Problem Set 1
ISE 407 – Computational Methods in Optimization
Due: September 10, 2013
Dr. Ralphs

This first assignment is to get you familiar with and using C, Python, and Matlab. Completing this assignment will require some research and some familiarization with the programming environments we’ll use during the rest of the semester. We will learn in more detail how to do the kinds of testing and analysis asked for in this assignment throughout the course. The objective of this assignment is simply to get familiar with the tools and take a first try at doing some computational experiments.

1. Install the following software.

(a) Install Python on your preferred platform. Note that Python comes standard with Linux and OSX. On OSX, however, it’s probably better to install Python yourself. I use the version from Macports because Macports also provides easy installation of a number of popular packages (including Pygame, which we may use during the semester). However, you can also install the official version of Python for OSX from the Python Web site or install a version from another package manager, such as Homebrew or Fink. On Windows, simply install the official version using the standard installer.

(b) Install gcc (and gdb) on your preferred platform. Again, gcc comes standard on Linux. On OSX, it comes with XCode, which is a free (though very large) download. Alternatively, you can install gcc from Macports, which has the advantage that it will be a much more recent version and can be upgraded easily. Macports also provides gfortran, which XCode does not. Again, Homebrew and Fink should also provide solutions. On Windows, install either CYGWIN or MINGW.

(c) Install Eclipse with PyDev and the C Development Tools on your preferred platform. You can also use another IDE if you like.

2. Based on the source code available on the Web site, write functions for multiplying two matrices of a given size in C, Python, and Matlab according to the following guidelines:

- Your functions should use the following API:

  C: 
  ```c
  int ** matmult (int ** A, int * dimA, int ** B, int * dimB)
  ```

  Python: 
  ```python
  def matmult(A, B)
  ```

  Matlab: 
  ```matlab
  function matmult()
  ```

  In all three cases, the return value should be the result of multiplying the two input matrices. For the C function(s), `dimA` and `dimB` should be arrays containing the dimensions of `A` and `B`, respectively. Your function should check to make sure that the dimensions of the matrices and conformable.

- For your Python implementation, you should code a “naive” implementation in addition to the one we saw in class.
• For your C implementation, you should code the various versions we saw in class.

3. Write a test program in each language that generates random matrices in some “reasonable” way. You’ll have to research how to generate random numbers on each platform. Use your wrapper programs to compare the speeds of your implementations on one or more graphs. The graphs should show the effect of the size of the matrices somehow and should show how the implementations compare to each other. It is up to you how to do this. One idea would be to use performance profiles, which we will look at more closely later in the semester.

4. Compare and contrast the three languages in light of your results and explain the outcomes of the experiments in as much depth as possible.