

# MATLAB

For research, classwork,  
and publications

---

# Introduction

- Easy to use tool for:
  - numeric computation
  - advanced graphics & visualization
  - high-level programming language
- On csehelp administered suns:
  - /opt/matlab/bin/matlab
- Student version ~\$100 at bookstore
  - 1 package -- both windows and linux

# Outline

- Basics of MATLAB
- Loading data and saving results
- Matrices and computation
- Graphing
- Exporting your results
- Final comments

# MATLAB Components

- Variables: C-like, automatically-allocated, overloaded functions simplify handling
- Language: high-level language, more intuitive than C constructs
- Programming:
  - Interactive command line interface
  - Script a set of commands in a .m file
  - Encapsulate a function in a .m file
- Graphics windows

# Getting around

- Accessing help:
  - ‘help’, ‘helpwin’ (GUI), ‘lookfor’ (search)
- Workspace information:
  - ‘whos’: list current variables
  - ‘what’: list MATLAB-specific files in cwd
  - ‘which’: locate files

# Getting around

- Understanding functions:
  - ‘type functionName’: prints out the code for the function
  - Works w/ most MATLAB functions
  - Learn from available code (or remember your math)
- OS commands:
  - ‘cd’, ‘copyfile’, ‘getenv’
  - ‘!’ (execute OS command in shell)

# Interacting with MATLAB

- The desktop
  - Easy to learn, comprehensive GUI environment
  - Slow rendering in Linux, Sun .... VERY slow over a network
- Command line only
  - More difficult to learn
  - No waiting for rendering, less 'mouse' work

# Demo: basics

---

# Loading data

- 'load myData.txt'
  - Use with plain text file with a full matrix
  - Comments begin with '%'
  - Automatically stored in variable 'myData'
- Text: 'fopen', 'fgetl', 'fscanf', 'input' (user input)
- HDF: NCSA's Hierarchical Data Format
- Others: 'imread' (images), 'wavread' (audio), 'aviread' (movie)

# Saving results

- Symmetric functions to reading
- 'save myWork.dat'
  - Save your MATLAB workspace (half-finished work, new results) in a binary file
  - Restore work with 'load myWork.dat'

# Matrices

- Syntax:
  - ‘[]’: Brackets define a matrix
  - ‘,’: commas separate columns (== space)
  - ‘;’: semicolons separate rows
  - X’: transpose of x
  - ‘:’: full range (i.e. all columns)
  - ‘1:5’: first 5 elements

# Example data set

- Measured iteration times for 100 iterations of Jacobi
- Mostly standard results, 1 extreme outlier
- File contains
  - A comment at top (starts with ‘%’)
  - Column 1: iteration #
  - Column 2: iteration time (sec)

# Demo: matrices, loading, & saving

---

# Demo: graphing

---

# Exporting to tables

- ‘save -ascii myFile.txt myMatrix’
  - Saves as plain text -- load into word, excel directly
- ‘writeTexTable(matrix, fileName, fmt)’
  - Writes a TeX formatted table
  - Ready for use in a LaTeX: `\input{...}`
  - ‘fmt’ describes the row format
  - Not MATLAB standard ... written by Otto Sievert

# Exporting to figures

- ‘print -dpsc2 myFig.ps’
  - Prints current figure window as level 2 postscript file
  - Ready for use in LaTeX: `\epsfig{...}`
  - Other possibilities: `-dpng`, `-dtiff`, `-djpeg`
  - See with ‘!gv myFig.ps’
- Output jpegs -> `mpeg_encode` -> portable movie

# Demo: exporting++

---

# Final comments

- Data analysis & visualization
- Figures and tables for publication
- Rapid prototyping
- Learning about and experimenting with techniques you don't know well

# Slides and demos online:

- [www.cs.ucsd.edu/groups/csegsa/enrichment.html](http://www.cs.ucsd.edu/groups/csegsa/enrichment.html)
- Questions & help: [hdail@cs.ucsd.edu](mailto:hdail@cs.ucsd.edu)
- Questions?