



MAX PLANCK INSTITUTE  
FOR DYNAMICS OF COMPLEX  
TECHNICAL SYSTEMS  
MAGDEBURG



COMPUTATIONAL METHODS IN  
SYSTEMS AND CONTROL THEORY

# Best Practices in Octave (for Model Reduction)

C. Himpe

2017-05-03

CSC Reading Group



- GNU **Octave** is a Matlab interpreter.
- Matlab is a high-level, array-based programming language.
- **Octave** is largely compatible with Mathworks MATLAB.
- **Octave** is open source and free, get it here:  
`http://octave.org`
- Octave-Forge provides many (MATLAB) toolboxes:  
`http://octave.sourceforge.io`
- **Octave** has advanced syntax and additional functionality.
- **Octave** is cross-platform: Linux (x86 & ARM), Windows, macOS.
- **Octave** has even an Android App.
- **Octave** has syntax highlighting in nano, vim and emacs.
- **Octave** is named after Prof. Octave Levenspiel.



- Language Best Practices
- Benchmarking Best Practices
- Pure Octave
- Pure MATLAB
- Don't!
- About MOR

## ■ **Pre-Allocation**

Always initialize arrays, treat array dimensions as immutable.

## ■ **Automatic Broadcasting / Implicit Expansion**

Operators  $+$ ,  $-$ ,  $\cdot$ ,  $*$  and others can be used row or column-wise. Available in Octave since version 3.6, in MATLAB since 2016b.

## ■ **Copy-on-Write**

Function arguments are only copied if changed, otherwise only referenced, treat array arguments as constant.

## ■ **In-Place Operations**

Operations can be performed in-place if argument and return value and target refer to the same array of same size. This is an exception to CoW.

## ■ **Currying**

Wrap functions in single argument lambdas.

## ■ **Closures**

Return a lambda function handle.

## ■ Reproducible randomness

Seed the random number generator via `rand('seed',seed);`.

## ■ Wall time

Measure wall-time via `th = tic; ... wt = toc(th);`.

## ■ Headless Computation

Run Matlab code remotely, i.e. over ssh.

MATLAB: `nohup matlab20XX -nodisplay -r progname < /dev/null &`

Octave: `nohup octave-cli --eval "progname(args)" > my.log &`

## ■ Maximum memory usage

Log the peak memory usage of Matlab code:

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

## ■ CPU Affinity

Pin Matlab interpreter to specific cores or sockets / memory banks.

OpenMP: `export OMP_PROC_BIND=true`

Multi-Core (UMA): `taskset -c 0,2 octave-cli`

Multi-Socket (NUMA): `numactl -N 0 -m 0 octave-cli`

## ■ Reporting

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

Email: `system('mutt -s "Done!" me@host.tld -a result.svg < my.log');`

## ■ FlexiBLAS

A BLAS and LAPACK wrapper library with runtime exchangeable back ends. FlexiBLAS **Version 2.0** has just been released:

<http://www.mpi-magdeburg.mpg.de/projects/flexiblas>

Use it with MATLAB:

```
LD_PRELOAD="/path/to/libflexiblas.so"; matlab2016b -nodisplay
```

## ■ By default, do not use a GUI

Terminal-based work promotes **replicability** as every step is part of the command history.

## ■ Process and System Monitoring

htop is an interactive process viewer and task manager.

## ■ Use a developer font (IOIO)

For example, hack is pretty good: <http://sourcefoundry.org/hack>



## ■ Don't be a jerk

`clear` all clears all variables, including those in the global workspace.

## ■ The awful `jet` colormap

The worst colormap in the history of mankind: non-monotonic gray-scale conversion, unsuitable for colorblind, perceptually non-uniform, bad contrast. Use `viridis` (default in Octave $\geq$ 4.2), for MATLAB: <http://bit.ly/2oIAG1K>

## ■ Graphical progress bar

The `waitbar` itself is a performance hog. It will distort your timings. But you are not using a GUI anyway, right?

## ■ Indent size: zero

Indent your code correctly and consistently. **Readability** of source code is paramount in science. Note: a function declaration induces a nesting level.

## ■ Plot in pixel formats

Use `.eps` or `.pdf` for document embedded plots (crop with `pdfcrop`), use `.svg` for web graphics.



## ■ Assignment Operations

Octave has the C assignment operators `+=`, `-=` and `*=`.

## ■ Default Arguments

Octave has default arguments via function `myfun(a=1,b=1,c=2,d=3,e=5)`.

## ■ Hash comments

Octave allows comments to be started by a hash symbol `#`. This allows bash executable octave scripts via the shebang `#!/usr/bin/octave-cli`.

## ■ Mathematical vectorization

Octave has a built-in `vec = @(m) m(:);` function.

## ■ Operator Aliases

Octave has an inequality operator `!=` and exponentiation operator `**`.

## ■ String Delimiters

Octave has double quote string delimiters `"..."`.

## ■ Specific endings

Octave has specific end statements such as `endif`, `endfor`, `endfunction`.

## ■ Vector-valued ranges

Octave allows vector arguments for `linspace` and `logspace`, MATLAB does not.

## ■ Testing if Octave runs

use: `if(exist('OCTAVE_VERSION','builtin'))`.



- **Memory profiling**

via `profile -memory`

- **BLAS and LAPACK versions**

Back end info via `version('-blas')` and `version('-lapack')`.

- **Static code analysis**

Linting for MATLAB m-files via `mlint('mfile.m')`.

- **McCabe Complexity**

Cyclomatic number of the control flow graph via  
`mlint('mfile.m', '-cyc')`.



## ■ **Compartmentalization**

Separate computation and visualization; save projections.

## ■ **Orthogonality**

svds sometimes returns non-orthogonal singular vectors; for an additional orthogonalization use `qr(U,0)`, not `orth`.

## ■ **HAPOD**

The Hierarchical Approximate Proper Orthogonal Decomposition:  
<http://arxiv.org/pdf/1607.05210> (morHimLR16)

## ■ **Discrete time-domain norms**

Fixed time step Lebesgue norms: `lnorm = @(m,1) norm(m(:),1);`

## ■ **Procedural test system matrix**

A Lehmer matrix is positive definite: `A = -gallery('lehmer',N);`

## ■ **Balanced gains**

Use scaled HSVs  $d_i := |\tilde{b}_i \tilde{c}_i| \sigma_i$  via:

```
[~,gi] = sort(abs(sum((V'*B).*(C*U)',2)).*diag(D),'descend');
```

```
Ug = U(:,gi); Vg = V(:,gi);
```



# Matlab MOR Toolboxes

- **drtoolbox** – Matlab Toolbox for Dimensionality Reduction
- **emgr** – EMpirical GRamian framework for model reduction of input-output systems
- **KerMor** – Model order reduction for nonlinear dynamical systems and nonlinear approximation
- **MatMOREMBS** – Matlab Model Order Reduction of Elastic Multibody Systems
- **M.E.S.S.** – Matrix Equation Sparse Solver
- **MORE** – a MOdel REduction Toolbox
- **MORLAB** – Model Order Reduction Laboratory
- **MORPACK** – Model Order Reduction PACKAge
- **Nonlinear Model Reduction Routines for MATLAB**
- **RBMATLAB** – a MATLAB library for model order reduction with Reduced Basis Methods
- **SiMpLify** – Structured Model reduction Toolbox for MATLAB
- **SLICOT** – Model and Controller Reduction Toolbox
- **sssMOR** – Sparse State-Space and Model Order Reduction Toolbox
- **SUMO** – SUrrogate MOdeling Toolbox
- **MRedTool** – a MATLAB toolbox for model reduction of multi-dimensional systems
- **SMORES** – A Matlab tool for Simulation and Model Order Reduction of Electrical Systems



## Links:

- <http://wiki.octave.org>
- <http://github.com/mikecroucher/awesome-MATLAB>
- <http://git.io/mtips>

## Books:

- R.K. Johnson. **The Elements of MATLAB Style**. Cambridge University Press, 2011. ISBN 9780521732581
- L. Fortuna and M. Frasca. **Optimal and Robust Control: Advanced Topics with MATLAB**. CRC Press, 2012. ISBN 9781466501911
- Y.M. Altman. **Accelerating MATLAB Performance: 1001 tips to speed up MATLAB programs**. CRC Press, 2015. ISBN 9781482211306

“Programs are meant to be read by humans and only incidentally for computers to execute.”

– Donald E. Knuth