

Programming an OCTAVE Higher (About the GNU Octave Matlab Interpreter)

Christian Himpe (christian.himpe@uni-muenster.de)

WWU Münster
Institute for Computational and Applied Mathematics

Software Tools für die Numerische Mathematik
04.11.2015

What is Octave?

- A calculator,
- a ~~Matlab~~ clone Matlab¹ **interpreter** (runtime),
- a high-level programming language,
- based around a matrix data type,
- for numerical computations,
- and scientific computing,
- under an open-source license,
- for cross-platform use.

octave.org

or in a Linux repository near you.

¹MatLab is short for MATrix LABoratory

Why Use the Matlab Programming Language?

- Reproducibility through expressiveness / accessibility
- Vectorization utilizes BLAS / LAPACK backends
- Proven code stability (> 20 years)
- Widespread²
- Fast prototyping

² Matlab is in the top 20 programming languages in the TIOBE index:
tiobe.com/index.php/content/paperinfo/tpci/MATLAB.html

(High-Level) Alternatives

- FreeMat, Barely alive Matlab interpreter
- SciLab, Deliberately less compatible Matlab interpreter
- NumPy & SciPy, Python's³ answer to Matlab
- R, an S interpreter similar to the Matlab language
- Julia, the new contender

³See Stephan's talk.

Why not use Mathworks Matlab?

Octave is Open Source!
(it's free, too!)

Version 4.0

- OpenMP is activated by default.
- OpenGL is now the default plot backend (instead of gnuplot)
- Octave now has a GUI (octave-cli opens the terminal)
- Better Matlab compatibility (i.e. sylvester deprecates syl)
- Many bugfixes and performance improvements

More Info: www.gnu.org/software/octave/NEWS-4.0.html

Packages (“Official” Toolboxes)

- control
- fem-fenics
- fuzzy-logic-toolkit
- general
- image
- linear-algebra
- miscellaneous
- mpi
- netcdf
- nurbs
- odepkg⁴
- optim
- parallel
- signal
- specfun
- statistics
- symbolic
- ...

more at octave-forge: octave.sourceforge.net/packages.php
Install from Octave terminal: `pkg install -forge packagename`

⁴Will be included into Octave 4.2 and provides ode45 etc.

Assignment Operators

Octave has assignment operators as in C, C++:

■ $+=$

■ $-=$

■ $*=$

■ $/=$

Automatic Broadcasting

Similar to NumPy / SciPy, Octave can:

```
A = rand(5,20);  
b = rand(5,1);  
C = A - b; % Matlab throws an error
```

In Matlab you do such operations with:

```
C = bsxfun(@minus,A,b);
```

which for compatibility Octave has, too.

Default Arguments

Octave allows default arguments:

```
function y = myfunc(a,b=1.0) % Matlab throws an error
```

In Matlab you use:

```
if( (nargin<2) || isempty(b) ), b = 1.0; end;
```

Comments

Octave can use # to introduce comments,
thus you can do:

```
#!/usr/bin/octave-cli
```

Additionally to the block comment #{ ... #}
there are also section comments ##.

Matlab only allows %, %{ ... %} and %%.

Extra Functionality

Octave has:

- the mathematical vectorization as a function `vec`
(Matlab has only `(:)`)
- the C-style not-equal-operator `!=`
(Matlab has only `~=`)
- the double-quote string delimiters `“mystring”`
(Matlab has only `'mystring'`)
- the double-star-operator for exponentiation `**`
(Matlab has only `^`)
- specific end-commands like `endfor`, `endif`, ...
(Matlab has only `end`)
- `linspace` and `logspace` accept vector-valued arguments
(Matlab only scalars)

Missing Functions

I miss:

- Static Code Analysis `mlint`
- McCabe Complexity `mlint('myfunc.m', '-cyc')`
- BLAS info `version('-blas')`
- LAPACK info `version('-lapack')`
- ~~`mtic`~~ ... ~~`mtoc`~~ (Deprecated by Matlab)

Overall, there are missing a few more, see:

http://hg.savannah.gnu.org/hgweb/octave/file/3ccc2d02e64b/scripts/help/__unimplemented__.m

Other Differences

- Octave has a profiler, but it works a little different.
- Octave can directly plot `.pdf` and `.svg`
- Octave's default ODE solver `lsode` swaps all arguments and return values for `t,y`
- Instead of `.mex`, Octave has `.oct` files

Startup

Octave does not have a `startup.m` but an `.octaverc`
Typical entries are:

```
page_screen_output(0);  
page_output_immediately(1);  
graphics_toolkit("gnuplot");
```

Detecting Octave

To detect if you are in Octave and not Matlab, use:

```
exist('OCTAVE_VERSION','builtin')
```


Headless Octave

Start Octave and execute function:

```
octave-cli --eval progname
```

Start Octave and execute function as background process:

```
nohup octave-cli --eval progname &
```

Set SMP cores⁵ to be used by Octave:

```
taskset -c 0,2,4,6 octave-cli
```

Record peak memory usage of Octave:

```
/usr/bin/time -f "%M KB" octave-cli
```

Alltogether:

```
nohup /usr/bin/time -f "%M KB" taskset -c 0,2,4,6 octave-cli --eval progname &
```

⁵Use `numactl` for NUMA machines (Thanks Andi!)

More Toolboxes

- MPITB - MPI Toolbox for Octave
www.ugr.es/~jfernand/mpitb.html
- BCT - Brain Connectivity Toolbox
www.brain-connectivity-toolbox.net
- OpenCL Toolbox
code.google.com/p/opencl-toolbox
- emgr - Empirical Gramian Framework (← that's mine!)
gramian.de

and many more ...

Compiling Octave Yourself

How to do it:

- 1 Install Octave from repositories
- 2 Download and extract source package
- 3 `./configure` # Read output (`config.log`) carefully!
- 4 `make`
- 5 `make check`
- 6 `make install`

Link to your Octave build:

- FlexiBLAS⁶, runtime exchangeable BLAS and LAPACK backends,
see: www.mpi-magdeburg.mpg.de/projects/flexiblas
- `tcmalloc_minimal`, part of `gperftools`,
see: goog-perftools.sourceforge.net/doc/tcmalloc.html

⁶ FlexiBLAS Version 1.3 is about to be released!

Summary:

- Octave is cool!
- More intuitive for programmers
- Well suited for **Open Science**

Remember:

- Always **INDENT** ! (Imagine writing Python code.)
- Always **VECTORIZE** ! (Octave is not slow!)
- Never use the **JET COLORMAP** ! (Try antijet: git.io/antijet)

More info:

- wiki.octave.org/FAQ
- www.ices.utexas.edu/sysdocs/Octave-Matlab
- git.io/mtips

Did you know ...

- Octave is named after Prof. Octave Levenspiel?
- there is an Octave Android app?
`play.google.com/store/apps/details?id=com.octave`
- Octave's first release was in 1988 only 4 years after Matlab?
- has now an official Windows installer?
- has syntax highlighting in emacs, nano and vim?
- you can do locally:

```
system(['notify-send "', mfilename, ':\ I am done!"]);
```

or remotely:

```
system('mutt -s "I am done!" me@host.tld -a result.svg < nohup.out');
```