Exercise 3.1 (May 8, 2020)

-B9TB1707

Question:

Suppose we have three points in 3D space and their coordinates are (x,y,z)=(0.2+rx1 , -0.1+ry1 , 1.0+rz1), (3.0+rx2 , 0.1+ry2 , -1.0+rz2), and (1.0+rx3 , -2.0+ry3 , -0.5+rz3), respectively. r is a random number between -0.1 and 0.1. Find a plane passing through these three points. Note that the equation of a plane that does not pass through the origin (0,0,0) is given by *ax + by + cz = 1*

Solution:

There are two ways to solve this problem: namely either, using an inverse or using the Gaussian elimination method. In this report, I am going to use the inverse to find the solution.

A simultaneous equation:

$$\begin{aligned} 2x + 2y + z &= 0 \\ 3x - y + 3z &= 3 \\ 2x - y - 3z &= -1 \end{aligned}$$

Its vector-matrix notation:

$$\begin{bmatrix} 2 & 2 & 1 \\ 3 & -1 & 3 \\ 2 & -1 & -3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ -1 \end{bmatrix}$$

The solution:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.19512 \\ -0.51220 \\ 0.63415 \end{bmatrix}$$

This is the hint we got.

My code for the solution is below.

```
Editor
File  Edit  View  Debug  Run  Help
* CAPS_03_B9TB1707_3.1.m
1 A= [.2,-.1,1.0;3.0,.1,-1.0;1.0,-2.0,-.5] + (.1 - (.2*rand(3,3)))
2 B = inv(A)*ones(3,1)
3 printf("the plane passing through (%d,%d,%d), (%d,%d,%d) and (%d,%d,%d) is given
4 by %dx + %dy + %dz = 1",A(1,1),A(1,2),A(1,3),A(2,1),A(2,2),A(2,3),A(3,1),A(3,2),A(3,3),
5 B(1,1),B(2,1),B(3,1))
```

The output is as follows. The randomly generated points are the stored in matrix A the points are displayed below.

```
Command Window
>> A =

   0.21272  -0.19886   0.97495
   3.01621   0.14475  -1.08241
   0.97981  -1.99493  -0.45161

B =

   0.64017
  -0.37033
   0.81049

the plane passing through (0.212719,-0.198857,0.974948), (3.01621,0.144751,-1.08241) and (0.9
.99493,-0.451614) is given by 0.640168x + -0.370329y + 0.810486z = 1

>>
```

How it works:

1. Line 1 initializes a matrix with the each point as a row on the matrix. It also uses the A + (B-A)*rand() algorithm, where A and B are the upper and lower limit of the set within which a random number must be chosen (0.1 and -.1 in this case).
2. Line 2 gets the solution and stores it in the array B. It gets the answer by inverting A and multiplying it with a matrix with all elements with 1.
3. Lines 3,4,5 deals with standard printing functionalities.

Conclusion:

Hence I solved the problem by using an inverse matrix. The Gaussian elimination method is more efficient and is better method overall, but this is only a three-dimensional problem, I feel that the inverse method sufficient since it simplifies the code by a lot. This is the reason I used the inverse method.