DGP Fall 2016 – Lab 0

Lab Objectives

- Learn how to write basic MATLAB code.
- Learn how to use the "Eigen" C++ linear algebra library.
- Learn how to build and run OpenGP projects.

Please complete the tasks below and show them to your TA to get your lab completion mark.

Part 1: MATLAB

MATLAB is a useful development environment for prototyping algorithms. You can use it on the lab machines, or you can download it on your own machines for free (https://matlab.engr.uvic.ca/).

Idiomatic MATLAB in a Nutshell

In MATLAB, operations can be done on every element of a matrix in parallel. This allows you to write clean and efficient code. The following exercises will teach you how to write such code.

1. Create a 1 by 10 matrix using the "zeros" function.
2. Add the two 1D matrices [1,2,3,4,5] and [6,7,8,9,10] to get the result [7,9,11,13,15].
    a. First, do it by writing a for-loop that iterates over the elements and adds the results.
    b. Do it again, using MATLAB's built-in matrix "+" operator to do the addition in parallel.
3. Extract all negative numbers in the matrix [1, -2, 3, 4, -5] and get the result [-2, -5]
    a. Hint: Use MATLAB's built-in "find" function, and use value-based indexing*.
4. Extract the first 5 elements of the matrix [1,2,3,4,5,6,7,8,9,10].
    a. Hint: Use value-based indexing* load using a range of integers as an index.

* see: http://www.mathworks.com/help/matlab/math/matrix-indexing.html

Visualization in MATLAB

MATLAB has built-in capabilities for visualizing data, which helps with designing and debugging algorithms. The following exercises will teach you how to use these visualization capabilities.

1. Plot the function y = sin(x).
2. Plot the 4 points with the following coordinates: x = [1,2,3,4], y = [2,3,2,1]

3. Render a histogram of the following data: [1,2,2,5,1,2,5,7,1,3,5,1]

Hint: Search the MATLAB online documentation (Google: “matlab how to X”)

Linear Algebra in MATLAB

MATLAB has built-in functions for many useful linear algebra functions we’ll use in the DGP course. MATLAB uses highly optimized linear algebra libraries (such as LAPACK) to implement these operations, so using these built-in functions is generally good practice. The following exercises will teach you how to use some of these matrix operations.

1. Multiply the two matrices [1,2,3;4,5,6;7,8,9] and [9,8,7;6,5,4;3,2,1]
2. Invert the matrix [2,0,0;0,2,0;0,0,1]
3. Transpose the matrix [1,2,3;4,5,6;7,8,9]
4. Solve the linear system Ax=B with A = [1,1,1;0.5,0.5,0;1,0.5,1] and B = [1,2,3]’
    a. Hint: Use the “\” operator.

Image Processing in MATLAB

MATLAB is a popular tool in the computer vision industry, so naturally it has image processing functions. If you’re a graphics programmer, you might also find these functions useful for prototyping. The following exercises will teach you how to use some of these image functions.

1. Load an image using imread
2. Display the image using imshow
3. Blur the image using imgaussfilt and display it again.

Part 2: Eigen

Eigen is a C++ library for linear algebra. It’s designed for tasks like solving large, sparse, linear systems. For comparison, this is more like what MATLAB does and less like what GLSL does.

Please study the following pages to get a feel for Eigen:

http://eigen.tuxfamily.org/dox/GettingStarted.html

http://eigen.tuxfamily.org/dox/group__TutorialMatrixClass.html

http://eigen.tuxfamily.org/dox/group__TutorialMatrixArithmetic.html

Please answer the following questions:

- What is the difference between a fixed-size and dynamic array?

    - When should you use which one?
- What is the difference between a dense and a sparse array?
    - Explain the difference in how they are stored
- Eigen is designed using expression templates. Explain how that affects the code you write.

Part 3: OpenGP

OpenGP is a framework for implementing DGP algorithms and displaying the results, maintained by Dr. Tagliasacchi. The assignments and labs in this course will use OpenGP, usually in the style of filling in some skeleton code.

As a warmup to get familiar with getting OpenGP set up, please download OpenGP from here:

https://github.com/OpenGP/OpenGP

Build OpenGP by following the instructions in the README. Make sure it works by running some of the example programs.

Note: For the homework assignments, OpenGP will be supplied inside the GitHub “ataiya/dgp” course repository. You won’t have to download it from GitHub “OpenGP/OpenGP”.

Please answer the following question, based on the sample program “examples/surface_mesh_iterators.cpp”:

- The surface_mesh_iterators.cpp program uses a vertex circulator to visit then neighbouring vertices of a vertex in the half-edge mesh data structure. Please explain the algorithm the vertex circulator uses internally to implement this traversal.