EEE4120F

High Performance Embedded Systems

REFERENCE GUIDE

Octave Setup & Basics

Course Team

Course Convener:	Associate Professor Simon Winberg
Teaching Assistant:	Travimadox Webb

University of Cape Town 2026

Octave Setup & Basics

Companion Guide for Practical 1

Who is this guide for?

This document covers the installation and basic operation of GNU Octave. If you are already comfortable with Linux terminals and MATLAB scripting, you may skip this guide and proceed directly to **Practical 1**.

1 Installation

You may use either **GNU Octave** (Open Source) or **MATLAB** (Commercial). We recommend you use MATLAB as you are familiar with it from other courses.

- **Octave:**
  - **Windows:** Download the installer from the official GNU website. When prompted, ensure you install the `octave-forge` collection (specifically the `image` and `signal` packages).
  - **Linux:** Install via terminal or alternatively check official GNU website

Terminal(Debian/Ubuntu)

```
sudo apt-add-repository ppa:octave/stablesudo apt-add-repository
ppa:octave/stable
sudo apt-get update
sudo apt-get install octave octave-image octave-signal
```

- **MATLAB:** Available via university site license. Ensure the ”Image Processing Toolbox” is installed.

2 File Management (Cheat Sheet)

Octave operates within a directory structure. You must navigate to your folder before running scripts.

Command	**Description**
`pwd`	**P**rint **W**orking **D**irectory. Shows where you currently are.
`ls` (or `dir`)	Lists all files in the current folder.
`cd foldername`	**C**hange **D**irectory. Enter a specific folder.
`cd ..`	Go back (up) one folder level.

3 Creating & Running Scripts

Do not type long programs directly into the command window. Use **Script (.m)** files.

1. **Create:** Type `edit myscript.m` in the command window.
2. **Write:** Enter your code in the editor window.
3. **Run:** In the command window, type the filename *without* the extension:

Octave Command Window

```
>> myscript
```

4 Essential Commands

- `clc` : Clears the command window screen (does not delete variables).
- `clear` : Deletes all variables from memory (use this at the start of scripts).
- `help function_name` : Displays documentation (e.g., `help tic`).
- `whos` : Lists all variables currently in memory and their sizes.

5 Code Example: Benchmarking

For Practical 1, you need to measure execution time. Use the `tic` and `toc` pattern:

```
% Start the timer
tic;

% --- Your Heavy Code Goes Here ---
A = rand(1000, 1000);
B = A * A';
% ---------------------------------

% Stop the timer and store the result
elapsedTime = toc();

% Display the result
disp(['Execution took ', num2str(elapsedTime), ' seconds.']);
```