## 1.3 Input File Formats

Finite Element codes usually have input files in sophisticated tekt formats. While simple in principle, the sheer amount of user options and the way these are expressed in the input file varies wildly between different codes. We make the common assumption that the input files consist of sections (sometimes called input cards). These sections always start with a \magic" word, followed by an arbitrary amount of lines containing the input data. The input file is parsed in a procedural way, calling a

### 1.3.3 Data composition

The data compositor method is used to write the data to a model input file. By

Subclass the FETextFile class:

```
    class SuperFEMTextFile(FETextFile):
        type = 'superfem'
```

create an input file descriptor file superfem.fe.

implement the extractor methods (and possibly the compose methods)

The name of the input file descriptor file should be the same as the the one given in the type variable of the SuperFEM class.

## 1.4 Output file formats

Extracting data from output files is much more involved than parsing input files. Depending on the fantasy of the model programmer, these files can be in any form of binary data format you could think of. If you have a description of the output file format (or the code) you're lucky off. Due to this, no general framework can be given. However I am are happy to provide a framework to read Fortran output