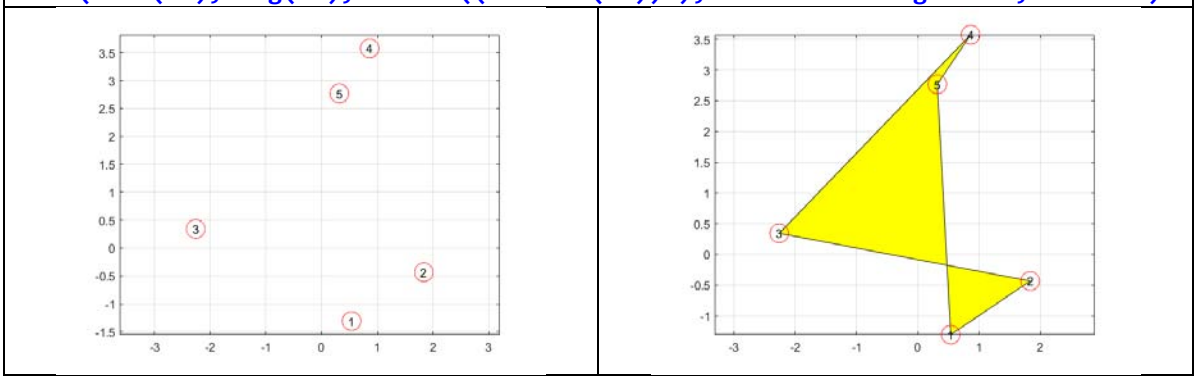
Exercises and Laboratories

SC2_01b – Solve, in MATLAB, the following exercises by using complex numbers.

1. The following code creates and displays N random complex numbers, which are considered as points in the real plane (see the left figure). We want to find a **simple non-intersecting walk** through them. But there is a problem: following the natural order of the points, the random path is often intertwined (see the right figure).

```
N=5;
rng('default') % comment this line to change the random seed each time
Zn=randn(N,1) + 1i*randn(N,1);
figure; patch(real(Zn),imag(Zn),'y'); grid on; box on; axis equal; hold on
plot(real(Zn),imag(Zn),'or','MarkerSize',16)
text(real(Zn),imag(Zn),num2str((1:numel(Zn))'),'HorizontalAlignment','center')
```

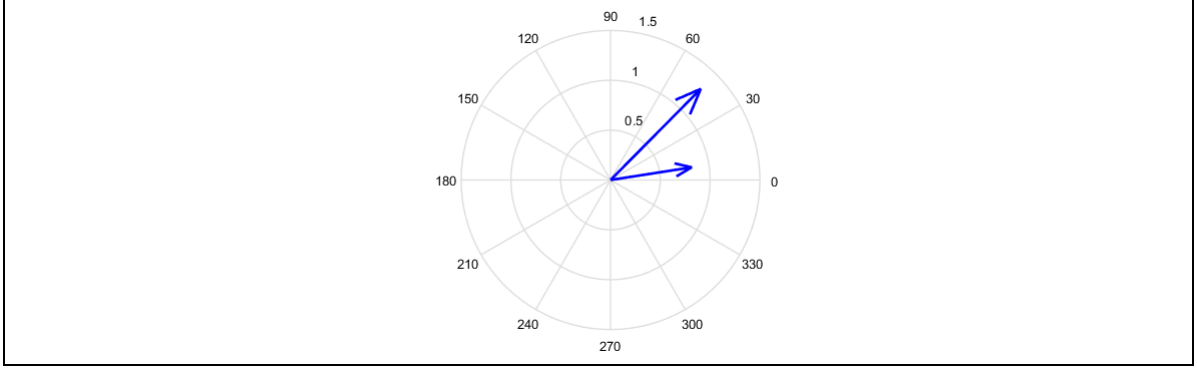


How to find a **simple non-intersecting walk**?

[Hint: compute the barycenter and make a suitable use of the **argument** of a complex number].

2. The following code creates and displays two random complex numbers, which are considered as vectors in the real plane (see the figure). They locate two half-lines.

```
N=2;
rng('default')
Zn=rand(N,1) + 1i*rand(N,1);
figure ; h=compass(Zn); set(h,'Color','b','LineWidth',2)
```



How to find their **bisector**?

[Hint: make a suitable use of the **argument** of a complex number].

3. Starting from a random integer value n, find all the n^{th} roots of a random complex number z by means of a primitive n^{th} root of unity, selected randomly. The MATLAB function `rand()` returns

uniformly distributed pseudorandom **real numbers** between 0 and 1; the MATLAB function `randi()` returns uniformly distributed pseudorandom **integers**.
[Hint: first of all, compute a particular n^{th} root of z; second, compute all the n^{th} roots of unity and, among them, put only the primitive ones into a vector].

4. Display the surfaces of the **real part**, **imaginary part** and **modulus** of the function
$$f(z) = z^4 - i$$
and add them the points corresponding to its zeros.

.