# An open-source ABAQUS Implementation of the scaled boundary finite element method to study interfacial problems using polyhedral meshes: Supplementary file

Shukai Ya[a], Sascha Eisenträger[a], Chongmin Song[a], Jianbo Li[b,*]

[a]*School of Civil and Environmental Engineering, University of New South Wales, Sydney 2052, Australia*

[b]*Institute of Earthquake Engineering, State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116024, China*

## 1. Introduction

A brief description of the utilization of ABAQUS polyhedral user elements based on the scaled boundary finite element method (SBFEM) is presented in this file. The user element is implemented through ABAQUS user subroutine UEL. In order to store the polyhedral mesh, the user subroutine UEXTERNALDB which is used to manage user external databases is involved in the current implementation. These two subroutines are written in FORTRAN language and incorporated into one single source code, which is named as *sbfem.for* in the current repository.

## 2. System requirements

- ABAQUS installation. In the current work, ABAQUS 2017 is used, which has the availability of pyramid element C3D5. The ABAQUS 6.16 and later versions are recommend. The pyramid element C3D5 is used for overlaying the volume sector scaled from a quadrilateral surface of a polyhedral element to define element-based surface.

- Intel FORTRAN compiler. Intel Parallel Studio XE2016 is used in this work, and it should be linked with the ABAQUS. The linking method could be found in the work by Dr. Petri Tanska (https://www.researchgate.net/publication/313924098).

## 3. Input files

In order to run an analysis in ABAQUS featuring the proposed user element, four files should be stored in the working directory. They are:

- The source code (sbfem.for) of the implementation containing UEL and UEXTERNALDB;

- An ABAQUS input file (*.inp) in which the user elements are defined;

- A text file (*.txt) as supplementary input file to store the polyhedral mesh in COMMON blocks;

- A used defined include file (inp_param.inc) specifying the dimensions of arrays in the COMMON blocks.

The detailed explanations of the implementation and input formats are given in the original paper.

*Corresponding author

*Email address:* jianboli@dlut.edu.cn (Jianbo Li)

## 4. Running the simulation

The actual analysis is carried out by executing the following command in the working directory:

```
abaqus job=<input file name> user=<fortran file name>.
```

An additional function provided in this work is allowing users to monitor the progress of a simulation. To active this function, users are suggested to execute the following command:

```
abaqus job=<input file name> user=<fortran file name> int.
```

In this case, the current analysis procedures will be shown in the command window.

## 5. Outputs

Except for the default output files of ABAQUS, such as *.obd, *.dat, *.msg, there are two user output files generated through UEXTERNALDB:

- A text file recording the analysis state history and used CPU time for each individual analysis procedure named ANA_STAT.txt;
- A text file providing recommendation of the parameters defined in the include file named PARA_SET.txt.

## 6. Examples

Five examples are presented in the current repository, which are the numerical examples presented in the original paper, i.e., the patch test in Section 4.1.1, the transient analysis of a cantilever beam in Section 4.1.3, the patch contact test in Section 4.2.1, the patch test including cohesive elements in conjunction with contact in Section 4.2.2, and the cube with spherical inclusion in Section 4.2.3.

One should note that, after performing these examples, the user elements are not visible in the result files (*.obd). For the two examples from Section 4.2.1 and Section 4.2.2, only the overlaid standard elements are visible in the result files, which have negligible stiffness and do not affect the analysis results. These overlaid standard elements are used to create element-based surfaces for the purpose of assigning pressures or establishing interactions at the interfaces. The SBFEM user elements are assigned to the polyhedral meshes at the interface for the example from Section 4.2.3.

The authors are not responsible for any adverse/beneficial outcomes of using the implementation by individuals for any purposes.