A Brief Introduction to Matlab

1 Data representation

MATLAB’s data representations are all interpretations of one basic structure, the matrix. This is a rectangular array of numbers, stored and manipulated internally using the computer’s floating point format and operations. The basic matrix data structure can have special interpretations in a number of situations, including:

**matrix** – A real matrix with number entries. For example,

```
>> A = [.1 .2; .3 .4]  % This creates a 2-by-2 real matrix and
                      % assigns it to variable A
```

**vector** – A matrix with only one row or column. For example,

```
>> v = [.1 .2 .3 .4]  % This creates a 4 component vector
```

**scalar** – A 1-by-1 matrix. For example,

```
>> a = .3
```

**complex scalar** – A 1-by-1 complex matrix. For example,

```
>> b = .3 +.4i
```

**complex matrix** – A pair of matrices, the real and imaginary parts. For example,

```
>> C = [.1 .2; .3 .4] + [.5 .6; .7 .8]*i
```

**flint** – A scalar whose value is an integer. Flints with fewer than nine decimal digits are printed without a decimal point or exponent. For example,

```
>> f = 3
```

**character** – A flint, usually in the range from 32 to 127, interpreted as the ASCII code for a printable character. For example,

```
>> ch = 'a'  % This create a character 'a' and assigns
            % it to variable ch
```
string - A row vector of characters, together with an internal "string" tag. For example,

    >> st = 'Hello World'

string matrix - A matrix with several rows, each of which is a string. For example,

    >> sm = ['Hello'; 'World']

The toolbox 'Symbolic Math Toolbox' extends MATLAB's operations on strings and string matrices to include:

symbolic expression - A string, representing a mathematical quantity. For example,

    >> se = '1/(5+4*cos(x))'

symbolic variable - An isolated character in the range 'a':'z', excluding 'i' and 'j'. For example,

    >> 'x'

symbolic constant - A symbolic expression which does not contain a symbolic variable. For example,

    >> '(1+sqrt(5))/2'

symbolic matrix - A string matrix each of whose rows contains the same number of symbolic expressions, separated by commas. For example,

    >> '[cos(t), sin(t); -sin(t), cos(t)]'

You can build expressions with the usual arithmetic operators and precedence rules:
+ - addition
- - subtraction
* - multiplication
/ - right division
\ - left division
^ - power.

2 How to plot data in MATLAB

To display 2-dimensional data in MATLAB, use the function plot(), whose argument needs to be a vector. Note that a command plot(sin(t)) makes no sense unless MATLAB knows what "t" is (It
does have its own algorithms for sine, cosine, etc.). Proceed thusly to graph the sine function on
the interval [0,4*pi] using 125 equally spaced points (enter the commands as written to the right of
the MATLAB prompt >>):

\[
\begin{align*}
\texttt{>> t = 0:4*pi/125:4*pi} & \quad \text{This creates a vector named t with entries} \\
& [0,4*pi/125,8*pi/125,\ldots,4*pi]
\end{align*}
\]

Notice that this entire list of numbers was displayed on the screen, which is not generally what you
want. To SUPPRESS the display, end the line with a semi-colon (;)

\[
\begin{align*}
\texttt{>> x = sin(t);} & \quad \text{This creates a vector named x with entries} \\
& [\sin(0),\sin(4*pi/125),\ldots,\sin(4*pi)]
\end{align*}
\]

\[
\begin{align*}
\texttt{>> plot(x)} & \quad \text{Notice the units...} \\
\texttt{>> plot(t,x)} & \quad \text{What's the difference?}
\end{align*}
\]

\[
\begin{align*}
\texttt{>> y = sin(t+.25);} & \quad \text{Phase shift} \\
\texttt{>> z = sin(t+.5);} & \quad \text{Another}
\end{align*}
\]

\[
\begin{align*}
\texttt{>> hold on} & \quad \text{The next graph drawn will not erase the present one.} \\
\texttt{>> plot(t,y,'--')} & \\
\texttt{>> plot(t,z,'.'),}
\end{align*}
\]

\[
\begin{align*}
\texttt{>> title('sine function with phase shift')} & \\
\texttt{>> xlabel('t')} & \\
\texttt{>> ylabel('sin(t+)')} & \quad \text{Self-explanatory}
\end{align*}
\]

\[
\begin{align*}
\texttt{>> hold off} & \\
\texttt{>> plot(x,z)} & \quad \text{What is this?}
\end{align*}
\]

\[
\begin{align*}
\texttt{>> subplot(3,1,1), plot(x)} & \\
\texttt{>> subplot(3,1,2), plot(y)} & \\
\texttt{>> subplot(3,1,3), plot(z)} & \quad \text{Subplot(m,n,p) creates an m by n array}
\end{align*}
\]
of plots on a single screen, and refers to the pth one, counting from the upper left. Experiment with this a bit, till you’re sure how it works.

To get a printout from MATLAB:

```
>>print -dps2
```
Sends whatever’s displayed to the printer; once something’s gone from the screen, it can’t be printed. The -dps2 makes it print faster; just plain "print" will also work.

```
>>print -dps2 filename.ps
```
Command to save the current image as a text file with the name "filename.ps". At some later time, this can be sent to the printer by using the command `%lp filename.ps` (this command is issued from outside MATLAB, in a cmdtool window.)

VERY IMPORTANT!! MATLAB has an extremely useful "help" facility. If you don’t know how the print command works, do

```
>>help print
```
and similarly for everything else. In the next section, you may want to type help eul at some point....

Before beginning the next section, do

```
>>subplot(1,1,1)
```
to get back to a single figure. There are other ways to do this, including

```
>>clf
```

3 Linear algebra

Example:
>> A = [2/3 2/3 1/3; 2/3 -1/3 -2/3; 1/3 -2/3 2/3]
This creates a 3-by-3 matrix and assigns
it to variable A

>> b = [1; 1; 1]
This creates a 3-component column vector
and assigns it to variable b

>> who
to check how many variables we have

>> x = A\b
to solve the linear system A*x = b

>> y = inv(A)*b
another way to solve A*x = b, where
inv(A) is the inverse of A

>> x-y
check if x is equal to y

>> z = det(A)
This calculates the determinant of A
and assigns it to variable z

>> [V,D] = eig(A)
What is this for? get information
by using 'help eig'. Can you obtain
the all eigenvalues and eigenvectors of
A?

>> A'*A
this calculates the multiplication
of A and its transpose. What can you say
about the matrix from the result?

Try to find the eigenvalues and eigenvectors of A and solve linear system Ax = b, where

\[
A = \begin{bmatrix}
    1 & -1 & 2 \\
    2 & 0 & -1 \\
    3 & 1 & 3
\end{bmatrix}, \quad b = \begin{bmatrix}
    1 \\
    0 \\
    1
\end{bmatrix}.
\]  \hspace{1cm} (1)

4 Symbolic Mathematics – Differentiation and Integration

Calculus Example:

>> f = '1/(5+4*cos(x))'
Stores the symbolic expression defining
the function in f

>> ezplot(f)
produces the plot of f(x)

>> f1 = diff(f,'x')
differentiates f(x) once with respect
to its symbolic variable x to produce
its first derivative

>> f2 = diff(f,'x',2)
differentiates f(x) twice with respect
to its symbolic variable x to produce
its first derivative

```
>> ezplot(f1)
>> hold on
>> ezplot(f2)
>> hold off
```
What are these commands for? Use the command ’help’ to find them

```
>> F = int(f)   calculate the indefinite integral of
            f(x) with respect to its symbolic
            variable x
>> ezplot(F)
>> F1 = int(f,’x’, 0, 1)   calculate the definite integral of f
                          with respect to x from 0 to 1
>> eval(F1)     get the numeric of F1
```

Could you try this function \( f(x) = e^{-x^2} \)? Calculate its first, second derivatives, first anti-derivative, and definite integral of \( f(x) \) from \( x = 0 \) to \( x = \text{inf} \) (the infinite symbol used in matlab). Don’t forget to plot these functions.

5 Packages of ODE Solvers

```
>> dfield   plot the direction fields for scalar
           ODE
>> ppp      plot the direction fields for systems
           of two ODEs
```

6 Exit from MATLAB

```
>> quit     Exit from MATLAB
```