

**ExCALIBUR  
10**

# **PROJECT NEPTUNE NESO: NEPTUNE EXPLORATORY SOFTWARE**

**James Cook (presenting this)  
Will Saunders (presenting next)  
+ UKAEA team**

**ExCALIBUR Workshop, Bristol, 11 Oct 23.**



---

# NESO: High level goals

## Neptune Exploratory Software

Bring together key technologies as foundational building blocks

- Nektar++ for its high-p DG/CG capabilities to solve generic PDEs
- NESO-Particles for SYCL-enabled charged and neutral particles
- NESO brings together Nektar++ and NESO-Particles to build mini-apps
  - An electrostatic PIC code
  - A mean field exhaust relevant system of equations
  - The turbulent reduced MHD Hasegawa-Wakatani equations
  - Capability in 2D and 3D!



# The NESO Repositories

## Software development

The screenshot displays six GitHub repository cards for NESO components. Each card includes the repository name, visibility status, programming language, star count, license, fork count, issue count, and last update date. A green activity line graph is shown to the right of each repository name.

Repository Name	Visibility	Language	Stars	License	Forks	Issues	Updated
NESO	Public	C++	4	MIT	3	49 (2 issues need help)	4 days ago
NESO-Particles	Public	C++	1	MIT	1	7	3 weeks ago
NESO-fame	Public	Python	1	GPL-3.0	0	2	3 weeks ago
NESO-Spack	Public	Python	3		0	1	Jul 12
NESO-UQ	Public	Python	0	MIT	0	0	May 26
NESO-Reactions	Private	Python	0	MIT	0	3	Feb 28

Top level framework

Particles with MPI & GPUs

Field Aligned Mesh Extrusion – for  
Gridding along magnetic field lines.

Make NESO, Nektar++, NESO-  
Particles easier to build with SYCL

Nascent Uncertainty Quantification  
repo

Placeholder for neutral particle  
reactions code

# NESO: Status

## Neptune Exploratory Software

- Prototype NEPTUNE code
  - Fluid (with Nektar++) and/or particles (with NESO-Particles)
  - Implemented in SYCL (both Intel oneAPI and hipSYCL)

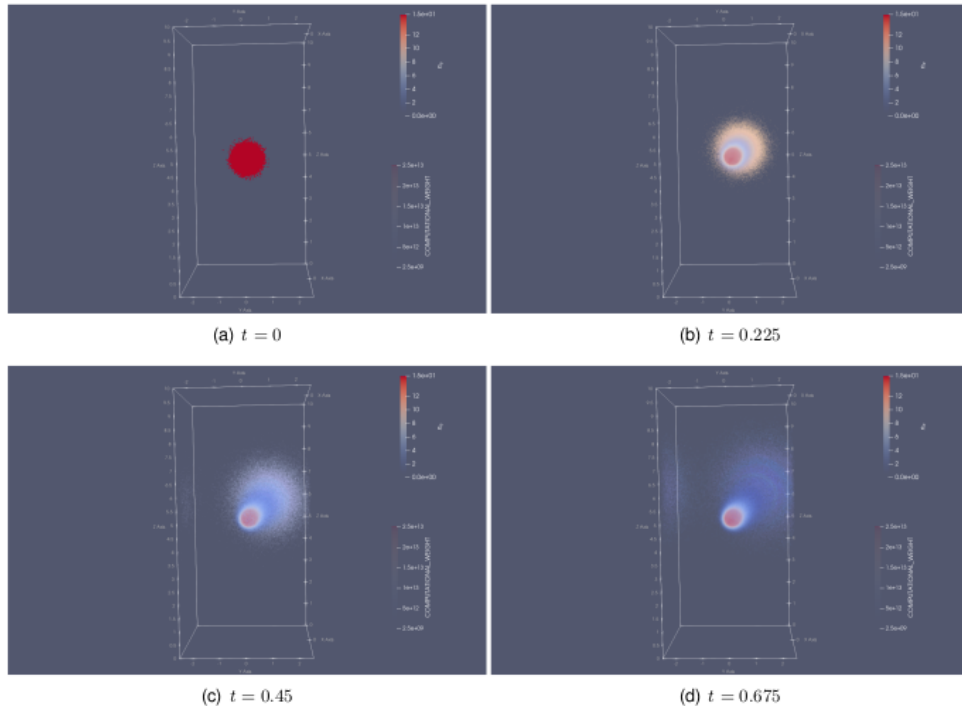


Figure 3: Evolution of the particle distribution (points) and electron density (coloured contours) in the coupled 2D-in-3D Hasegawa Wakatani solver. Particles are coloured according to their computational weight, corresponding to the number of physical neutrals that they represent. Captions under each panel indicate the simulation time.

[http://en.wikipedia.org/wiki/Neso\\_\(moon\)](http://en.wikipedia.org/wiki/Neso_(moon))

<http://github.com/ExCALIBUR-NEPTUNE/NESO>

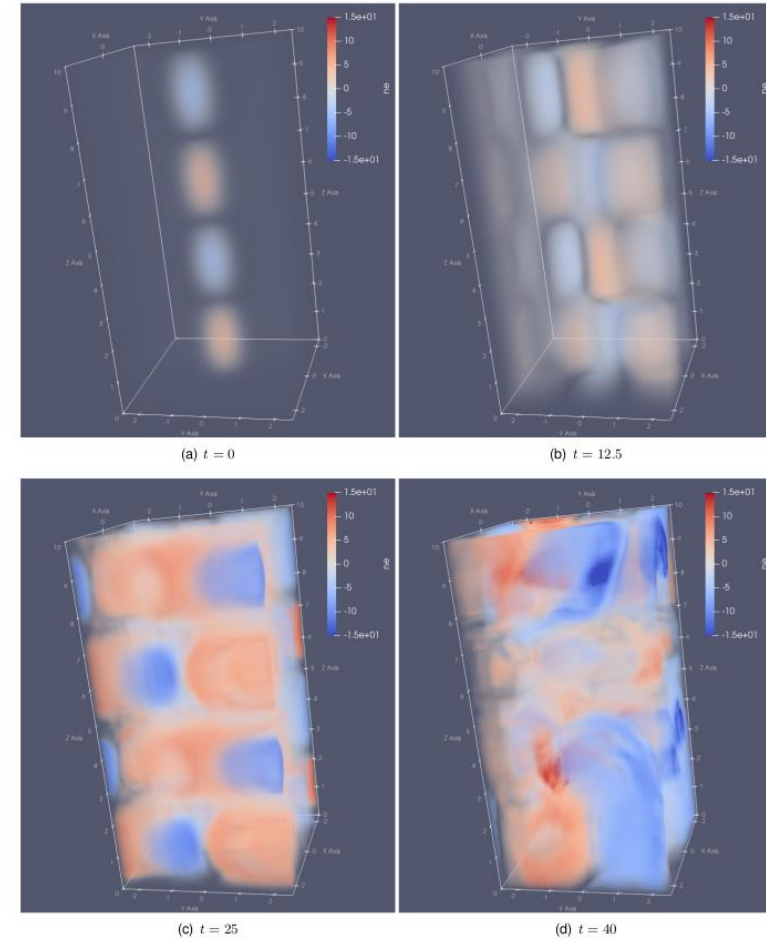


Figure 2: Evolution of the electron density in the fluid-only 2D-in-3D Hasegawa Wakatani solver.

# ExCALIBUR-NEPTUNE

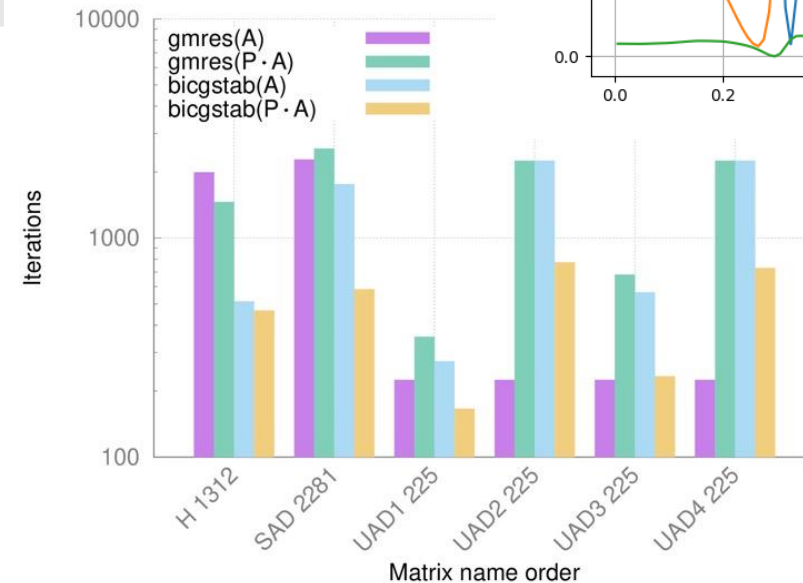
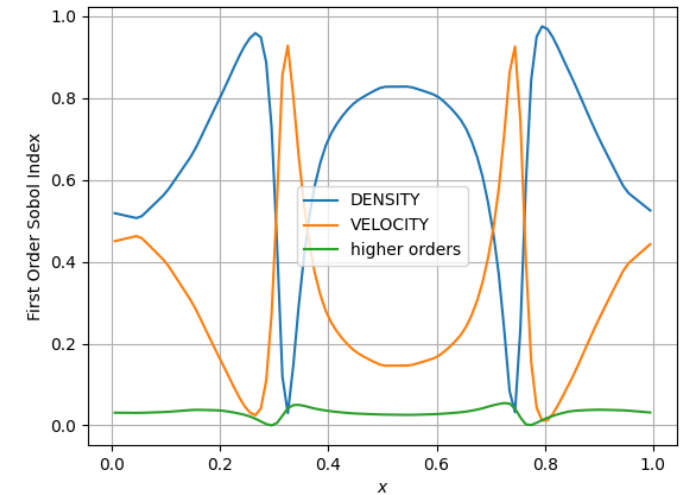
## NEutrals & Plasma TUBulence Numerics for the Exascale

- Harness the code with a DSL for rapid physics exploration

```
1 ddt(rho) = -V_dot_Grad(v, rho) - rho*Div(v);  
2 ddt(p)   = -V_dot_Grad(v, p) - g*p*Div(v);  
3 ddt(v)   = -V_dot_Grad(v, v) + (cross(Curl(B),B) - Grad(p))/rho;  
4 ddt(B)   = Curl(cross(v,B));
```

- Unobtrusive UQ ("error bars") and surrogates ("fits" that are quick to evaluate) will render the code base actionable for engineering design.

- Leverage fundamental research at the forefront of science from advanced preconditioner methods, parallel-in-time and neural methods.



# NESO-Spack

## Building dependencies from scratch

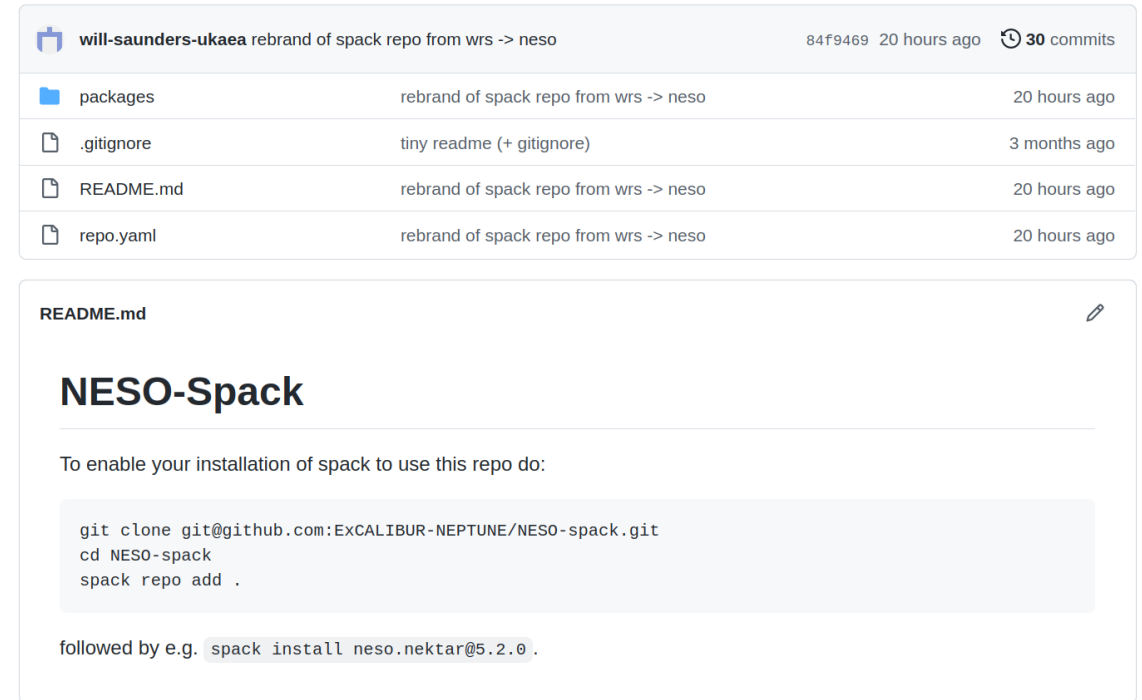
Users are able to build nektar++ against:

- Intel's **oneapi** SYCL implementation
- AMD's **hipsycl** SYCL implementation
- particular compilers, e.g.
  - **gcc@11.2.0**
    - **oneapi@2022.1.0** (i.e. dpcpp / icpx)
- **python** for NekPy
- your choice of MPI e.g. **mpich**

One can make a module file; `module load neso-hipsycl`

```
spack install --keep-stage -j 4 neso.nektar@5.2.0-f1598d ^python ^mpich %gcc@11.2.0
spack install --keep-stage -j 4 neso.hipsycl@0.9.2 %gcc@11.2.0
```

```
spack install --keep-stage -j 4 neso.nektar@5.2.0-f1598d +mkl %oneapi@2022.1.0 ^intel-
oneapi-mpi %oneapi@2022.1.0 ^intel-oneapi-mkl %oneapi@2022.1.0 ^python
```



will-saunders-ukaea rebrand of spack repo from wrs -> neso 84f9469 20 hours ago 30 commits

File	Commit	Time
packages	rebrand of spack repo from wrs -> neso	20 hours ago
.gitignore	tiny readme (+ gitignore)	3 months ago
README.md	rebrand of spack repo from wrs -> neso	20 hours ago
repo.yaml	rebrand of spack repo from wrs -> neso	20 hours ago

README.md

### NESO-Spack

To enable your installation of spack to use this repo do:

```
git clone git@github.com:EXCALIBUR-NEPTUNE/NESO-spack.git
cd NESO-spack
spack repo add .
```

followed by e.g. `spack install neso.nektar@5.2.0`.

# FIN

## UKAEA NEPTUNE:

Rob Akers  
Wayne Arter  
Matthew Barton  
James Cook  
Joseph Parker  
Owen Parry  
Will Saunders  
Ed Threlfall

*The support of the UK Meteorological Office  
and Strategic Priorities Fund is acknowledged.*

# ExCALIBUR-NEPTUNE

## NEutrals & Plasma TUBulence Numerics for the Exascale

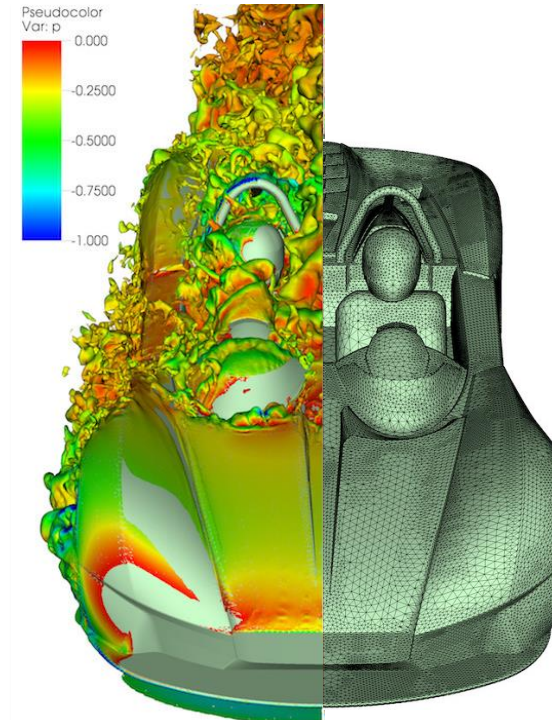
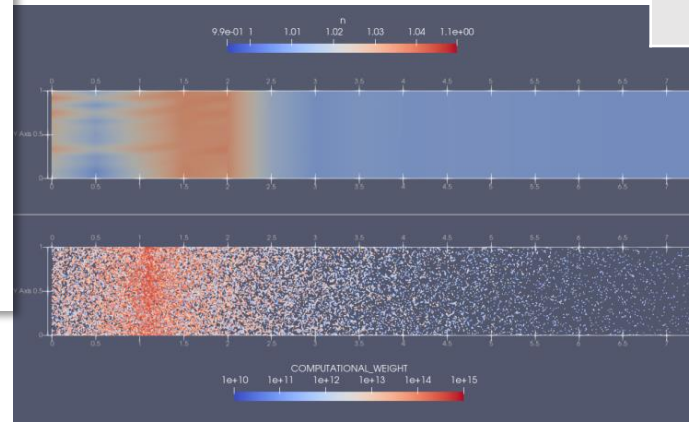
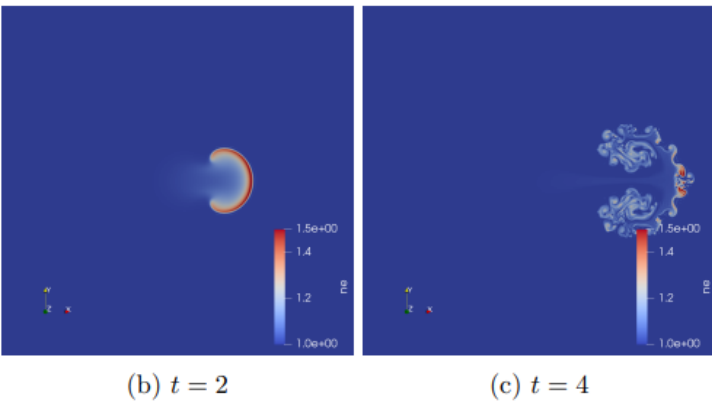


NEKTAR++

SPECTRAL/HP ELEMENT FRAMEWORK

• Fusion reactors are a strongly coupled, multi-physics system of systems problem - The plasma itself requires several different models to accommodate enormous range in time- and length-scales.

• NEPTUNE and its satellite repositories together comprise a highly scalable, arithmetically complex, performance portable next gen code, based on Nektar++ and SYCL.



• One such problem is the **coupling** of plasma to the first wall, which requires **3 velocity components** of the plasma to be represented near the wall (hence > billions of particles).

