Additional Topics

Jiseok Chae

Department of Mathematical Sciences
KAIST

Week 14

Contents

To avoid confusion, we have tried avoiding the use of complex numbers in MATLAB, sticking to real vectors and matrices.

But since now we have learned that eigenvalues of real matrices can be complex, it seems to be a good time to see how we can use complex numbers in MATLAB.

Using `i` or `j` with a number in front of it without a multiplication symbol generates a complex number.

```
>> z = 3 + 4i

z =

   3.0000 + 4.0000i

>> exp(1j * pi) + 1 % e^{i\pi} + 1 = 0
      % a.k.a. ''the most beautiful equation''

ans =

   0.0000e+00 + 1.2246e-16i
      % ooh... not so beautiful in MATLAB.
```

`real` computes the real part, and `imag` computes the imaginary part.

```
>> z = 3 + 4i;
>> real(z)

ans =

     3

>> imag(z)

ans =

      4
```

For a complex input, `conj` computes the complex conjugate, and `abs` computes the absolute value.

For $z = a + bi$ with $a, b \in \mathbb{R}$, we define $|z| = \sqrt{a^2 + b^2}$.

```
>> z = 3 + 4i;
>> conj(z)

ans =

   3.0000 - 4.0000i

>> abs(z)

ans =

    5.0000
```

An example of complex numbers arising from eigenvalues of a real matrix:

```
>> A = [1, 0, 0;
        0, 0, -1;
        0, 1, 0];
>> v = eig(A)

v =

   0.0000 + 1.0000i
   0.0000 - 1.0000i
   1.0000 + 0.0000i
```

Contents

When defining a function by ourselves, we made an `.m` file, with the format

`function` $\langle output \rangle = \langle function\ name \rangle(\langle inputs \rangle)$
[*function body*]

Functions defined in this way have an innate name associated with it.

In contrast, anonymous functions are—as the name indicates—functions without a name.

Anonymous functions are not stored as a separate file, but associated with a variable. This enables us to pass a function as an input of another function. This can be very useful in some cases.

However, anonymous functions should be declared with its return value(s) right away.

The format for declaring an anonymous function is

`@`($\langle$*inputs*$\rangle$) [*return value*]

Let us see an example. Say we want to define an anonymous function

$$\mathtt{x} \mapsto \frac{\mathtt{x}\sin(\mathtt{x})}{1+\cos^2(\mathtt{x})}.$$

To use it later, let us associate this anonymous function with the variable named `f`.

```
>> f = @(x) x.*sin(x)./(1+cos(x).^2)

f =

  function_handle with value:

    @(x)x.*sin(x)./(1+cos(x).^2)
```

A *function handle* is a data type in MATLAB that represents a function.

You can use `f` as if it is a user-defined function.

```
>> f(0)

ans =

     0

>> f(pi/2)

ans =

    1.5708
```

Moreover, you can use `f` as any other variable. Especially, you can pass it to another function as an input.

Suppose we create a function file as follows:

eval_at_0.m

```
function res = eval_at_0(f)
res = f(0);
```

Recall that

```
>> f = @(x) x.*sin(x)./(1+cos(x).^2);
```

The function `eval_at_0` takes a function handle `f` and returns the evaluation $\texttt{f}(0)$.

```
>> eval_at_0(f)

ans =

     0

>> eval_at_0(@(x) sech(x))

ans =

     1
```

A representative example of where function handles become useful is numerical integration.

The command `integral(f, a, b)` takes a function handle `f`, and numerically evaluates the integral $\int_{\mathtt{a}}^{\mathtt{b}} \mathtt{f}(x)\,\mathrm{d}x$.

```
>> integral(f, 0, pi)

ans =

    2.4674
```

In fact, $\displaystyle\int_0^{\pi} \frac{x \sin x}{1+\cos^2 x}\,dx = \frac{\pi^2}{4}$.

For anonymous functions to have multiple outputs, we have to pack the outputs into a single array and pretend as if there is only one output.

```
>> g = @(x) [2*sin(x).*cos(x), sin(2*x)]

g =

  function_handle with value:

    @(x)[2*sin(x).*cos(x),sin(2*x)]

>> g(pi/12)

ans =

    0.5000    0.5000
```

Meanwhile, having multiple inputs is not a problem.

```
>> h = @(x, y) 1./(1 - x.*y)

h =

  function_handle with value:

    @(x,y)1./(1-x.*y)
```

The variable `h` contains the anonymous function $(x, y) \mapsto \dfrac{1}{1 - xy}$.

It is known that $\int_0^1 \int_0^1 \frac{1}{1-xy}\,dxdy = \frac{\pi^2}{6}$.

The command `integral2(f, a, b, c, d)` computes the double integral $\int_{\mathtt{c}}^{\mathtt{d}} \int_{\mathtt{a}}^{\mathtt{b}} \mathtt{f}(x,y)\,dxdy$ numerically.

```
>> integral2(h, 0, 1, 0, 1)

ans =

    1.6449
```

There is also a function named `integral3`, which can numerically compute a triple integral. A triple integral is an integral which is done with respect to three variables.

Thank you!