



F. Cabot, G. Fourestey - SCITAS

C. Sommariva, M. Toussaint, P. Ricci - SPC

S. Mannane, L. Serafin, E. Larrieux, S. Kenderdine - eM+

OUTLINE

- Real-time visualization : the FusReal Unreal Engine plugin
 - Summary of what state it was in last year
 - Updates
- New project : Digital Twin

UNREAL PLUGIN : FUSREAL

FUSREAL : INTRODUCTION

- Build real-time visualization solutions for Plasma Physics
- Science communication (both external and internal) visualization

FIRST VIDEO

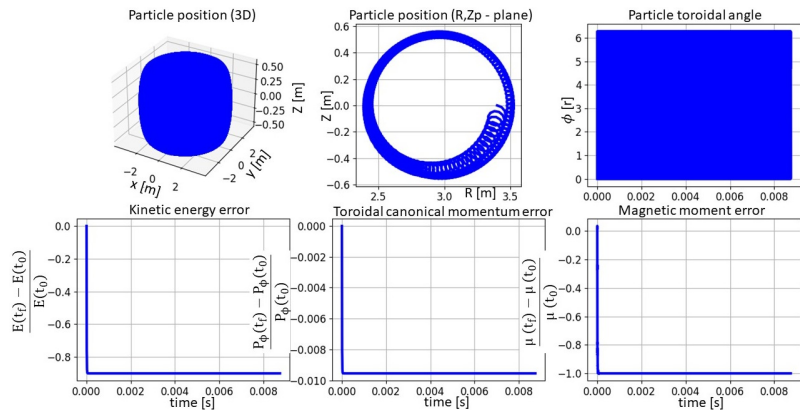
SECOND VIDEO

FUSREAL : MULTIMODAL

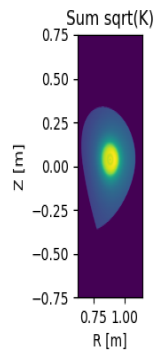
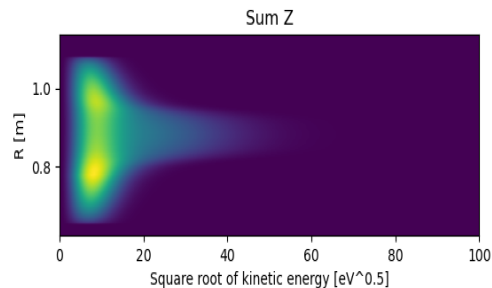
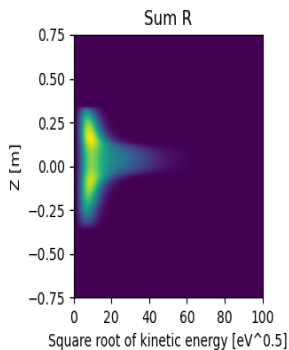


Use Boris method for electromagnetic force integration

Phase-space particles initialization
(separate or combined
position/velocity possible)



Used importance function (from $\sqrt{K}$ expression)



FUSREAL : COULOMB COLLISIONS

Coulomb collisions of tracked particles with background plasma particles

→ *Takizuka, T., & Abe, H. (1977). A binary collision model for plasma simulation with a particle code. Journal of computational physics, 25(3), 205-219.*

FUSREAL : REACTIONS

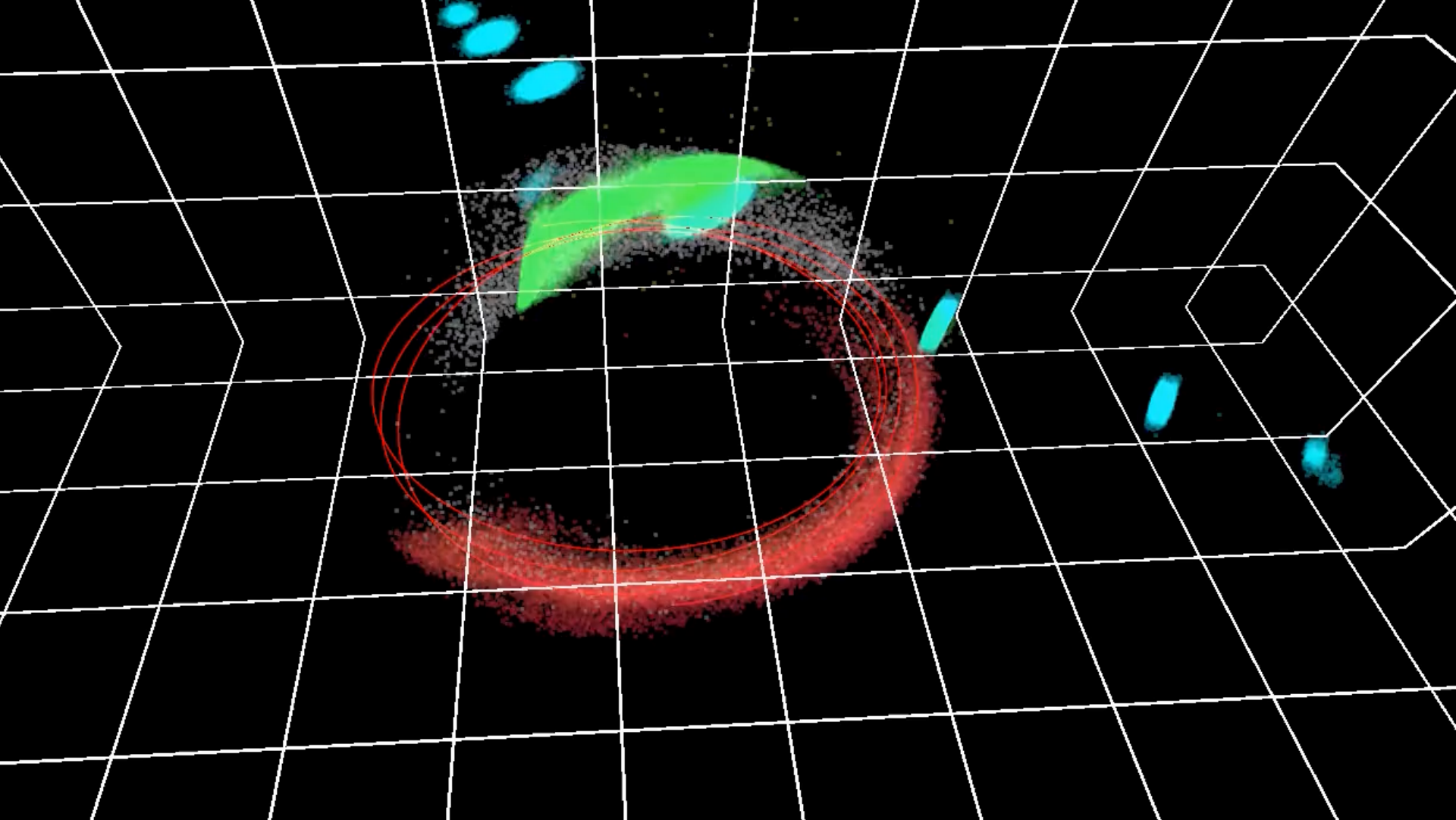
- NBI ionization using Suzuki method for reaction rates
→ *S Suzuki et al 1998 Plasma Phys. Control. Fusion 40 2097*
- Ported ASCOT5's BBNBI code to GPU for real-time ionization

FUSREAL : REACTIONS

- NBI ionization requires dynamic particle types
- Implemented a particle dynamic allocator/deallocator on the GPU (Pool allocator)
- Useful for other types of reactions

FUSREAL : REACTIONS

- Fusion reactions using Bosch-Hale cross section computation method for reaction rates
→ *Paula Sirén et al 2018 Nucl. Fusion 58 016023*
- Ported part of ASCOT5's `boschhale.c` code to GPU for real-time nuclear reactions
- Only reaction used for now is
 $D(+) + D(+) \rightarrow He3(2+) + n$
- Fast ions are missing so there's no natural fusion in our simulations yet



FUSREAL : REACTIONS

- CX reactions using OpenADAS interpolated cross sections
- No neutral particles in background plasma yet
- The rest is similar to other types of reactions
- Hard to visualize as-is : we thought of using another OpenADAS table to compute the amount of light emission associated to this reaction and color particles in an emission region for fast rendering

FUSREAL : BOUNDARY CONDITIONS

- Wall-particle collisions can be detected using Unreal collision detection directly
- For now, particles just bounce back with a perfect mirror-like equation from the wall but we are working on a physics-based model of specular reflections and absorption/emission based on the tile material and the particle type.

FUSREAL : INPUT FILES

- We want to distribute the software so we need a well-defined input format
- Our simulation has a lot of similarities with what ASCOT does, we pretty much need the same input
 - So let's use the same input
- ASCOT HDF5 input files can now be used as-is for visualization (2D magnetic fields only, for now)
- Added optional extensions to ASCOT format to specify X points for example (useful to compute background plasma boundaries)

FUSREAL : OPEN SOURCING

- The ASCOT HDF5 format made us rewrite most of the data structures (which was a good thing)
- RAI wrapper of the HDF5 C lib + high level loader from HDF5 dataset to GPU array to accelerate dev of future features
- Some of the logic was also rewritten to be more modular, which made testing and validation easier ; the test suite does « physics unit tests » and a Python pipeline can produce all the plots for validation from the written files

FUSREAL : OPEN SOURCING

<https://gitlab.epfl.ch/spc/public/fusreal>

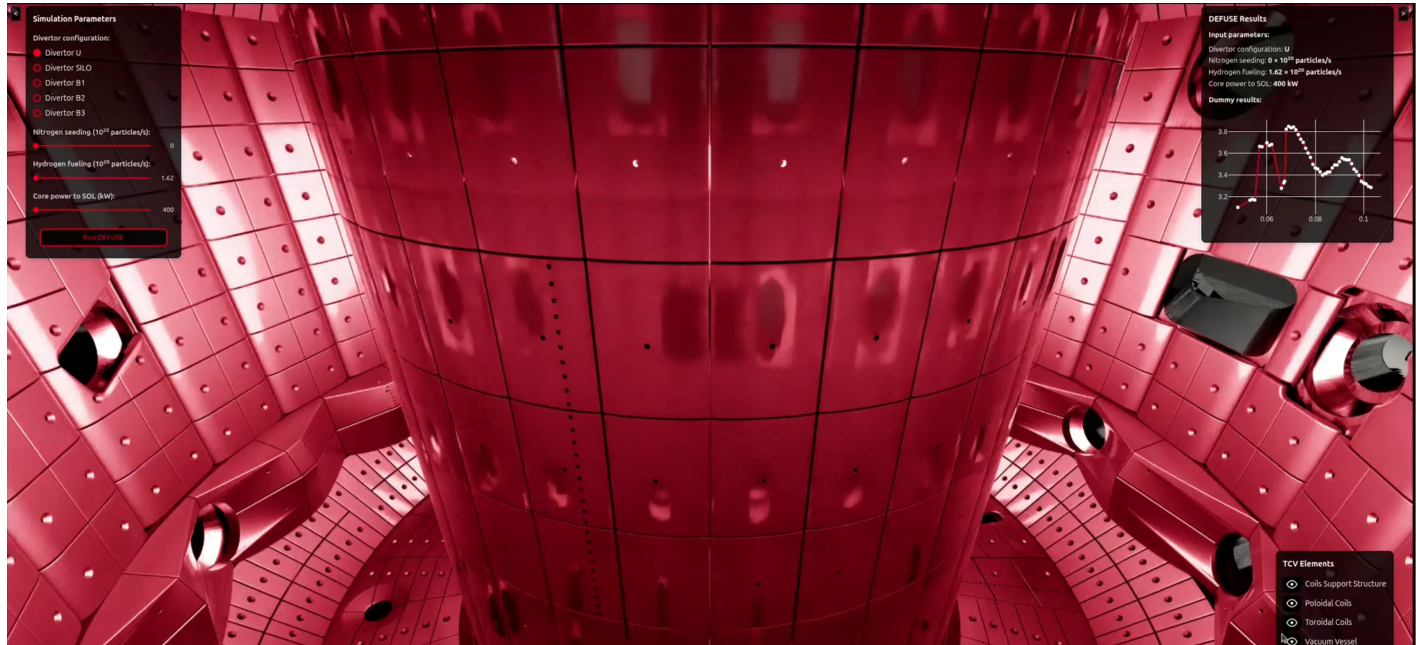
- We are still waiting on validation from the SPC to actually make it public, hopefully it will happen soon.
- Unreal Engine plugin : Bring Your Own Data, only the computation is open
- In principle : install the plugin in Unreal, drag and drop your ASCOT file and you're good to go (you'll also need the CAD of your machine of choice)
- Benefit of the plugin : you can write your own Unreal application with it

DIGITAL TWIN PROJECT

DIGITAL TWIN PROJECT

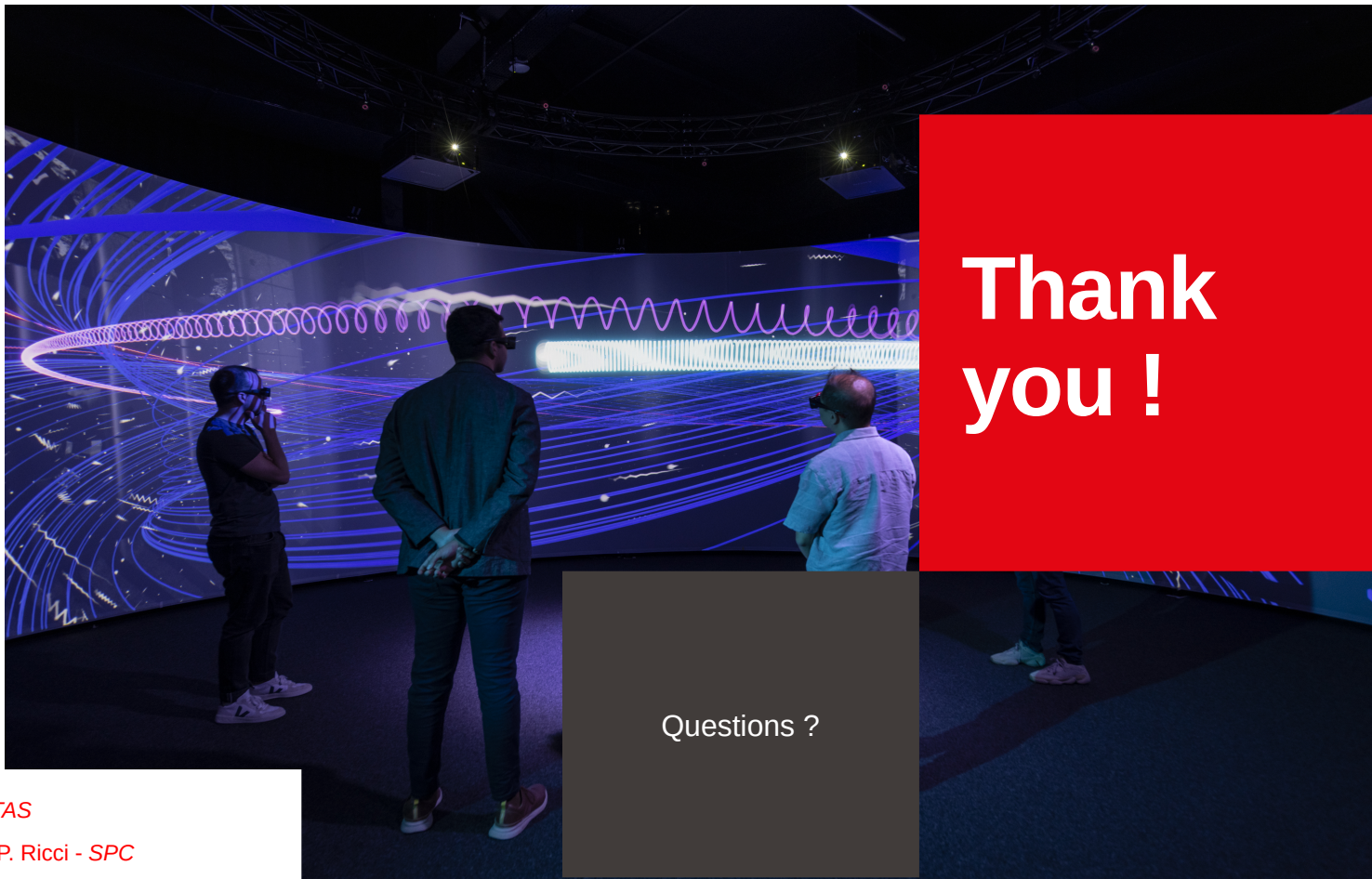
- Integration of multiple simulation codes in a single workbench, with visualization
- Our first goal is to simulate the behavior of a new piece of TCV : Tightly Baffled Long Legged Divertor (TBLLD)
- We need to integrate, at least : DEFUSE, Ansys, SOLPS and a visualization tool ; all running on different machines
- We collaborate with Nvidia for the backbone of the Digital Twin, using Omniverse
- My work is focused on ANSYS for now and some Omniverse-specific code

DIGITAL TWIN PROJECT (image from Cyrille Favreau - NVIDIA)



CONCLUSION

- Real-time GPU simulation of particle interactions with a background plasma model :
Boris integration, Coulombs collisions, NBI emission+ionization, magnetic field lines analysis, CX, nuclear reactions, wall interactions
- Missing one big feature to fully enable the WIP ones : fast ions + background neutrals ; the data structures are ready, we still need to recover the data
- Open-source ready, only requires administration approval
- We also have nice sound (Eric Larrieux)



**Thank
you !**

Questions ?

F. Cabot, G. Fourestey - SCITAS

C. Sommariva, M. Toussaint, P. Ricci - SPC

S. Mannane, L. Serafin, E. Larrieux, S. Kenderdine - eM+