

SUPPLEMENTARY MATERIAL INFO

As supplementary material, to our article “**Towards an automated framework for the finite element computational modelling of directed energy deposition**” published in **Finite Elements in Analysis and Design**, we provide the in-house developed Fortran/Python code and all resulting Abaqus files used in the numerical case study.



Please cite this work:

D. Kovšca, B. Starman, D. Klobčar, M. Halilović, N. Mole, "Towards an automated framework for the finite element computational modelling of directed energy deposition," *Finite Elem. Anal. Des.* 221 (2023), 103949. <https://doi.org/10.1016/j.finel.2023.103949>.

Developed Fortran/Python code

The in-house developed Fortran/Python code is used to pre-process the activation criterion (activation time). The main code is contained in the `gcode_reader.py` file as a Python class. Therefore, an object should be created as follows:

```
gcode = gcode_reader(delta_t)
```

where `delta_t` represents the time increment argument. Several other attributes can be set via the `gcode.set_methods`, while the main object method is called as follows:

```
Atime, Bpos = gcode.main(gcode_lines, centroids)
```

Input arguments to this method represent a list of G-code lines `gcode_lines` and a list of finite element (FE) centroids `centroids`. During the code execution, a Fortran subroutine contained in the `gcode_activation.f90` file is called several times. The `gcode.main()` method returns a list of activation times `Atime` and a list of beam welding coordinates in time `Bpos`. These two lists should then be exported to `Atime.dat` and `Bpos.dat` files for use with the Abaqus subroutines. The source G-code for the numerical case study is included in the `cylindrical_part.gcode` file.

The `gcode sele.f90` Fortran subroutine is used to search for adjacent FEs by comparing mutual nodes. As an input, the subroutine takes an array of nodes belonging to each FE and returns an array of neighbouring elements which should be exported to `Sele.dat` file for use with the Abaqus subroutine.

Abaqus files used in the numerical case study

The implemented Abaqus/Standard subroutines (USDFLD, DFLUX and DFILM) are contained in separate files for heat `user-DED-Heat.for` and stress `user-DED-Stress.for` analysis. `AKT_TIME`, `BEAM_POS`, and `IS_ELE` are variables that should be passed in from the `Atime.dat`, `Bpos.dat` and `Sele.dat` files, respectively.