# MathTeX.m

Stellan Östlund
Gothenburg University

March 16, 2014

### Abstract

The package `MathTeX.m` enables `TeX` output from Mathematica.

## General:

- remove references to `layout` and `showframe` to remove visible page geometry
- the package `listings` must be available

## Files:

- `mtexdemo.mtex` which should be run with `$ math < mtexdemo.m`
- `MathTeX.m` which does the hard job and is run as a package.
- `listings` is a directory which includes the listings macros
- `mlistings.tex` is a file which loads som fragile listings commands

## Example:

The file `mtexdemo.mtex` should be run with the command

```
$ math < mtexdemo.mtex
```

It will generate two files, `mtexdemo.tex` and `mtexdemo.m` .

The file `mtexdemo.tex` can be handled with the command `pdflatex mtexdemo` whereas `mtexdemo.m` is suitable as input to the command line interface in mathematica and has all the special `mtex` constructions stripped.

## Usage:

The easiest way to get the hang of it is to look at `mtexdemo.tex` and compare it to tex output, mtexdemo.m and mtexdemo.tex to see what is going on.

This file is itself a demo using the `MathTeX.m` hack. Special escapes are

1. `((((  expression ))))`

   firsts lists 'expression' then executes it resulting in a listing in the document
2. `(*(( expression ))*)`

   lists 'expression" but does not execute it result
3. `(*$   lhs = TeX[ rhs ]  $*)`

   formats rhs and places between $ signs resulting in an inline mathematical expression in the document

4. `(*[   lhs = TeX[ rhs ]  ]*)`

   formats rhs and places between [ ] signs resulting in an equation in the document

5. `(*!   lhs = TeX[ rhs ]  !*)`

   formats rhs and places between quotes so that it is interpreted as a latex expression rather than a math expression.

6. an optional period or comma can precede the closing $.

7. `TeX[struct_]`

   can be defined to override `TeX` command

8. `TeXFormat[ mybeta ] := "\\beta_{my} "`

   specifies formatting

9. `TeXFormat[ hh[a_,b_] ] := \...`

   would special format `hh[x,y]` .

The logic here is that the file should execute fine even if running the math script without TeX; there should be no side effects from `mtex` special constructions since they end up being parsed as comments. Thus only type (1) expressions are actually executed if `TeX.m` is missing.

With `TeX.m` two files are created; a `.tex` file containing the `TeX` code and a `.m` file containing the mathematica code. These should be usable separately. Since debugging is more difficult with the `.mtex` file this can be useful.

Note that if the file `file.mtex` is input *inside* mathematica, the `.mtex` commands are ignored.

The `lhs = TeX[rhs]` construction is somewhat fragile. `lhs` should not be interpreted and `lhs =` should be optional. The `=` sign is fragile. First example (1). Note that the code is executed so that f and g acquire definitions.

Listing 1: *Here a simple bit of code* *mtexdemo.mtex*

```

(* Here a simple bit of code *)
f[x_] := Cos[ x ];
g[x_] := Expand[ ( x - 3)^2 ];
alpha = f[Pi/6 ];
```

Here is a broken out equation, with an optional period at the end.

$$g[f[x]] = \cos^2(x) - 6\cos(x) + 9.$$

Note that the construction `lhs = rhs` does not put through `lhs` through any tex parsing, so that it must be done explicitly

- $(3) alphabeta = \frac{\sqrt{3}\beta}{2}$
- (5) :

```
alpha beta  =\frac{\sqrt{3} \beta }{2}
```

- (4) :

$$\alpha\beta = \frac{\sqrt{3}\beta}{2}$$

Graphics in the commandline interface is a bit fragile. To have graphics output to screen, you must run the command `<< JavaGraphics‘` . However this loads slowly and is not necessary when using the batch command.We now plot a simple figure and insert into the text. Note that FigInsert and FigSave violate the rule that the mathematica file runs with or without TeX, since the functions are defined in TeX `TeX` .

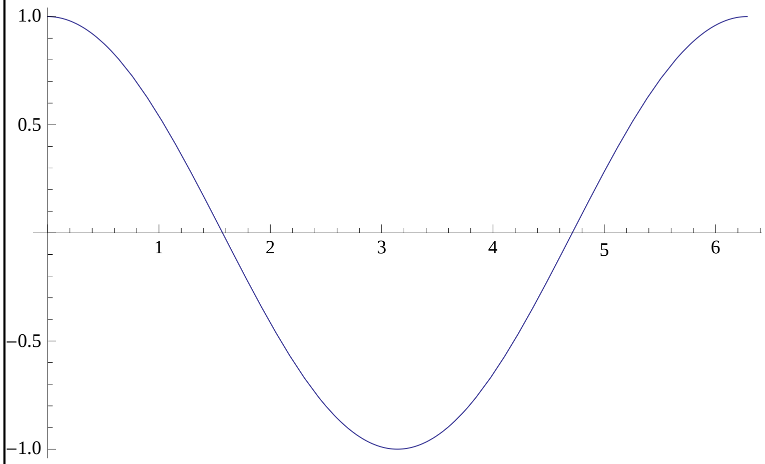


Figure 1: here is the captin

Listing 2: *A small example showing that TeXForm is dumb* *mtexdemo.mtex*

```

(* A small example showing that TeXForm is dumb *)
(* It does not know how to format the variable mybeta  and the function hh *)
sigma2  = Expand[ mybeta PauliMatrix[2]  + hh[pp,qq] PauliMatrix[3 ] ];
```

$$\sigma_2 = \begin{pmatrix} hh[pp,qq] & -imybeta \\ imybeta & -hh[pp,qq] \end{pmatrix}$$

Listing 3: *We format the variable mybeta* *mtexdemo.mtex*

```

(* We format the variable mybeta *)
TeXFormat[mybeta] := "\\beta_{my}";
TeXFormat[ hh[a_, b_] ] := "h_{"<>TeX[a]<>"}^{"<>TeX[b]<>"}";
```

$$\sigma_2 = \begin{pmatrix} h_{pp}^{qq} & -i\beta_{my} \\ i\beta_{my} & -h_{pp}^{qq} \end{pmatrix}$$

You can define yourself how `TeX[ exp ]` handles `exp` . In this file, I simply put in `TeX[ ex_] := ToString[ TeXForm[ ex ]] ;` so that the program sends it to `TeXForm` . You can instead intercept the expression and parse it as you like. For example adding the lines below will intercept the handling of arrays.

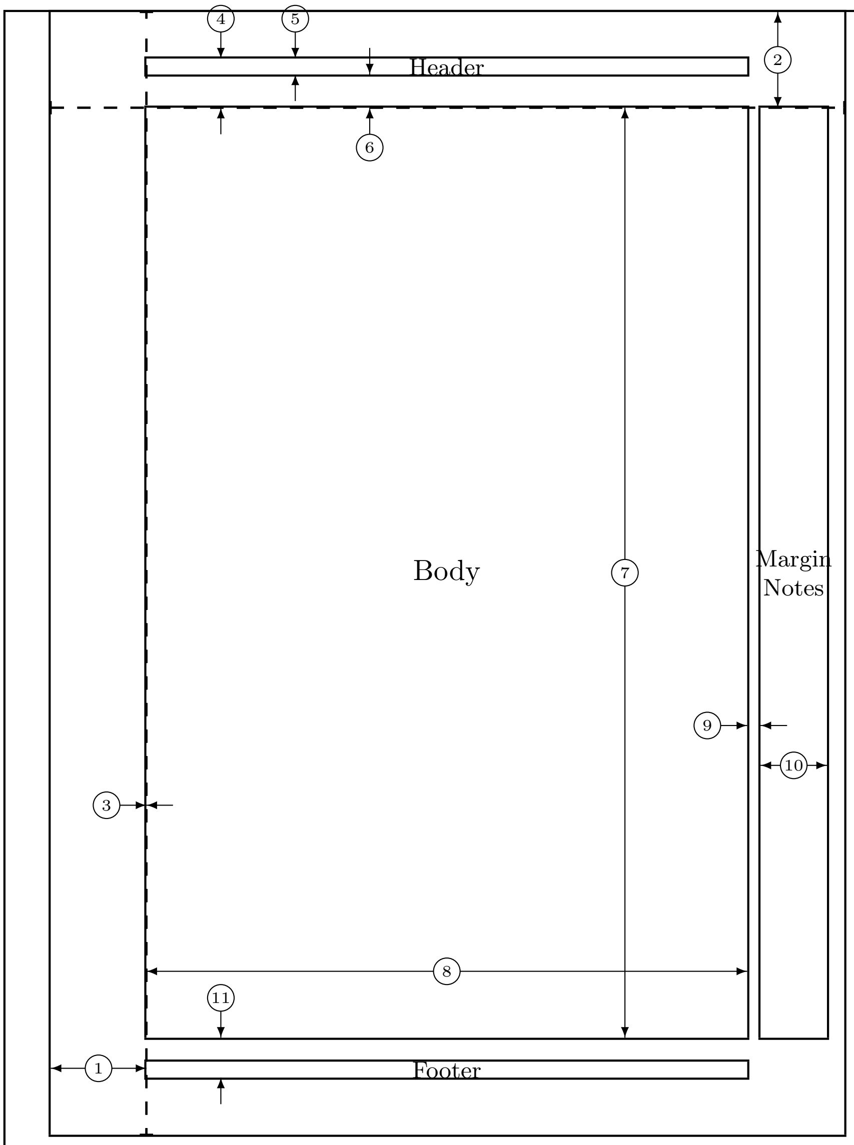
Listing 4: *Override TeXForm for matrices* *mtexdemo.mtex*

```

(* Override TeXForm for matrices *)

 TeX[a_?MatrixQ] := Module[{exp,len,i},
          len = Length[ a[[1]] ];
          exp = "\\left[\\begin{array}{"<>StringJoin[Table["c",{Length[a[[1]] ]
            }]]<>"}\n";
          Do[
          Do[ exp = exp<>ToString[TeX[a[[i,j]]]]<>If[ j < len," & ",""], {j,1,len}];
          If[ i < Length[a], exp = exp<>"\\\\\n"],{i,1,Length[a] }];
          exp = exp<>"\n";
          exp = exp<>"\\end{array}\\right]";
          Return[exp];
          ];
```

Now print this again

$$\sigma_2 = \left[\begin{array}{cc} h_{pp}^{qq} & -i\beta_{my} \\ i\beta_{my} & -h_{pp}^{qq} \end{array}\right]$$

So now the matrix is intercepted and printed with square brackets instead.



| | | | |
|---|---|---|---|
| 1 | `one inch + \hoffset` | 2 | `one inch + \voffset` |
| 3 | `\oddsidemargin = 0pt` | 4 | `\topmargin = -37pt` |
| 5 | `\headheight = 12pt` | 6 | `\headsep = 25pt` |
| 7 | `\textheight = 700pt` | 8 | `\textwidth = 452pt` |
| 9 | `\marginparsep = 10pt` | 10 | `\marginparwidth = 50pt` |
| 11 | `\footskip = 30pt` | | `\marginparpush = 5pt (not shown)` |
| | `\hoffset = 0pt` | | `\voffset = 0pt` |
| | `\paperwidth = 597pt` | | `\paperheight = 845pt` |