EUROfusion

Memory optimization for the response matrices in JOREK

F. Cipolletta

HPC ACHs 2023 | 15-16 Nov. 2023

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.

Outline

1. Brief introduction to JOREK and the response matrices
2. Organization of the work
3. Matrix Compression
    a. Chosen method
    b. How it is implemented
4. How the method preforms within applications
    a. Memory Savings
    b. Issues and Idea on How we are addressing

Outline

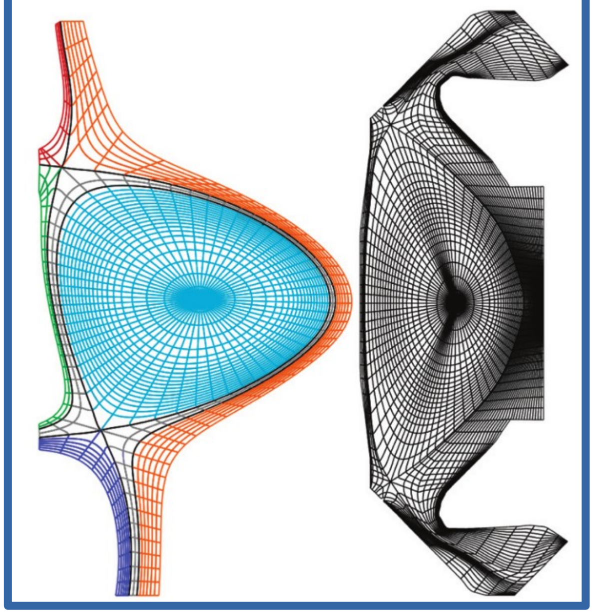
1. **Brief introduction to JOREK and the response matrices**
2. Organization of the work
3. Matrix Compression
    a. Chosen method
    b. How it is implemented
4. How the method preforms within applications
    a. Memory Savings
    b. Issues and Idea on How we are addressing

The JOREK code

- Non-linear MHD + many extensions including kinetic & hybrid models
- Bezier finite element + toroidal Fourier expansion
- Fully implicit time-evolution
- Divertor tokamaks including X-point(s) – **(1)**
- Adopted for simulations of plasma instabilities – **(2)**
- Originally developed at CEA Cadarache:
  - Czarny and Huysmans (2008); Huysmans and Czarny (2007)
- In the present work Reduced MHD for simplicity
- Fortran 90/95
- MPI + OpenMP hybrid parallelization

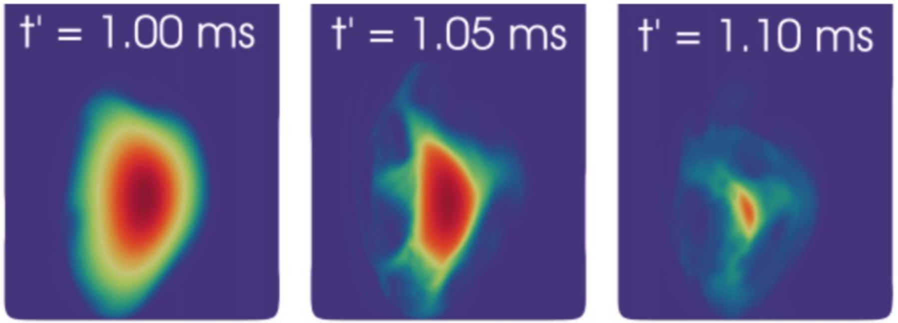
More details in the presentation M. Hoelzl (2012)

(1) Typical grids used in JOREK



See Hoelzl et al (2021)

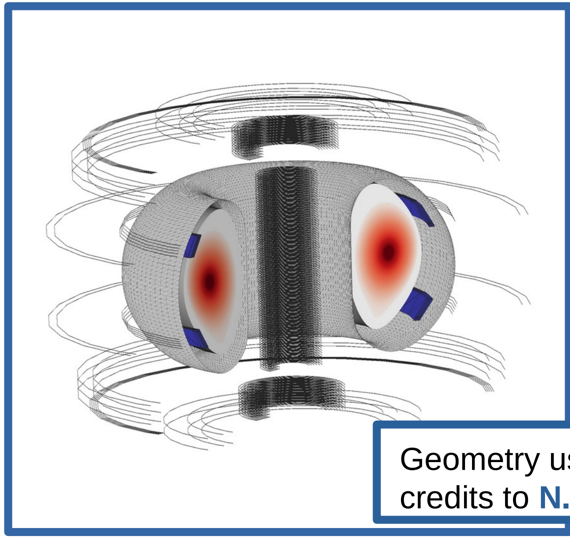
(2) Snapshots of evolution of the pressure in the $\phi = 0$ plane in a 3D Vertical Displacement Event (**VDE**)



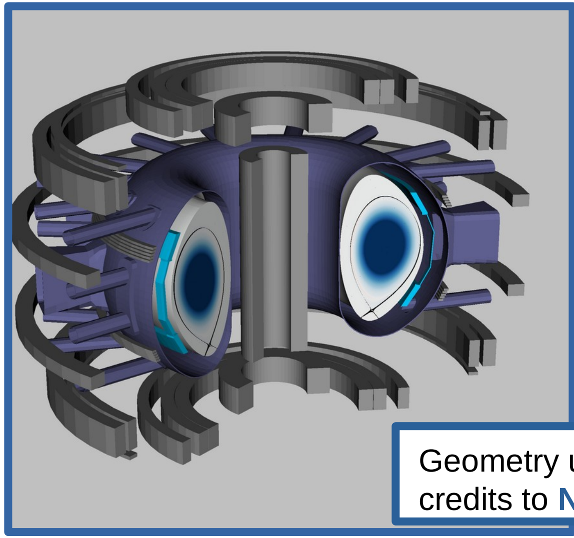
STARWALL and CARIDDI

The free-boundary and resistive wall extension of JOREK are obtained through coupling to external codes, which provides **response matrices**, in particular:

- **STARWALL** → 3D <u>thin</u> resistive walls response – see Merkel, Strumberger (2015)
- **CARIDDI** → 3D <u>volumetric</u> resistive walls – see Isernia et al (2022) + in preparation



Geometry used in **STARWALL**
credits to **N. Schwarz**



Geometry used in **CARIDDI**
credits to **N. Schwarz**

Work requested to BSC-ACH

Goal

Enable modeling capabilities of realistic and accurate 3D wall structures within MHD simulations of plasma instabilities inside a Tokamak

Objective related to JOREK

Reduce Memory required by response matrices and **Improve Performance** → Apply factorization and compression techniques to matrices provided by STARWALL or CARIDDI

Challenges

TASK-RELATED – connected to matrix dimensions (could become big for realistic geometries)

CODE-RELATED – complex collection of several Fortran files with ID computed externally

PROJECT-RELATED – many parallel developments → "*code-orthogonality*" is mandatory

Outline

1. Brief introduction to JOREK and the response matrices
2. **Organization of the work**
3. Matrix Compression
   a. Chosen method
   b. How it is implemented
4. How the method preforms within applications
   a. Memory Savings
   b. Issues and Idea on How we are addressing

Organization of the work

- Compilation on HPC
  - **MN4 -** efforts by **myself** and **Operation** → used in case of maintenance of Marconi
  - **Marconi -** libraries and makefile provided by **I. Holod** → used in production
- Code development
  - Attending TSVV-8 Meetings
  - Attending code seminars
  - Attending progress meeting of JOREK-CARIDDI
  - Interaction with developers
    - Initial discussions with **S. Ventre**, **N. Isernia**, and **G. Rubinacci** (method)
    - Weekly email and reports with **M. Hoelzl**, **N. Schwarz** (implementation&testing)
    - Online meeting as per needed with **M. Hoelzl**, **N. Schwarz**
    - Discussions, report and coordination with **A. Soba**
  - Attended JOREK General Meeting 2023
  - Will attend JOREK General Meeting 2024
- Outreach
  - EFTC 2023
  - Fusion HPC 2023 → writing of publication for PPCF in progress

Outline

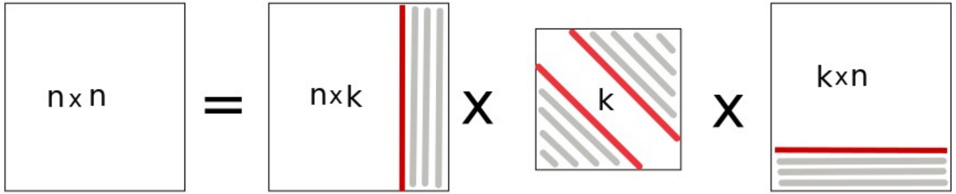
1. Brief introduction to JOREK and the response matrices
2. Organization of the work
3. **Matrix Compression**
   a. **Chosen method**
   b. How it is implemented
4. How the method preforms within applications
   a. Memory Savings
   b. Issues and Idea on How we are addressing

Singular Value Decomposition (SVD)

Given a dense matrix, with one accurate SVD, we could write

$$A = (U \times \Sigma) \times V^T$$

A dense
U, V orthogonal
Σ diagonal



Required Dimensions for the factorized representation:

$rank(\Sigma) = k$, storing (UΣ) and V^T => $(2nk)$ elements instead of n^2

A rectangular (m,n) => $(mk+nk)$ elements instead of (mn) (size scales linearly with k)

Powerful Features in view of applications:

1. The SVD can be always performed
2. An SVD with singular values in descending order always exists
3. The SVD is an optimal approximation with respect to the residual

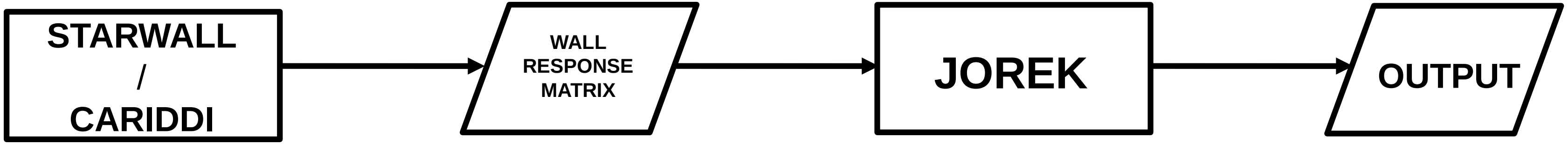
Implementation:

The Scalable Linear Algebra PACKage (ScaLAPACK) parallelized library offers the routine `pdgesvd` to compute the SVD of a given matrix (https://netlib.org/scalapack/)

Outline

1. Brief introduction to JOREK and the response matrices
2. Organization of the work
3. **Matrix Compression**
   a. Chosen method
   b. **How it is implemented**
4. How the method preforms within applications
   a. Memory Savings
   b. Issues and Idea on How we are addressing

Starting point for this work



What we do

How the compression code works

- At compilation, the user selects:
  - Which matrices to compress
  - What fraction of singular values of the SVD to retain
- Reads STARWALL/CARIDDI response file
- ScaLAPACK performs SVD and returns decreasing order Singular Values (SVs)
- **Compressing** = *Retaining the biggest Svs*
- An output file is printed with all the required information (*factorization in* $(U\Sigma)$ *and* V^T)
- Possibility to re-aggregate the SVD decomposition (*VALIDATION*)
- Runs are typically fast (minutes), since they rely on the optimized ScaLAPACK library
- The compression needs to be run only one time per JOREK simulation (*separate module*)

Notes on the current implementation in JOREK

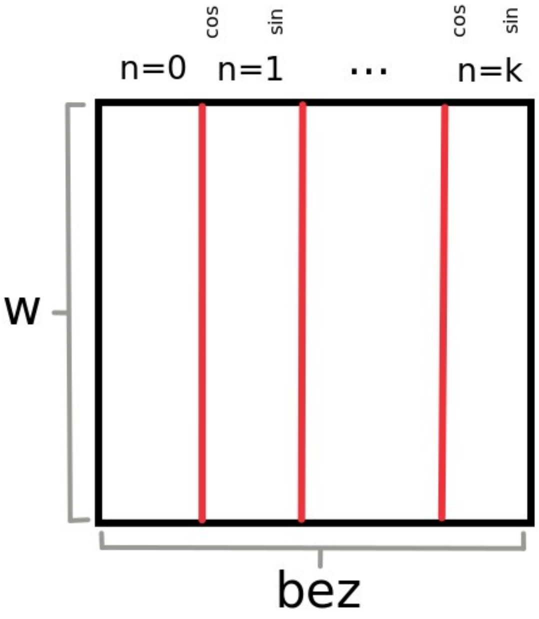
ADAPTATION of JOREK

- Need to choose one or two matrices as a starting point
- Need to adapt JOREK to treat the compressed and factorized matrices
- Already implemented hybrid MPI/OpenMP parallelization for additional operation

The chosen matrices have dimesions (*w*, *bez*) or (bez, *w*)

w = wall dof → <u>kept constant for every order *k* of toroidal harmonics</u>

bez = Bezier dof times Fourier harmonics → <u>**changes** with different *k*</u>



NOTE: the compression is applied on the direction of bez

Outline

1. Brief introduction to JOREK and the response matrices
2. Organization of the work
3. Matrix Compression
   a. Chosen method
   b. How it is implemented
4. **How the method preforms within applications**
   a. **Memory Savings**
   b. Issues and Idea on How we are addressing

Memory compression for selected test cases

r_r = rate of retained singular values → smaller = more compression

OK
ISSUES

Tearing Mode (TM)	
r_r	Size [GiB]
-	0.121
1.0	0.126
0.75	0.0944
0.5	0.0629
0.25	0.0315
0.2	0.0252
0.15	0.0189
0.1	0.0126
0.05	0.00629
0.025	0.00315

Vertical Displacement Event (VDE)			
r_r	Size n=0 [GiB]	Size n=9 [GiB]	Size n=19 [GiB]
-	0.0273	0.245	0.518
1.0	0.0275	0.265	0.605
0.75	0.0206	0.199	0.454
0.5	0.0137	0.132	0.302
0.4	0.0110	0.106	0.242
0.3	0.00825	0.0795	0.181
0.25	0.00687	0.0662	0.151

TAKEAWAY MESSAGES

- Linear scaling of matrix size is respected
- VDE test case needs more development

Outline

1. Brief introduction to JOREK and the response matrices
2. Organization of the work
3. Matrix Compression
    a. Chosen method
    b. How it is implemented
4. **How the method preforms within applications**
    a. Memory Savings
    b. **Issues and Idea on How we are addressing**

Toroidal harmonics representation and compression

Given a response matrix A, it is composed of blocks as the sketch on the side, for each node element at the boundary of the plasma

Issues shown for the previous tests of VDE are probably due to the fact that the n=0 part is sensitive to compression

The current development is testing the possibility to leave the n=0 parts of the matrices uncompressed