EUROfusion

3rd Annual Meeting of EUROfusion HPC ACHs

Cray's inferno: a journey through the porting of the JOREK code on LUMI-C

C. Sommariva, M. Kong, L. Édes, C. Wang, E. Lanti and M. Hoelzl

École Polytechnique Fédérale de Lausanne (EPFL), Swiss Plasma Center (SPC), CH-1015 Lausanne, Switzerland

MaxPlanck Institute for Plasma Physics, Boltzmannstr. 2, 85748 Garching b. M., Germany

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.

EPFL Swiss Plasma Center

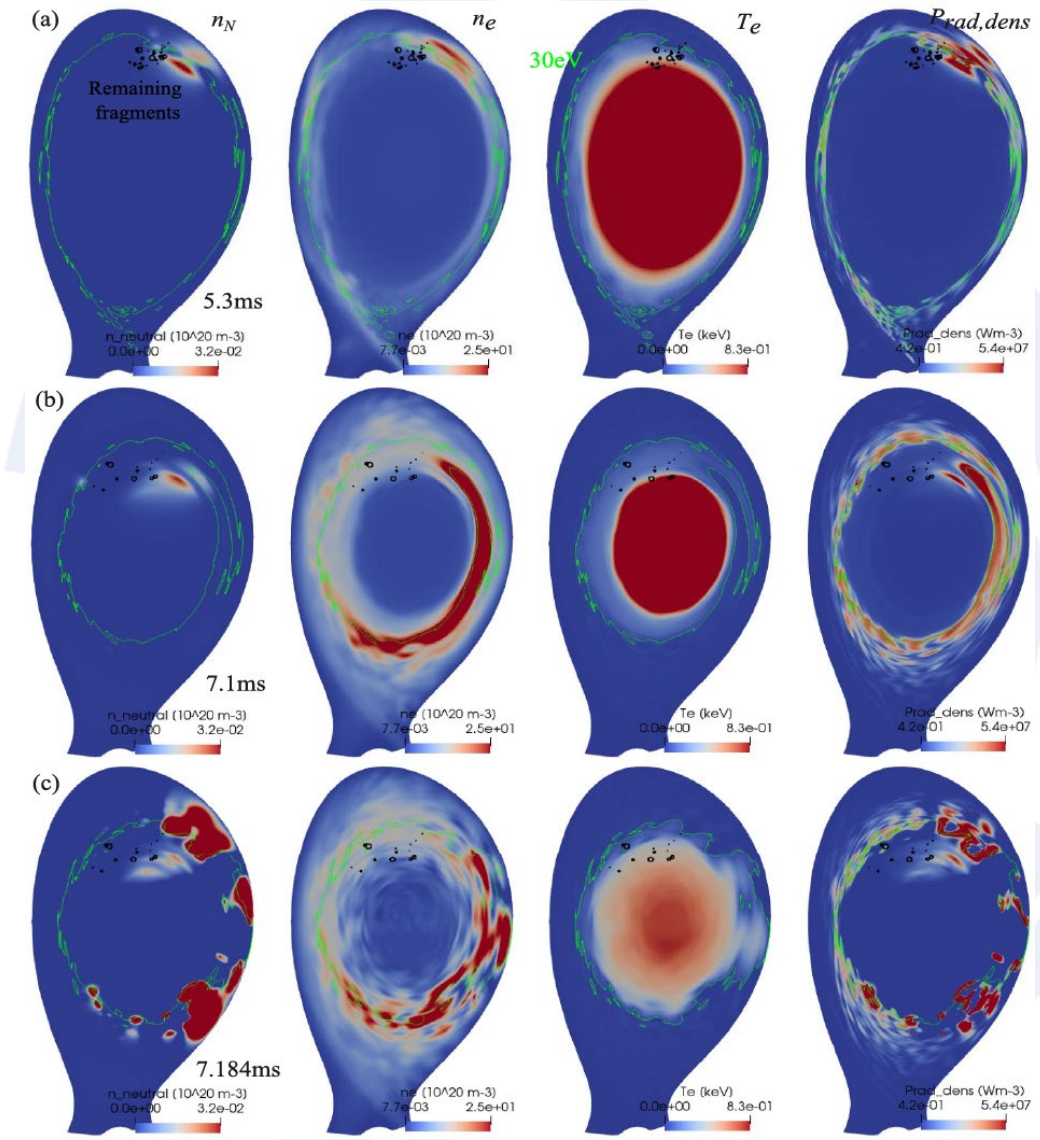
JOREK

A brief overview of the JOREK code

JOREK is an extended 3D-time varying (reduced and full) MHD code applied to the simulation of fusion plasma transients in realistic 3D geometries [1,2].

Key features of the JOREK code:

- Spatial integration via finite element method on unstructured grid
  - Fourier polynomials (toroidal) and G_n-continuous elements (poloidal)
- Fully non-linear time-marching temporal integration
- Multiple linear solvers (MUMPS [3], PasTix [4], STRUMPACK [5], …)
- Hybrid MPI-OpenMP parallelization (GPU porting underway)
  - MPI parallelization mainly w.r.t. the toroidal harmonics
  - Additional parallelization within the linear solvers
- Free boundary simulation via coupling with external wall solvers
  - STARWALL [6], CARIDDI [7]
- Kinetic solver via Monte-Carlo integration
  - Particle tracking (6D & 5D), collisions, atomic physics, …



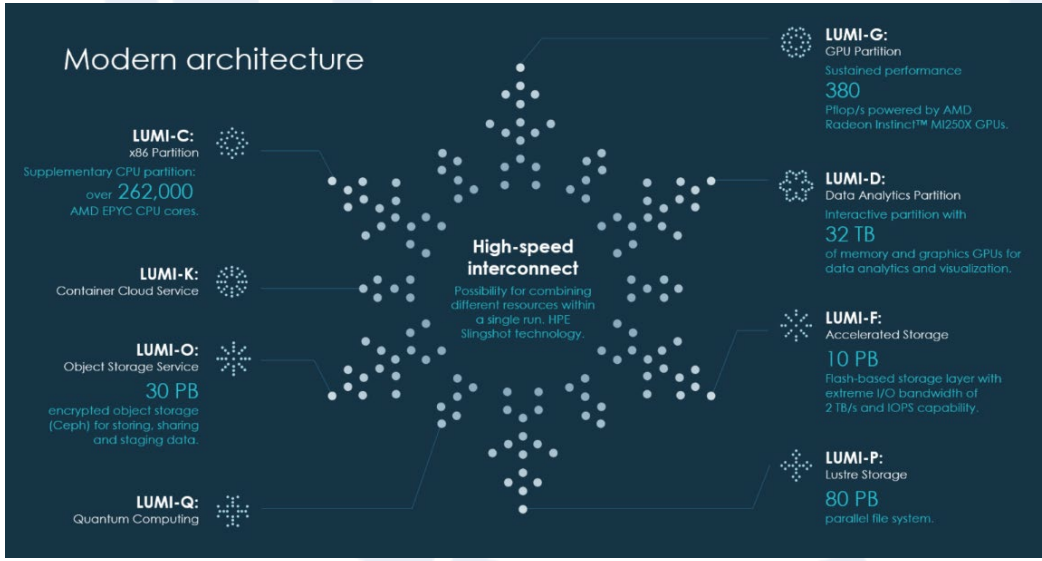
JOREK simulated MHD activity due to shattered pellet injection in a JET-like plasma (courtesy of M. Kong, [8])

The European pre-exascale system: LUMI

LUMI is a pre-exascale European HPC system in production since 2023

- Cray cluster @ 9th place in the top500 [9] list with 379.70 PFlop/s (Linpack)
- Multiple available partitions for computing:
  - LUMI-G: 2978 nodes (1x64 cores AMD EPYC 7A53 Trento, 512GB RAM, 2x2 MI250 GPUs, 1x 200GB/s HPE slingshot-11)
  - LUMI-C: 1888 nodes w 256GB RAM, 128 nodes w 512GB RAM, 32 nodes w 1024GB RAM (2x64 cores AMD EPYC 7763, 1x 200GB/s HPE slingshot-11)
- Multiple available storage options:
  - LUMI-P: Lustre disk based parallel filesystem w 20PB at 240GB/s
  - LUMI-F: Lustre solid state parallel filesystem w 8PB at 740GB/s
  - LIMI-O: object storage filesystem accessible via webservice and S3 API w 30PB of storage (not tested in this task)
- Local support for basic installation tasks provided by the LUST team
  - Possible to have more extensive support through the EuroHPC JU (EPICURE, not used here)

LUMI HPC [10]



LUMI HPC architecture [10]

Why porting JOREK on LUMI? What are the requirements & initial conditions?

2024 has been a difficult year for a fraction of the JOREK users:

- Marconi & Pitagora unavailable
- Part of JOREK operation on Leonardo hindered due to bug in the Intel OneAPI - 2021 compiler (solved with OneAPI 2023)
- Leonardo computation resources in fast depletion

⇒ Need to diversify the HPC providers of the JOREK community

Dependencies required by the JOREK code (not exhaustive):

- Graph partitioning libraries: Metis, ParMetis, Scotch, PtScotch
- Mathematical libraries: MKL, Cray Scientific Library, ….
- Dense matrix solvers: MUMPS
- Sparse matrix solvers: STRUMPACK, PasTix
- FFT library: FFTW 3.3.10
- I/O Library: HDF5 1.12 or above
- Object oriented enabled FORTRAN compiler (2011 or above)
- MPI library (better if with full support to multi-threading)

Available resources on LUMI

- GPU-enabled JOREK not in production yet

  ⇒ JOREK installation performed on LUMI-CPU (LUMI-C)
- Granted pre-production resources on LUMI-C
  - 2000Nh for pre-production project

Available environment on LUMI-C

- HPE Cray – Cray Programming Environment 24.03
  - Graph: Scotch 7.0.4, Metis 5.1.0, ParMetis 4.0.3
  - Math: Cray LibSci, Boost 1.83.0
  - IO: parallel HDF5 1.12.2
- HPE GNU – Cray Programming Environment 24.03
  - Graph: Scotch 7.0.4, Metis 5.1.0, ParMetis 4.0.3
  - Math: cray-libsci, Boost 1.83.0, FFTW-3
  - IO: parallel HDF5 1.12.2

EasyBuild repository for: MUMPS 5.5.1 & STRUMPACK 7.0.1

First test: Library installation using Cray – CPE environment

First test: compile & install MUMPS and STRUMPACK using HPE Cray – CPE 24.03

- Linked libraries: Metis, ParMetis, Scotch, Cray LibSci from Cray environment

  ⇒Compilation was successful

  ⇒Both MUMPS and STRUMPACK tests returned segmentation fault

Second test compile & install: GKLib, Metis, ParMetis, MUMPS, STRUMPACK using HPE Cray – CPE 24.03

- Linked libraries: Cray LibSci from Cray environment

  ⇒Compilation was successful for all the libraries

  ⇒Both MUMPS and STRUMPACK tests returned segmentation fault

    ⇒Support ticket to LUST was opened

Third test: compile & install MUMPS using HPE Cray – CPE 24.03 via LUST EasyBuild receipt

- Linked libraries: Metis, ParMetis, Scotch, Cray LibSci from Cray environment

  ⇒Compilation was successful

  ⇒Both MUMPS and STRUMPACK tests returned segmentation fault (again ….)

After a series of failures in compilation, installation & testing:

- Identified an issue with the Cray – CPE compiler and/or Cray LibSci with the handling of multi-threading

  ⇒Compilation and testing of MUMPS was successful imposing multi-threading optimization -Othread1

  ⇒Tested compilation of FFTW-3: successful

  ⇒STRUMPACK testing still failing with segmentation fault (works only for serial applications)

First test: installation of JOREK using Cray – CPE Environment

JOREK installation using Cray compilers is not straight forward

⇒ JOREK source code must be modified for compilation

1. Fortran “implicit” statements must be removed in SIMD declaration
2. Order of compiling flags must be changed
3. Module access (“use”) statements must be moved after SIMD declaration
4. Commas after “write” statements must be removed
5. Modifications to internal function calls for complying with FORTRAN standard
6. Renaming of variables in “select type” statements

⇒ Compilation of JOREK using LUMI Cray-CPE succeeded!

⇒ Segmentation faults when running standard test cases!

⇒ Opened second discussion with LUST for solving execution problems

```
#if _OPENMP >= 201511
    !$omp declare simd
#ifndef _CRAYFTN
    implicit none
#endif
#else
    implicit none
#endif
```

```
#ifdef _CRAYFTN
  use phys_module, only: T_1, T_min_neg, corr_neg_temp_coef
#endif
  !$omp declare simd
#ifndef _CRAYFTN
  use phys_module, only: T_1, T_min_neg, corr_neg_temp_coef
#endif
#else
  use phys_module, only: T_1, T_min_neg, corr_neg_temp_coef
```

```
#ifndef _CRAYFTN
      !> WARNING: access=append IS NOT standard FORTRAN language
      open(ifile, file=filename, form='formatted', access='append', iostat=ioerr)
#else
      open(ifile, file=filename, form='formatted', position='append', iostat=ioerr)
#endif
```

```
select type (pa => particles(j))
select type (part=>particles(j))
  type is (particle_kinetic_leapfrog)
    pa%v = ran(5:7) ! save other components of this point for velocity init in a later routine
    part%v = ran(5:7) ! save other components of this point for velocity init in a later routine
end select
```

Please note that this also means that i will help you get your code running on LUMI using ONE compiler, e.g. the GNU-compiler. If you then afterwards want to try out the Cray-compiler you will have to do that by yourself

Second test: JOREK installation using GNU-CPE environment

Starting again: new compilation of MUMPS and STRUMPACK using the GNU-CPE environment

- Compilation of both MUMPS and STRUMPACK with "self"-compiled Metis and ParMetis:

  ⇒Segmentation fault!

- Compilation of MUMPS with Metis and ParMetis available in the GNU-CPE software stack:

  ⇒ Success (no-optimization reduction needed)!

- Compilation of STRUMPACK: segmentation faults!

  ⇒Tested compilation using GNU-CPE software stack

  ⇒Tested installation with EasyBuild

  ⇒Tested job submission for reducing memory footprint

  ⇒Clues suggest a possible bug in the Cray LibSci library

  ⇒Require compiling all software stack with OpenBLAS

  ⇒Solution dismissed due to project time limit …

Installation of the JORKE code using GNU-CPE environment

- JOREK installed mostly using libraries available in the GNU-CPE software stack of LUMI-C
- Only available linear solver: MUMPS

  ⇒ Expected reduced performance with sparse matrix

Nevertheless:

⇒ JOREK successfully installed on LUMI-C

⇒ All non-regression tests involving MUMPS passed successfully

⇒ All unit tests passed successfully

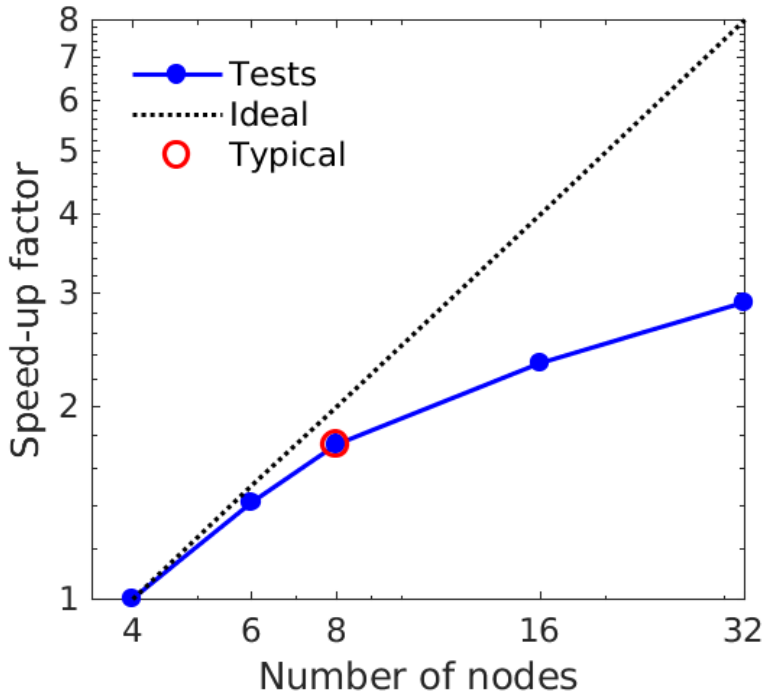
⇒ Residual segmentation fault issues persist for JOREK branches not merged with the main develop branch

What about using different sparse solvers than STRUMPACK?

⇒ Tested installation of PasTix 5.2.3

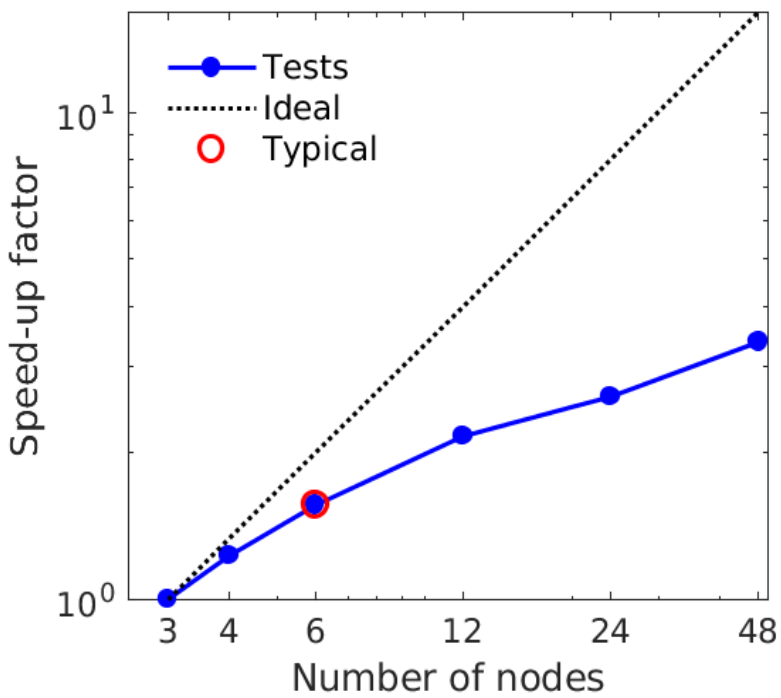
⇒ Failure due to segmentation faults of the tests

Code performance on LUMI-C using GNU-CPE



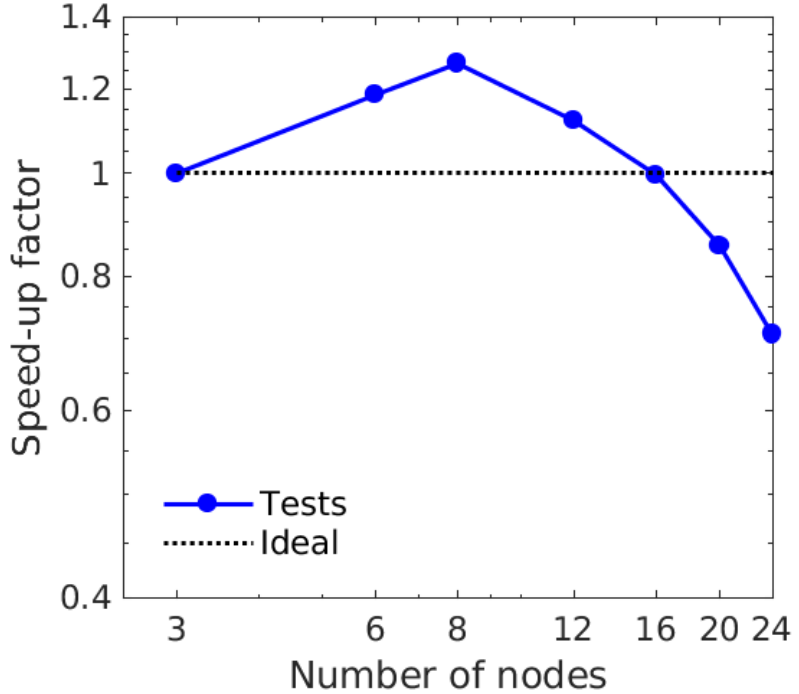
N# cores	Wall time [s]	Speed-up	N# nodes
512	48.0263	1.0	4
768	33.9507	1.4146	6
1024	**27.5718**	**1.7419**	**8**
2048	20.5685	2.3349	16
4096	16.5369	2.9042	32

Strong scaling fixed boundary



N# cores	Wall time [s]	Speed-up	N# nodes
384	113.8380	1.0	3
512	92.6006	1.2293	4
768	**72.9053**	**1.5615**	**6**
1536	52.6347	2.1628	12
3071	43.6548	2.6077	24
6144	33.7233	3.3756	48

Strong scaling free-boundary



N# cores	Wall time [s]	Speed-up	N# nodes
384	48.4957	1.0	3
768	40.9261	1.1898	6
1024	**38.2421**	**1.2549**	**8**
1536	43.2622	1.1730	12
2048	48.6455	1.0618	16
2560	56.7384	0.8951	20
3072	68.6783	0.7023	24

Weak scaling fixed boundary

⇒ Wall time and scaling considered satisfactory by the users!

Conclusions

- **JOREK code successfully ported on LUMI CPU partition**, but:
  - Impossible to compile using HPE Cray compilers
    - MUMPS compiled but with reduced multi-threading optimization level
    - STRUMPACK compiled but with segmentation fault in testing
    - JOREK code modified for complying with Cray compiler standard
      - Still segmentation fault while testing
    - Successfully compiled and tested the JOREK code with GNU compilers
    - STRUMPACK compiled but with still segmentation fault in testing (works for serial applications applications)

      ⇒Most probably, the HPE Cray Scientific Library is the source of errors

      ⇒Maybe some issue related with the HPE slingshot interconnect?
- **Wall time, strong and weak scaling considered satisfactory by the user**
  - Tested production fluid Runaway electron test case
    - Both fixed and free – boundary simulation tested
- **Production project for JOREK allocation on LUMI-C submitted and under review**

Artistic representation of the Cray-CPE compilation steps [11]

Lessons learned and future work

- Identify source of issue with STRUMPACK and JOREK

  ⇒Test new release of HPE – Cray compilers & MPI

  ⇒Test new release of HPE – Cray Scientific Library

  ⇒Test open-source mathematical libraries (BLAS, BLACS, LAPACK, ScaLAPACK)

- Code porting on new systems becomes more challenging with time
  - Variegate architectures are appearing in the scene
  - More complex libraries with challenging installation
  - Support from help desks may be doubtful (similar issue with CINECA-Leonardo …)
  - Code installation and testing may require tens of days of work from several people

    ⇒Investigate the containerization of codes for efficient porting on different machine should be studied and tested

THANK YOU!

References

[1] M. Hoelzl et al., Nucl. Fus., vol. 64, p. 112016, 2024

[2] M. Hoelzl et al., Nucl. Fus., vol. 61, p. 065001, 2021

[3] P. R. Amestoy et al., SIAM J. Matrix Anal. Appl., vol. 23, p. 15, 2001

[4] P. Hénon et al., Parallel Comput., vol. 28, p. 301, 2002

[5] P. Ghysels et al., IEEE Int. Parallel and Distrib. Processing Symp. (IPDPS), p. 897, 29 May – 2 June, 2017

[6] P. Merkel et E. Strumberger., arXiv 1508.04911, 2015

[7] R. Albanese et al., IEEE Proc. A, vol. 135, p. 457, 1988

[8] M. Kong et al., Nucl. Fus., vol. 64, p. 066004, 2024

[9] https://top500.org/

[10] https://lumi-supercomputer.eu/lumi_supercomputer/

[11] https://www.storicang.it/a/viaggio-nellinferno-di-dante_15320