



## FESTIM, a modelling code for hydrogen transport in materials for nuclear fusion applications

*R. Delaporte-Mathurin<sup>1,2</sup>, E. A. Hodille<sup>1</sup>, J. Dark<sup>2</sup>, F. Leblond<sup>1</sup>, J. Mougnot<sup>2</sup>, Y. Charles<sup>2</sup>, C. Grisolia<sup>1</sup>*

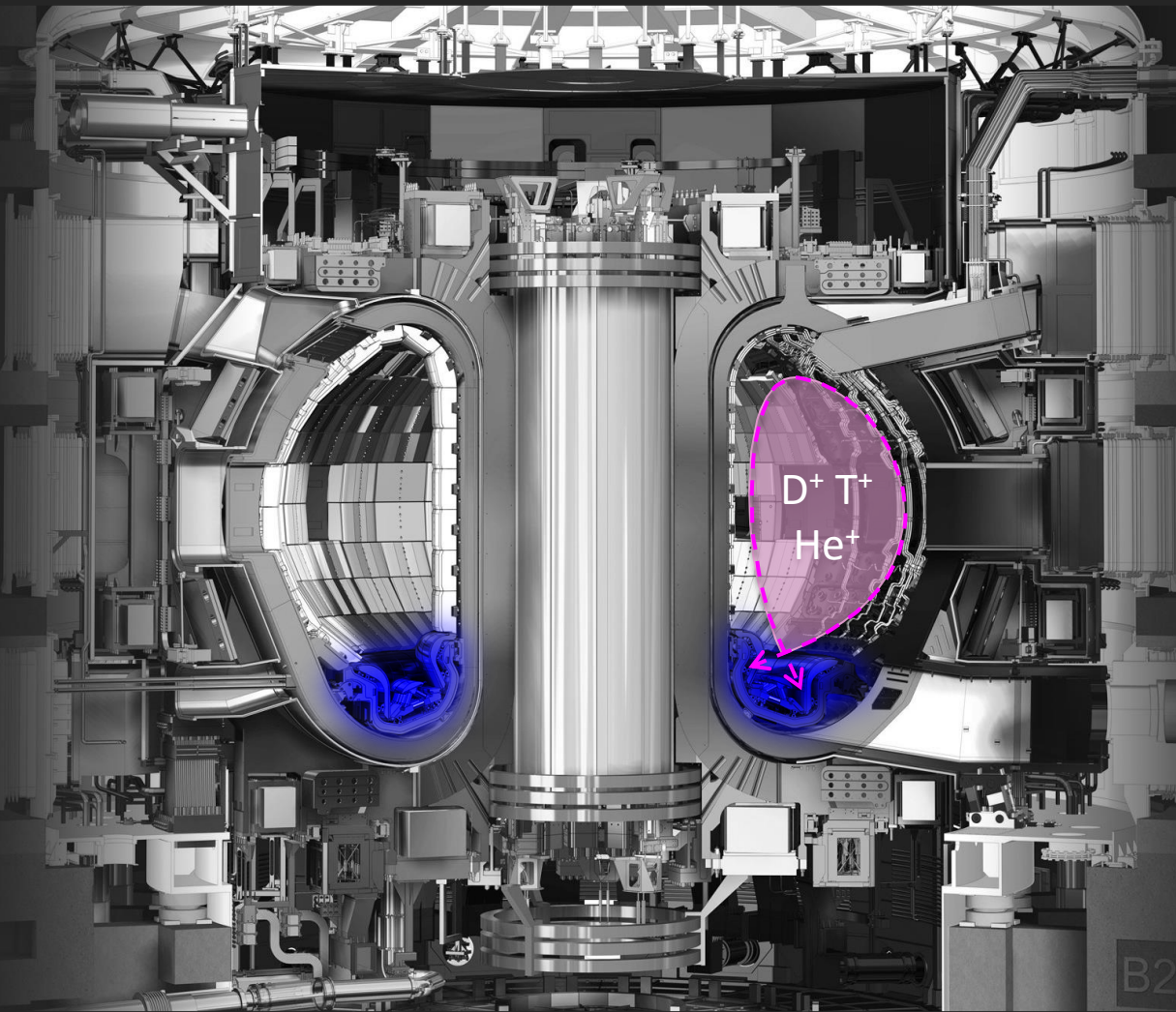
<sup>1</sup> CEA, IRFM/GCFPM, F-13108 Saint-Paul-lez-Durance, France

<sup>2</sup> Université Sorbonne Paris Nord, Laboratoire des Sciences des Procédés et des Matériaux, LSPM, CNRS, UPR 3407, F-93430, Villetaneuse, France



NUMFOCUS  
OPEN CODE = BETTER SCIENCE





Tritium  
contamination



Material  
embrittlement



Recycling  
fluxes



B2

$$\begin{aligned} \partial_t c_m &= \nabla(D(T) \cdot \nabla c_m) - \sum_i \partial_t c_{t,i} \quad \text{on } \Omega \\ \partial_t c_{t,i} &= k(T) \cdot c_m (n_i - c_{t,i}) - p(T) \cdot c_{t,i} \quad \text{on } \Omega \\ \frac{c_m^-}{S(T)^-} &= \frac{c_m^+}{S(T)^+} \quad \text{on } \Omega_i \cap \Omega_j \\ \rho C_p \partial_t T &= \nabla(\lambda \cdot \nabla T) + Q \quad \text{on } \Omega \end{aligned}$$

Hydrogen transport  
 McNabb & Foster - Trans.  
 Metall. Soc. (1963)

Conservation of  
 chemical potential  
 at interfaces

Energy equation

- ▶  $c_m, c_{t,i}, T$  H concentrations and temperature
- ▶  $i$  corresponds to a type of sink

## FESTIM

- ▶ Finite Element Simulation of Tritium In Materials
- ▶ Based on FEniCS
- ▶ 1/2/3D
- ▶ Multi-materials

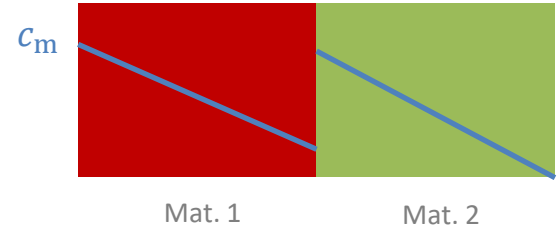


For more info:

Delaporte-Mathurin *et al*, NME (2019)



$$\frac{c_m^-}{S^-} = \frac{c_m^+}{S^+} \quad \text{at interfaces}$$



► Modelling discontinuities in FEniCS

$$\partial_t c_m = \nabla(D \cdot \nabla c_m) - \sum_i \partial_t c_{t,i} \quad \text{on } \Omega$$

$$\theta = c_m / S$$

1. Solve :  $\partial_t(\theta S) = \nabla(D \cdot \nabla(\theta S)) - \sum_i \partial_t c_{t,i}$  on  $\Omega$
2. Post-processing:

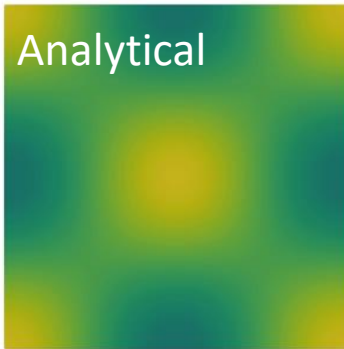
$$c_m = \theta \cdot S$$

project on DG1 space

```
V_DG1 = FunctionSpace(mesh, 'DG', 1)
c_m = project(theta*S, V_DG1)
```

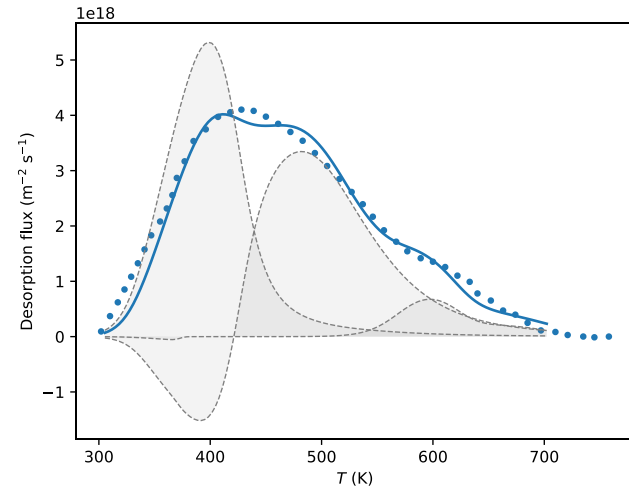
## Verification using MMS

- ▶  $c_{m,D} = 1 + \cos(2\pi x) \cos(2\pi y) + \cos(2\pi t)$
- ▶  $T = 500 + 30 \cos(2\pi x)$
- ▶ Multi-material



## Experimental validation

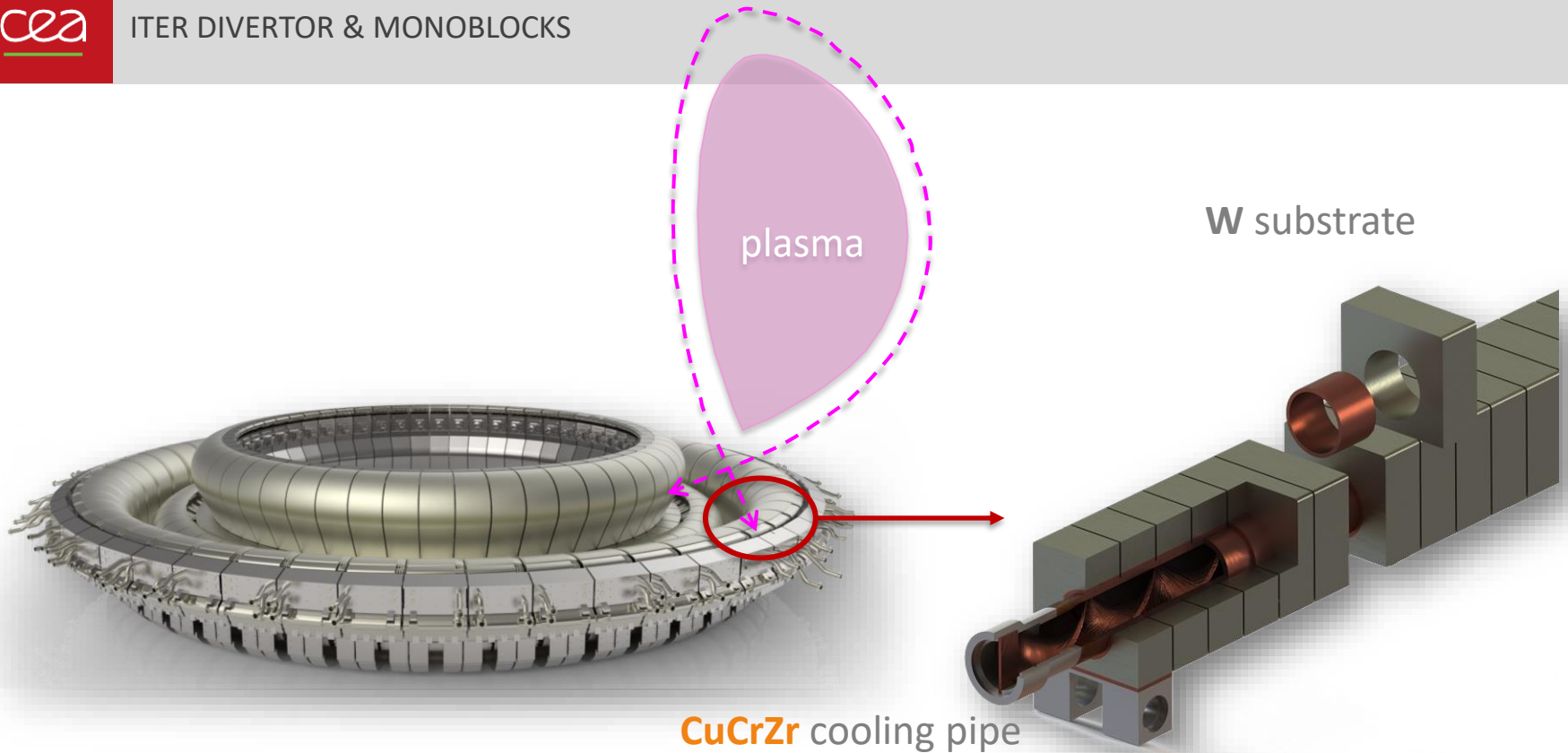
- ▶ Thermo-desorption experiments
- ▶ Parametric optimisation



Delaporte-Mathurin *et al*, NME (2021)

# Application:

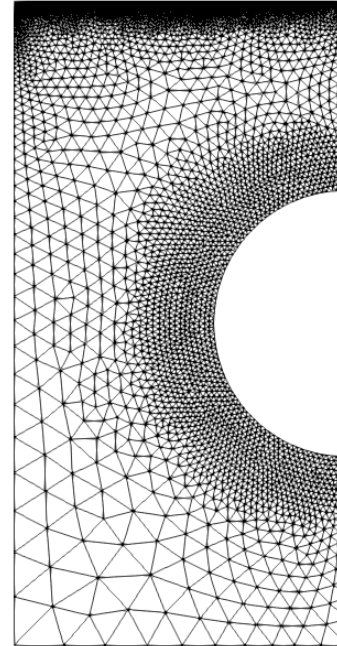
## Tokamak components



How much hydrogen  
is retained ?

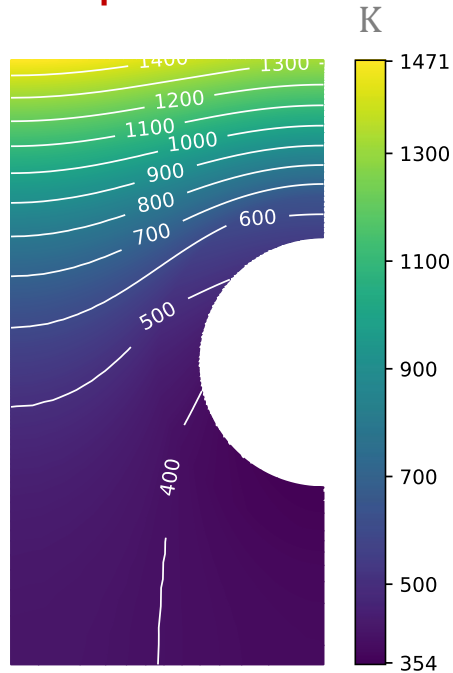


- ▶ Meshed with SALOME (open-source)
- ▶ Converted from .med to .xdmf with meshio [1]
- ▶ High refinement:
  - on the top surface
  - at interfaces
- ▶ Planned: using Adaptive Mesh Refinement

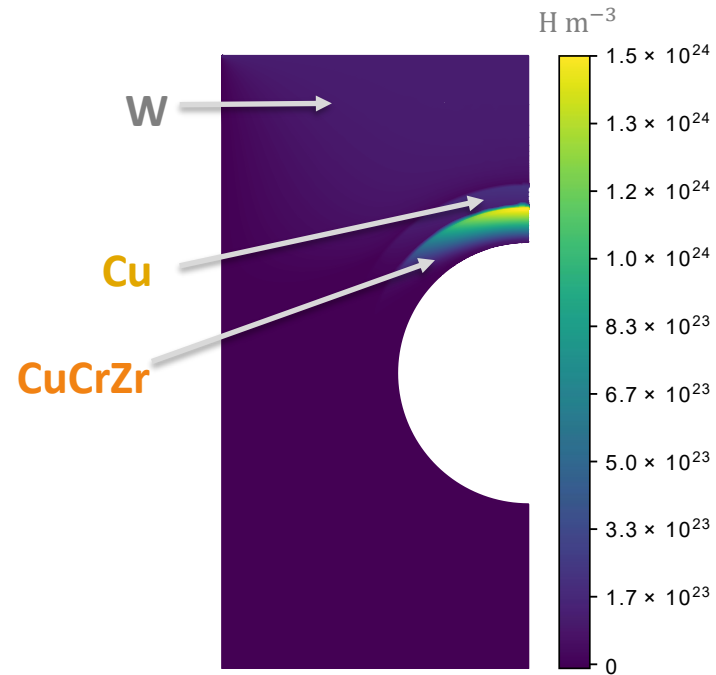


[1] Meshio: [10.5281/zenodo.4590119](https://zenodo.org/record/105281)

## Temperature field



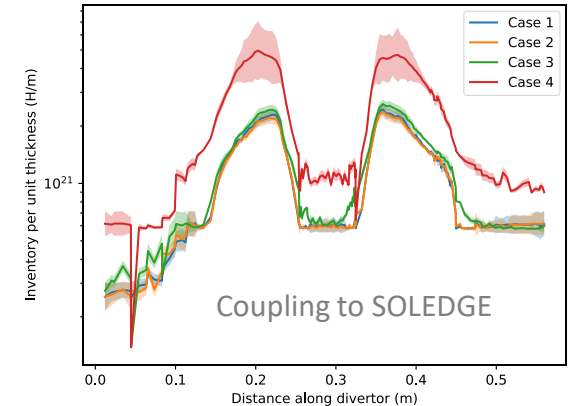
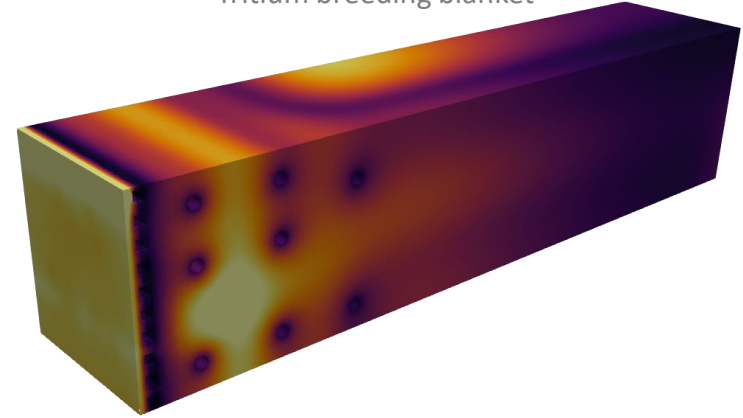
## Hydrogen concentration



- ▶ Total inventory of H :  $\int (c_m + c_t) dx$
- ▶ Coolant contamination :  $\int D(T) \nabla c_m \cdot n dS$

- ▶ Applications to more complex tokamak components
- ▶ Coupling with other physics
  - CFD
  - MHD,
  - co-deposition models
  - He transport...
- ▶ Coupling with external plasma codes

Tritium breeding blanket





Thank you for your attention

Plots were made with Matplotlib and Paraview

- ▶ **FESTIM** is a FEniCS-based simulation interface
  
- ▶ Hydrogen transport (including diffusion and trapping) is modelled and coupled to heat transfer.
  
- ▶ **FESTIM applications:**
  - Simulating fusion reactors components
  - Identifying materials properties
  
- ▶ **Perspectives:**
  - Applications to more complex tokamak components (tritium breeding blankets)
  - Coupling with other physics (CFD, MHD, co-deposition models, He transport...)
  - Coupling with external plasma codes (SOLEEDGE, SOLPS)

```
import FESTIM

parameters = {
    "materials": [
        {
            "E_D": 0.1,
            "D_0": 1,
            "id": 1
        }
    ],
    "traps": [],
    "mesh_parameters": {
        "initial_number_of_cells": 200,
        "size": 1,
        "refinements": [
        ],
    },
    "boundary_conditions": [
    ],
    "temperature": {
        "type": "expression",
        "value": 300
    },
    "solving_parameters": {
        "final_time": 100,
        "initial_stepsize": 0.1,
        "newton_solver": {
            "absolute_tolerance": 1e-10,
            "relative_tolerance": 1e-9,
            "maximum_iterations": 50,
        }
    },
}

output = FESTIM.run(parameters)
```