Skoltech

Skolkovo Institute of Science and Technology

SVD compression of results from the Finite Element Method

NLA project, 2019

Vladimir Dmitriev, Veronika Zorina,
Alexander Shumilov, Daria Gazizova

Plan

- Idea
- SVD compression algorithm
- FEM
- Results and discussion

Idea

- Amount of data produced by a complex finite element analysis can be **enormous**!

- Want to **save storage capacity** and also **accelerate the data transfer** between computers as the analysis itself and **make visualization of the results easier** even in the personal computer.

SVD compression

The idea of the SVD compression: taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space that exposes the substructure of the original data more clearly and orders it from the largest variation to the least.

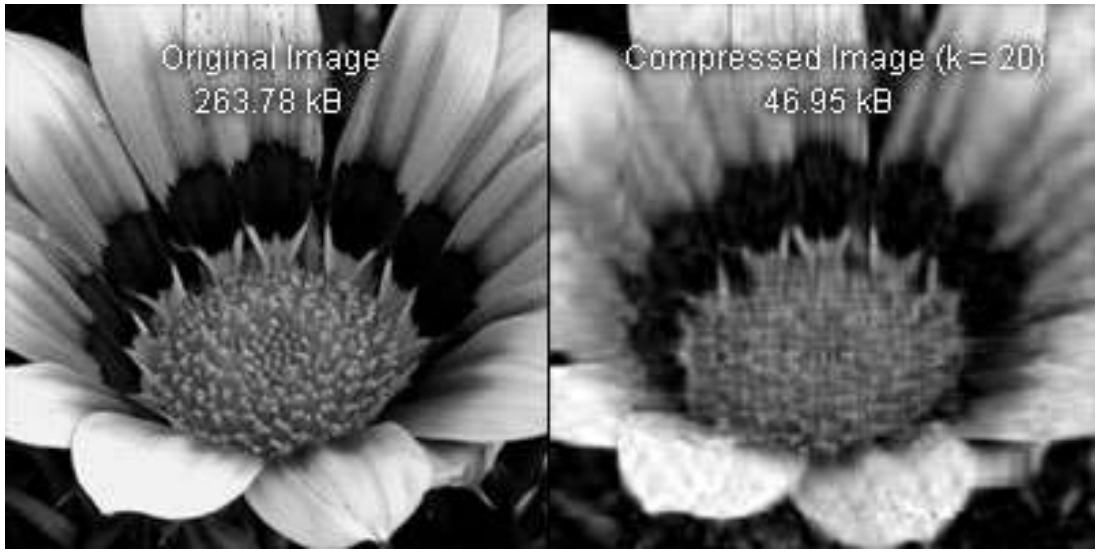

Fig.1 Usage in image compression

SVD compression algorithm

1) Construct a low-dimensional subspace that captures the action of the matrix. In other words, in this stage we need to compute an approximation basis for the range of input matrix A:

$$A \approx QQ^T A$$

r vectors of Q could be found from randomize method: The result of applying A to any vector is a vector in the range of A, and if the matrix is applied to r random vectors, the results will nearly span the range of A with extremely high probability

SVD compression algorithm

2) Use Q to find approximate SVD factorization of A:

- Construct $B = Q^T A$
- Compute an exact SVD of the small matrix: $B = WS'V'^T$
- Set $U' = QW$

$$A \approx QQ^T A = QB = QWS'V'^T = U'S'V'^T$$

For a dense input matrix, randomized SVD algorithm requires $O(m*n*\log(r))$ floating-point operations, substantially less than classical algorithms.

Error estimation

In this work we
used **Normalized Rooted Mean Square Deviation**:

$$NRMSD = \frac{RMSD}{X_{max} - X_{min}} = \frac{\sqrt{MSE}}{X_{max} - X_{min}}$$

Application in finite element method

Since a complex finite element analysis can produce large amount of data, SVD compression algorithm is suggested to help post-process results from finite element solvers. Namely, compress images representing solutions of PDE.

As for solver, we chose **FEniCS** - open-source computing platform for solving PDEs.

FENICS PROJECT

Problem - Poisson equation

We want to compute the deflection D(x,y) of a two-dimensional, circular membrane, subject to a load p over the membrane:

$$-T\nabla^2 D = p \quad \text{in } \Omega = \{(x,y)\,|\,x^2+y^2 \leq R\}$$

T - tension in the membrane (constant), p - is the external pressure load, boundary conditions – D = 0.

A localized load can be modeled as a Gaussian function:

$$p(x,y) = \frac{A}{2\pi\sigma}\exp\left(-\frac{1}{2}\left(\frac{x-x_0}{\sigma}\right)^2 - \frac{1}{2}\left(\frac{y-y_0}{\sigma}\right)^2\right)$$

Solutions of PDE*

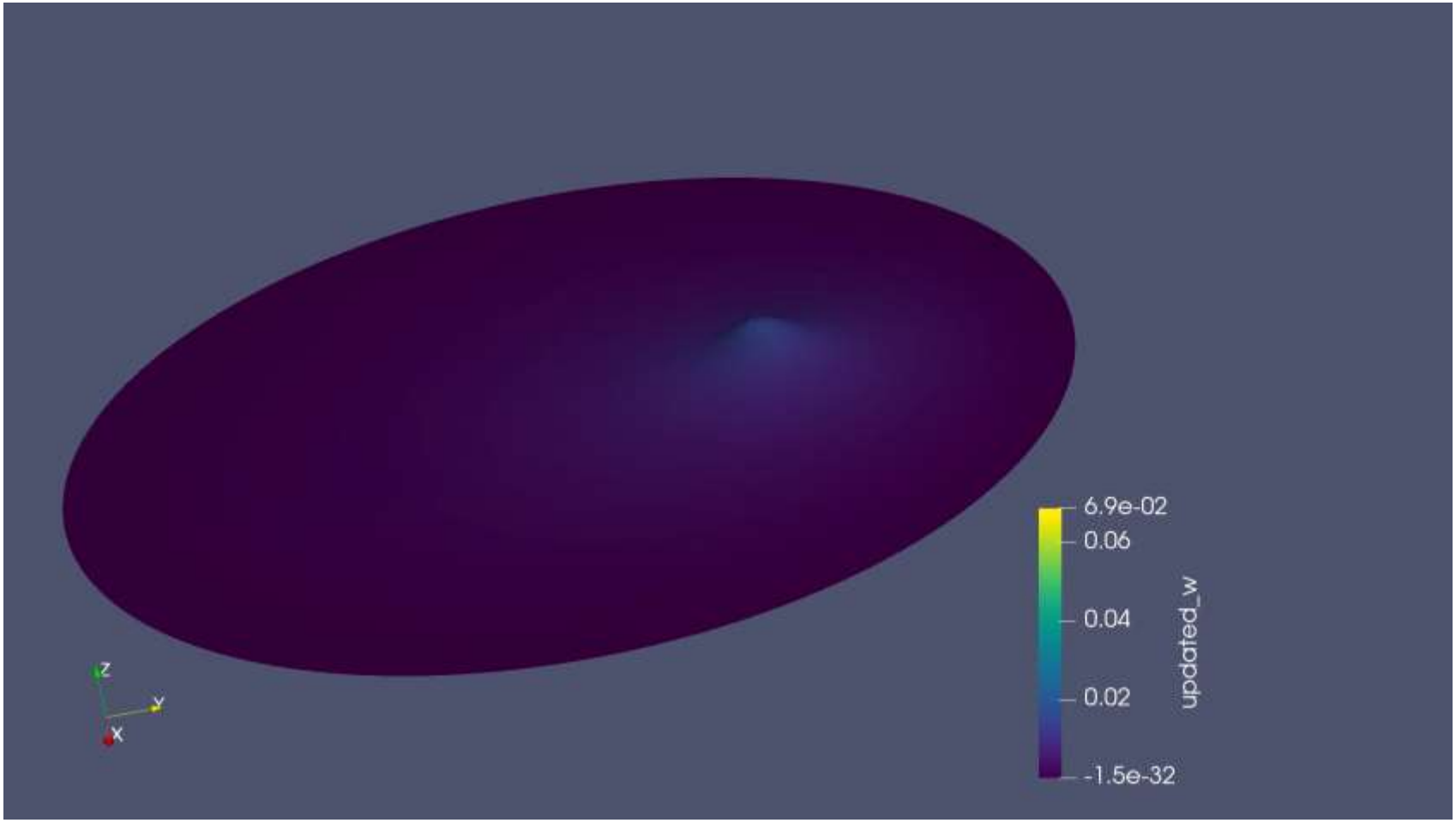

Fig.3 Solution with increasing load. Optimal rank = 8

* Visualized in Paraview

Solution of PDE

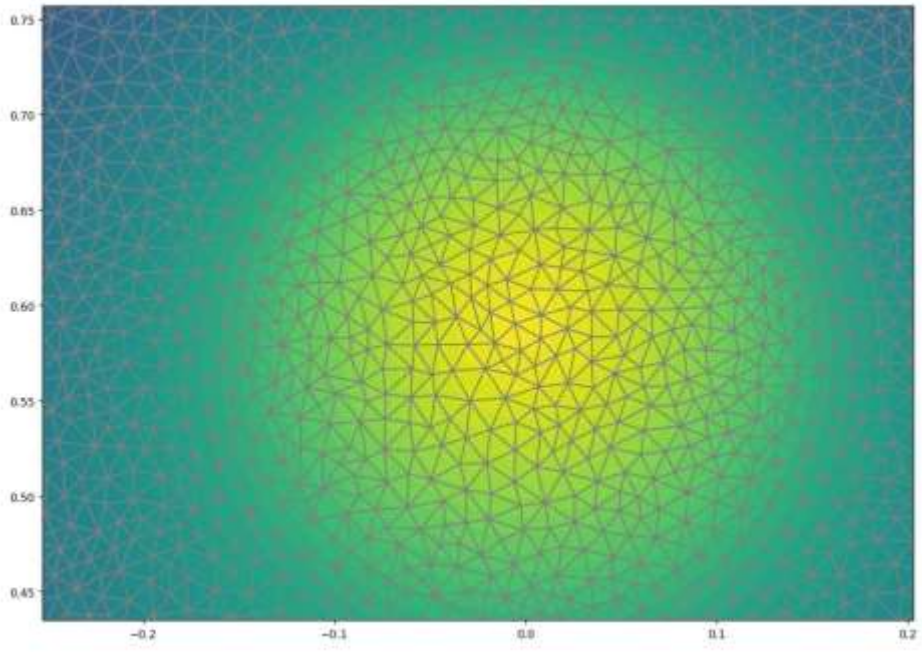

Fig.4 Example of mesh for our FEM

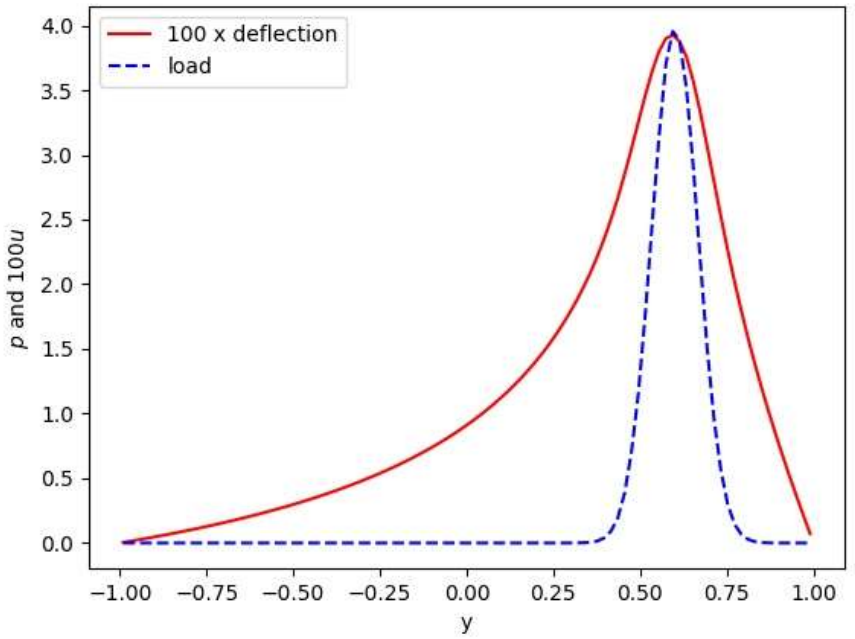

Fig.5 Comparison between load and corresponding membrane deflection

Dependence of NRMSD on optimal rank for results of FEM

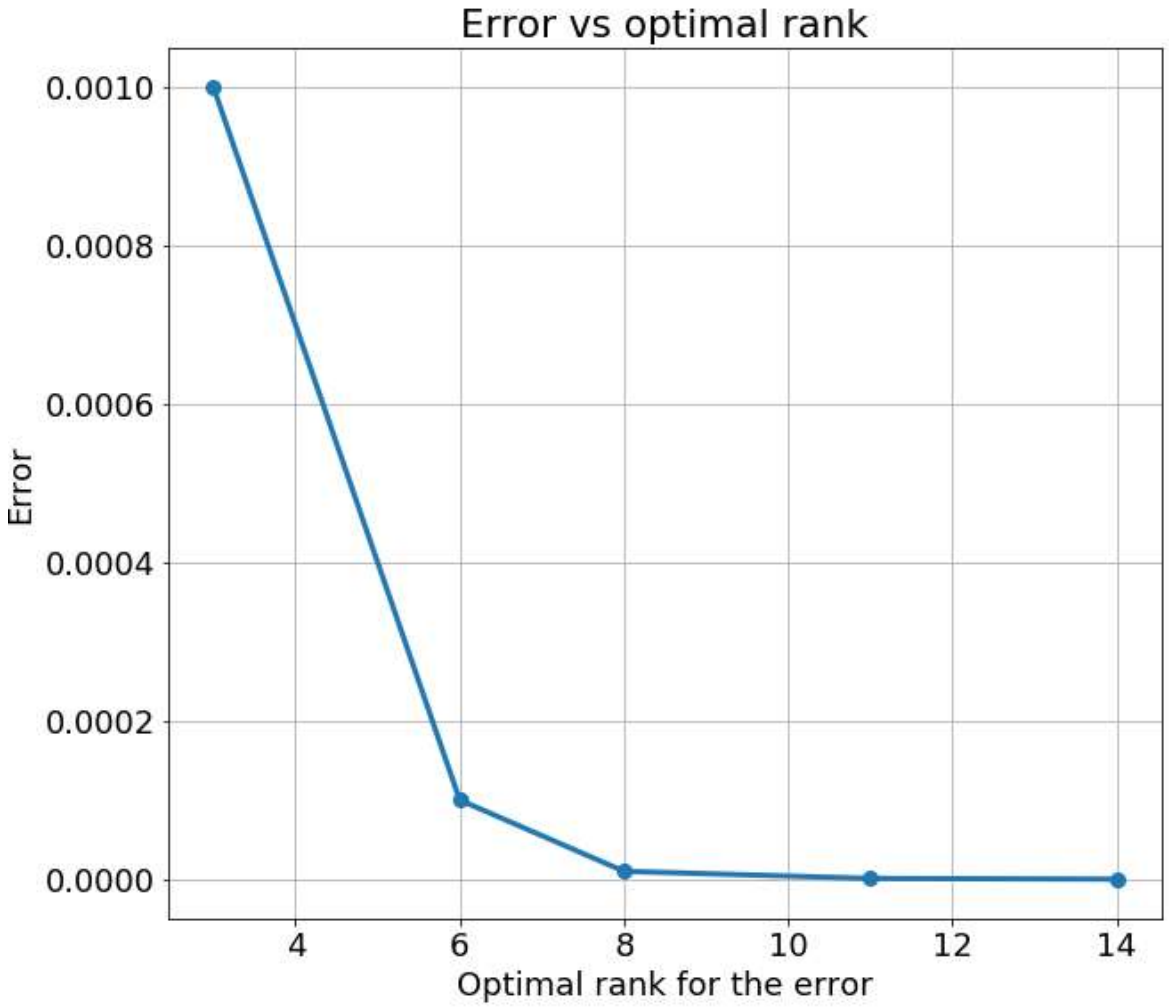

Fig.6

Comparison of standard and randomized SVD

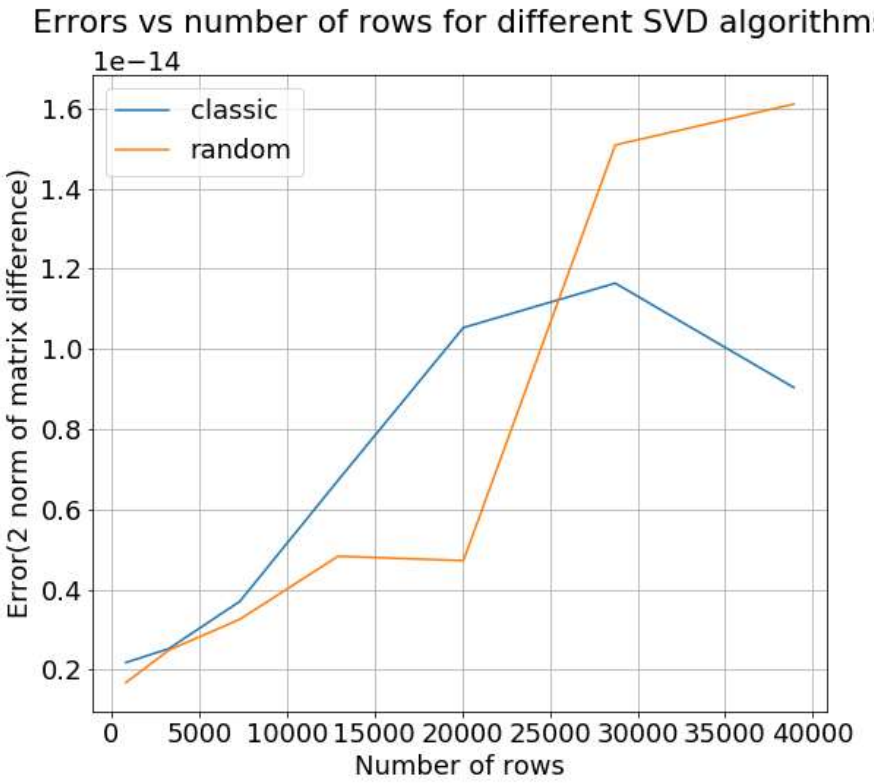

Fig. 7

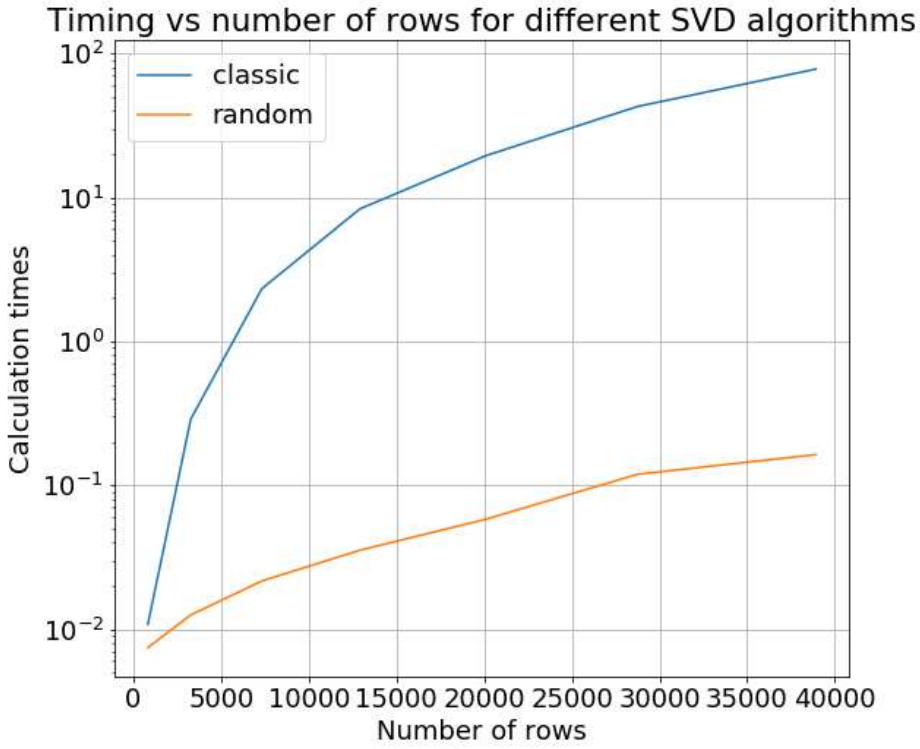

Fig. 8

Difference for solution from FEM

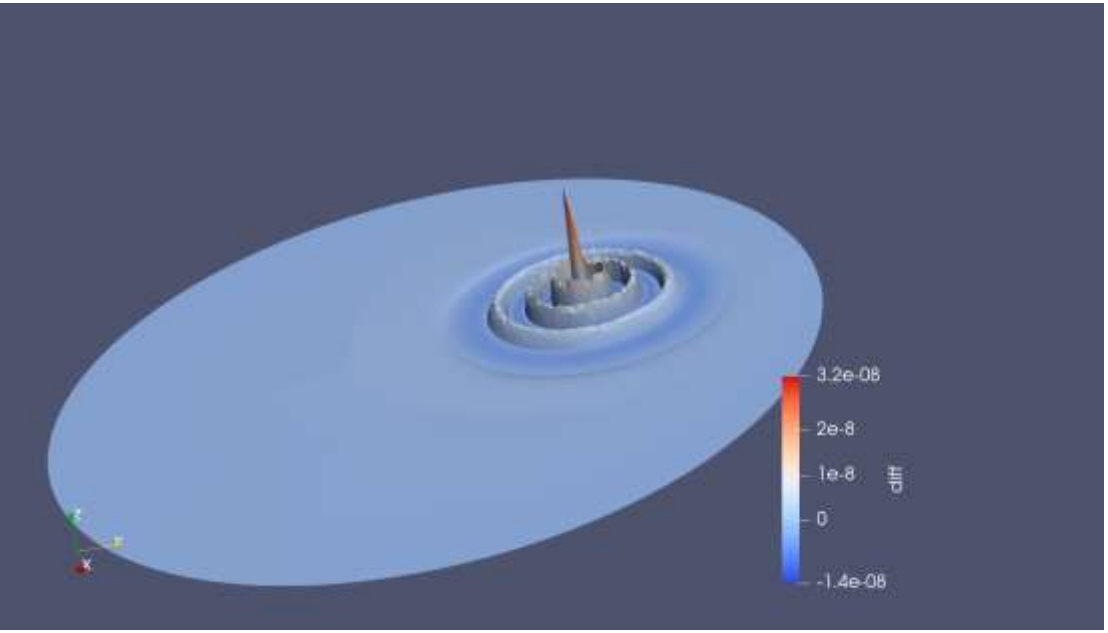

Fig.9.1 Difference between original and compressed solution with increasing load. Optimal rank = 8

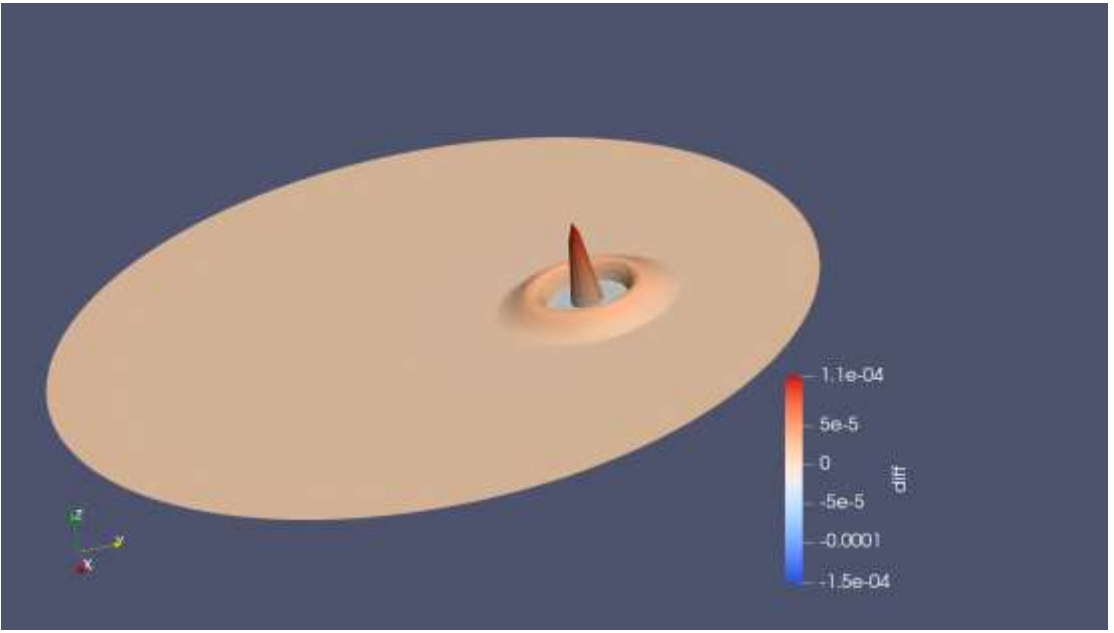

Fig.9.2 Difference between original and compressed solution with increasing load. Optimal rank = 3

References

- Štěpán Beneš, Jaroslav Kruis. Singular Value Decomposition used for compression of results from the Finite Element Method, *Advances in Engineering Software 117 (2018) 8–17, 2017.* *https://doi.org/10.1016/j.advengsoft.2017.12.007*
- Martinsson P, Rokhlin V, Tygert M. A randomized algorithm for the decomposition of matrices. *Appl Comput Harmon Anal 2011;30(1):47–68*. https://doi.org/10.1016/j.acha.2010.02.003.
- Szlam A, Kluger Y, Tygert M. An implementation of a randomized algorithm for principal component analysis. *J ACM (JACM)* 2014;1(1).
- Halko N, Martinsson P, Tropp J. *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev 2011;53(2):217–88*
- Witten R, Candes E. *Randomized algorithms for low-rank matrix factorizations: sharp performance bounds. Algorithmica 2015;72(1):264–81.*
- Logg, Mardal, Wells. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book (Lecture Notes in Computational Science and Engineering),* 2012. ISBN-13: 978-3642230981

Thank you for your attention!