Creating awesome .tex tables

table

Marius Hötting
Marius.Hoetting@udo.edu

Matthias Jaeger
Matthias.Jaeger@udo.edu

20. November 2015

TU Dortmund – Fakultät Physik

Inhaltsverzeichnis

1 Einführung **2**

2 Funktionen **3**
2.1 write 3
2.2 make_SI 3
2.3 make_table 3
2.4 make_composed_table 3
2.5 make_full_table 3

3 Beispiele **3**

1 Einführung

Dieses Modul definiert Funktionen, welche die Generierung von *.tex* Dokumenten aus dem internen Python Code heraus ermöglichen sollen. Insbesondere wird ein Augenmerk auf die Generierung von Tabellen gelegt. Grundlage ist die Funktion *make_table*, welche Daten inklusive Fehlergrößen aufnimmt und einen tex-formatierten string zurückgibt. Falls nun mehrere solcher Tabellen zusammengefügt werden sollen, so kann dies über *make_composed_table* geschehen. Sodann ist es zur Zeit vorgesehen, den return Wert dieser beiden Funktionen in eine .tex Datei abzuspeichern. Dies geschieht mit *write*. Es empfiehlt sich, hierfür einen /build Ordner zu verwenden!

Die Funktion *make_full_table* übernimmt dann den Teil, aus der generierten Tabelle – welche momentan aus Zahlenwerten, getrennt durch & -Zeichen, besteht – automatisch eine standardisierte, vollständige .tex Tabelle zu erstellen. Der Rückgabewert dieser Funktion ist vom Typ string und startet stets mit ”
begintable”. Dieser Rückgabewert sollte erneut über *write* in eine .tex File gespeichert werden. Nun ist in den entsprechenden tex Kapiteln nur noch ein ”
inputbuild/filename.tex” einzufügen.

2 Funktionen

2.1 write

2.2 make_SI

2.3 make_table

2.4 make_composed_table

2.5 make_full_table

Call: make_full_table (caption, label, source_table, stacking, units)

caption	[string]	Set the caption of your tabel, displayed usually above the table
label	[string]	Set the label used to reference your table in tex code
source_table	[string]	Specify the path/filename of the previously generated .tex table (generated by *make_table*)
stacking	[list of type int]	Specifiy which columns are error related values. Here you chose the column(s) (starting from 0) that you will finally see at the table head line. See the examples for fast understanding.
units	[list of type string]	Here you put in the descriptions of the head line, ordered in a list according to the columns.

3 Beispiele

First make table:

```
write('build/Tabelle_b.tex', make_table(
    [Wert_b, C2*1e9, R2, R34, Rx, Cx*1e9], [0, 1, 1, 1, 1, 1]))
```

Now take your table and go to make full table. Here you have the first column (0) without any error and the following columns [1,2,3,4,5] with an error connected to them. This will make 11 columns in total, but only 6 are displayed in the table head line.

```
write('build/Tabelle_b_texformat.tex', make_full_table(
    'Messdaten Kapazitätsmessbrücke.',
    'table:b',
    'build/Tabelle_b.tex',
    [1,2,3,4,5],
    ['Wert', '$C_2 \\:/\\: \\si{\\nano\\farad}$', '$R_2 \\:/\\: \\si{\\ohm}$',
    '$R_2 / R_4$', '$R_x \\:/\\: \\si{\\ohm}$',
    '$C_x \\:/\\: \\si{\\nano\\farad}$']))
```