IMSIL Manual and Database

Thomas Bernhard

June 2019

Project report for PR 362.156. This document gives insight into the making of the project. For usage and detailed documentation of the code please check the README file and documentation in the GitHub Repository.

1 Introduction

The goal of the project was to generate a SQLite database for the parameters used by the Implant Simulator IMSIL. Currently the parameters are stored as part of the manual written in LaTeX. The parameters should be converted to reStructuredText and checked for validity in the browser.

2 Architecture

The program is split into the following modules:

- manualToDb: This is the main entry point for the application. It contains the global variables and paths.
- parameter: The parameter class represents one parameter with all its attributes including the record where it is stored.
- database: The database module provides functions for generating the tables, writing and reading parameters.

- parsing: This module contains a *parseFile* function which makes the conversion from LaTeX to reStructuredText and returns a list of parameters
- supply: This module collects the parameters from the database and generates a .rst file, which can then be displayed as a part of the documentation.

3 Parsing

After going through the project of another student dealing with a similar kind of task I decided not to use a 3rd-party library for parsing the files and rather rely on python's built in string methods.

3.1 Approach

In the *parseFile* function every LaTeX File containing parameters gets processed. In order to do so there are a few helper functions dealing with specific syntax conversions. For example the `\texttt{}` command has no counterpart in reStrucuredText so there is a function that recursively removes all of those occurrences.
After parsing the file as a whole it is split into a list of parameters and every parameter is split into its attributes. For some attributes there is some special formatting. E.g. dealing with lists and tables in *Default* and *Range* attributes.

3.2 Citations and References

Some of the parameters have references to other parts of the manual or citations. Labels are converted in the following convention:

References:
LaTeX: `\ref{s:sput}`
reStructuredText: `:ref:‘s_sput‘`

The manual has to contain: `.. _s_sput:`

Citations:
LaTeX: `\cite{I9502}`
reStructuredText: `[I9502]_`

The bibliography has to contain: `.. [I9502] Some Book`

As converting the whole manual including the bibliography was not part of this project the references and citations only refer to dummy-labels at the moment. See Chapter 5 for more information.

3.3 Math

As equations stay in LaTeX syntax, the only thing to do was to make sure not to convert the sections enclosed by $ - signs into reStructuredText.

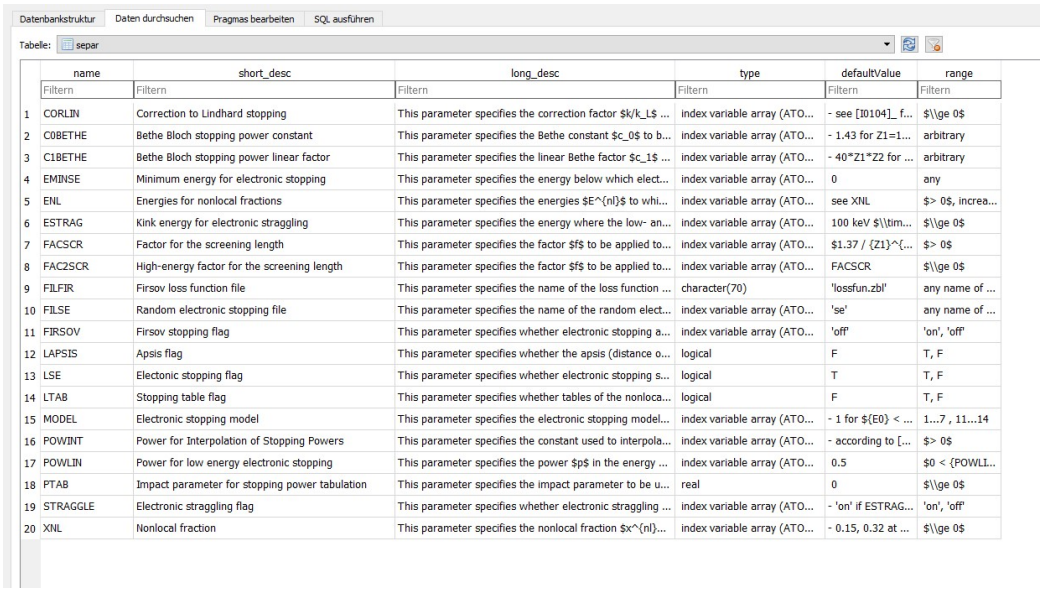
4 Database

The database is structured in tables where each table corresponds to a file containing parameters. A table consist of multiple parameters where each parameter consists of the following attributes:

- `name (str)`: Name of the input parameter (e.g TILT)
- `short_desc (str)`: Short description of the input parameter
- `long_desc (str)`: Long description of the input parameter
- `type (str)`: Type of the input parameter (e.g real)
- `defaultValue (str)`: Default value of the parameter (e.g 0)
- `range (str)`: Range (e.g $0 < TILT < 90$)

The database is generated and edited using the sqlite3 package. See the documentation for the database module for detailed information about how to write/read parameters to/from the database.

Additionally you can edit an existing .db file using the DB Browser for SQLite.



Datenbankstruktur | Daten durchsuchen | Pragmas bearbeiten | SQL ausführen

Tabelle: separ

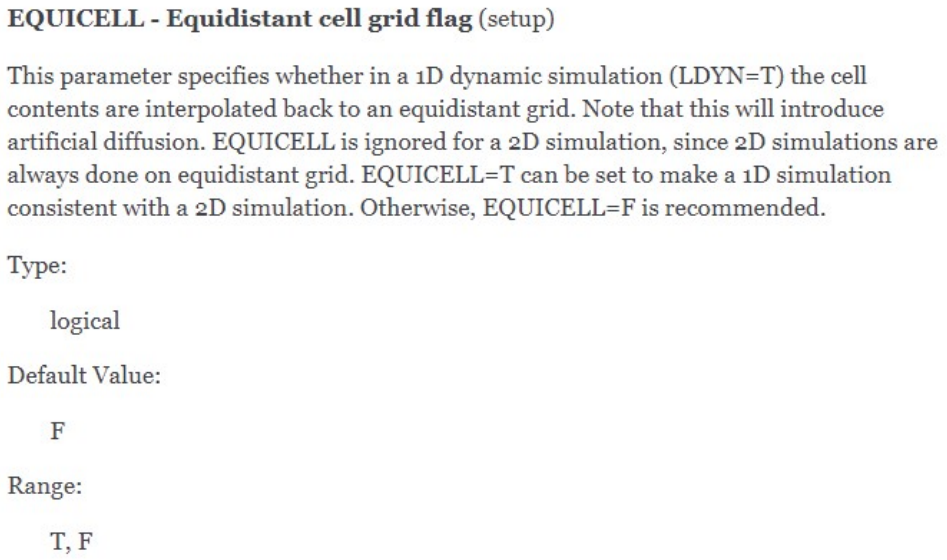
	name	short_desc	long_desc	type	defaultValue	range
	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern
1	CORLIN	Correction to Lindhard stopping	This parameter specifies the correction factor k/k_L ...	index variable array (ATO...	- see [I0104]_ f...	$\\ge 0$
2	C0BETHE	Bethe Bloch stopping power constant	This parameter specifies the Bethe constant c_0 to b...	index variable array (ATO...	- 1.43 for Z1=1...	arbitrary
3	C1BETHE	Bethe Bloch stopping power linear factor	This parameter specifies the linear Bethe factor c_1 ...	index variable array (ATO...	- 40*Z1*Z2 for ...	arbitrary
4	EMINSE	Minimum energy for electronic stopping	This parameter specifies the energy below which elect...	index variable array (ATO...	0	any
5	ENL	Energies for nonlocal fractions	This parameter specifies the energies E^{nl} to whi...	index variable array (ATO...	see XNL	> 0, increa...
6	ESTRAG	Kink energy for electronic straggling	This parameter specifies the energy where the low- an...	index variable array (ATO...	100 keV $\\tim...	$\\ge 0$
7	FACSCR	Factor for the screening length	This parameter specifies the factor f to be applied to...	index variable array (ATO...	$1.37 / {Z1}^{...	> 0
8	FAC2SCR	High-energy factor for the screening length	This parameter specifies the factor f to be applied to...	index variable array (ATO...	FACSCR	$\\ge 0$
9	FILFIR	Firsov loss function file	This parameter specifies the name of the loss function ...	character(70)	'lossfun.zbl'	any name of ...
10	FILSE	Random electronic stopping file	This parameter specifies the name of the random elect...	index variable array (ATO...	'se'	any name of ...
11	FIRSOV	Firsov stopping flag	This parameter specifies whether electronic stopping a...	index variable array (ATO...	'off'	'on', 'off'
12	LAPSIS	Apsis flag	This parameter specifies whether the apsis (distance o...	logical	F	T, F
13	LSE	Electonic stopping flag	This parameter specifies whether electronic stopping s...	logical	T	T, F
14	LTAB	Stopping table flag	This parameter specifies whether tables of the nonloca...	logical	F	T, F
15	MODEL	Electronic stopping model	This parameter specifies the electronic stopping model...	index variable array (ATO...	- 1 for ${E0} < ...	1...7 , 11...14
16	POWINT	Power for Interpolation of Stopping Powers	This parameter specifies the constant used to interpola...	index variable array (ATO...	- according to [...	> 0
17	POWLIN	Power for low energy electronic stopping	This parameter specifies the power p in the energy ...	index variable array (ATO...	0.5	$0 < {POWLI...
18	PTAB	Impact parameter for stopping power tabulation	This parameter specifies the impact parameter to be u...	real	0	$\\ge 0$
19	STRAGGLE	Electronic straggling flag	This parameter specifies whether electronic straggling ...	index variable array (ATO...	- 'on' if ESTRAG...	'on', 'off'
20	XNL	Nonlocal fraction	This parameter specifies the nonlocal fraction $x^{nl}...	index variable array (ATO...	- 0.15, 0.32 at ...	$\\ge 0$

Figure 1: Example of a table displayed in the DB Browser

5 Displaying Parameters

In order to check the parameters for validity you can either use the DB Browser for SQLite or display the parameters in a browser as part of the documentation written with sphinx. To do so, the supply module collects all the parameters from the database and writes them into a single .rst file. After rebuilding the documentation (See README for detailed information) you can inspect the parameters.
As already mentioned above the references and citations point to dummy-labels which are also included in the project documentation. If at some point the whole manual gets converted to reStructuredText the labels have to be

converted as described in Chapter 3.2.
The equations are rendered using MathJax.



EQUICELL - Equidistant cell grid flag (setup)

This parameter specifies whether in a 1D dynamic simulation (LDYN=T) the cell contents are interpolated back to an equidistant grid. Note that this will introduce artificial diffusion. EQUICELL is ignored for a 2D simulation, since 2D simulations are always done on equidistant grid. EQUICELL=T can be set to make a 1D simulation consistent with a 2D simulation. Otherwise, EQUICELL=F is recommended.

Type:

logical

Default Value:

F

Range:

T, F

Figure 2: Example of a parameter displayed in the browser as part of the documentation

6 Conclusion

During the time of developing this program I learned some valuable lessons on how to work with databases, documenting code, python's string methods and reStructuredText. While the part concerning the database was pretty straight forward, parsing and displaying the parameters was quite challenging in spite of it seeming like a trivial task at first. It did not take long to see some results but a lot of work went into the details. Despite putting attention to designing the parsing functions as general as possible I did not cover the whole LaTeX syntax which means that future changes to the manual may cause problems.