Ensemble propagation for efficient uncertainty quantification: Application to the thermomechanical modeling of a first mirror for the ITER core CXRS diagnostics



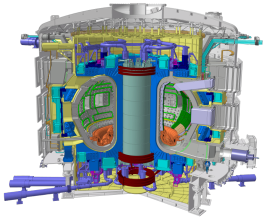
K. Liegeois[1], R. Boman[1], E. T. Phipps[2], Ph. Mertens[3], Y. Krasikov[3] and M. Arnst[1]

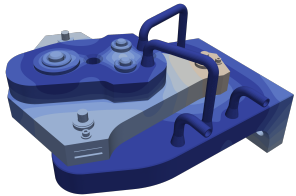
[1]Aerospace and Mechanical Engineering, Université de Liège, Belgium
[2]Sandia National Laboratories, USA
[3]Forschungszentrum Jülich, Germany

UNCECOMP 2019
Hersonissos, Greece
June 25, 2019

Ongoing PhD: New methods for parametric computations with multiphysics models on HPC architectures with applications to design of opto-mechanical systems

ITER

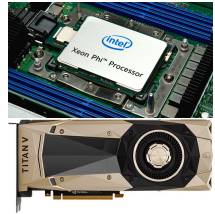




High performance computing library

Clusters

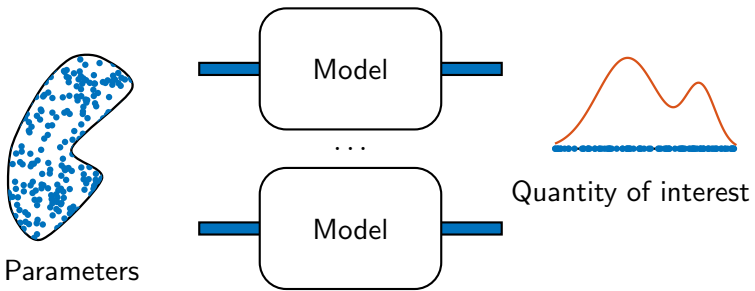
Emerging architectures



Parametric computations

Sampling-based parametric computations typically require numerous calls of potentially **costly models**.

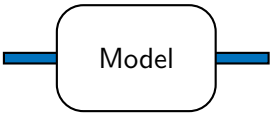
Example: Monte Carlo for uncertainty quantification.

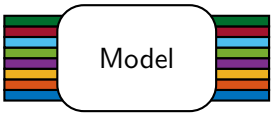


Goal of this work: to **reduce** the **CPU time** to evaluate a given set of samples.

Ensemble propagation

In sampling-based parametric computation, instead of individually evaluating each instance of the model, Ensemble propagation (EP) consists of **simultaneously evaluating** a **subset of samples** of the model.





EP was introduced by [Phipps, 2017], made available in **Stokhos** a package of **Trilinos**, and implemented using a **template-based generic-programming** approach:

```
template <typename T, int ensemble_size>
class Ensemble{
    T data[ensemble_size];
    Ensemble<T,ensemble_size> operator+ (const Ensemble<T,ensemble_size> &v);
    Ensemble<T,ensemble_size> operator- (const Ensemble<T,ensemble_size> &v);
    Ensemble<T,ensemble_size> operator* (const Ensemble<T,ensemble_size> &v);
    Ensemble<T,ensemble_size> operator/ (const Ensemble<T,ensemble_size> &v);
    //...
}
```

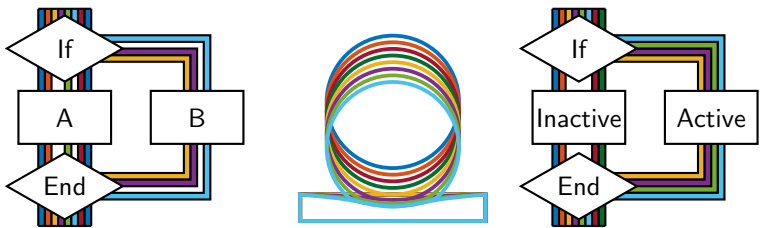
Ensemble propagation

Advantages of the EP:

- Reuse of common variables,
- More opportunities for SIMD (more data parallelism),
- Improved memory usage,
- Reduction of Message Passing Interface (MPI) latency per sample.

Challenges of the EP:

- Increased memory usage,
- Ensemble divergence:
  - control flow divergence: if-then-else divergence and loop divergence,



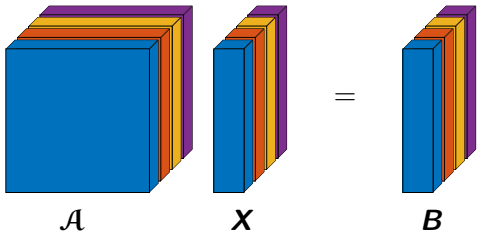
  - function call divergence.

Parametric linear systems

We want to solve a **parametric linear system** for a subset of s samples of the parameters together:

$$\boldsymbol{A}_{::\ell}\, \boldsymbol{x}_{:\ell} = \boldsymbol{b}_{:\ell} \quad \text{for all} \quad \ell = 1, \ldots, s.$$

Representation of a system for $s = 4$:



How to solve efficiently the parametric linear system with EP?

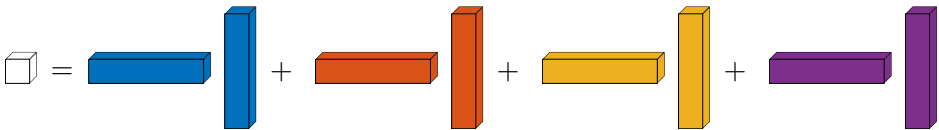
Reduced inner product used in conjugate gradient method

First approach [Phipps, 2017]: to gather the sample matrix into a block diagonal matrix

$$\begin{bmatrix} \boldsymbol{A}_{::1} & & \\ & \ddots & \\ & & \boldsymbol{A}_{::s} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{:1} \\ \vdots \\ \boldsymbol{x}_{:s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_{:1} \\ \vdots \\ \boldsymbol{b}_{:s} \end{bmatrix},$$

and to apply the conjugate gradient method on the block diagonal system.

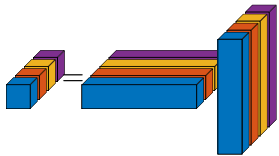
Mathematically equivalent to defining a **reduced inner product**:



Advantages: No ensemble divergence and possibility to use efficient BLAS implementations,
Challenges: The samples are coupled together, the spectra are gathered, and the condition number increases.

Ensemble-typed inner product used in conjugate gradient method

Second approach [D'Elia, 2017]: to avoid the coupling of the samples together using an **ensemble-typed inner product**:



It was first introduced for grouping purpose.

Advantages: No coupling: each sample converges as fast as if it was propagated alone,
Challenges: Every ensemble divergence has to be managed explicitly.

Occurrence of ensemble divergence in GMRES

$\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(0)}$
$\beta = \|\boldsymbol{r}^{(0)}\|$
$\boldsymbol{v}_{:1} = \boldsymbol{r}^{(0)}/\beta$
for $j = 1, \ldots, m$ **do**
  $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{M}^{-1}\,\boldsymbol{v}_{:j}$
  $\boldsymbol{h}_{(1:j)j} = \boldsymbol{V}_{:(1:j)}^{\mathrm{T}}\,\boldsymbol{w}$
  $\boldsymbol{v}_{:(j+1)} = \boldsymbol{w} - \boldsymbol{V}_{:(1:j)}\,\boldsymbol{h}_{(1:j)j}$
  $h_{(j+1)j} = \|\boldsymbol{v}_{:(j+1)}\|$
  if $h_{(j+1)j} \neq 0$ **then**
    $\boldsymbol{v}_{:(j+1)} = \boldsymbol{v}_{:(j+1)}/h_{(j+1)j}$
  else
    $m = j$
    break
  if $\boldsymbol{q}_{:(j+1)}^{\mathrm{T}}\boldsymbol{e}_1 \leq \varepsilon$ **then**
    $m = j$
    break
$\boldsymbol{y} = \arg\min_{\boldsymbol{z}} \|\beta\,\boldsymbol{e}_1 - \boldsymbol{H}_{(1:m+1)(1:m)}\,\boldsymbol{y}\|$
$\boldsymbol{x}^{(m)} = \boldsymbol{x}^{(0)} + \boldsymbol{M}^{-1}\boldsymbol{V}_{:(1:m)}\,\boldsymbol{y}$

Ensemble divergence in GMRES:

1. an Arnoldi vector can require a normalization or not: **if-then-else divergence**,
2. different samples may require different numbers of iterations to converge: **loop divergence**,
3. called BLAS functions, such as GEMV for the dense matrix-vector operations, may not support ensemble-typed inputs, leading to **function call divergence**.

Efficient ensemble GMRES with ensemble-typed inner product

In [Liegeois, in preparation], we describe and implement an efficient ensemble GMRES without ensemble reduction.

The control flow divergence and the function call divergence have been solved by:

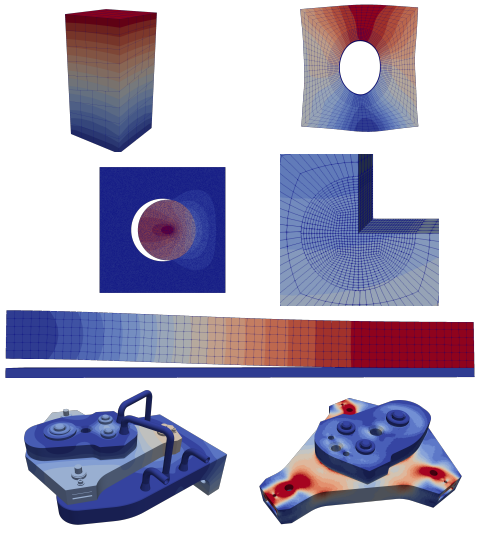
- Implementing a Mask class which is used to apply masked assignment and logical reduction;
- Implementing an efficient ensemble GEMV for the orthogonalization process.

Those two contributions lead to:

- An equivalent cost per iteration of ensemble GMRES with and without reduction;
- A safe implementation which is able to deal with converged samples.

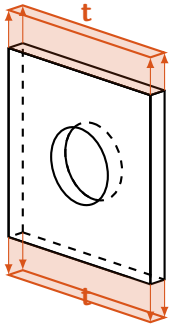
Implemented code and its capabilities

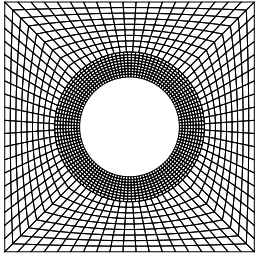
- **Fully templated C++** code heavily based on **Trilinos** which provides a fully templated solver stack;
- Embedded in a **Python** interface. This eases the looping around samples, the grouping of samples together, etc;
- **Hybrid parallelism** based on **Tpetra** with **MPI** for distributed memory and **Kokkos** with **OpenMP** for shared memory;
- Uses **Gmsh** [Geuzaine, 2009] to import 3D meshes and **VTK** to write the output files;
- Has already generated preliminary results for **industrial thermomechanical contact problems**.



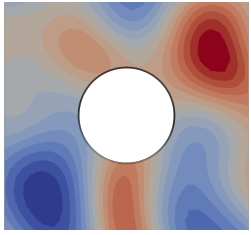
Test case 1: mesh tying problem

- Plate with a hole pulled on two opposite sides;
- Two meshes glued with the Mortar finite element method in saddle point formulation;
- Lamé parameters represented as a Gaussian random field;
- Multigrid preconditioner with saddle point matrix on each multigrid level;
- Solved on Intel(R) Xeon(R) Platinum 8160 CPU.

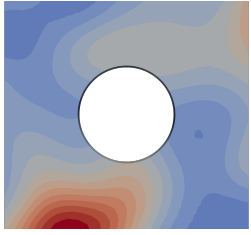




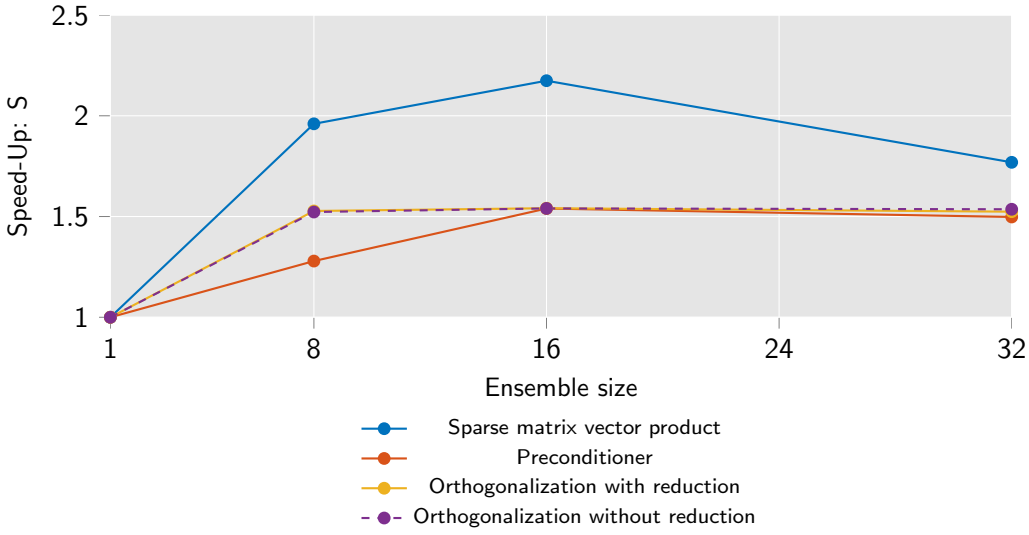
Realization 1:



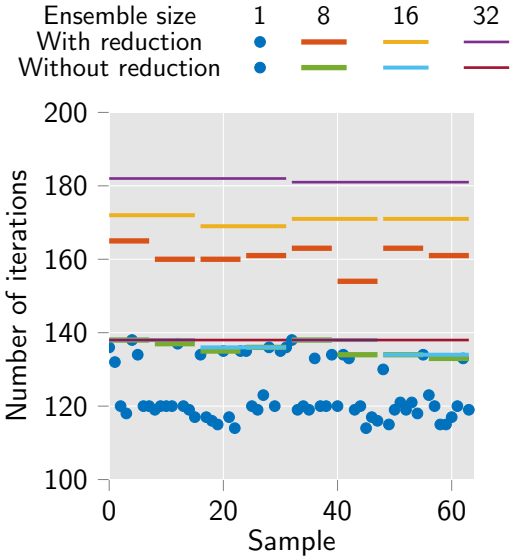
Realization 2:

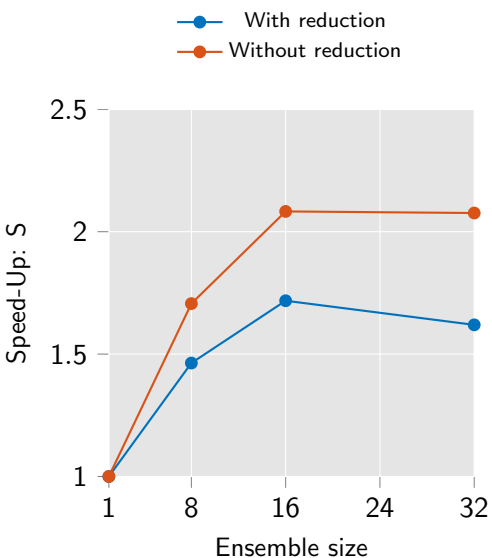


Test case 1: speed-up of one GMRES iteration



Test case 1: total speed-up

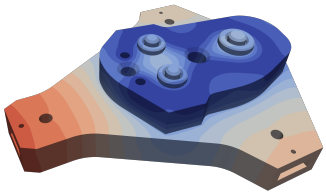


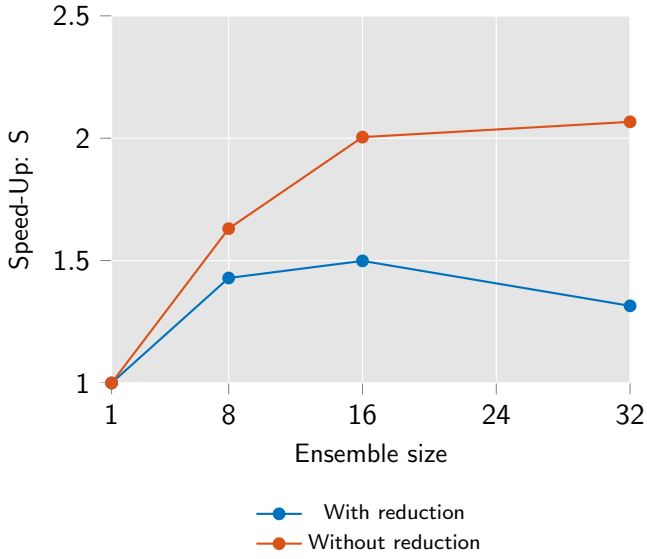


Test case 2: first results on the ITER mirror

MPI rank:

Temperature:





Conclusion

Conclusion and contributions:

- Two variants of GMRES can currently be used: with reduced inner product and with ensemble-typed inner product;
- Cost per iteration of ensemble GMRES is independent of coupling the samples together with ensemble reduction;
- Ensemble GMRES without reduction is faster due to an improved convergence compared to ensemble GMRES with reduction.

Future work:

- Finalize the application of the method on engineering problems relevant for ITER in collaboration with FZ. Jülich;
- Finalize the testing on more than one computational node to leverage the increased memory usage.

References

- M. D'Elia, E. T. Phipps, A. Rushdiz, and M. S. Ebeida, Surrogate-based Ensemble Grouping Strategies for Embedded Sampling-based Uncertainty Quantification. arXiv preprint arXiv:1705.02003, 2017.

- K. Liegeois, R. Boman, E.T. Phipps, T. Wiesner, and M. Arnst, Efficient GMRES with Ensemble Propagation: Application to non-Symmetric-Positive-Definite Mechanical Problems, In preparation.

- E.T. Phipps, M. D'Elia, H C. Edwards, M. Hoemmen, J. Hu, and S. Rajamanickam, Embedded ensemble propagation for improving performance, portability, and scalability of uncertainty quantification on emerging computational architectures. SIAM Journal on Scientific Computing, 2017, vol. 39, no 2, p. C162-C193.

Acknowledgement

The first author, Kim Liegeois, would like to acknowledge the Belgian National Fund for Scientific Research (FNRS-FRIA) and the Federation Wallonia-Brussels (FW-B) for their financial support.