The Journal of Open Source Software

Paraqus: Exporting Finite Element Simulation Results from Abaqus to VTK

Tim Furlan [1*¶], **Jonathan Stollberg** [1*], and **Andreas Menzel** [1,2*]

1 Institute of Mechanics, Department of Mechanical Engineering, TU Dortmund University, Leonhard-Euler-Str. 5, 44227 Dortmund, Germany **2** Division of Solid Mechanics, Department of Construction Sciences, Lund University, P.O. Box 118, SE-22100 Lund, Sweden ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.05729

Software

- Review
- Repository
- Archive

Editor: Prashant Jha

Reviewers:

- @lelaus

Submitted: 28 February 2023
Published: 30 January 2025

Summary

The finite element (FE) method is one of the approaches available for solving different types of partial differential equations in the various fields of physics, such as solid mechanics, heat transfer, and electromagnetics. Visualization and manipulation of results play a crucial role in the interpretation and analysis of simulation results, especially when complex and advanced problems are considered. Paraqus is a Python package that exports simulation results from the commercial FE software Abaqus to the open VTK file format, allowing researchers to use tried and tested pipelines for the visualization, and to exchange results in a format that is independent of the software they were created with. Paraqus is modular in structure, therefore the VTK exporting capability can be used independently of Abaqus, and exporters for other FE software can be added as well. Since VTK is not limited to FE results, the package could be used to export results from other simulation methods, such as CFD simulations, by implementing appropriate exporters. The Paraqus documentation is available at https://paraqus.readthedocs.io.

Statement of need

Abaqus is an example of commercial FE software that is widely used in the academic engineering community. ScienceDirect shows more than 7000 articles tagged with the keyword “Abaqus” and published in 2022. Exporting simulation results to the VTK format serves two purposes: - the data can be shared with other researchers and users without access to expensive software, and - specialized open source software like ParaView can be used to create visualizations that would be tedious or impossible to generate in the post-processing module of Abaqus.

Some software packages similar to Paraqus have been released over the last years. However, these tools suffer from great limitations and are, therefore, not as versatile and straightforward as Paraqus. A script to export Abaqus results to VTK format was published in Liu et al. (2017) under the name `odb2vtk`. This script offers limited options to customize the export data, and supports only a subset of the finite elements available in Abaqus. *AbaPy* is a Python package that is not aimed solely at post-processing, but also at the automatic creation of Abaqus models and run of Abaqus simluations (Charleux et al., 2016). Although *AbaPy* offers the option to export field outputs to VTK format, the wider range of applications comes with increased complexity in terms of both implementation and usage. Paraqus falls between these existing options: it is organized as a package with a small and intuitive API, but allows a wide range of customization of exports, and writes the more efficient binary version of the VTK format. It also provides options to group exports from multiple simulations, time steps, or bodies, using ParaView data (PVD) files. Furthermore, Paraqus performs exports noticeably faster than *AbaPy*, especially in case of large models.

Syntax example

While extensive examples for the usage of Paraqus can be found in the documentation, a short syntax example is given in this section. After specifying the variables for the output file path, instance name and step name, the code below can be executed in the Abaqus Python shell or as a script file in Abaqus Python. The reader is referred to the examples folder of the repository for examples that include directions to create example Abaqus output database (ODB) files.

```
# import the Paraqus classes to read Abaqus output and
# to store it as an ASCII-based VTK file
from paraqus.abaqus import OdbReader
from paraqus.writers import AsciiWriter

# set some constants based on the ODB that will be exported
ODB_PATH = "my_abaqus_output.odb"  # path to the Abaqus ODB
MODEL_NAME = "My Model"  # can be chosen freely for the output
INSTANCE_NAMES = ["PART-1-1"]  # choose which instances will be exported
STEP_NAME = "Step-1"  # name of the step that will be exported
FRAME_INDEX = -1  # export the final frame (timestep) of the step

# the class OdbReader is used to read results from Abaqus ODBs
reader = OdbReader(odb_path=ODB_PATH,
                   model_name=MODEL_NAME,
                   instance_names=INSTANCE_NAMES,
                   )

# request output of displacements at node points
reader.add_field_export_request("U", field_position="nodes")

# request output of temperature at element centroids
reader.add_field_export_request("TEMP", field_position="elements")

# the class AsciiWriter is used to write results in ASCII-based VTK format
vtk_writer = AsciiWriter("my_output_directory", clear_output_dir=True)

# there might be multiple instances, so we create a list of the models
instance_models = list(reader.read_instances(step_name=STEP_NAME,
                                             frame_index=FRAME_INDEX))
instance_model = instance_models[0]  # this is a ParaqusModel instance

# use the writer to write instance_model to disk
vtk_writer.write(instance_model)
```

Contributions

Tim Furlan: Conceptualization; Software - Design, Implementation (focus on reading Abaqus ODB files), Documentation; Writing - Original Draft; Example generation. **Jonathan Stollberg:** Conceptualization; Software - Design, Implementation (focus on writing VTK files), Documentation; Writing - Original Draft; Example generation. **Andreas Menzel:** Conceptualization; Supervision - Project direction; Writing - Review & Editing; Example generation

Acknowledgements

The authors are grateful to Isabelle Noll and Lennart Sobisch for beta-testing Paraqus and providing valuable feedback on the examples and documentation. Furthermore, the detailed review of the paper and code by Gabriele Ottino (newcleo S.p.A.) is gratefully acknowledged.

Funding

Paraqus was developed in the context of two projects funded by the German Research Foundation (DFG) under project IDs 403857741 and 278868966, which is gratefully acknowledged.

References

Charleux, L., Bizet, L., Keryvin, V., & mousta. (2016). *Abapy: abapy_v1.1*. https://doi.org/10.5281/zenodo.50550

Liu, Q., Li, J., & Liu, J. (2017). ParaView visualization of Abaqus output on the mechanical deformation of complex microstructures. *Computers & Geosciences*, *99*, 135–144. https://doi.org/10.1016/j.cageo.2016.11.008