Complex Numbers

A complex number is generally written in the form

z = a + bi

where a is called the real part of the number z, b is the imaginary part of z, and i is $\sqrt{-1}$. This is the way mathematicians usually write a complex number;

in engineering it is often written as a + bj, where j is $\sqrt{-1}$. A complex number is purely imaginary if it is of the form z = bi (in other words, if a is 0).

We have seen that in MATLAB both i and j are built-in functions that return$\sqrt{-1}$ (so, they can be thought of as built-in constants). Complex numbers can be created using i or j, for example, 5 + 2i or 3 – 4j.

The multiplication operator is not required between the value of the imaginary part and the constant i or j.

Quick question!

Is the value of the expression 3i the same as 3 * i?

Answer: It depends on whether i has been used as a variable name or not. If i has been used as a variable (for example, an iterator variable in a for loop), then the expression 3 * i will use the defined value for the variable, and the result will not be a complex number. Therefore, it is a good idea when working with complex numbers to use 1i or 1j rather than just i or j. The expressions 1i and 1j always result in a complex number, regardless of whether i or j have been used as a variable.

```
>> i = 5;
```

```
>> i
i=
    5
>> 1i
ans =
    0 + 1.0000i
```

MATLAB also has a function complex that will return a complex number. It receives two numbers, the real and imaginary parts in that order, or just one number, which would be the real part (so the imaginary part would be 0). Here are some examples of creating complex numbers in MATLAB:

```
>> z1 = 4 + 2i
z1 =
4.0000 + 2.0000i
>> z2 = sqrt(-5)
z2 =
0 + 2.2361i
>> z3 = complex(3,-3)
z3 =
3.0000 - 3.0000i
>> z4 = 2 + 3j
z4 =
2.0000 + 3.0000i
>> z5 = (-4) ^ (1/2)
```

ans =

0.0000 + 2.0000i

>> myz = input('Enter a complex number: ')

Enter a complex number: 3 + 4i

myz =

3.0000 + 4.0000i

Notice that even when j is used in an expression, i is used in the result. MATLAB shows the type of the variables created here in the Workspace Window (or using whos) as double (complex). MATLAB has functions real and imag that return the real and imaginary parts of complex numbers.

>> real(z1)

ans =

4

>> imag(z3)

ans =

-3

To print an imaginary number, the disp function will display both parts automatically:

>> disp(z1)

4.0000 + 2.0000i

The fprintf function will print only the real part unless both parts are printed separately:

>> fprintf('%f\n', z1)

4.000000

```
>> fprintf('%f %f\n', real(z1), imag(z1))
4.000000 2.000000
>> fprintf('%f + %fi\n', real(z1), imag(z1))
4.000000 + 2.000000i
```

For the variable z5, even though it shows the answer as 3, it is really stored as 3 + 0i, and that is how it is displayed in the Workspace Window.

Equality for Complex Numbers

Two complex numbers are equal to each other if both their real parts and imaginary parts are equal. In MATLAB, the equality operator can be used.

```
>> z1 == z2
ans =
0
>> complex(0,4) == sqrt(-16)
ans =
1
```

Adding and Subtracting Complex Numbers

For two complex numbers z1 = a + bi and z2 = c + di,

z1 + z2 = (a + c) + (b + d)i

z1 – z2 = (a – c) + (b – d)i

As an example, we will write a function in MATLAB to add two complex numbers together and return the resulting complex number.

addcomp function

In most cases, to add two complex numbers together you would have to separate the real and imaginary parts, and add them to return your result.

```
function outc = addcomp(z1, z2)

% Adds two complex numbers and returns the result

% Adds the real and imaginary parts separately

realpart = real(z1) + real(z2);

imagpart = imag(z1) + imag(z2);

outc = realpart + imagpart * i;

>> addcomp(3+4i, 2-3j)

ans =

5.0000 + 1.0000i
```

addcomp2 function

MATLAB will automatically do this in order to add two complex numbers together (or subtract).

```
function outc = addcomp2(z1,z2)

% Adds two complex numbers and returns the result

outc = z1 + z2;

>> addcomp2(3+4i, 2-3j)

ans =

5.0000 + 1.0000i
```

Multiplying Complex Numbers

For two complex numbers z1 = a + bi and z2 = c + di,

z1 * z2 = (a + bi) * (c + di)

= a*c + a*di + c*bi + bi*di

= a*c + a*di + c*bi – b*d

= (a*c – b*d) + (a*d + c*b)i

For example, for

z1 = 3 + 4i

z2 = 1 - 2i

z1 * z2 = (3*1 - -8) + (3*-2 + 4*1)i = 11 -2i

This is, of course, automatic in MATLAB:

```
>> z1*z2
ans =
11.0000 - 2.0000i
```

Complex Conjugate and Absolute Value

The complex conjugate of a complex number z = a+bi is $\bar{z}$ = a - bi. The magnitude, or absolute value of a complex number z is z = $\sqrt{a^2 + b^2}$. In MATLAB, there is a built-in function *conj* for the complex conjugate, and the abs function returns the absolute value.

```
>> z1 = 3 + 4i
z1 =
3.0000 + 4.0000i
>> conj(z1)
ans =
3.0000 - 4.0000i
>> abs(z1)
ans = 5
```

Polar Form

Any complex number z = a + bi can be thought of as a point (a,b) or vector in a complex plane in which the horizontal axis is the real part of z, and the vertical axis is the imaginary part of z. So, a and b are the Cartesian or rectangular coordinates. Since a vector can be represented by either its rectangular

or polar coordinates, a complex number can also be given by its polar coordinates r and θ, where r is the magnitude of the vector andθ is an angle.

To convert from the polar coordinates to the rectangular coordinates:

a = r cosθ

b = r sinθ

To convert from the rectangular to polar coordinates:

r = $|z|$ = $\sqrt{a^2 + b^2}$

θ=arctan$\left(\frac{b}{a}\right)$

So, a complex number z = a + bi can be written as r cosθ + (r sinθ)i, or z = r (cosθ + i sinθ)

Since $e^{i\theta}$ = cos θ+ i sinθ , a complex number can also be written as z = r$e^{i\theta}$. In MATLAB, r can be found using the abs function, and there is a special built-in function to find θ, called angle.

```
>> z1 = 3 + 4i;
r = abs(z1)
r = 5
>> theta = angle(z1)
```

```
theta =
0.9273
>> r*exp(i*theta)
ans =
3.0000 + 4.0000i
```