GNU Octave/TeXmacs Interface

by Michael Graffam (mgraffam@mathlab.sunysb.edu)

Octave and TeXmacs communicate via a synchronous pipe interface. This allows the Octave console to be presented in a typeset region of a TeXmacs document. To display typeset materials in this TeXmacs region, Octave code can print escape sequences to transfer Scheme, LaTeX, or Postscript materials for display.

This Octave interface is implmeneted a a set of m-files for interacting with TeXmacs via Scheme. As of v1.0.1 TeXmacs includes the required interface code and should automagically detect the presense of Octave. These m-files generate TeXmacs content from Octave and perform related functions:

tmdisp.m Displays an Octave variable "pretty-printed" via the TeXmacs interface. It supports scalar, matrix, structure, list and string types. You must use tmdisp, instead of disp, to get TeXmacs formatted variable output.

mesh.m Modified mesh routine for use with TeXmacs. Also included are modified back-end plot routines using gnuplot to transfer the embedded Postscript to TeXmacs. High level interfaces like plot work as expected.

polyout.m Modified polynomial printer to use a LaTeX formula.

scheme.m Executes a string as a Scheme expression via TeXmacs.

num2scm.m Converts a number (scalar) to a TeXmacs Scheme expression.

mat2scm.m Converts a matrix to a TeXmacs Scheme expression.

str2scm.m Converts a string to a TeXmacs Scheme expression.

struct2bullet.m Converts a structure to a bulleted TeXmacs list.

struct2tree.m Converts a structure to a TeXmacs tree. The leaves of the tree are "switchable" and can be switched to the variable name or content for easy traversal of complicated Octave structures.

struct2scm.m Converts a structure to a TeXmacs Scheme expression by calling either struct2bullet or struct2tree, depending on the user's configuration. The TMSTRUCT global Octave variable is used. If TMSTRUCT = 0 a tree is used, else a bulleted list is used.

list2scm.m Converts a list to a TeXmacs Scheme expression.

obj2scm.m Front end interface to the other converters to convert an arbitrary Octave variable to a TeXmacs Scheme expression. In most cases, user programs should use this function.

The appearance of the TeXmacs output, namely the colors used, can be customized by setting the appropriate Octave global variables. These global variables should be defined in your ~/.octaverc or in the system-wide octaverc:

```
global TMSTRUCT=0; ## Use tree output for structures.

global TMCOLORS=["black"; "red"; "magenta"; "orange"; "green"; "blue";];

global TMCOLIDX=6; ## number of colors
```

The TMCOLORS array lists the available colors, in order, to be used by struct2bullet.m it will change to each color in turn as it indents the bulleted list. Additionally, the first color in the list is used as the color for the leaves of struct2tree.m output.

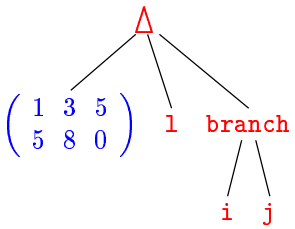
As an example of using these functions and of their behavior, consider the following sample Octave session:

```
GNU Octave, version 2.1.36 (i686-pc-linux-gnu).
Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002 John W. Eaton.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTIBILITY or
FITNESS FOR A PARTICULAR PURPOSE.

Report bugs to <bug-octave@bevo.che.wisc.edu>.
```

octave> $M.\text{mat} = \begin{pmatrix} 1 & 3 & 5 \\ 5 & 8 & 0 \end{pmatrix}$; $M.l = \text{list}([1,2,3], 2, 3)$; $M.\text{branch}.i = 3$; $M.\text{branch}.j = 4$;

octave> tmdisp(M); ## M.mat has been "switched"



octave> TMSTRUCT = 1;

octave> tmdisp(M)

→ mat = $\begin{pmatrix} 1 & 3 & 5 \\ 5 & 8 & 0 \end{pmatrix}$

→ l = 1. (1 2 3)
2. 2
3. 3

→ branch =
- i = 3
- j = 4

octave> tmpolyout([4, 5, 6])

$4 \cdot s^2 + 5 \cdot s^1 + 6$

```
octave> X=0:.1:3;
octave> Y=sin(X);
octave> plot(X,Y);
```