EQ Simulator Input Friction File Format

Michael Barall 06/28/2010
Version 0.4

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

1. Overview

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

If the file includes elastic parameters, it contains for the entire model:

- Lame constant, lambda.
- Lame constant, mu.

If the file includes fault strength parameters, it contains for each element:

- Static yield strength.
- Dynamic sliding strength.

If the file includes rate-state parameters, it contains for each element:

- Rate-state parameter A.
- Rate-state parameter B.
- Rate-state characteristic distance L.
- Rate-state friction coefficient f0.
- Rate-state reference velocity V0.

2. File Format

The input friction 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 friction 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 fault friction 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: elastic parameters, fault strength parameters, and/or rate-state parameters.
3	Elastic parameters	One line that gives the elastic parameters (record kind 201) for the entire model, if the file includes elastic parameters.
4	Fault strength parameters	One line that gives the fault strength parameters (record kind 202) for a single element, if the file includes fault strength parameters.
5	Rate-state parameters	One line that gives the rate-state parameters (record kind 203) for a single element, if the file includes rate-state parameters.
6		Repeat 4 and 5 for each element.
7	End-of-file	One line that marks the end of the file, as described in the container file specification.

The signature for an input friction file is "EQSim_Input_Friction_2". The specification level of this document is 1. So, the first line of the file contains the following signature record:

```
101 EQSim_Input_Friction_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 comes the elastic parameters (if present), then 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 (202 or 203). 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 friction 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 friction file.

Kind	Name	Description
200	summary	Fault friction summary record
201	elastic_param	Elastic parameters record
202	fault_strength	Fault strength parameters record
203	rate_state	Rate-state parameters 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. Fault Friction Summary Record

```
200 n_element elastic_flag strength_flag rate_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 4 data fields, described in the following table.

	Name	Type	Description
1	n_element	integer	The total number of elements.
2	elastic_flag	integer	A flag which indicates if the file includes elastic parameters. A value of 1 indicates it is included, a value of 0 indicates it is not included.
3	strength_flag	integer	A flag which indicates if the file includes fault strength parameters. A value of 1 indicates it is included, a value of 0 indicates it is not included.

4	rate_state_flag	integer	A flag which indicates if the file includes rate-state parameters. 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 three flags must have the value 1.

3.2. Elastic Parameters Record

```
201 lame_lambda lame_mu
    comment_text
```

This record gives the elastic parameters for the entire model.

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

	Name	**Type**	**Description**
1	lame_lambda	real	The Lame modulus lambda, in Pascal.
2	lame_mu	real	The Lame modulus mu, also known as the shear modulus, in Pascal.

The names "lame_lambda" 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. Fault Strength Parameters Record

```
202 index static_strength dynamic_strength
    comment_text
```

This record gives information about the fault strength in 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	static_strength	real	The static yield strength of this element, in Pascal. This is the level of shear stress at which the fault begins to slip (assuming that the normal stress is equal to its initial value).
3	dynamic_strength	real	The dynamic sliding strength of this element, in Pascal. This is the level of shear stress during steady-state sliding (assuming that the normal stress is equal to its initial value).

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 conventional friction laws.

Some codes do not compute absolute levels of shear stress, only changes in shear stress from some unspecified initial state. Such codes may use a single value for fault strength, which can be computed as static strength minus dynamic strength.

3.4. Rate-State Parameters Record

```
203 index A B L f0 V0
    comment_text
```

This record gives parameters for rate-state friction in an element.

The record contains 6 data fields, described in the following table.

	Name	Type	Description
1	index	integer	The index number of the element.
2	A	real	Rate-state parameter A for this element.
3	B	real	Rate-state parameter B for this element.

4	L	real	Rate-state characteristic distance L for this element, in meters.
5	f0	real	Rate-state friction coefficient f0 for this element.
6	V0	real	Rate-state reference velocity V0 for this element, in meters/second.

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.