### Lab08 – Complex 2D Rotation

# Content

# Learning objectives

## Exam objectives

By the end of this lab you should be able to (pen and paper):

* Perform 2D rotations through multiplication of complex numbers
* Practice the previous in the trigonometric shape of the complex numbers
* Design **non-standard** composite transformations as well

We advise you to **make your own summary of topics** which are new to you.

## Supportive objectives

### Self-support by GeoGebra

More specifically related to the above you should in GeoGebra:

* Compute 2D rotations through multiplication of complex numbers
* Practice the previous in the trigonometric shape of the complex numbers
* Compute non-standard composite transformations as well
* Visualize the original polygon and its rotation image in the complex plane

# Exercises

Dependent of the lab session you may work individually or teamed (organized by the lab attendant). In either case make sure that throughout the course of this lab, you backup sufficiently your solution file on your local machine as

**1DAExx-0y-name1**(+name2+name3).GGB given **xx**=groupcode, **0y**=labindex

Use the online **GeoGebra Classic:** <https://www.geogebra.org/classic>

## Basic exercises

### Exercise1

This exercise prepares for later on expressing transformations into complex numbers.

* Convert the following complex numbers from the euler notation to algebraic notation. Try first with pen and paper, then verify your finding by GeoGebra.

*z1 =*

*z2 =*

*z3 =*

* Create in GeoGebra the list with naming *shape = { z1 , z2 , z3}*
* Show in GeoGebra/2Dgraphics its corresponding *Polygon(shape)*

*A screenshot of a computer

Description automatically generated with medium confidence*

### Exercise 2

This exercise practices expressing transformations into complex numbers. In this exercise we calculate the result of a TRS transformation. The general formula for such a transformation with complex numbers is:

In other words, *Cp* is the input complex number we transform.

*CSR* is the complex number that represents a rotation and scale.   
The **uniform scaling** corresponds to the **modulus** of this complex number.   
The **rotation** corresponds to the **argument** of this complex number.

*CT* is the complex number that does the translation.

**Given**:

*Cp* *= 1 + 3i*

* Calculate *Cp’* given following scale, rotation and translation scenarios:

1. Scale:2, Rotation 30° and translation *CT* *= 4 + 3i*
2. Scale:3, Rotation -75° and translation *CT* *= -1 + 2i*
3. Scale: 0.5, Rotation 125° and translation *CT* *= -2 -2i*

* Visualize each of these three image points Cp’ in the same Gauss plane by showing them within GeoGebra/2Dgraphics

## Bridging exercises

### Exercise 3

Create a .GGB-file that uses a complex number *Cp* to create a rotation around the origin of the axis system. The radius of this rotation is 100 units. The rotation comes with a period of 360 days per full rotation. Start from a given reference point on which you perform translation and rotation (see Exercise 2)

Visualize this Planet rotation in GeoGebra/2Dgraphics by a Point with Pointsize 9 in colour red, and a slider to run the variable **days**.

### Exercise 4

Extend the previous .GGB-file by adding a Moon to the former red Planet. The Moon, complex number *CM*, circles around the planet with a radius of 40 units and makes a full rotation around the planet in 60 days.

*To chain* both transformations, you need to apply the rotation and translation in the same order as used for the solar system scenegraph types, given

Visualize this Moon rotation in GeoGebra/2Dgraphics by a Point with Pointsize 6 in colour green. Of course the slider already running the **days**, remains in use.

**Bonus:**

Finalize this .GGB-file by adding a Satellite to the former green Moon. This Satellite, complex number *CS*, circles around the planet with a radius of 10 units and makes a full rotation around the planet in 30 days.

Visualize this Satellite rotation in GeoGebra/2Dgraphics by the Pointtype Diamond with Pointsize 6 in colour blue. The slider already running the days, stays in.

**Hint**: it may be of major assistance to sketch the object tree of this scenegraph.

Chart, scatter chart

Description automatically generated

## Contextual practice

### Exercise 5

Create a .GGB-file for an off-center spinning polygon, hereafter outlined.

Setup both in CAS and 2Dgraphic these three complex numbers

*z1 =*

*z2 =*

*z3 =*

* Create in GeoGebra the list with naming *shape = { z1 , z2 , z3}*

Show in GeoGebra/2Dgraphics its corresponding *Polygon(shape)*

* Calculate *zm =* (*z1 + z2 + z3)/3*
* To nowsetup the composite transformation which pivots the polygon around its own centroid, you need to apply the same steps as for action matrices, like

The rotation comes with a period of 30 pulses per full rotation.

**Practical hint:** in GeoGebra many calculations are listable meaning you can apply them on a complete list as well, instead of on just one single operandus

* Visualize this off-center rotation in GeoGebra/2Dgraphics by the rotation image *Polygon(shape’’)* and a slider to run the variable **pulses**.

### Exercise 6

Create a .GGB-file for a three limb bone structure, hereafter outlined.

Setup both in CAS and 2Dgraphic these four complex numbers

*z1 =*   
*z2 =*

*z3 =*

*z4 =*

* Create in GeoGebra the list with naming *vertices = { z1 , z2 , z3, z4 }*

Show in GeoGebra/2Dgraphics its corresponding *diamond =* *Polygon(vertices)* which is centered on the origin. It may be helpful to visualize it temporarily.

* To nowsetup the scenegraph, you need to follow the same steps as for action matrices by applying TRS transformations. The general formula for such a transformation with complex numbers is:

**Hint**: it is of major assistance to sketch the object tree before the scenegraph.

* Mount the first limb named *biceps* at the origin by an appropriate shift

**Practical hint:** in GeoGebra many calculations are listable meaning you can apply them on a complete list, therefore *biceps = vertices +* applies.

* Nextly, rotate this upper limb by an angular slider clipped between 0 en Pi (so it can not be overstretched whilst animating it) leading to image *biceps’*

Visualize this moving upper limb in GeoGebra/2Dgraphics by *Polygon(biceps’)* and the slider to adjust its rotation angle.

* Mount the second limb named *forearm* at the end of biceps’ and make it rotational by an angular slider clipped between 0 en Pi/2 (to prevent overstretching). Visualize this moving middle limb in GeoGebra/2Dgraphics by *Polygon(forearm)* and the slider to adjust its rotation angle.
* Mount the last limb named *hand* at the end of the *forearm* and make it rotational by an angular slider clipped between - Pi/4 en Pi/4 (to prevent overstretching). Visualize this waving hand in GeoGebra/2Dgraphics by *Polygon(hand)* with its slider to wave this lab session goodbye.

Chart, line chart

Description automatically generated

# References

## Books

**Animation Maths**, chapter Hypercomplex numbers, par Complex numbers and transformations

## Softwares

<https://wiki.geogebra.org/en/Complex_Numbers>

<https://wiki.geogebra.org/en/Category:Tools>

<https://www.geeksforgeeks.org/complex-numbers-c-set-1/>

<https://en.cppreference.com/w/cpp/numeric/complex>