

EQ Simulator Input Condition File Format

Michael Barall 06/28/2010

Version 0.4

Note: Input condition format version 0.4 is identical to version 0.3.

1. Overview

The condition file specifies the initial conditions for the fault surface. For each element, the file can contain either or both of the following: *initial stress*, and *initial state*. The file format is flexible enough to include initial conditions for both conventional friction laws and rate-state laws, and it can be expanded in the future to include additional initial conditions as needed.

If the file includes initial stress, it contains for each element:

- Initial shear stress.
- Initial normal stress.

If the file includes initial state, it contains for each element:

- Initial state value for rate-state friction.

2. File Format

The input condition file is a container, as described in the EQ Simulator Container Format. The container format lets us store different kinds of records in a single file.

2.1. Overall File Structure

The following table shows the overall structure of an input condition file. Specific kinds of records are described later.

	Part	Description
1	Header	File header that contains the file signature, metadata, and record descriptors, as described in the container file specification.
2	Summary	One line that contains the initial condition summary record. This record gives the total number of elements. It also contains flags that indicate which kinds of data are included in the file: initial stress, and/or initial state.
3	Initial stress	One line that gives the initial stress (record kind 201) for a single element, if the file includes initial stress data.
4	Initial state	One line that gives the initial state (record kind 202) for a single element, if the file includes initial state data.
5		Repeat 3 and 4 for each element.
6	End-of-file	One line that marks the end of the file, as described in the container file specification.

The signature for an input condition file is “EQSim_Input_Condition_2”. The specification level of this document is 1. So, the first line of the file contains the following signature record:

```
101 EQSim_Input_Condition_2 1
```

Refer to the container file specification for an explanation of signature and specification level.

Notice that data records are required to appear in a specific order. After the summary come all the data records for element 1, then all the data records for element 2, then all the data records for element 3, and so on. For each element, the data records must appear in order according to the kind of record (201 or 202). By imposing a specific order, we make it possible for a reader to always know what kind of record to expect next.

2.2. Indexing

Each element (triangle or rectangle) is assigned an index number. Indexes start at 1, and increase consecutively throughout the entire file. These are Fortran-style indexes.

The element index numbers appearing in the condition file match the index numbers in the geometry file.

3. Record Formats

The following table shows the standard kinds of data records for the input condition file.

Kind	Name	Description
200	summary	Initial condition summary record
201	initial_stress	Initial stress record
202	initial_state	Initial state record

These are all data records, which means that each record contains a series of data fields. Each kind of record is explained below.

The names “summary” and so forth must be listed in the descriptor part of the file header.

3.1. Initial Condition Summary Record

```
200 n_element stress_flag state_flag  
comment_text
```

This must be the first data record in the file. It gives the total number of elements, and it indicates which kinds of data are included in the file.

The record contains 3 data fields, described in the following table.

	Name	Type	Description
1	n_element	integer	The total number of elements.
2	stress_flag	integer	A flag which indicates if the file includes initial stress. A value of 1 indicates it is included, a value of 0 indicates it is not included.
3	state_flag	integer	A flag which indicates if the file includes initial state. A value of 1 indicates it is included, a value of 0 indicates it is not included.

The names “n_element” and so forth must be listed in the descriptor part of the file header.

As in any data record, the fields must be separated by one or more blank spaces. The *comment_text* is optional, but if included it must be separated from the last field by one or more blank spaces.

At least one of the two flags must have the value 1.

3.2. Initial Stress Record

```
201 index shear_stress normal_stress
    comment_text
```

This record gives the initial stress acting an element.

The record contains 3 data fields, described in the following table.

	Name	Type	Description
1	index	integer	The index number of the element.
2	shear_stress	real	The initial shear stress of this element, in Pascal.
3	normal_stress	real	The initial normal stress of this element, in Pascal.

The names “index” and so forth must be listed in the descriptor part of the file header.

As in any data record, the fields must be separated by one or more blank spaces. The *comment_text* is optional, but if included it must be separated from the last field by one or more blank spaces.

3.3. Initial State Record

```
202 index rs_theta
    comment_text
```

This record gives the initial state variable for rate-state friction an element

The record contains 2 data fields, described in the following table.

	Name	Type	Description
1	index	integer	The index number of the element.
2	rs_theta	real	The initial value of the rate-state variable theta for this element, in seconds.

The names “index” and so forth must be listed in the descriptor part of the file header.

As in any data record, the fields must be separated by one or more blank spaces. The *comment_text* is optional, but if included it must be separated from the last field by one or more blank spaces.

This record is typically used by codes that implement rate-state friction laws.