About user postprocessing of Abaqus results files

You can write and utilize postprocessing programs to manipulate data stored in the Abaqus results file.

This page discusses:

- Initialization
- Data processing

The Abaqus results file is identified by the file extension `.fil` and contains results based on user-specified output requests; see Output to the Data and Results Files for more information on the types of data that are written to the results file. The standard file format is binary, but it can be changed to ASCII format for each run by user request. Alternatively, it can be set to a default ASCII format during site installation. Abaqus uses Fortran unit `8` to communicate with the results file.

Sample postprocessing programs that perform commonly exercised tasks are presented in separate sections in this chapter. These include merging multiple results files and converting the resulting results file from binary format to ASCII, or vice-versa; computing principal values and directions of stress and strain; and computing a perturbed mesh for a collapse analysis by incorporating a user-specified geometric imperfection in the form of the critical buckling mode shape.

Each postprocessing program must be linked using the **make** parameter when running the Abaqus execution procedure (see Making User-Defined Executables and Subroutines). To link properly, the postprocessing program cannot contain a Fortran PROGRAM statement. Instead, the program must begin with a Fortran SUBROUTINE with the name ABQMAIN.

General programming concepts, Abaqus Fortran interfaces, and data processing concepts are described below. Refer to File Output Format for more information. The program listings in each section provide details on the program flow, how to interface with various computer platforms that use different operating systems and Fortran compilers, and how to interface with Abaqus subroutines to handle data files and records.

Refer to Using the Abaqus Scripting Interface to access an output database or Using C++ to access an output database for information on accessing data stored in the Abaqus output database.

Initialization

Details about the variables that are used in the postprocessing programs are discussed in Accessing the Results File Information. Abaqus uses a 512-word buffer named `ARRAY` for the reading and writing of data on the results file. This is dimensioned as `ARRAY(513)`. The integer equivalent is `JRRAY(513)` for a 64-bit computer or `JRRAY(2,513)` for a 32-bit computer. The `EQUIVALENCE` statement is used to equivalence `ARRAY` and `JRRAY` to simplify manipulation of real and integer numbers in the data record stored in the buffer.

The information concerning the Fortran unit number and format of the results file that is read is defined in `LRUNIT(2,NRU)`, where `NRU` is the number of files to be processed. The Fortran unit number for the *n*th file is stored in `LRUNIT(1,n)`. The information about the file format is stored in `LRUNIT(2,n)`, which is initialized to 1 for ASCII format and to 2 for binary format. If a new results file is to be created by the postprocessing program, the file format of the output file is defined similarly via the variable `LOUTF`, which is also initialized to 1 for ASCII format and 2 for binary format. The root file name for both input and output results files is defined through the character variable `FNAME`. The root file name case will be the same as the case in which `FNAME` is defined; Abaqus defines the file extensions to be lowercase letters. See Accessing the Results File Information for a discussion of the naming convention for the file extensions.

The final initialization phase is done internally by calling the Abaqus subroutines `INITPF` and `DBRNU`. The Fortran interfaces are

```
CALL INITPF(FNAME, NRU, LRUNIT, LOUTF)
CALL DBRNU (JUNIT)
```

where the arguments in the call to `INITPF` are as described above, and `JUNIT` is the Fortran unit number connecting the file.

These integer variables must be defined before the subroutines are called.

Data processing

Data manipulation requires knowledge of each data record. Details of these records are found in Results File.

The data organization in the results file uses a sequential format. Each record must, therefore, be retrieved in a sequential manner via a call to `DBFILE` using the interface

```
CALL DBFILE(0, ARRAY, JRCD)
```

This call can be placed inside a `DO`-loop, and the loop count should exceed the number of records stored in the file. Alternatively, `DBFILE` can be called as long as `JRCD` is equal to `0`. The first argument, `0`, indicates that a record is to be read. Each record that is read is stored in the buffer `ARRAY` and returned to the calling program for manipulation. The last argument, `JRCD`, is a return code that is set to `0` unless an end-of-file condition or an incomplete record is processed, in which case `JRCD` is set to `1`.

If it is desirable to extract or modify certain records and save them in a new results file with the same data organization as an Abaqus-generated results file, then the subroutine `DBFILW` should be called with the interface

```
CALL DBFILW(1, ARRAY, JRCD)
```

The new results file will be written with the file extension `.fin`. Refer to Utility Routines for Accessing the Results File.