ExCALIBUR 10

# PROJECT NEPTUNE (FUSION USE CASE)

Ed Threlfall, UKAEA (on behalf of the NEPTUNE team)

ExCALIBUR SIMLInt Workshop

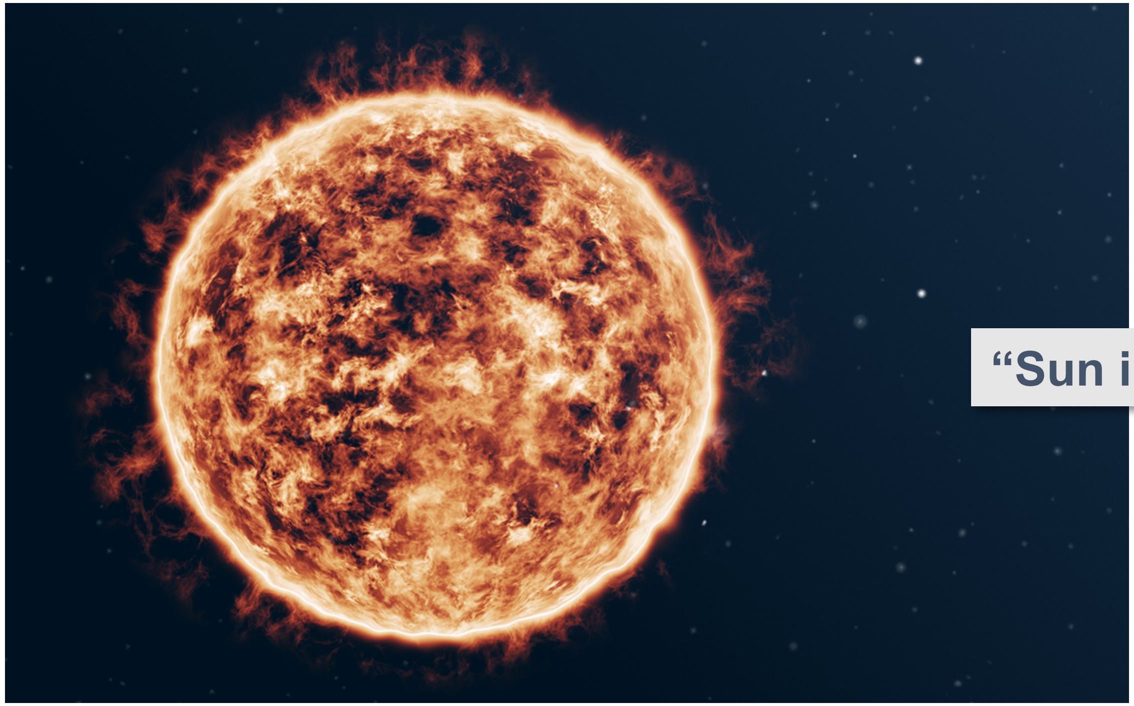
Edinburgh / Teams hybrid, 5 June 2023

Met Office

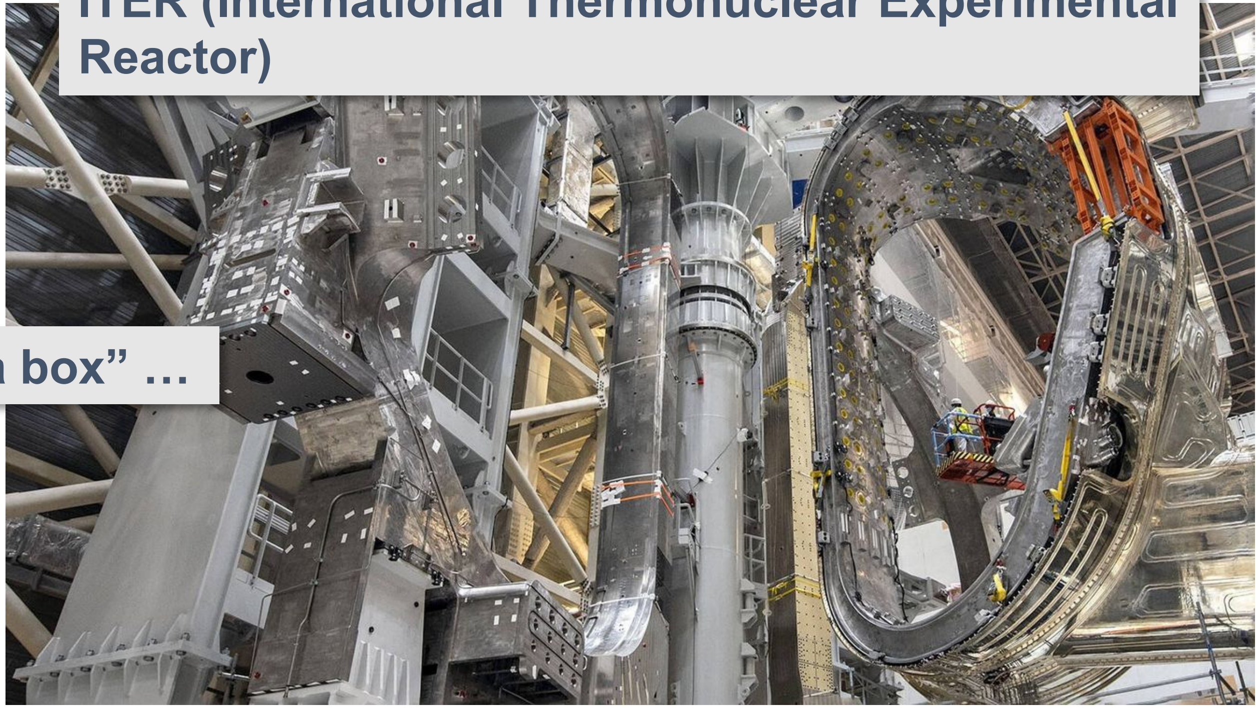
UK Research and Innovation

UK Atomic Energy Authority

# (Controlled) fusion energy



ITER (International Thermonuclear Experimental Reactor)



"Sun in a box" …

- Fusion is a potential sustainable energy solution.
- Magnetically-confined approach.
- Contained in a tokamak - a toroidal `box'.

# Challenges

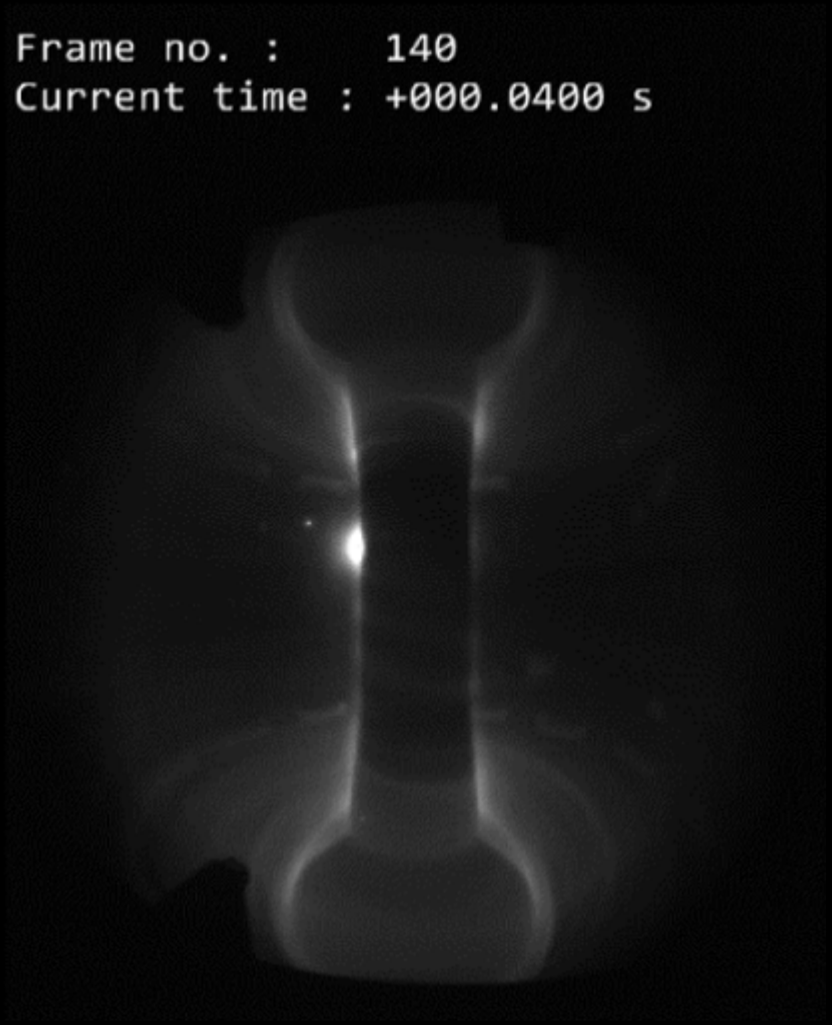
- ***Nature, and Nature's Laws lay hid in Night …***

100 million degrees (c.10X hotter than centre of sun)

10,000,000,000,000,000,000+ particles

1 million amps (c.30 lightning strikes)

some of the world's strongest magnets




non-equilibrium matter, sometimes not like a gas

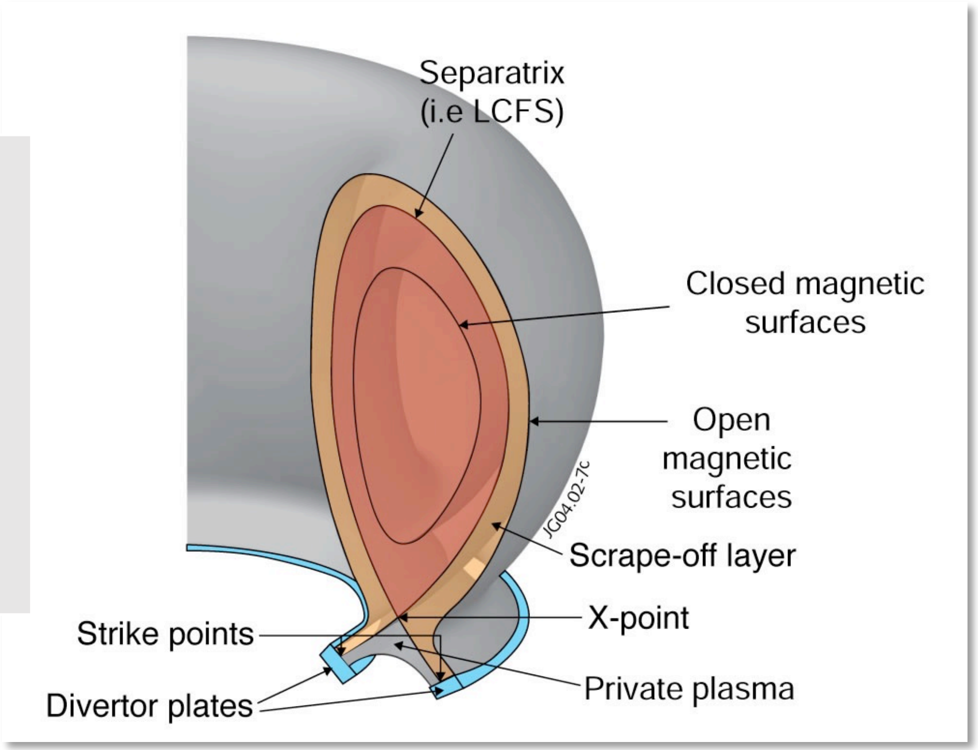
atomic physics of hundreds of different ions / atoms, etc.

turbulence and chaos - unsolved mathematics

# NEPTUNE (1)

- NEPTUNE focus is on “edge” region of plasma (the bit near the tokamak wall), not the core.
- Is open field lines region … means plasma interacts directly with the divertor.

- Division into two main simulation modules:

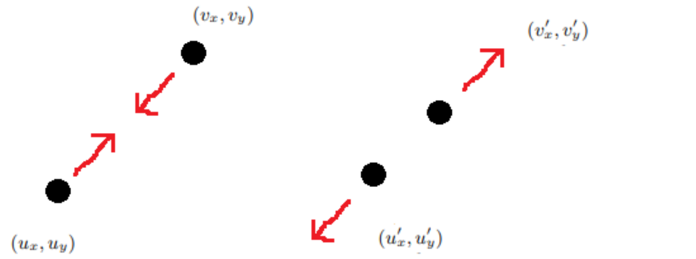



### Fluid physics

- Collisional regime.
- Navier-Stokes-like equations (resembles weather, with additional electromagnetic forces).

### Kinetic physics

- Non-collisional regime.
- Mean free path long …
- Boltzmann-like equations.




$$u'_x = \frac{1}{2}(u_x(1+\cos\chi) + v_x(1-\cos\chi) + \sin\chi(u_y - v_y))$$

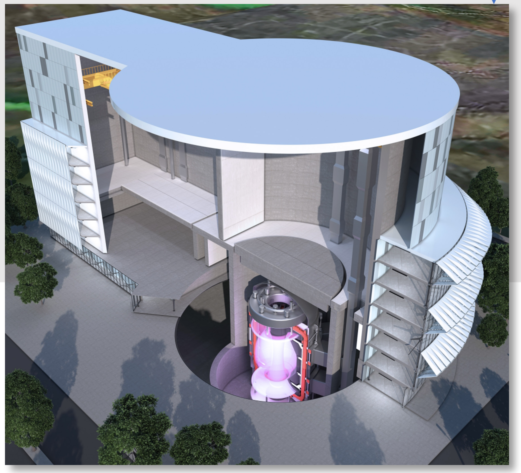
# NEPTUNE (2)

| | |
|---|---|
| Why digital models? | Tokamaks expensive! (ITER ~€20bn, initiated 1988). **Cannot test extreme scenarios.** |
| Existing software limitations | Finite-difference methods, not best suited to new HPC architectures. |

**NEutrals and Plasma TUrbulence Numerics for the Exascale**

- *'Once-in-a-generation' opportunity for UK to improve fusion plasma physics software processes in order to develop `actionable' code i.e. code suitable for incorporation in an engineering design workflow as required for STEP.* (Wayne Arter, Technical Lead.)
- Interdisciplinary team; 60-70% of work defrayed to UKRI.
- **Proxyapps development philosophy** (each solves part of the problem).
- **NEPTUNE is not currently one code.**

STEP (Spherical Tokamak for Energy Production) – fusion power on UK grid by 2040s.



# NEPTUNE – anticipated structure

- Fluid physics module - C++ finite element library, modelling plasma turbulence.
- Particles module – C++ library for out-of-equilibrium matter.
- Uncertainty quantification for actionable code – non-intrusive.

| | |
|---|---|
| Use | • Engineers, plasma physicists, software developers.<br>• Run on anything from single laptop to exascale.<br>• Aim for 30-year life-cycle. |
| Domain-specific language | DSL for user interaction - compose library components. |
| Languages | C++17 / SYCL 2020 for main code, Python for user DSL. |

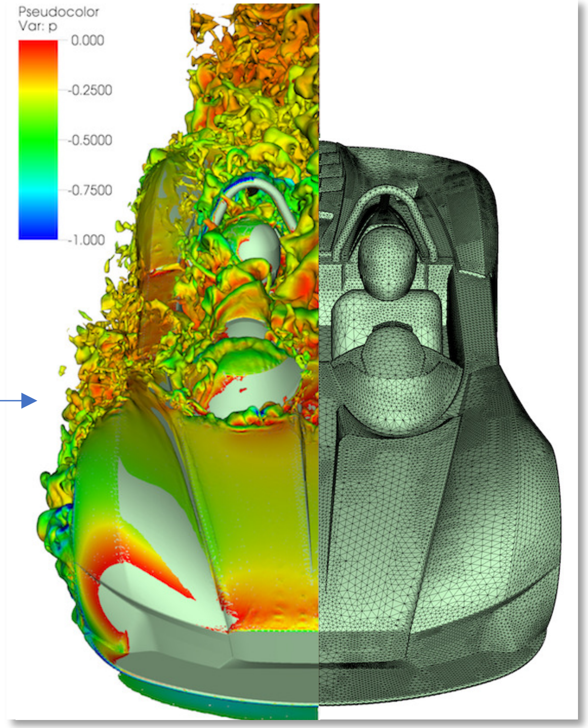
# Fluids: Nektar++ spectral / hp elements

NEKTAR++
SPECTRAL/HP ELEMENT FRAMEWORK

*Nektar*++ spectral / hp code [1] for solving plasma fluid equations.

- Arbitrary convergence order *p* – error ∝ $h^p$ ($h$~element size).
- Arithmetic intensity – increased number of operations on same data …
  **makes method suitable for modern HPC.**
- Supports complicated geometries, curved elements.

| | |
|---|---|
| Structure | • Set of libraries.<br>• C++ code with MPI parallelism for CPUs.<br>• Refactoring for performance portability / GPUs / C++17. |
| Provenance | • Proven scaling to c.100k cores.<br>• Well-tested code.<br>• Established community of developers / users. |




CFD simulation of Elemental RP1 track car.

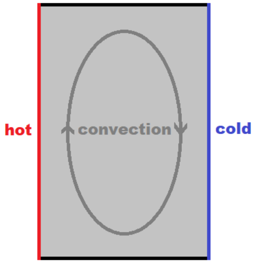
1. https://www.nektar.info

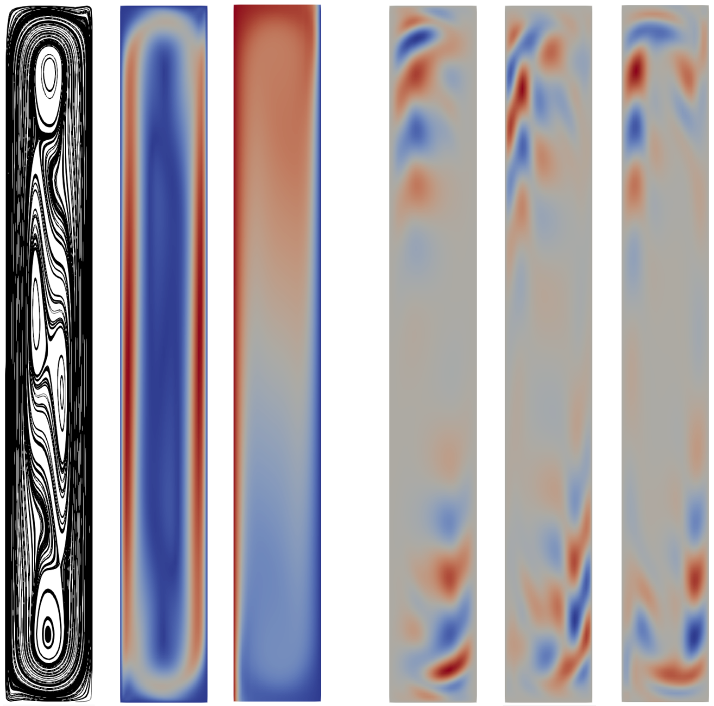
D. Moxey (King's College London); C.D. Cantwell, S.J. Sherwin (Imperial College London)

# Nektar++: modelling heat transfer

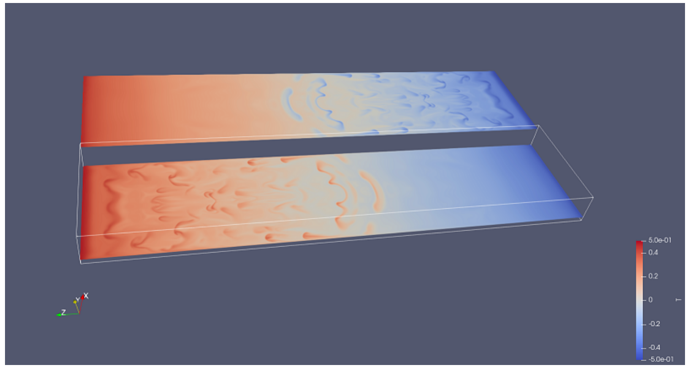
•Fluid tank heated from side (SmalLab apparatus, D. Buta, Leeds Univ., W. Arter, UKAEA). New capabilities to measure details of convective flow.

Very rich and interesting problem, related to heat transfer in fusion machine …
(see e.g. *Onset of interchange instability in a coupled core-SOL plasma* Wilczynski *et al,* 2020).

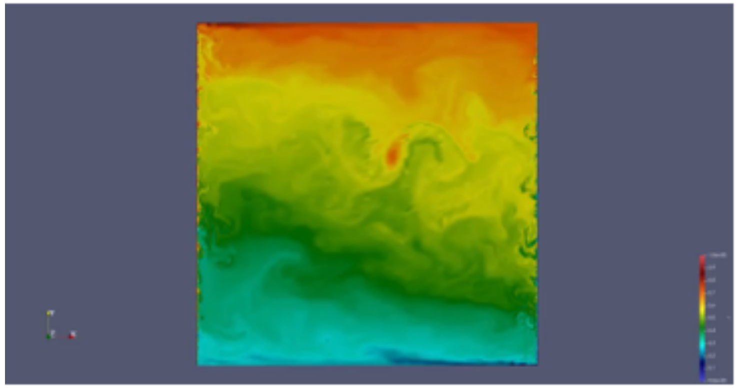


(L-R) Nektar++ streamlines, flow speed, temperature of background laminar flow; x- and y- velocity components and temperature for emerging time-dependent flow at first Hopf bifurcation. After Zucatti V., Lui H.F.S., Pitz D.B., Wolf W.R. (2020).



Nektar++ 3D flow simulation, actual tank geometry, run on Archer2 (Seimon Powell, UKAEA).

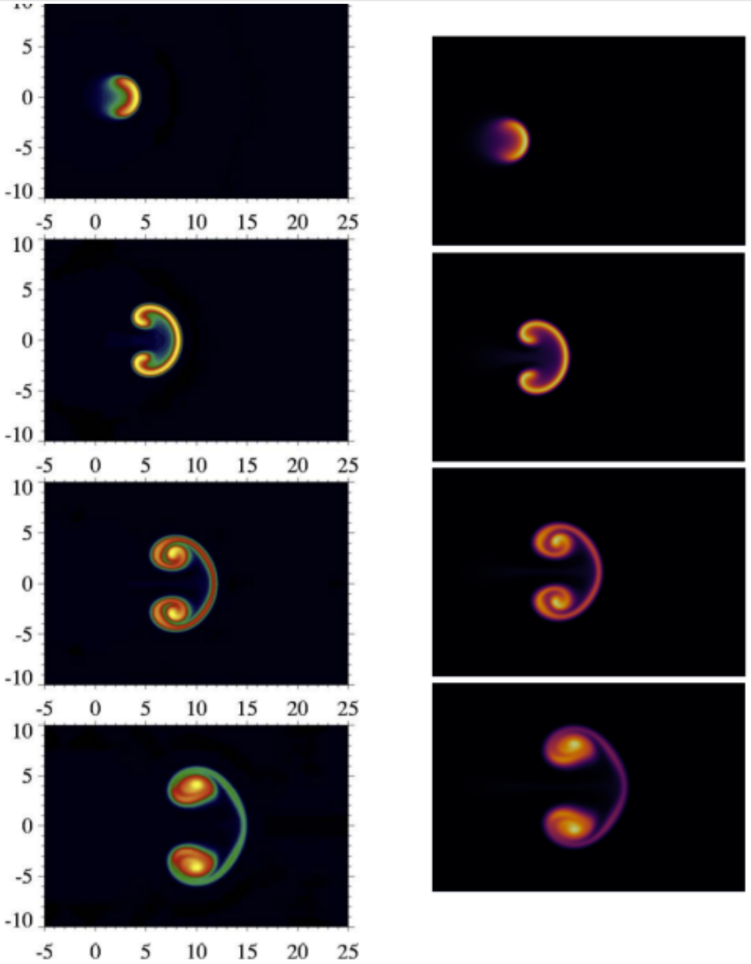


Flow in square slot heated on left, cooled on right (animation).

ExCALIBUR

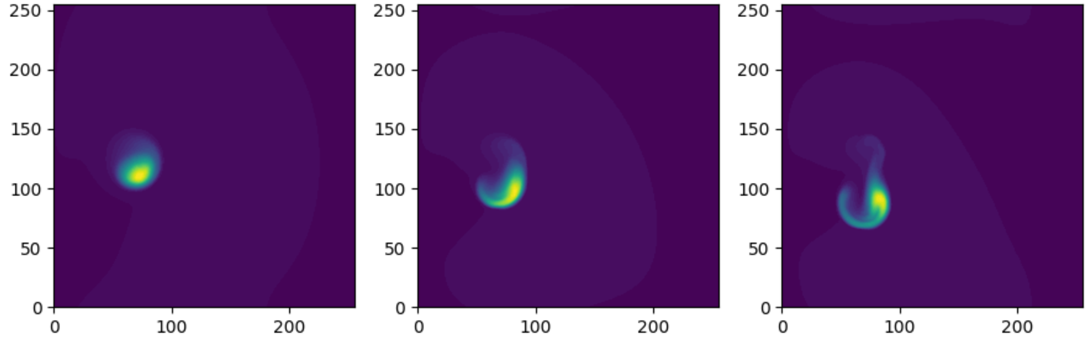
# Nektar++ plasma applications

- Edge plasma filament in Nektar++, cf. literature. Left: Garcia O.E., Bian N.H., Fundamenski W. 2006; right: Nektar++, 2023.

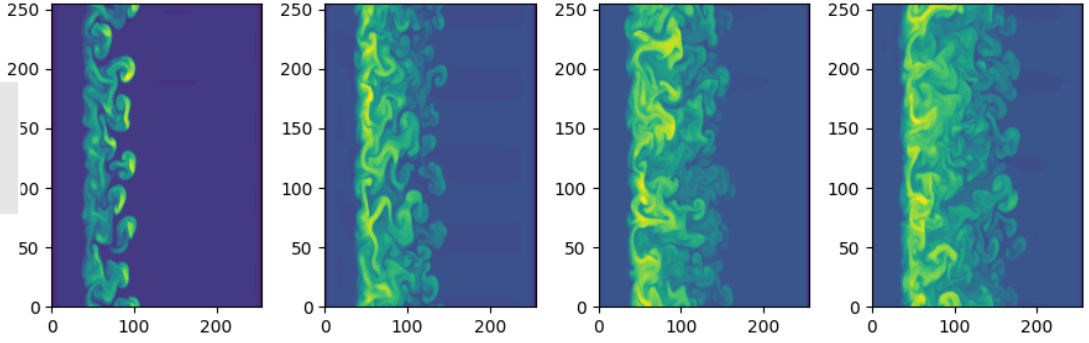


- Currently working towards implementing Nektar++ versions of examples from Hermes-3 finite-difference-based plasma code (Hermes-3 BOUT++ -based):

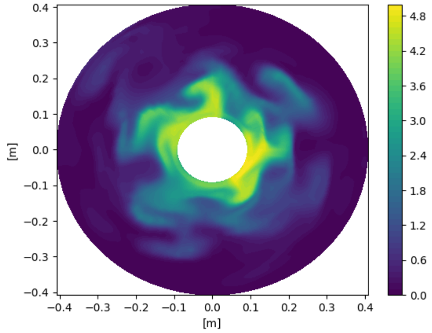
2D filament



2D turbulence

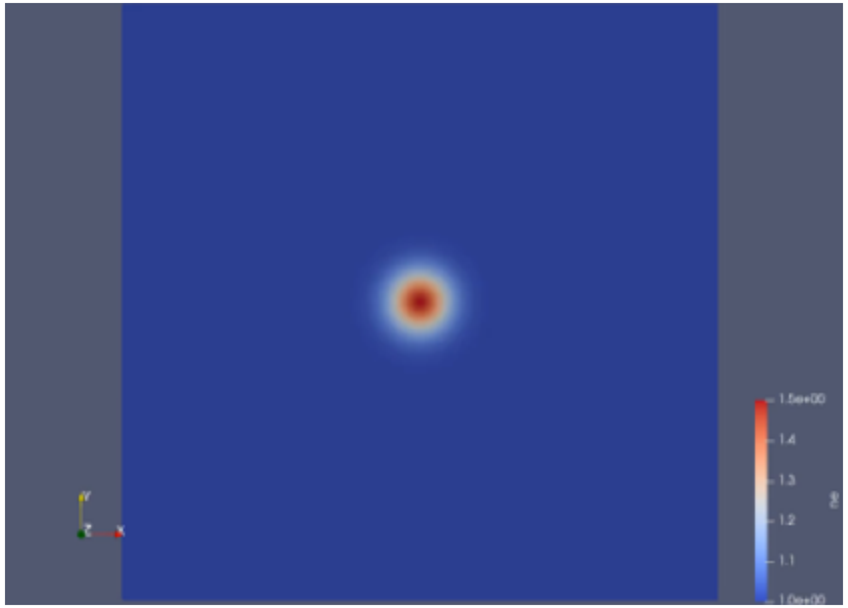


3D linear fusion device, nice 3D turbulence example, runs in 8 core-hours …



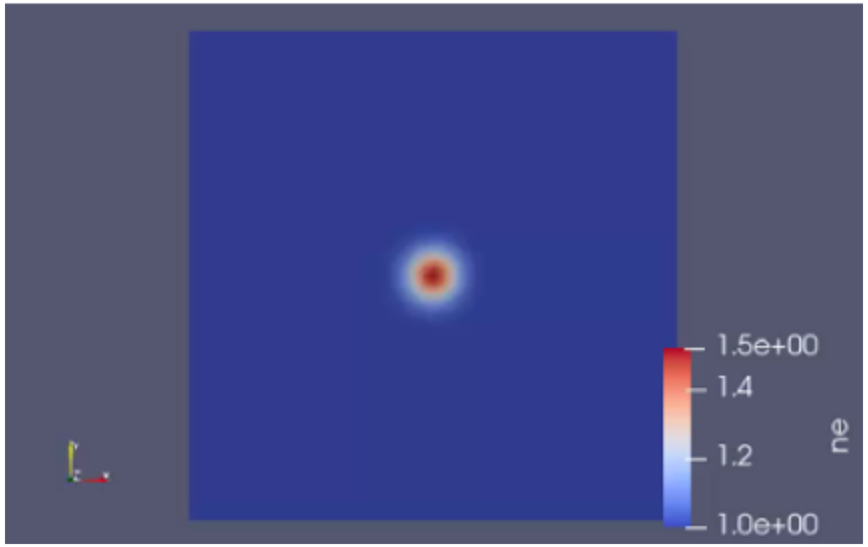
# Nektar++ plasma applications

- 2D examples from Hermes-3 implemented in Nektar++

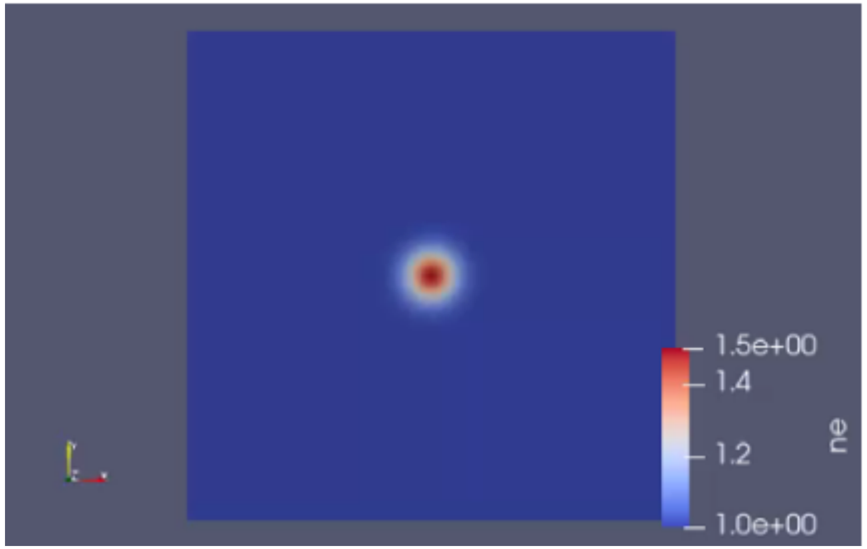


Blob2d isothermal

blob2D Te / Ti

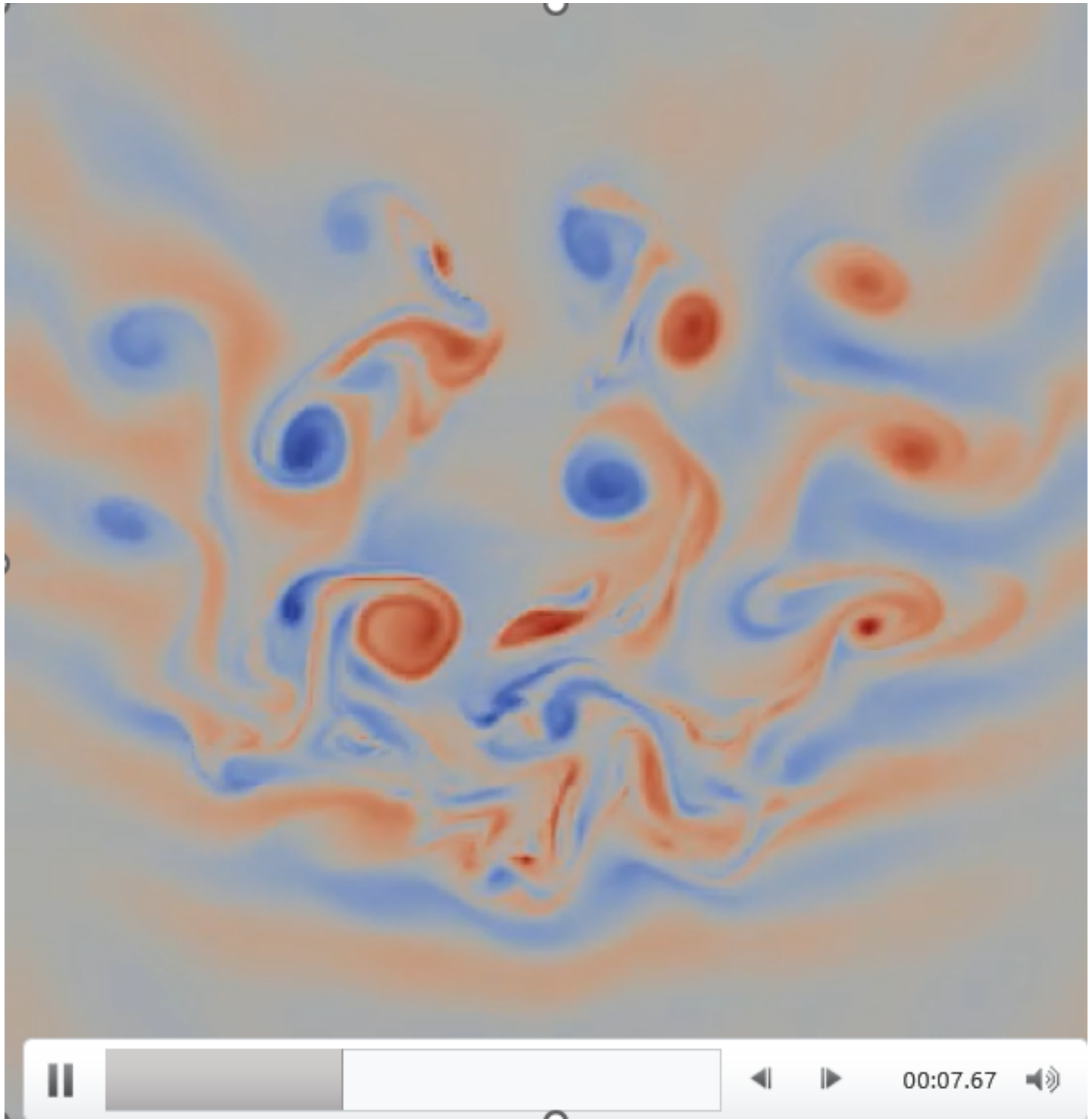



2 modes




3 modes

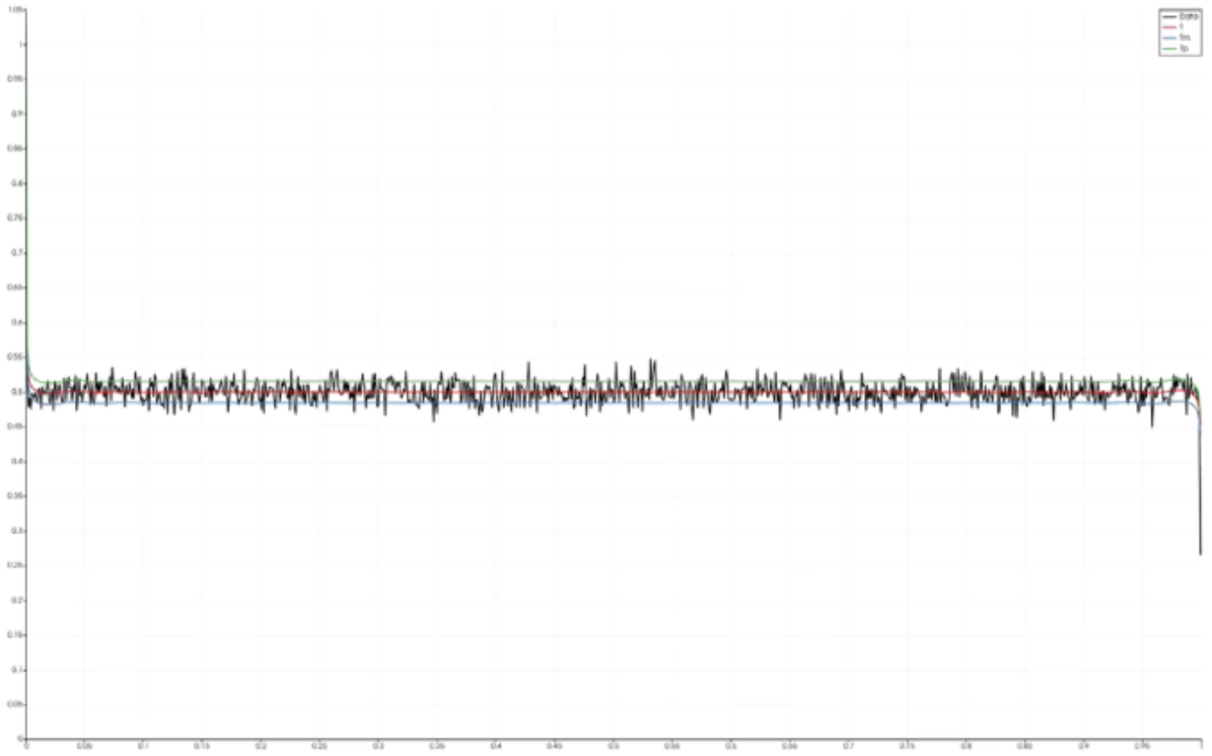
# Nektar++ plasma applications




- Hasegawa-Wakatani equations implemented in Nektar++

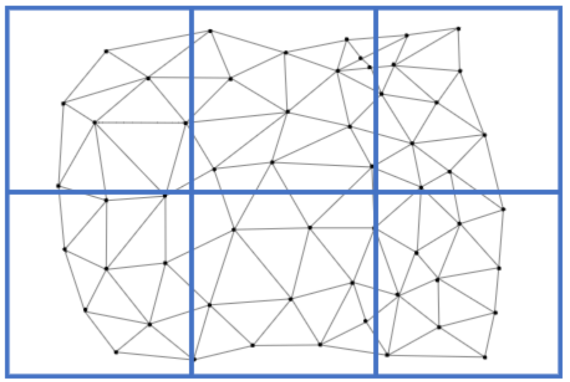
# Kinetics: particles

- Particle methods can be used to solve PDEs …
- Choose particles if continuum too expensive (high dimensions: 6D phase space for 3 space dimensions).
- Pedagogical example: diffusion / heat equation $\frac{\partial T}{\partial t} = D\frac{\partial^2 T}{\partial x^2}$, constant flux at boundaries …
- Main problem is statistical noise (example has $1024^2$ particles).



# NESO-Particles

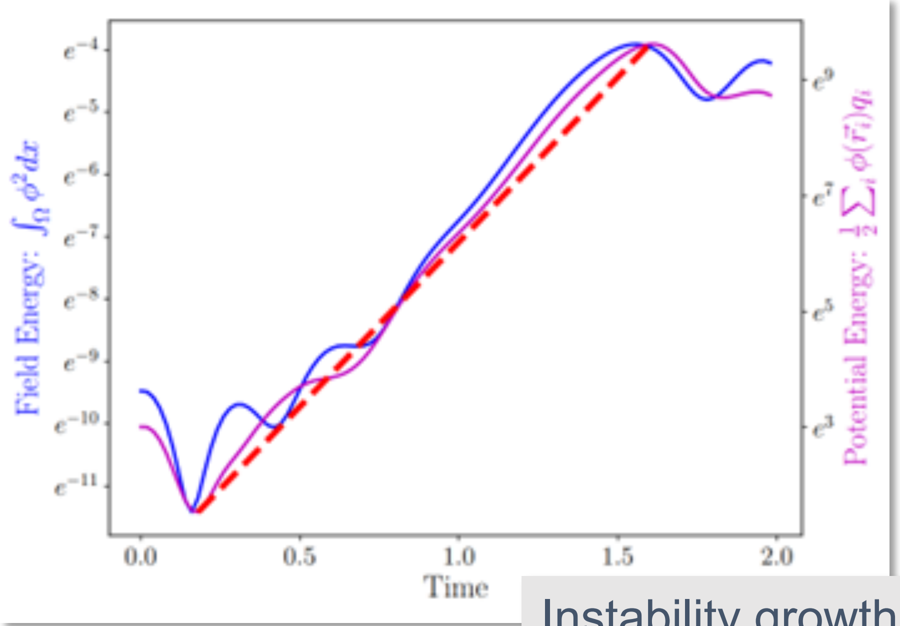
| | |
|---|---|
| **NESO-PARTICLES [1]**<br><br>**(NESO = NEPTUNE Experimental SOftware)** | • UKAEA header-only C++ library for particle data and moving particles between MPI ranks on unstructured meshes.<br>• Particle-mesh interface abstract - different mesh implementations possible (including *Nektar*++ meshes).<br>• Charged and neutral particles. |
| **Coupling** | Couples to finite-element *Nektar*++ code (tightly coupled currently; also exploring coupling frameworks). |
| **Dependencies** | • CMake<br>• SYCL 2020 (tested with hipSYCL 0.9.2 and Intel DPCPP 2022.1.0): **performance-portable**<br>• MPI 3.0 (tested with MPICH 4.0 and IntelMPI 2021.6)<br>• HDF5 (optional, if particle trajectories required) |



1. https://github.com/ExCALIBUR-NEPTUNE/NESO-Particles

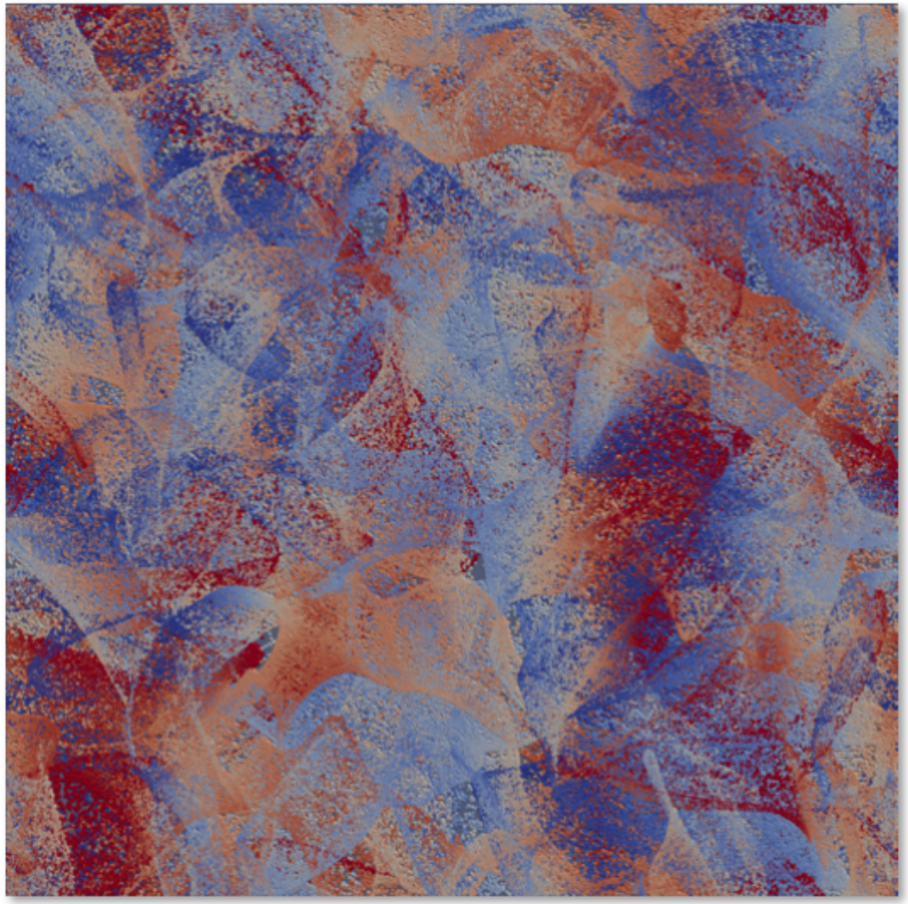
# Particles proxyapps

| | |
|---|---|
| **NESO [1]** | • Test implementations integrating particle capabilities and FEM.<br>• Can be built using Spack package manager.<br>• 2D2V electrostatic particle-in-cell solver.<br>• Nektar++ provides Poisson solve. |
| Tests | • Linear growth rates of unstable modes.<br>• Energy conservation. |



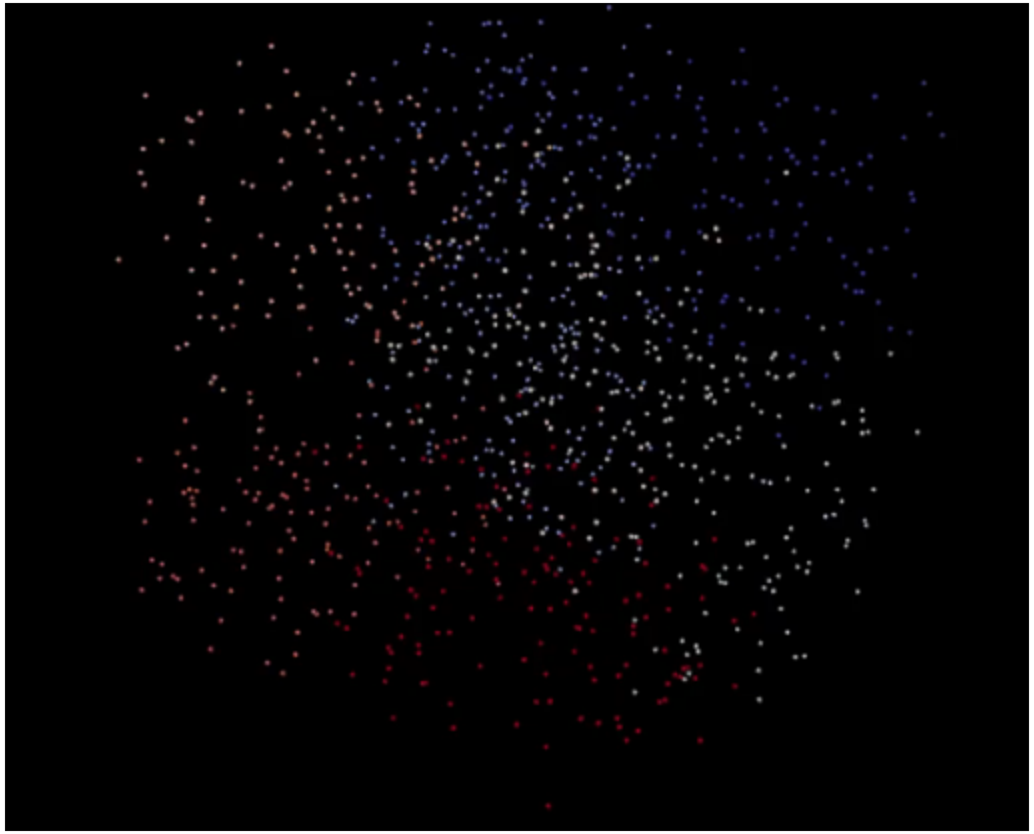

Instability growth rate vs theory

Time evolution of 512k interacting particles.



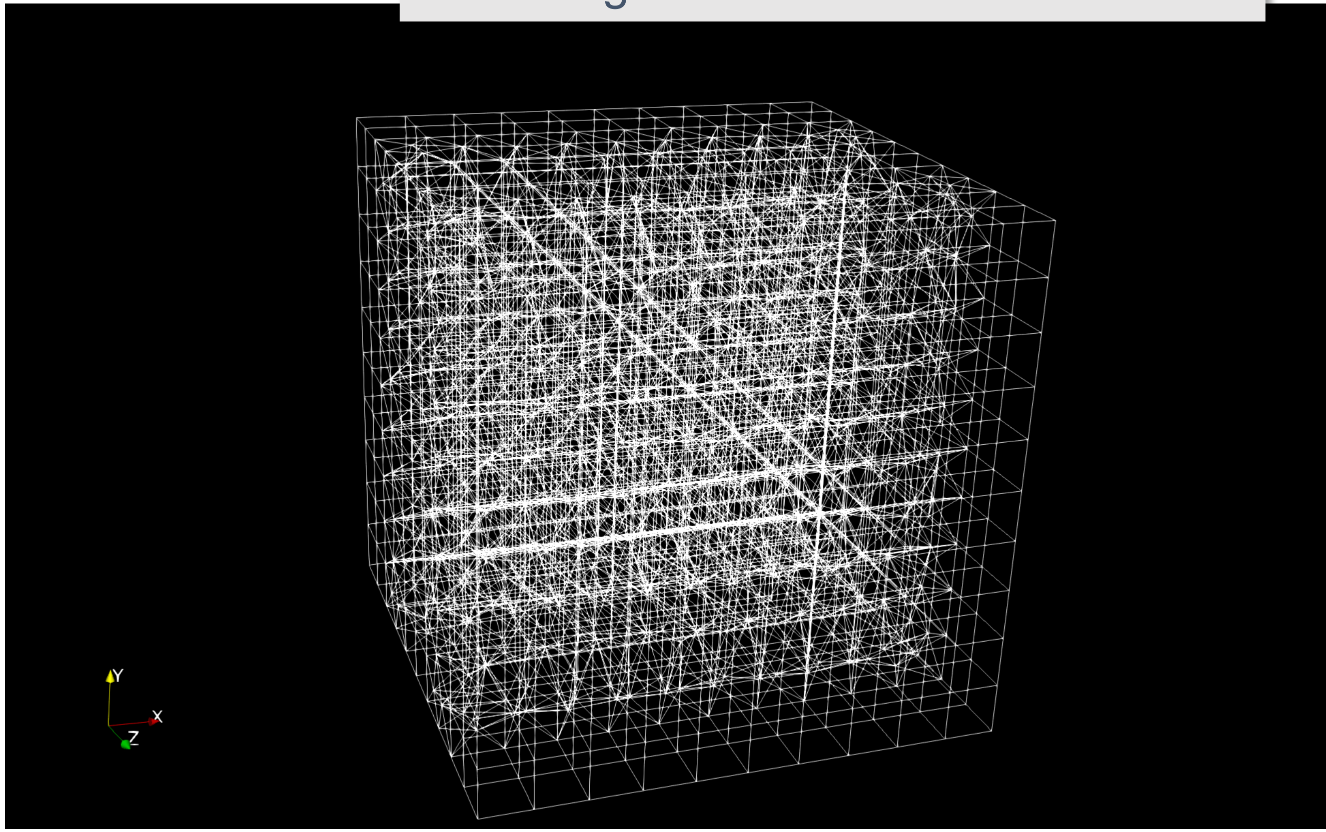
1. https://github.com/ExCALIBUR-NEPTUNE/NESO

# Particles proxyapps



Time evolution colour-coded by MPI rank.

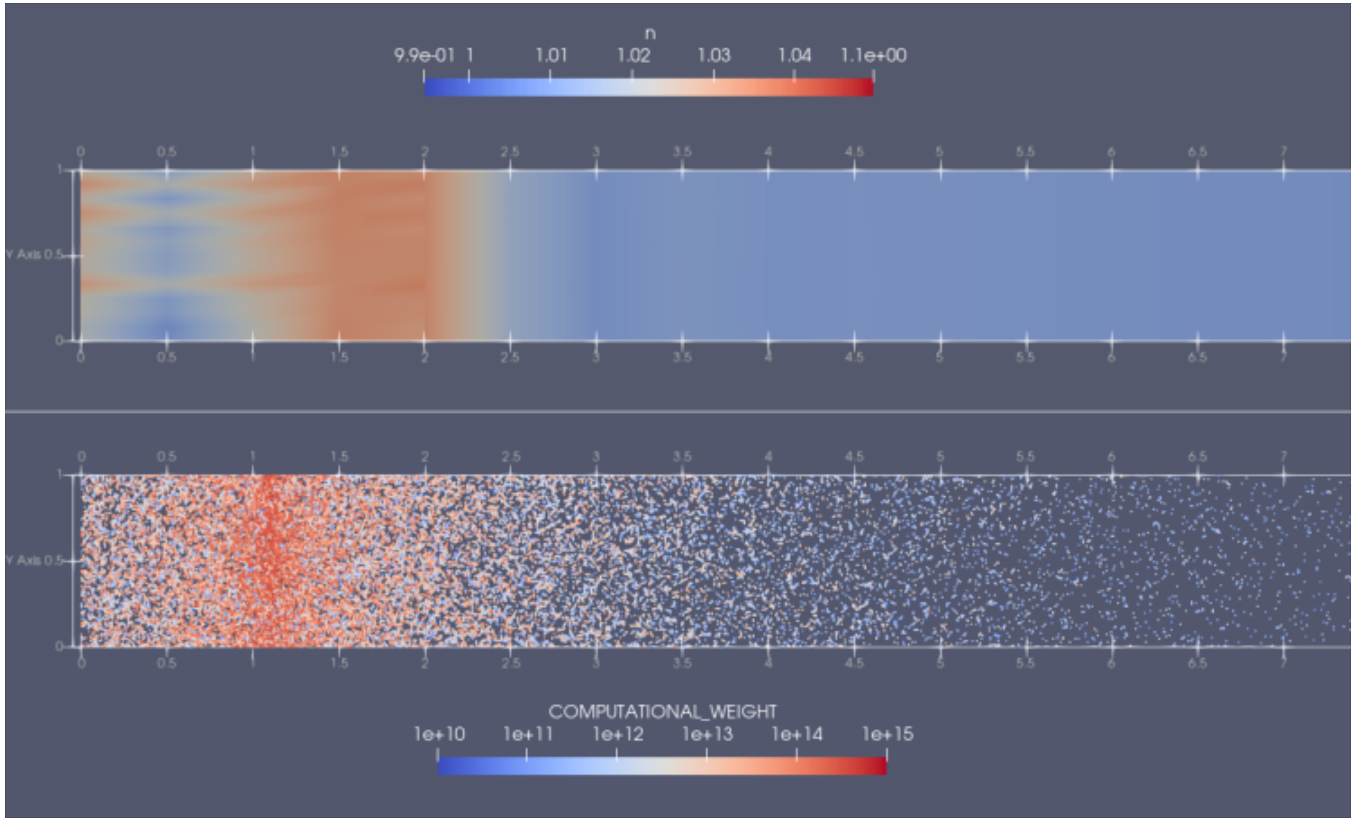
Currently implementing all required element geometries for 3D …




1. https://github.com/ExCALIBUR-NEPTUNE/NESO

# Coupled particles-fluids proxyapp

- Proxyapp coupling Nektar++ fluid code and neutral particles from NESO-Particles.
- Neutrals interact with plasma background, ionize to modify local density.
- Fluid model is advection-diffusion-reaction, turbulence to be added.




1. https://github.com/ExCALIBUR-NEPTUNE/NESO

# Actionable code requires Uncertainty Quantification

| | |
|---|---|
| VVUQ | • Meaningful bounds on error in code outputs given statistical uncertainty in inputs.<br>• Ensemble-based execution patterns.<br>• Non-intrusive UQ – separation of concerns. |
| SEAVEA project synergy, SEAVEA toolkit [1] | • UQ campaigns on HPC.<br>• Construction of surrogate models e.g. Gaussian process models, machine-learning.<br>• Modern data science techniques e.g. data assimilation, Bayesian analysis. |
| **FabNEPTUNE [2]** | • FabSim3 plugin created specifically for NEPTUNE.<br>• Easy execution of NEPTUNE simulations, integrated with SEAVEA toolkit.<br>• Currently drives *Nektar*++ 2D and 3D convection proxyapps - but more specific plasma applications very soon. |

*Nektar*++ convection proxyapp + VVUQ to support *Smallab* experiments (Arter, Buta – Univ. Leeds)

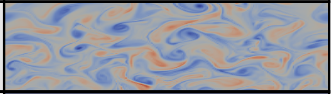
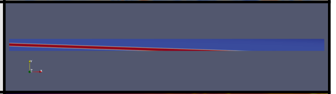
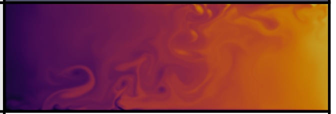
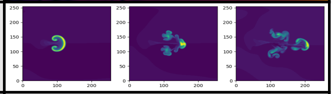
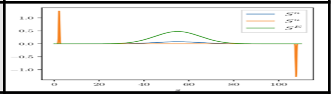
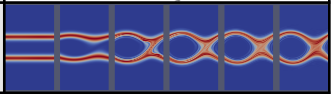
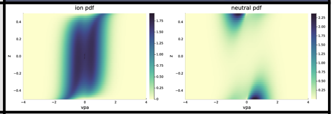
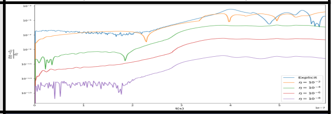
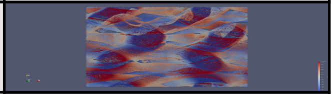
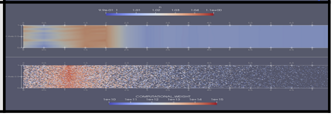
FabNEPTUNE


1. https://www.seavea-project.org/seaveatk
2. https://github.com/UCL-CCS/FabNEPTUNE


S. Guillas, P.V. Coveney, K. Bronik (UCL)

ExCALIBUR

# Proxyapps inventory

| Proxyapp | Framework | Language | Comments | Sample output |
|---|---|---|---|---|
| nektar-driftwave | *Nektar++* | C++ | 2D Hasegawa-Wakatani equations |  |
| nektar-diffusion | *Nektar++* | C++ | strongly anisotropic diffusion |  |
| vertical natural convection in spectral / hp, 2D and 3D | *Nektar++* | C++ | incompressible Navier-Stokes with buoyancy |  |
| 2D plasma turbulence equations in spectral / hp | *Nektar++* | C++ | *Hermes-3* equation system |  |
| 1D fluid solver with UQ and realistic boundary conditions | *Nektar++* | C++ | 1D model of scrape-off layer |  |
| 1D1V &1D3V Vlasov-Poisson kinetic solver in spectral / hp | *Nektar++* | C++ | Due May 2023 |  |
| moment-kinetics | new code (Univ. Oxford) | Julia | moment-kinetic gyro-averaged code |  |
| minepoch | *EPOCH* (Univ. Warwick) | Fortran | used for testing particle implementations |  |
| electrostatic PIC proxyapp | NESO-Particles | C++ / SYCL | |  |
| 2D3V coupled fluids-neutral particles proxyapp | *Nektar++* and NESO-Particles | C++ / SYCL | neutrals in plasma fluid background with ionization |  |

# Community overview

| | |
|---|---|
| UKAEA TEAM | Rob Akers, Wayne Arter, Matthew Barton, James Cook, John Omotani, Joseph Parker, Owen Parry, Will Saunders, Ed Threlfall. |
| UKRI GRANTS | • University of Exeter (VVUQ, surrogate models): Peter Challenor, Tim Dodwell, Louise Kimpton.<br>• King's College London (Nektar++): Mashy Green, David Moxey.<br>• Imperial College London (Nektar++): Chris Cantwell, Bin Liu, Spencer Sherwin.<br>• University of Oxford: Michael Barnes, Patrick Farrell, Michael Hardman.<br>• STFC Hartree Centre: Vasil Alexandrov, Hussam al-Daas, Tyrone Rees, Emre Sahin, Andrew Sunderland, Sue Thorne.<br>• University College London (VVUQ): Kevin Bronik, Peter Coveney, Matt Graham, Serge Guillas, Tuomas Koskela, Yiming Yang.<br>• University of Warwick (DSLs): Gihan Mudalige.<br>• University of York (plasma physics, support & coordination, DSLs): David Dickinson, Ed Higgins, Chris Ridgers, Steven Wright. |
| ALUMNI | • University of Oxford: Felix Parra-Diaz.<br>• University of Warwick (EPOCH): Ben McMillan, Tom Goffrey.<br>• University of York: Ben Dudson. |
| OUTPUT (INC. CODE) | • Proxyapps code (MIT licence): see repositories on https://github.com/ExCALIBUR-NEPTUNE (some, inc. NESO and NESO-Particles, are public).<br>• Large body of supporting documents and reports – https://github.com/ExCALIBUR-NEPTUNE/Documents (currently private).<br>• Developer website in development. |