Ay190 – Worksheet 9

John Pharo

Date: February 10, 2014

Scooby Gang: David Vartanyan, Cutter Coryell

Problem 1

We have five linear systems of varying sizes. In order to solve the systems, we first need to check a couple of things. First, A needs to be a square matrix whose dimensions are the same as the length of b. Then we need to make sure that A is invertible, which is the same as checking that det(A) != 0. The script read.py checks the sizes (they're fine) and the invertibility (they also all pass this), and the five data sets should be solvable. LSE1 is 10x10, LSE2 is 100x100, LSE3 is 200x200, LSE4 is 1000x1000, and LSE5 is 2000x2000.

Problem 2

In lieu of writing my own Gauss Elimination function, I found one online from MIT:
http://ine.scripts.mit.edu/blog/2011/05/gaussian-elimination-in-python/

To test the Gauss Elimination function, I'm going to use two 3x3 LSE's I found on Wikipedia:

$$3x + 2y - z = 1 \qquad 2x - 2y + 4z = -2 \qquad -x + \frac{1}{2}y - z = 0$$

The known solution to this system is $x = 1, y = -2, z = -2$. Another example found on the Gaussian elimination page is

$$2x + y - z = 8 \qquad -3x - y + 2z = -11 \qquad -2x + y + 2z = -3$$

The known solution for this one is $x = 2, y = 3, z = -1$. Testing the myGauss function on these, I recover the known solutions. Then I may proceed to running the function on the supernova LSE's, and to figure out how long it takes to do so.

LSE	Times(s)
1	0.0013191699981689453
2	1.191711187362671
3	9.732844114303589
4	1259.8038818836212
5	10298.86017203331

Problem 3

LSE	Times(s)
1	$4.982948303222656e-05$
2	0.0007839202880859375
3	0.0040018558502197266
4	0.4685628414154053
5	3.8332340717315674

Numpy's linalg.solve function uses the lapack gesv code, which uses LU decomposition with partial pivoting and row interchanges to solve the system.