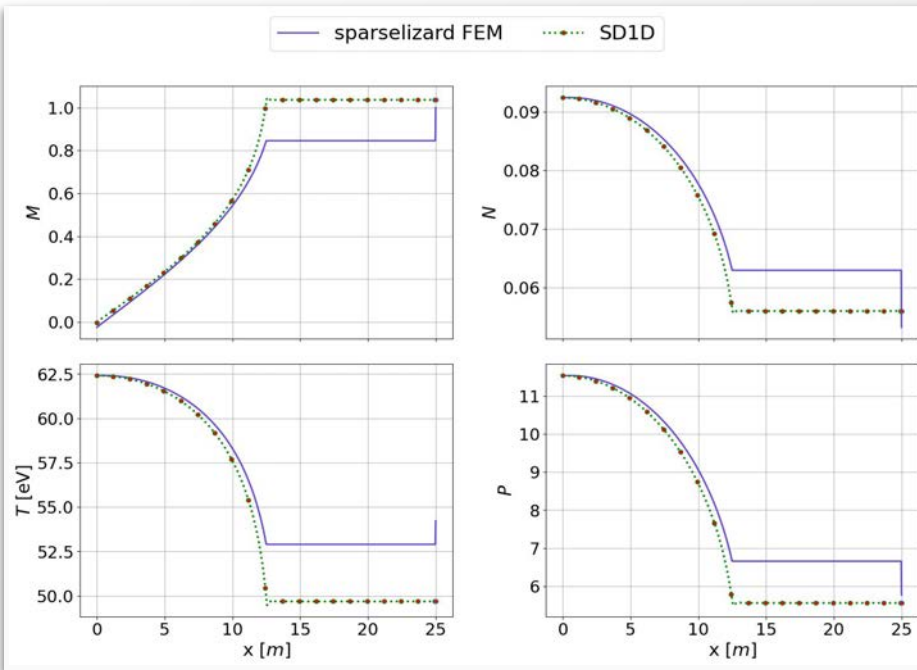
Convection in heat transfer mechanism

- Include convection in the energy equation.
- Results compared against SD1D:
 - It is a one-dimensional one-fluid SOL solver
 - Built on the framework of BOUT++ plasma simulation code (finite difference method).
 - Solves for Density (n), Momentum (nv) and Pressure (p).
 - SD1D is a transient solver.

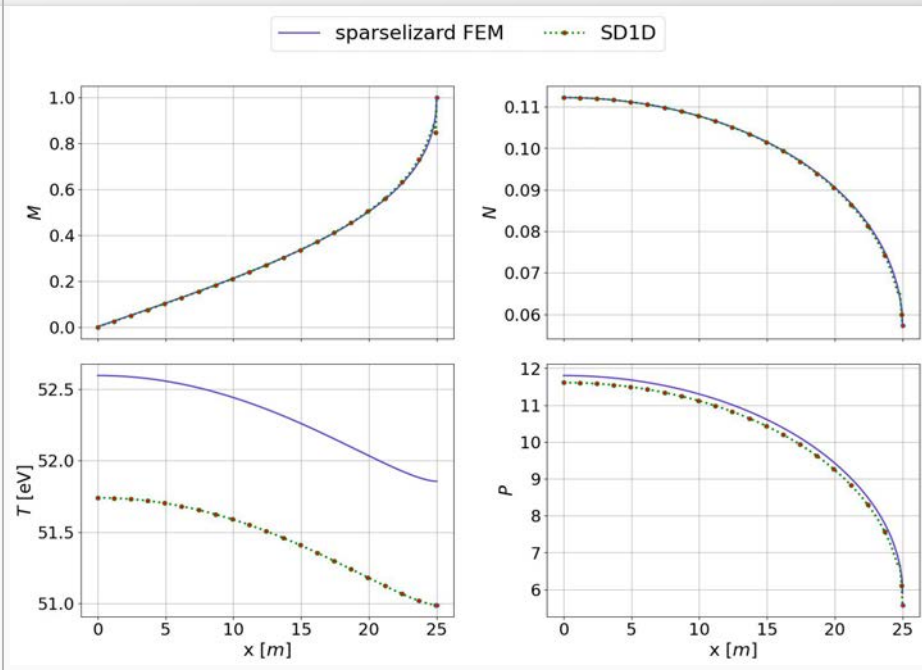


Heat transfer mechanism : convection only
Volumetric source : uniform throughout the domain

Convection in heat transfer mechanism



Heat transfer mechanism : convection only
 Volumetric source : uniform and only in the first half of the domain



Heat transfer mechanism : convection + conduction
 Volumetric source : uniform throughout the domain

Self-consistent source: neutral diffusion and atomic physics

$$\frac{d}{dx} \left(-D_n \frac{dn_n}{dx} \right) = S_n(x) \quad \rightarrow \quad \nabla \cdot (-D_n \nabla n_n) = S_n(x)$$

$$\rightarrow \quad \nabla \cdot (-D_n \nabla N_n) = S_{nn}(x)$$

Depends on atomic process such as recombination, ionization, charge exchange, etc.

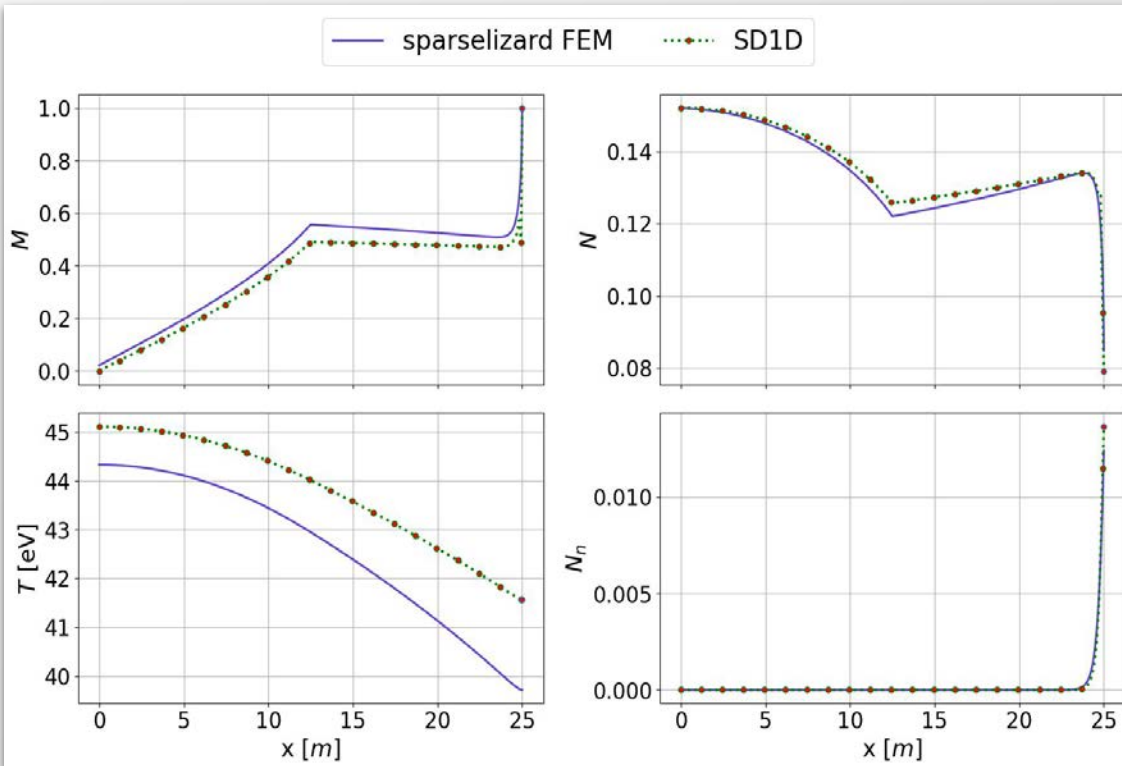
Include loss/sink terms arising from atomic process.

Particle, momentum, energy conservation with loss terms:

$$\frac{d}{dx} (nv) = S_{n_ext} - S_{n_atomic}$$

$$\frac{d}{dx} (m_i n v^2 + 2nT) = -F_{atomic}$$

$$\frac{d}{dx} \left(\left(\frac{1}{2} m_i v^2 + 5T \right) nv - k_{0_e} T_e^{5/2} \frac{dT_e}{dx} \right) = S_{p_ext} - S_{p_atomic}$$



SUMMARY:

- Implementing a new single-fluid 1D SOL solver using Sparselizard library:
 - Verified against 1D isothermal model.
 - Self-consistent temperature verified against two-point model.
 - Comparison of result against SD1D for different configurations:
 - Conduction, convection, neutral diffusion and atomic physics.

FUTURE WORK:

- Improve the robustness of the code:
 - Identifying and fixing the cause of deviation in temperature profiles.
- Simulating a case of fully detached plasma conditions.
- Test cases to compare the computational performance of Sparselizard and UEDGE.
- Extending the implementation from:
 - diffusive → fluid equations for neutral evolution.
 - 1-fluid model → 2-fluid model: separate equations for ions and electrons.
 - 1D → 2D

References

- [1] Reiter, D., Baelmans, M., & Boerner, P. (2005). The EIRENE and B2-EIRENE codes. *Fusion science and technology*, 47(2), 172-186.
- [2] Rognlien, T. D., Milovich, J. L., Rensink, M. E., & Porter, G. D. (1992). A fully implicit, time dependent 2-D fluid code for modeling tokamak edge plasmas. *Journal of nuclear materials*, 196, 347-351.
- [3] Simonini, R., Corrigan, G., Radford, G., Spence, J., & Taroni, A. (1994). Models and Numerics in the Multi-Fluid 2-D Edge Plasma Code EDGE2D/U. *Contributions to Plasma Physics*, 34(2-3), 368-373.
- [4] Wiesen, S., Reiter, D., Kotov, V., Baelmans, M., Dekeyser, W., Kukushkin, A. S., ... & Voskoboynikov, S. (2015). The new SOLPS-ITER code package. *Journal of nuclear materials*, 463, 480-484.
- [5] Vesey, R. A., & Steiner, D. (1995). A Two-Dimensional Finite Element Model of the Edge Plasma. *Journal of Computational Physics*, 116(2), 300-313.
- [6] Zanino, R. (1997). Advanced finite element modeling of the tokamak plasma edge. *Journal of computational physics*, 138(2), 881-906.
- [7] Marchand, R. (2002). Finite element modelling of transport in a tokamak edge and divertor. *Plasma physics and controlled fusion*, 44(6), 871.

References

- [8] Kainz, A., Weimann, G., & Kamelander, G. (2003). *Strategies in edge plasma simulation using adaptive dynamic nodalization techniques* (No. IAEA-CSP--19/CD).
- [9] Puetz, T., Reiter, D., & Wuttke, M. W. (1994). Finite element modelling of the plasma edge. *Contributions to Plasma Physics*, 34(2-3), 404-409.
- [10] Dudson, B. D., Umansky, M. V., Xu, X. Q., Snyder, P. B., & Wilson, H. R. (2009). BOUT++: A framework for parallel plasma fluid simulations. *Computer Physics Communications*, 180(9), 1467-1480.
- [11] Halbach, A. (2017). Sparselizard - the user friendly finite element c++ library.
- [12] Backmeyer, Merle, et al. "Modeling non-planar coils in a full-scale stellarator." *IEEE Transactions on Applied Superconductivity* (2024).