

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

# Progress report on work done with CARIDDI and JOREK-DREAM

F. Cipolletta,

N. Isernia, S. Ventre,

P. Karpov, A. Marek,

F. Virgulti, M. Hoppe,

M. Hoelzl

4th Annual Meeting of EUROfusion HPC ACHs

27/10/2026

# Outline

- **Requests**
  - JOREK-CARIDDI – Improve the eigenvalue solver
  - JOREK-DREAM – Support DREAM parallelization
- **JOREK-CARIDDI**
  - Report
- **JOREK-DREAM**
  - Report
- **Conclusions**
  - Takeaways and Perspectives

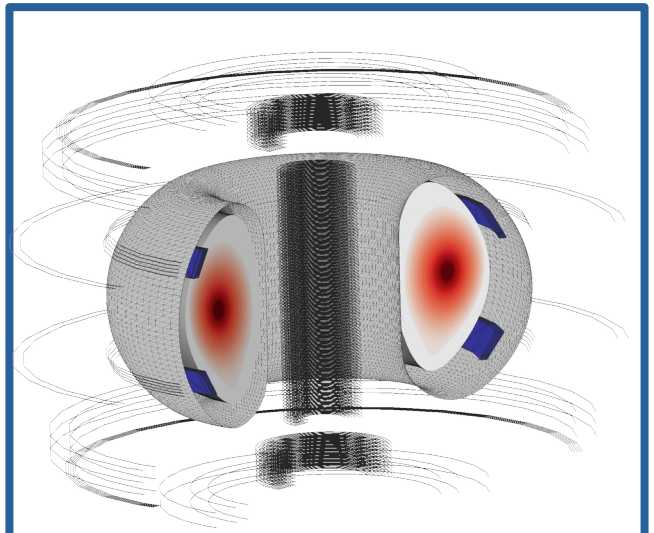
# Requests



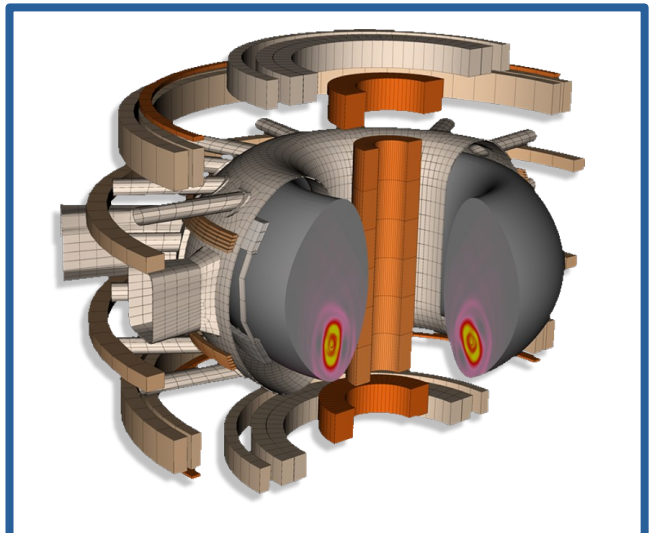



# JOREK-CARIDDI – Improve the eigenvalue solver

- An extension of JOREK is coupled to STARWALL (**J-S**) or CARIDDI (**J-C**) (plasma-wall **interactions**)
- Both STARWALL and CARIDDI rely on the solution of a full **generalized eigenvalue problem** (**gEP**)
- The gEP is solved by means of **ScaLAPACK** (pdsygvx) in the current implementation
- The usage of **ELPA** as an alternative solver was tested, with promising results



3D thin wall model used in **STARWALL**



3D volumetric model used in **CARIDDI**

**OBJECTIVE:** implement **ELPA in CARIDDI as an alternative to ScaLAPACK**

# JOREK-DREAM – Support DREAM parallelization







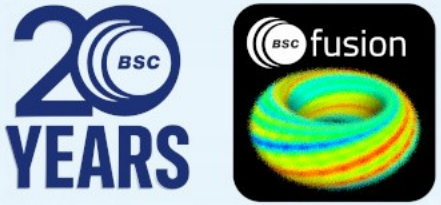
- J-S or J-C beyond Thermal Quench (**TQ**) and Current Quench (**CQ**) to get Runaway Electrons (**RE**)
- DREAM can evolve the RE kinetically and more efficiently than in JOREK
- Use the plasma evolution from JOREK in DREAM every n time step (one-way coupling)
- The RE distribution from DREAM feed back JOREK (two-way coupling)

**OBJECTIVE:** Cross long time scales:

- without 3D plasma instabilities
- with JOREK simulating the plasma vertical motion
- DREAM evolving the RE phase space distribution.

The **idea** is to complement the developments by M.Sc. student at KTH with ACH support activities

# JOREK-CARIDDI



# The CARIDDI workflow

1) **PREPROC:** counts the number of DoF from the mesh (serial computation)
2) **PROC:** build the matrices for the **gEP** (4 kinds of problems), solve it, and compute stuff (MPI parallelized)
3) **POSTPROC:** (possibly) compute other quantities of interest (MPI parallelized)

The following notes should be kept in mind:

- CARIDDI requires **meshes** obtained from CAD models
- CARIDDI requires **squared grid of processes** to run

Therefore, it is not trivial to do scaling tests

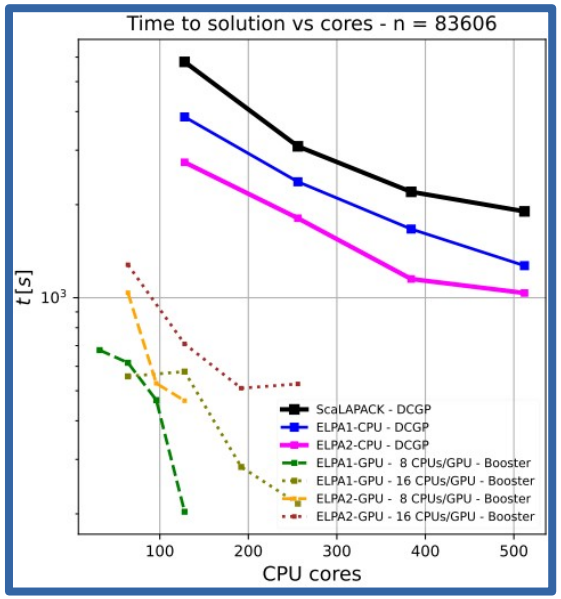
# Implementation steps

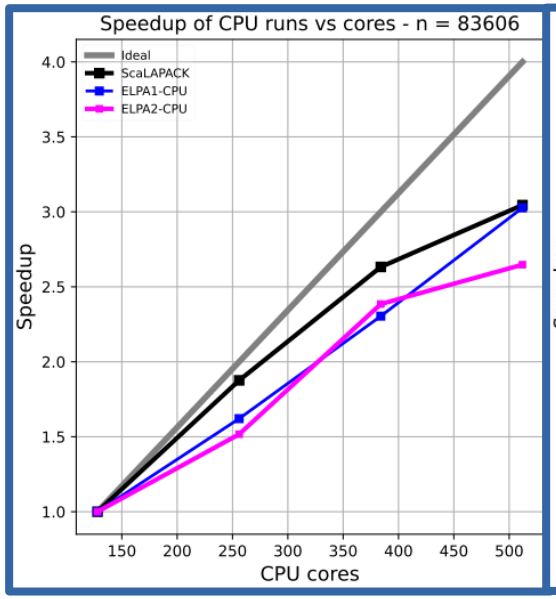
Gradual implementation of calls to ELPA in codes of increasing complexity, as follows:

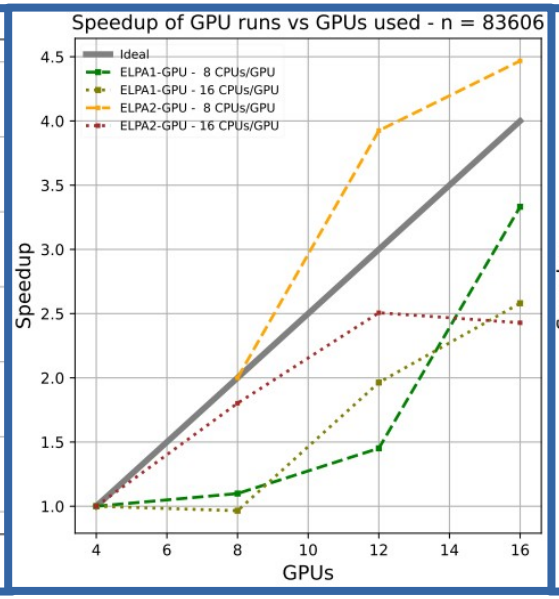
1. **standalone FORTRAN code:** comparison ELPA vs ScaLAPACK – in gEP from STARWALL/CARIDDI (2025)
2. **`generalizzato`:** solves the gEP (as done in CARIDDI) formulated on randomly defined matrices of given size
3. **`cariddi_test`:** avoids PREPROC and only focuses on solving gEP (simplified CARIDDI)
4. **`CARIDDI-JOREK`:** includes CARIDDI standalone (**Cs**) and the JOREK-CARIDDI coupling (**J-C**)

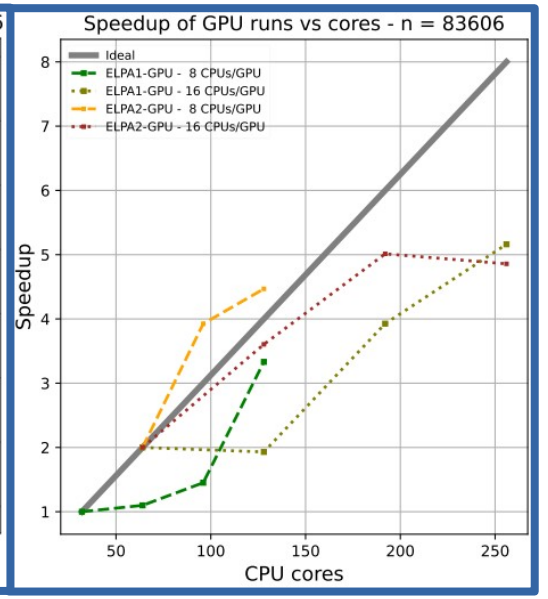
**Note on the HARDWARE:** Pitagora DCGP and Booster were used (the Makefile of CARIDDI has been adapted)

# Scaling tests with `cariddi_test` (no mesh)

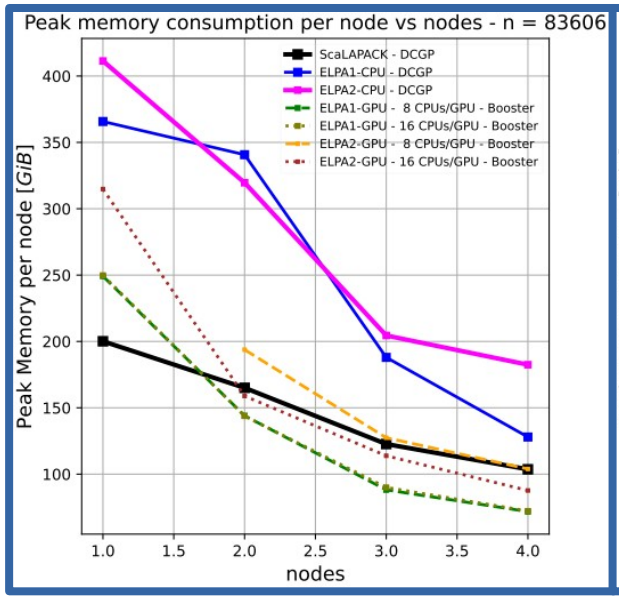


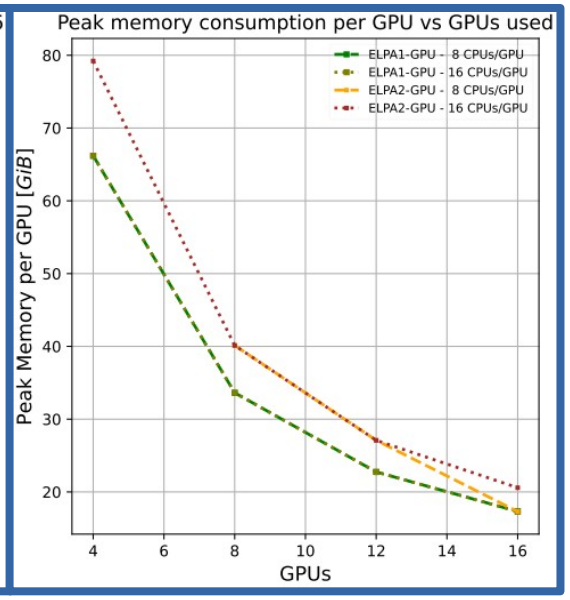



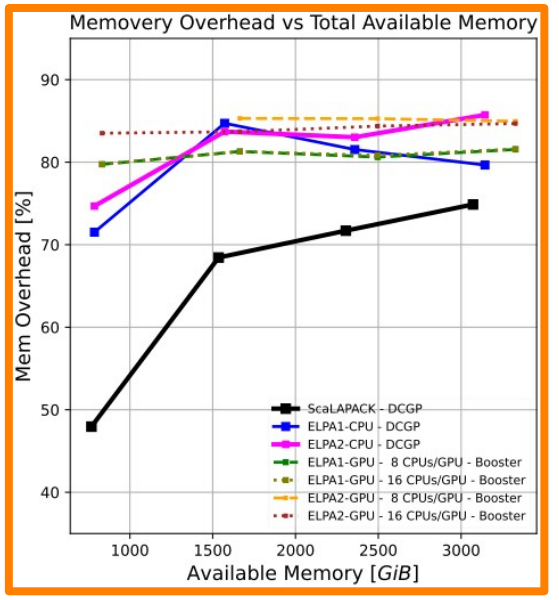




**NOTE:**
**ELPA** is used without OpenMP to keep coherence with CARIDDI

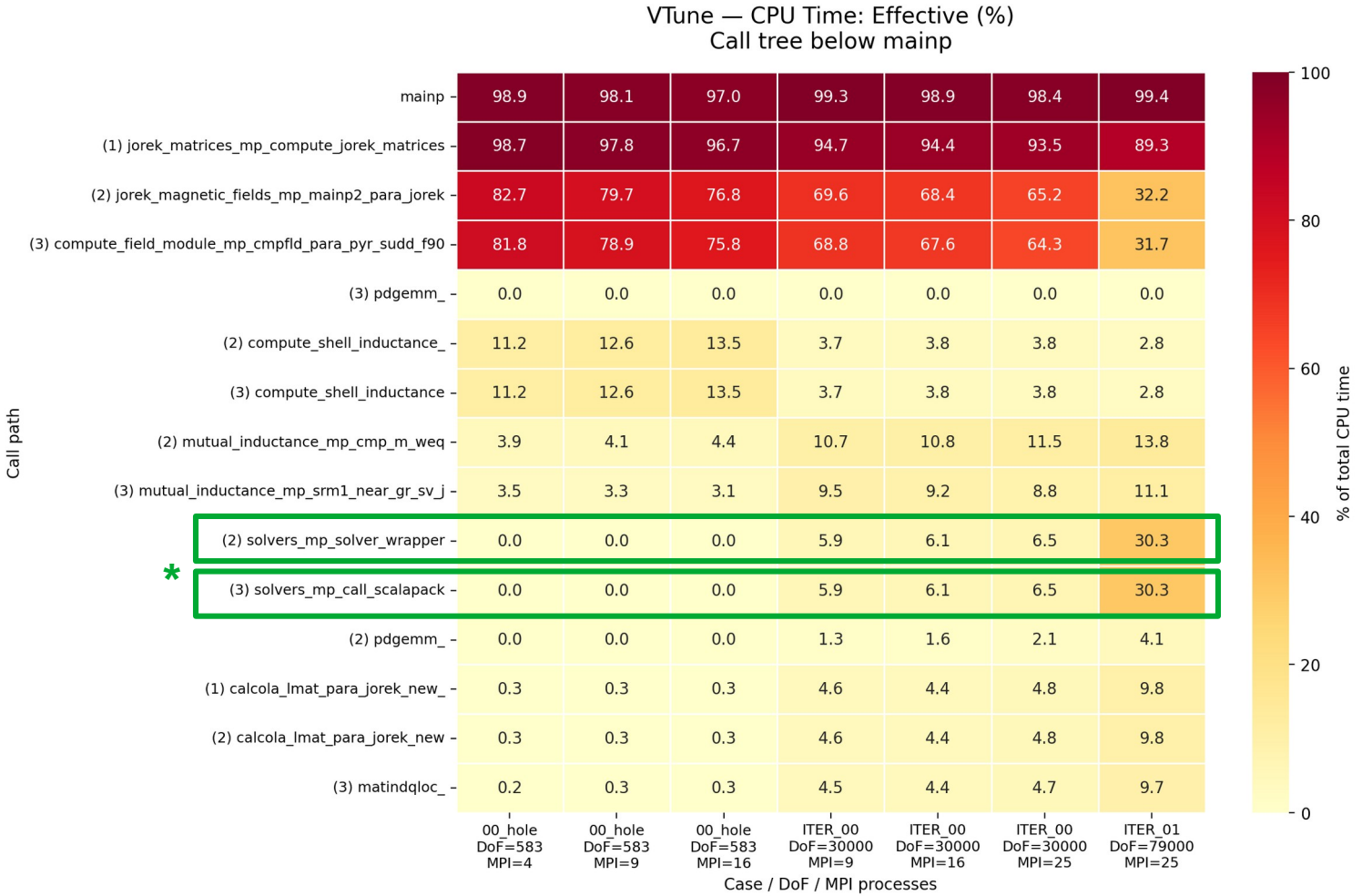



20 BSC YEARS

# Test cases and DoF for profiling CARIDDI (ScaLAPACK only)




*

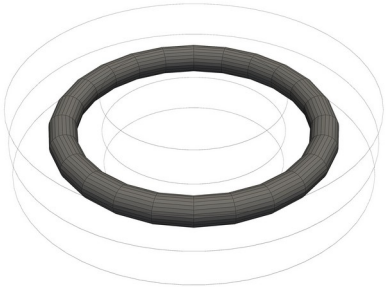
**Intel Vtune** (`topdown`)

Numbers in the call stack indicates the level from `mainp`

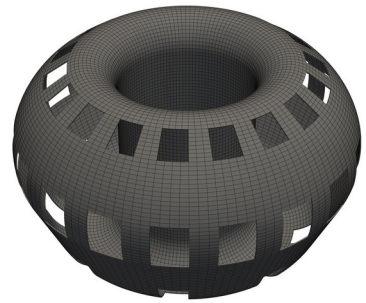
Only a selected set of subroutines

**NOTE*: The gEP gets more expensive with higher DoF**

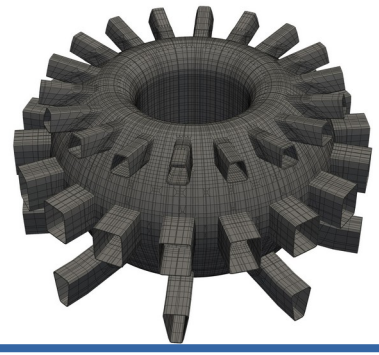
`00_hole`
**583** rows and cols



`ITER_00`
**30863** rows and cols

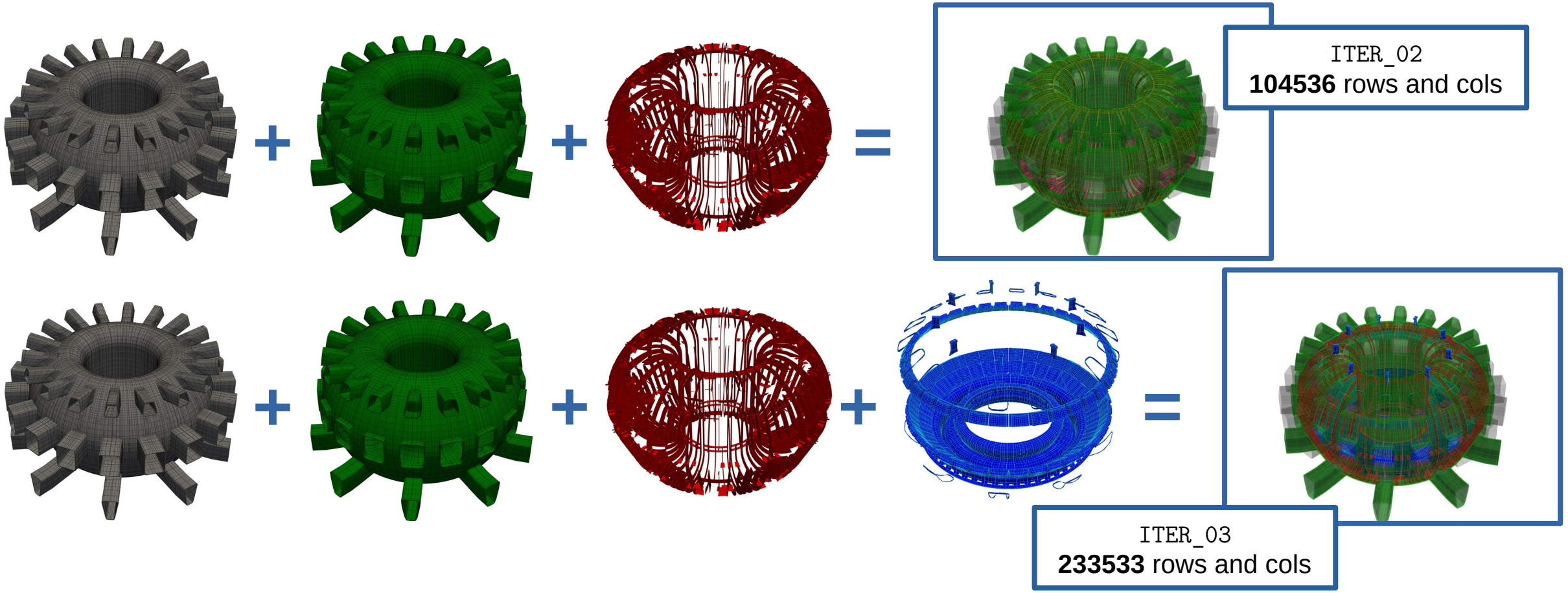


`ITER_01`
**79220** rows and cols



# Other test cases with CARIDDI

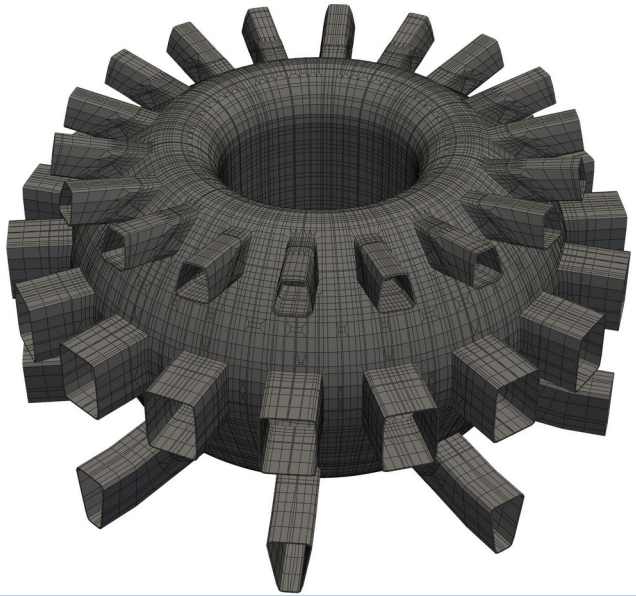
Bigger test cases obtained adding components to the `ITER_01` case
**COST: increase** DoF but with **huge** memory overhead (>20GiB per MPI task)



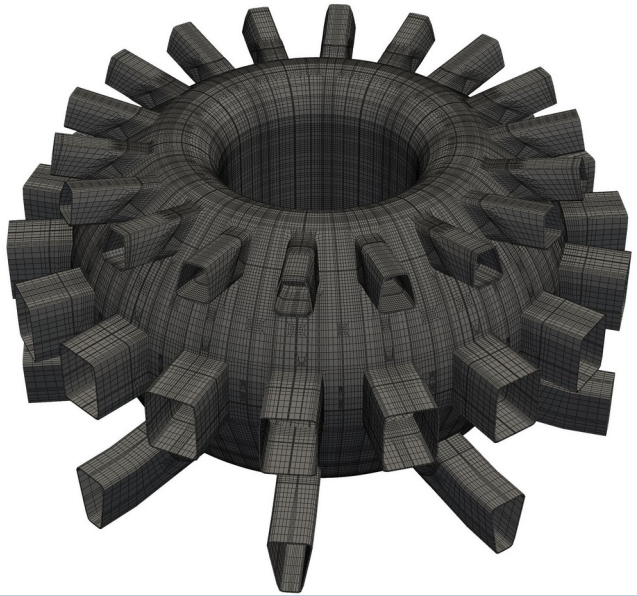
# Other test cases with CARIDDI

The CARIDDI developers has a code to increase the mesh resolution (4x)
**RESULT: increased** DoF but **halved** the memory overhead (∼11 GiB per MPI task)

`ITER_01`
**79220** rows and cols



`ITER_01_4x`
**352334** rows and cols



# Preliminary results for CARIDDI

Some of the test case have been run with ScaLAPACK and ELPA, others should be performed

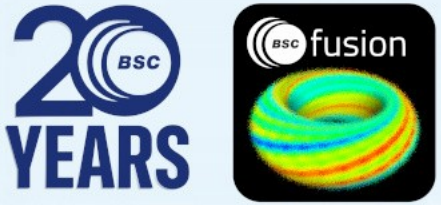
**ELPA** is confirmed to **improve the time to solution of gEP** in CARIDDI, wrt ScaLAPACK, e.g.

| Test_Name | Matr size | Machine | solver | nodes | tasks-per-node | cpus-per-tasks | tasks | solver_wrapper-time-[s] |
|---|---|---|---|---|---|---|---|---|
| ITER_03 | 233533 | Pitagora-DCGP | ScaLAPACK | 16 | 16 | 1 | 256 | 19386.588414 |
| ITER_03 | 233533 | Pitagora-DCGP | ELPA1-CPU | 16 | 16 | 1 | 256 | 14808.937639 |
| ITER_03 | 233533 | Pitagora-DCGP | ELPA2-CPU | 16 | 16 | 1 | 256 | 11794.074212 |
| ITER_03 | 233533 | Pitagora-Booster | ELPA1-GPU | 16 | 4 | 8 | 64 | 8147.718819 |
| ITER_03 | 233533 | Pitagora-Booster | ELPA2-GPU | 16 | 4 | 8 | 64 | 8184.941728 |

The memory overhead introduced by ELPA is **not a real problem** within CARIDDI

**Several new proposals** to improve CARIDDI further have been identified

# JOREK-DREAM



# JOREK-DREAM – Support activities

*“Preliminary” and Support phase*

- The DREAM code has been **compiled** on Pitagora DCGP
- The installation has been tested and debugged against the **“runaway”** example
- The -fno-finite-math-only CXX flag was **needed** for the optimized version with Intel compilers
- Linking to **PETSc compiled with MPI** capabilities has been performed (including Hypre)
- Also provided a **PETSc compilation with MPI+OpenMP** (should be debugged)
- Help **debugging** and **finding new ways** to proceed

# JOREK-DREAM – Development steps

*M.Sc. Thesis work at KTH (by Francesco Virgulti)*

- **Profiling** → linear sparse system solution is identified as a bottleneck
- **Parallelization** via PETSc (**KSP**)
- **Fixed r** and physics →
  - Tested different solver
  - Selected the **HypreAMG** preconditioner + **GMRES** linear solver
  - Observed some limits in scaling
- Switched to **several r** values
  - **Tested** preconditioning of aggregated and decoupled (among r values) matrices
  - The **radially-decoupled** version is performing best (Block-Jacobi, **BJ**, or Additive Schwarz Method, **ASM**)
  - These versions show some scaling (only 2 ranks – **need to be tested on Pitagora...**)

# Conclusions






# CARIDDI

- **ELPA** has been **implemented** in CARIDDI (great support from developers):
  - Modularity facilitates code maintenance and further improvement
  - The original coding style (conditional compilation, parameters, etc) is maintained
  - ELPA accelerates the solution of the gEP wrt ScaLAPACK

- Several **possible improvements** have been **identified**:
  - Parallelization of the PREPROC
  - Implementation of OpenMP (memory overhead, parallelization, complement ELPA)
  - Allow general process grid (requiring deep code modernization)

- Future **developments of interest** under discussion:
  - Krylov Model Order Reduction (**MOR**) in J-C (tested in Cs)
  - "**n-fold**" method → Block Circulant Structure of matrices
  - **EUROfusion standard software**

# DREAM

- DREAM has been **parallelized** with PETSc
- Several **preconditioners** are actually under consideration and testing

Next steps:

- Considering **different MHD fields**, to test of they impact the preconditioner
- Considering the **magnetic field varying with r** (the literature reports possible issues)
- Testing on **Pitagora-DGCP** to assess scaling capabilities
- Applying (and debugging) **OpenMP**
- Considering **MHD fields from JOREK**

# …anticipations…






# Experience with Pitagora

- **Tickets** are usually dealt fairly **quickly**
- The software environment is managed through **spack**
- The machine has suffered some **down periods**
- Training on **AMD hardware**?

fusion.bsc.es

@Fusion_BSC

fusion-bsc

EUROfusion

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

# Thank you

# Backup






# The generalized eigenvalue problem

- Both STARWALL and CARIDDI rely on the solution of a generalized eigenvalue problem (**gEP**):

$$L_w^* \, S_\lambda = \lambda \, R_w \, S_\lambda ,$$

$L_w^*$ → Inductance (**dense**) matrix for the conductive structures

$S$ → Basis of eigenvectors

$R_w$ → Resistance (**dense**) matrix

$\lambda$ → Eigenvalue

- Once this problem is solved, the system of equations treated in STARWALL/CARIDDI becomes **diagonal**
- Both the inductance and resistance matrices have a **size of the walls' DoF squared**
- No cut can be done, because a **complete basis of eigenvectors** is needed

See 10.1063/5.0167271 for further details

20 BSC YEARS

# JOREK-DREAM – Development current status

*Some results from tests on the local machine (due to Pitagora unavailablity):*

| Full system size | Agg/Decoup | radii/block | rtol | Decoupling | Preconditioner | Solver | ranks | Setup Time | Solver Time | ranks | Setup Time | Solver Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2432000 x 2432000 | Agg | 16 | 1e-12 | - | AMG(def) | GMRES | 1 | 2.633 | 17.769 | 2 | 30.698 | 11.891 |
| 2432000 x 2432000 | Agg | 16 | 1e-10 | - | AMG(def) | GMRES | 1 | 2.649 | 14.661 | 2 | 30.868 | 9.909 |
| 2432000 x 2432000 | Agg | 16 | 1e-10 | - | AMG(HMIS/ext+i) | GMRES | 1 | 1.460 | 9.889 | 2 | 7.343 | 5.615 |
| 2432000 x 2432000 | Decoup | 8 | 1e-10 | BJ | AMG(def) | GMRES | 1 | 2.297 | 13.280 | 2 | 1.171 | 7.548 |
| 2432000 x 2432000 | Decoup | 8 | 1e-10 | ASM | AMG(def) | GMRES | 1 | 2.842 | 14.204 | 2 | 1.655 | 8.118 |
| 2432000 x 2432000 | Decoup | 4 | 1e-10 | BJ | AMG(def) | GMRES | 1 | 1.553 | 11.135 | 2 | 0.880 | 6.401 |
| 2432000 x 2432000 | Decoup | 4 | 1e-10 | ASM | AMG(def) | GMRES | 1 | 2.723 | 14.811 | 2 | 2.031 | 8.530 |
| 2432000 x 2432000 | Decoup | 2 | 1e-10 | BJ | AMG(def) | GMRES | 1 | 0.990 | 9.132 | 2 | 0.540 | 5.320 |
| 2432000 x 2432000 | Decoup | 2 | 1e-10 | ASM | AMG(def) | GMRES | 1 | 3.122 | 16.176 | 2 | 2.543 | 10.765 |
| 2432000 x 2432000 | Decoup | 1 | 1e-10 | BJ | AMG(def) | GMRES | 1 | 0.596 | 5.314 | 2 | 0.357 | 3.263 |
| 2432000 x 2432000 | Decoup | 1 | 1e-10 | ASM | AMG(def) | GMRES | 1 | 4.103 | 21.649 | 2 | 3.473 | 12.777 |
| 2432000 x 2432000 | Decoup | 1 | 1e-10 | BJ | GMRES+AMG(def) | FGMRES | 1 | 0.605 | 37.795 | 2 | 0.336 | 22.902 |

20 BSC YEARS