In this lab, we will examine some basic concepts in using MATLAB to solve linear systems of equations (*i.e.* solve $Ax = b$ for the vector x)

1 Index of Important MATLAB Commands

Here is a list of operations that will be needed in this lab.

- `eye(n)` - generates an $n \times n$ identity matrix.
- `inv(A)` - generates the inverse of a square matrix A.
- `det(A)` - computes the determinant of a square matrix A.
- `x = inv(A)*b` - solves the system $Ax = b$ by formally generating the inverse of A and multiplying by the vector b.
- `x = A\b` - another way to solve the system $Ax = b$.

In addition, you will need to download the chapter on Linear Equations from the online text at

`http://www.mathworks.com/moler/chapters.html`

2 Solving Systems

1) To solve the system

$$\begin{aligned} 2a - b + 3c &= 3 \\ 4a + 2b - c &= -4 \\ -2a - 3b + 5c &= 9 \end{aligned}$$

a) First write the system in matrix-vector form.

$$\begin{pmatrix} 2 & -1 & 3 \\ 4 & 2 & -1 \\ -2 & -3 & 5 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 3 \\ -4 \\ 9 \end{pmatrix}$$

b) Enter the following commands at the MATLAB command prompt:

```
>> format rat
>> A = [2 -1 3; 4 2 -1; -2 -3 5]
>> b = [3 -4 9]'
>> AI = inv(A)
>> x = AI*b
```

Note that turning on the rational formatting using `format rat` will display the results using the closest available rational number for the results. The underlying computations are still done using double precision floating point calculations.

c) The vector `x` should be equal to

$$x = \begin{pmatrix} -1 \\ 1 \\ 2 \end{pmatrix}$$

Notice that, relative to the original variables a, b and c, we have $a = x(1) = -1, b = x(2) = 1$ and $c = x(3) = 2$ This is due to the manner in which the columns were aligned.

d) You can check the validity of the solution by computing

```
>> A*x
```

The result should be equal to the vector `b`.

e) The matrix `AI` is the inverse of A. Check this by computing

```
>> AI*A
>> A*AI
```

In both cases, you should have obtained a 3×3 identity matrix.

f) Another way to solve the system is to use the *backslash* operator.

```
>> xb = A\b
```

The vector `xb` should be the same as the vector `x`.

What is the difference between the two approaches to solving the system? In the first case, we formally generated A^{-1} and computed the solution from

$$x = A^{-1}b.$$

As we see later on, this is not a practical approach. In the second case, we clearly did something equivalent to computing the inverse. This is what we will discuss over the next couple of weeks.

2) Using both approaches in part 1), solve the system

$$\begin{aligned} 3y + 2x - z &= 3 \\ 2z + x + 5y &= -2 \\ 4x - 2z + 3y &= 5 \end{aligned}$$

What is the solution to this system? Verify the solution by computing Ax and comparing the result to b. Also verify that `AI` is the inverse of `A`.

3) Solve the system

$$\begin{aligned} 2x - y &= -1 \\ -4x + 2y &= 2 \end{aligned}$$

What happens in this case? What is the geometric interpretation of the error?

NOTE: Before doing the last 2 problems, return MATLAB to standard formatting:

```
>> format short
```

4) Do Problem 2.3 from chapter you downloaded from the website. Write out the system of equations to be solved in matrix vector form before entering them into MATLAB. Solve the system both by formally generating the inverse of the coefficient matrix and using the backslash operator. What are the resulting forces?

5) There is a theorem in mathematics that says that if you have n coordinates in the x-y plane and all the x-coordinates are different, then there is a unique polynomial of degree at most $n-1$ that goes through the points. For example, if you have 3 points with distinct x-coordinates, there is a unique (at most) quadratic polynomial that goes through them. This process is called *interpolation* and is the basis for many curve fitting techniques.

Suppose you have 3 points: $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and you want to find the quadratic that goes through them. This quadratic has the form

$$y = ax^2 + bx + c.$$

To find values for a, b and c, substitute the known values of the points into this equation as shown below:

$$\begin{aligned} a\,x_1^2 + b\,x_1 + c &= y_1 \\ a\,x_2^2 + b\,x_2 + c &= y_2 \\ a\,x_3^2 + b\,x_3 + c &= y_3 \end{aligned}$$

By writing the above set of equations in matrix-vector form, it can be seen that this now becomes a 3×3 system of equations for a, b and c.

$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Use the process above to compute the quadratic that goes through the points $(-1, 2), (1, 4)$ and $(2.5, -3)$. What are the resulting values of a, b and c?

For the last problems, you should hand in a copy of your MATLAB script files along with the requested output.