Riemann Sphere and Complex Plane Transformations with Sphere Movements (Software Aspect)

Vladimir Nodelman
Holon Institute of Technology
Israel
nodelman@hit.ac.il, vladinod@gmail.com

Abstract: *There are various ways to visualize complex functions, but most of them provide only passive show, even with support for manually controlled and/or parameterized animations. The Riemann sphere offers a very different instrument for modeling complex plane transformations, often allowing them to be defined as the result of a transformation of the sphere itself. In this paper, we present the use of the Riemann sphere model in discovering the features of complex functions and the construction of these functions as results of sphere movements with the help of the author's non-commercial software VisuMatica.*

1. Introduction. The Riemann Sphere

Bernhard Riemann proposed a geometric interpretation of the set of complex numbers. He placed them on a sphere, called the **Riemann sphere**.

To show how each complex number is associated with a point on the sphere position the sphere with the "south pole" at the origin of the complex plane, and from the "north pole" draw rays to each point of the plane. The point at which the ray going to the number z intersects the sphere is called the point z on the Riemann sphere.

At the same time, one point of the sphere, namely the "north pole", does not correspond to any complex number. However, if we make the number "move" along the complex plane to infinity (in any direction), then its image on the Riemann sphere will approach the "north pole".

This allowed Riemann to call the "north pole" the "*point of infinity*" and to add it to the complex plane, thus obtaining **an extended complex plane** $\overline{\mathrm{C}}=\mathrm{C}\cup\{\infty\}$.

The concepts of the real and imaginary parts of the point ∞ not defined. Arithmetic operations of complex numbers extended by defining, for any $z\in\mathrm{C}$, $z+\infty=\infty$, $z\cdot\infty=\infty$ for any $z\neq 0$, $\infty/0=\infty$ and $0/\infty=0$, and $\infty\cdot\infty=\infty$. Herewith, $\infty+\infty$, $\infty-\infty$ and $0\cdot\infty$ are undefined.

The images in Fig.1 a, b)[1] show the mapping of a large blue point on the plane, corresponding to the position of the mouse pointer in the domain view, onto the sphere of radius 0.5. If an opaque sphere does not cover this point, then the visible part is depicted as a solid, bold red segment, and the remaining, hidden part, as a thin dashed red line that connects the blue point of the plane with the north pole of the sphere.

The projection onto the sphere (the point where the ray intersects with it) is shown as a bold red dot if it is visible and a small red dot if it is invisible.

The equator of the sphere is drawn in orange.

[1] These and the following images of the complex plane mapping onto the Riemann sphere received after definition of some complex function, say, $w = z$.

The red arrows in Fig.1 c), d) present the "*Mapping* 2D" settings that allow this to show as an <u>*empty*</u> mapping onto an <u>*opaque*</u> <u>*sphere*</u> with the <u>*domain plane*</u> touching the sphere <u>*at the south*</u>.

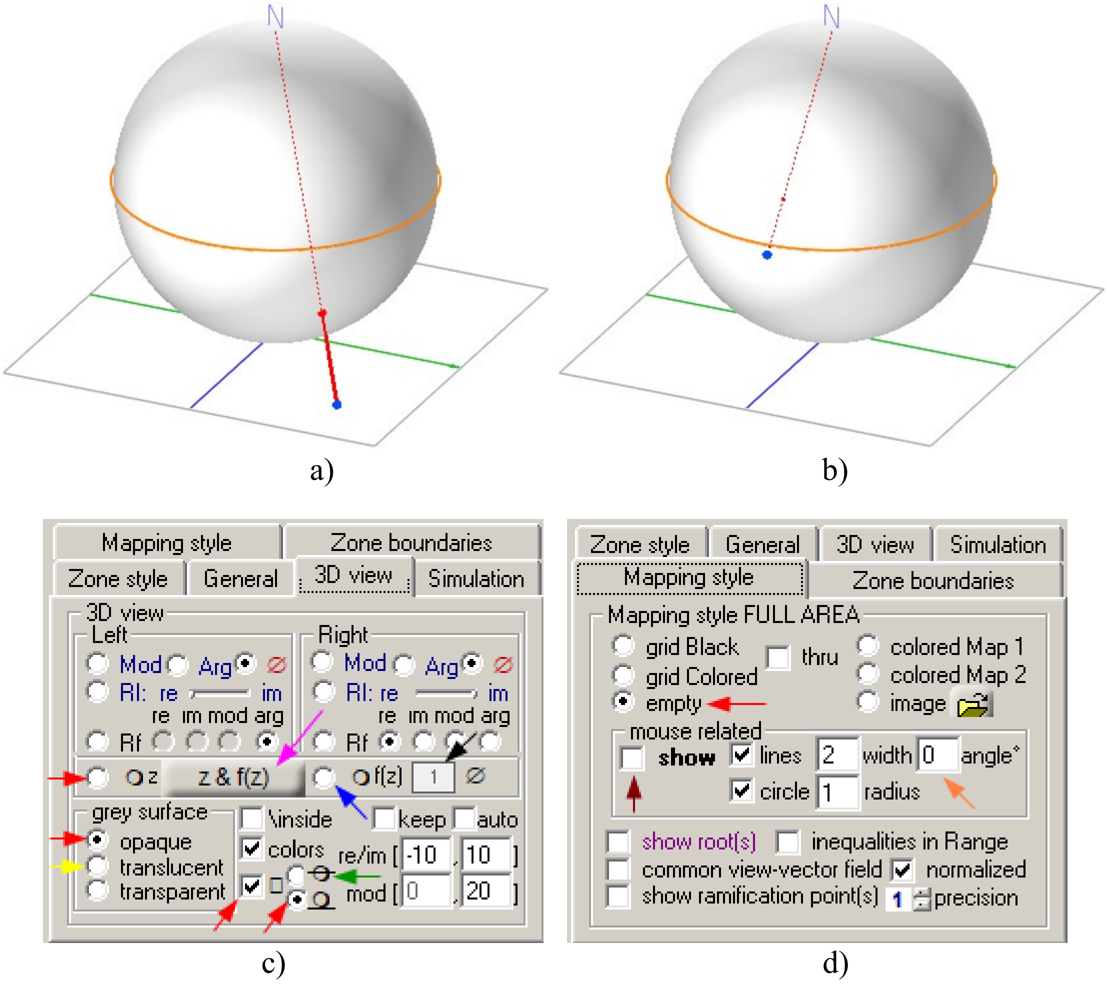


Figure 1

Frequently used another projection model with the center of the Riemann sphere at the zero of the complex plane (the sphere radius is 1). Fig.2 shows the corresponding picture obtained by switching to the mode corresponding to the green arrow. In this case, one can sequentially press the **F6** keyboard button to see only one semi-sphere: top, bottom, or return to the entire view to obtaining a clear show (Fig.3).

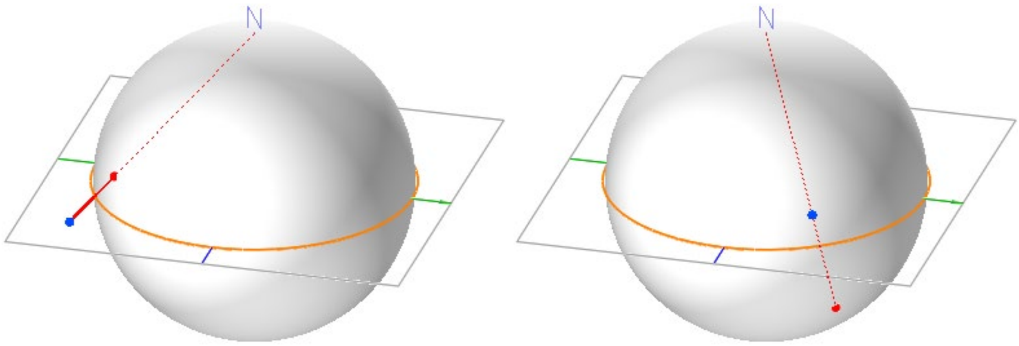


Figure 2

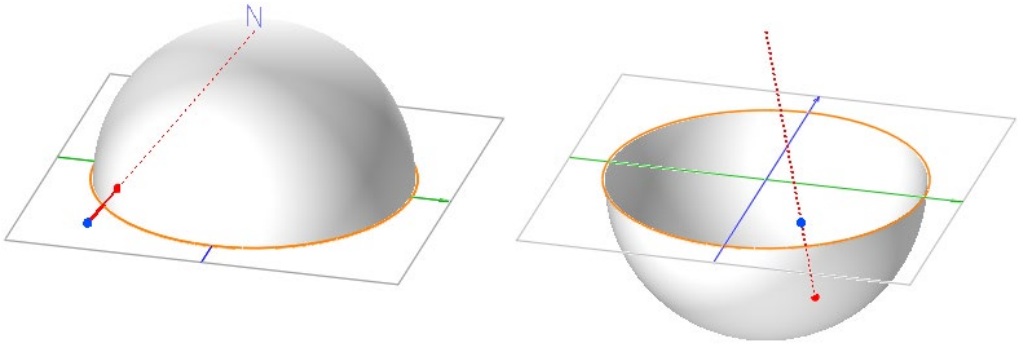


Figure 3

Another way to see the correspondence, especially if it is hidden, is to set the sphere drawing to *translucent* instead of *opaque* (Fig.4). (See the yellow arrow in Fig.1).

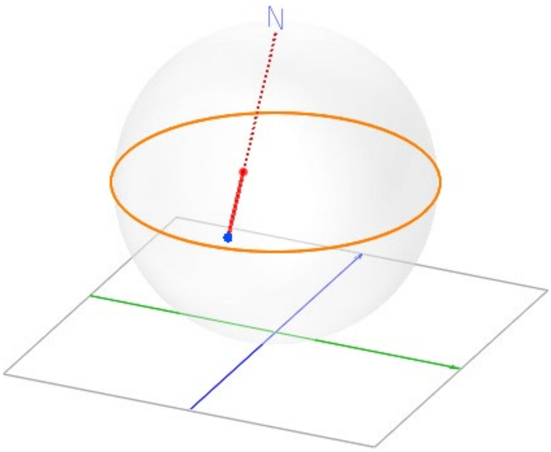


Figure 4

All these ways of representing the projection onto the Riemann sphere concerned the universal mechanism for constructing the image of a complex number - the point of the complex plane pointed to by the mouse. By replacing the "*Mapping style of FULL AREA*" with some other, not "*empty*" mode, we obtain a more expressive show.
The color coding emphasizes the correspondence (Fig.5).
The left image highlights the fact that the complex numbers inside the circle of radius 1 are projected onto the lower hemisphere, while the others – are onto the upper one.
One can rotate the 3D scene by pressing the ←, ↑, ↓, → buttons and moving the mouse pointer to grasp better the mapping features.
Some inconvenience of such visualization is associated with the relation of the boundaries of the domain area with its image on the plane in the 3D scene. While the rectangular image of the current domain area is projected entirely onto the sphere (wrapped, not completely covering it), the domain itself may be far from being completely visible on the plane in 3D. In this case, changing of viewing angle will help us.

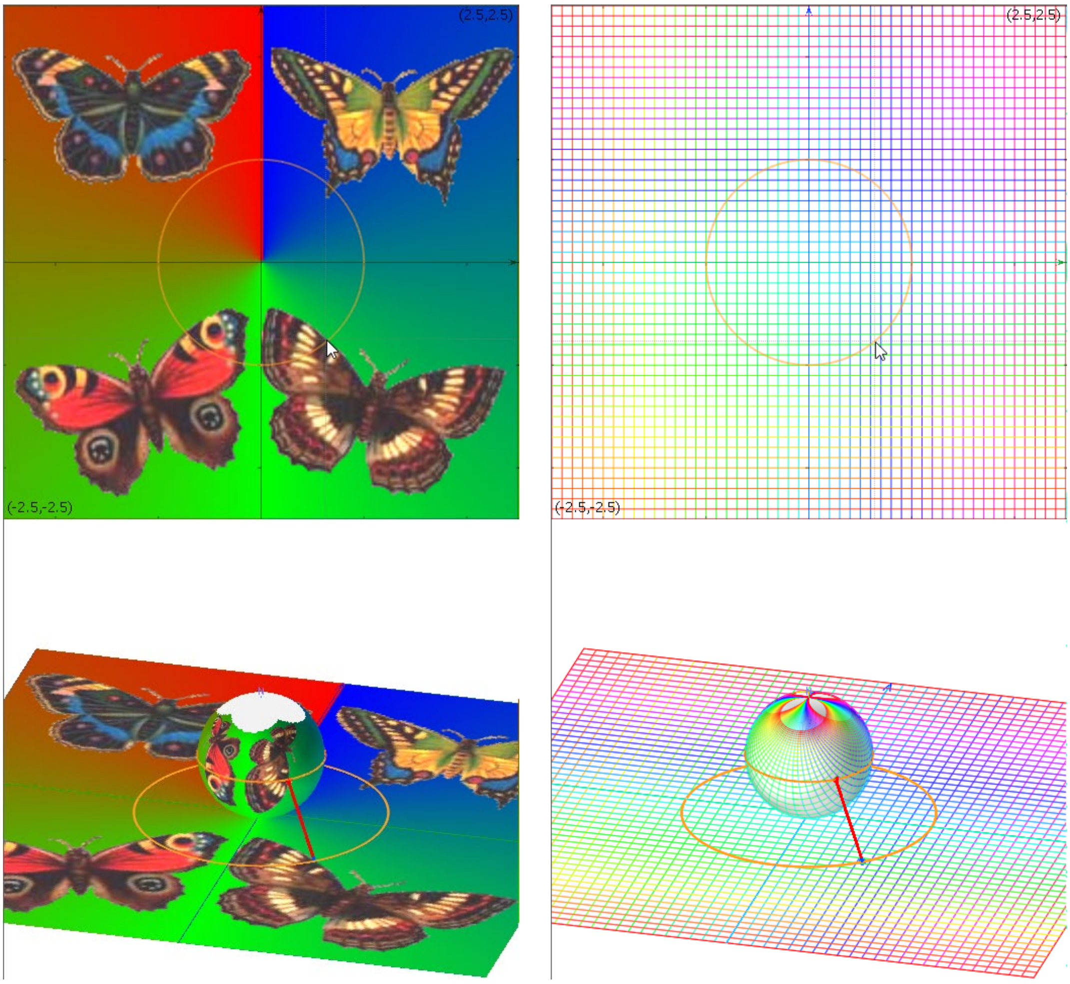


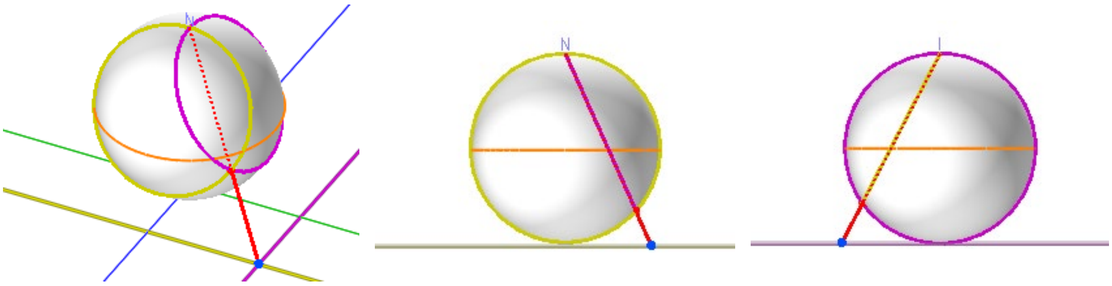
Figure 5

If the center of the sphere is located at the plane origin, and the intervals of the domain belong to the segment [-1, 1], the entire domain picture of the complex plane (except maybe its corners) lies inside the sphere. This will require either to set the sphere as translucent/transparent or to show only one hemisphere. Of course, expanding the domain boundaries can also help.

The right picture in Fig.5 indicates an interesting feature of the images of horizontal and vertical lines when the "*Mapping style of FULL AREA*" is set to "*grid…*". They are looking like circles on a sphere. Moreover, these circles pass through the north pole!

To check this observation let us use the show of "*mouse-related*" objects. Check the checkbox, marked by the brown arrow in Fig.1, and uncheck the "*circle*" check box. Move the mouse pointer and observe the scene (Fig.6 a). We got two perpendicular lines parallel to the coordinate axes passed through the mouse pointer. Rotate the scene and check if the lines are projected to planar figures (hold the Ctrl button while pressing the arrow keys to get smooth rotations). Fig.6 b) and c) are the result of a front and right view of the scene (received by selecting proper options of the "3D *view*" submenu of the "*View*" menu).

- Use Fig.6 to *prove* geometrically that the lines' images are circles that include the north pole.



a) b) c)

Figure 6

- Prove that this result is valid in the case of an arbitrary line.

- Prove that the preimage of any circle on the Riemann sphere passing through the north pole is a straight line.

- What do a straight line and its image on the Riemann sphere have in common?

- Prove that the image of a straight line passing through the origin is a great circle centered at the center of the sphere.

Enter a non-zero value of the angle in the text box "*angle* °" (see the orange arrow in Fig.1). Repeat the experiment this time with the straight lines rotated.

We already know the definition of the angle between planar/spatial curves at the point of their intersection as the angle between the tangents to them at that point.

- How - to "see" the angle between two curves on a sphere?
  Play with the mouse and the arrow keys to check your answer concerning circles.

- What is the angle between the circles-images of our mutually perpendicular lines? Does it depend on the position of the point of intersection of the lines-preimages?

- Are the values of these angles equal at each of the two points of the circles' intersection?

Fig.7 shows the positions of the three-dimensional scene, illustrating the angles between the circles. The left image is a top view of the north pole. The right view is a suitably rotated scene, allowing us to see the angle between the circles at the second intersection point.

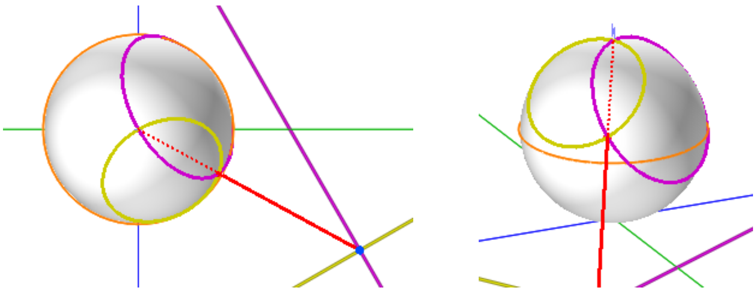


Figure 7

- How does the right picture in Fig.8[2], obtained as a result of turning the scene shown in the left picture, help to grasp the idea of proving the equality of both angles at the intersection points between the circles-images of mutually perpendicular lines?

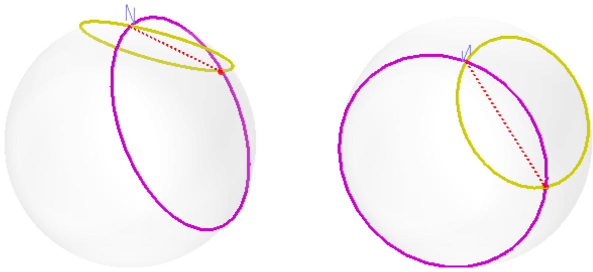


Figure 8

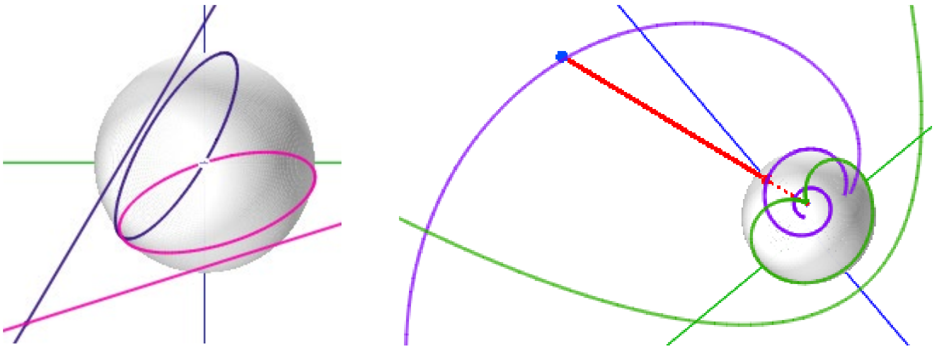
To check the angle between images of any two lines:
- add two lines using the "*linear element*" tab of the "*Geometry*" dialog,
- disable the "*mouse-related*" show,
- set the scene view to "*3D top view*",
- select both lines in the Domain view and start dragging them.

Compare the angles between lines and the circles at the north pole (Fig.9 a) – top view). Change the lines' directions and repeat the observation.

Add two parallel lines (define the first one by two free points, and the second - as parallel to it passing through a third free point). Move the points.
- What is the angle between the circles-images of parallel lines?

  Prove the correctness of your answer.

Use Fig.9 a) to grasp the idea and prove that the angles of the lines and their circles-images are equal.



a) b)

Figure 9

- Is the obtained result also valid concerning the angles between smooth curves on the plane and their images on the sphere at their intersection point (Fig.9 b)?

[2] The sphere is set to translucent. The show of the plane and equator is disabled.

To study the images of a circle when mapped onto a Riemann sphere, cancel “*lines*” and turn on “*circle*” (see Fig.1 d). Move the mouse in the domain area and observe the location of the image of the circle. Change the circle radius and repeat the mouse movement.

- What is the image of a circle?
- How to prove this fact visually?
- Can the image of a circle pass through the north pole of the sphere?
- Under what condition does the center of the circle on the sphere belong to the segment connecting the center of the prototype circle with the north pole?
- What is the preimage of a circle on the sphere?

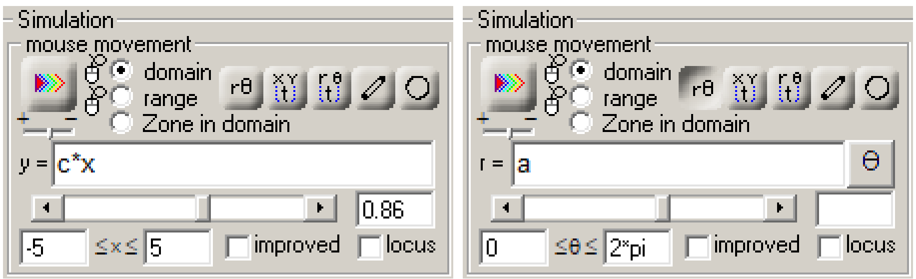
Consider the dependence of the image of a circle on the position of the center of the preimage more precisely, not manually. To that end, set the sphere to be translucent, and enable the show of the bottom plane. In the “*Simulation*” tab set the simulation of mouse movement to the domain, and movement trajectory as $y = cx$ (Fig.10 a). Reposition the slider and observe the behavior of both circles. Change the value of parameter c and repeat the observation.

Set the “*3D top view*” option. It will help to understand the studied dependence.

- How does the size and position of the image circle depend on the distance of the center of the circle on the complex plane to zero?

Change the nature of the simulation by making the mouse move along a circle of radius a around zero (Fig.10 b). Reposition the slider and observe the behavior of both circles. Change the value of parameter a, and the circle's radius and repeat the observation.

- Conclude.



a) b)

Figure 10

- What is the radius of the image of a circle of radius r whose center is at a distance a from zero?

  Solve the problem for the Riemann sphere in both cases: when it touches the complex plane at zero (the sphere radius equals 0.5), and when its center coincides with the complex zero (the sphere radius equals 1).

2. Inversion Transformation

Consider the character of mapping onto the Riemann sphere in the case of an inversion transformation $w = \frac{1}{z}$.

As we know, this transformation maps lines and circles to circles, except for lines passing through zero whose images are straight lines. We have also recognized that mapping onto the Riemann sphere maps lines and circles to circles on a sphere. Therefore, it is obvious that the spherical images of lines and circles under the inversion transformation are circles (Fig.11).

Start moving the mouse pointer in the upper left view of the domain and observe the images it points to in the other windows. Click the mouse and start rotating the 3D scene using the keyboard arrows. Compare the images of the two spheres during rotation.

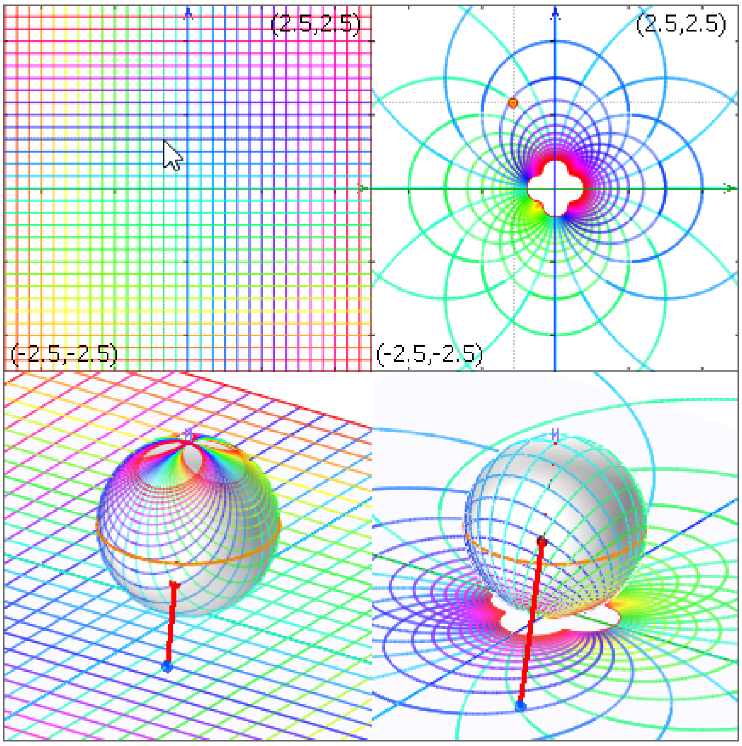


Figure 11

Repeat these activities with zones: drag them and rotate the scene (Fig.12).

- Has our conclusion been confirmed?

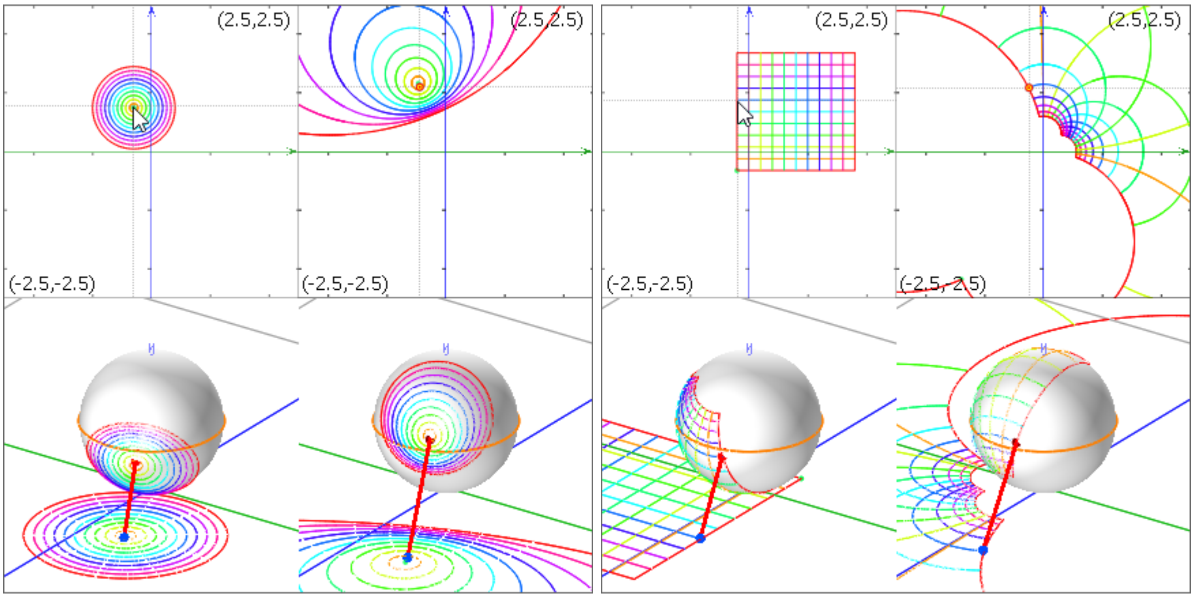


Figure 12

Comparison of the images of the left and right spheres leads us to a remarkable conjecture that the inversion mirrors the spherical image of the domain concerning the equator plane. To verify this, let us enable the "3D *front view*" menu option and start rotating the resulting scene around a vertical axis passing through the north and south poles using the arrows ←, → (Fig.13).

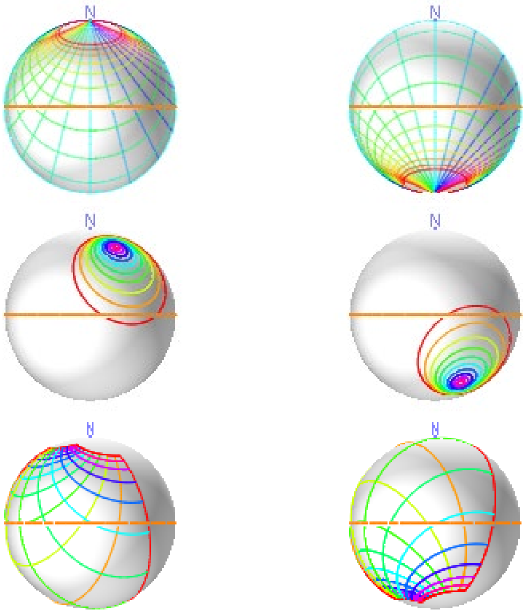


Figure 13

VisuMatica allows observing the common view by pressing the button "z & $f(z)$" pointed by the magenta arrow in Fig.1 c). Fig.14 presents the results, confirming our conjecture.

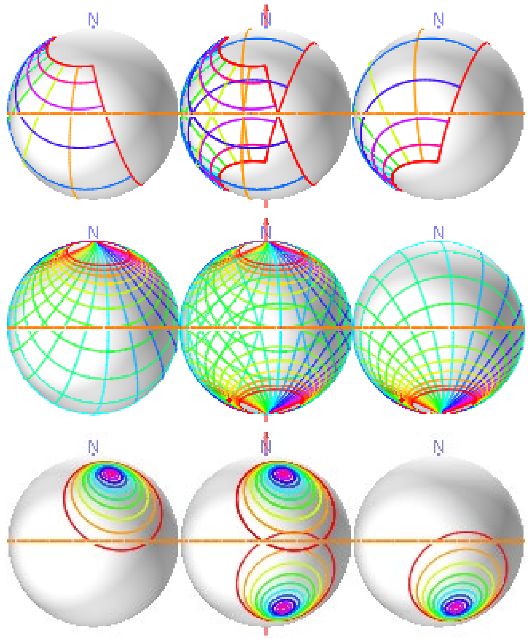


Figure 14

Now we can definitely call the inversion symmetry.
Let us prove this conjecture.
Since, during inversion, the correspondent points P_1 and P_2, which do not lie on the inversion circle[3], lie on a ray passing through zero, we consider a section of the 3D scene by a plane passing through the center of the sphere, the complex zero (aka the south pole) and points P_1 and P_2 (Fig.15).

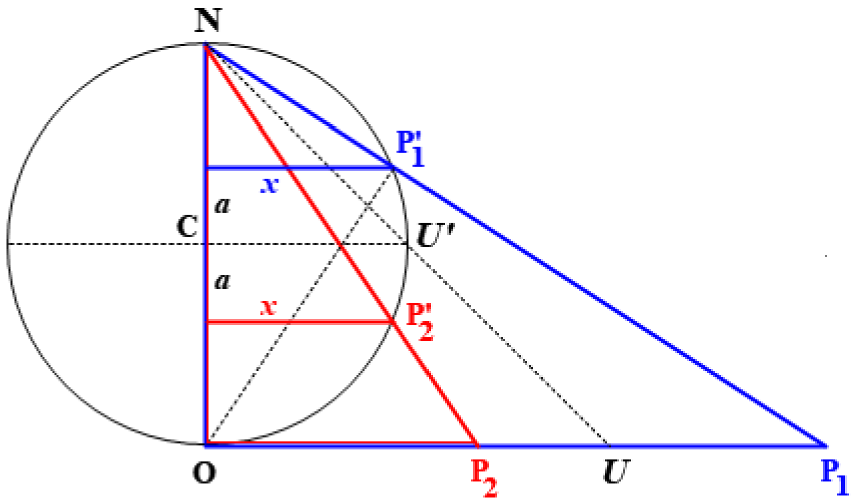


Figure 15

Let the points P_1' and P_2' - projections of points P_1 and P_2 onto the sphere - be symmetric about the equator plane (the horizontal diameter of the circumference of the sphere section). Then the corresponding blue and red perpendiculars dropped from these points to the diameter ON

[3] If the points belong to the circle, then they coincide and their image clearly lies on the equator.

are equal. Let us denote their length as x, their distances to the horizontal diameter as a, and the radius of the circle as R.

Considering pairs of similar red and blue triangles, we obtain the following equalities: $\frac{OP_2}{x} = \frac{2R}{R+a}$, $\frac{OP_1}{x} = \frac{2R}{R-a}$. Thus, $OP_2 = \frac{2R \cdot x}{R+a}$, $OP_1 = \frac{2R \cdot x}{R-a}$. The product of distances from points P_1 and P_2 to point O – the complex zero becomes $OP_2 \cdot OP_1 = \frac{4R^2x^2}{R^2 - a^2}$.

On the other hand, from a right triangle $\triangle NP_1'O$ we have: $x = \sqrt{(R-a)(R+a)}$ or $x^2 = R^2 - a^2$. Substituting this expression into the product formula, we have $OP_2 \cdot OP_1 = (2R)^2$.

Note that point U is a point on the circle of inversion since it is projected to a point on the equator U'. From the similarity of triangles $\triangle NCU'$ and $\triangle NOU$ we notice that the radius of the inversion circle is $OU = 2R$.

Thus, $OP_2 \cdot OP_1 = (2R)^2$ corresponds to the definition of inversion.

We have proved that points that are symmetric concerning the equator are the images of a point in the complex plane and its image under inversion. Since the mapping onto the Riemann sphere is one-to-one, the converse is also true: the images of the corresponding points P_1 and P_2 in inversion are symmetric on the sphere about the equator plane.

3. The Möbius transformation

The *Möbius transformation* of the complex plane is *a rational function* $\boldsymbol{M(z) = \frac{az+b}{cz+d}}$ with such complex coefficients *a, b, c,* and *d* that $ad - bc \neq 0$.

It has a lot of interesting features. We will consider just a few of them here.

When $\underline{c=0}$ we have $M(z) = \frac{a}{d}z + \frac{b}{d}$, i.e. a familiar linear transformation of the complex plane that can be expressed as a composition of rotation by the angle of $\arg\left(\frac{a}{d}\right)$, scaling by $\mathrm{mod}\left(\frac{a}{d}\right)$, and translation by $\frac{b}{d}$.

In the case of $\underline{c \neq 0}$ we have the following expression: [4]

$$M(z) = \frac{az+b}{cz+d} = \frac{bc-ad}{c^2} \cdot \frac{1}{z + \frac{d}{c}} + \frac{a}{c} .$$

[4] The reason for the condition $ad - bc \neq 0$ in the definition of the *Möbius transformation* becomes clear: in this case, all points of the plane are mapped to a single point $\frac{a}{d}$,- nothing interesting.

Thus, $M(z)$ can be presented as the following sequence of plane transformations:

$$f_1(z) = \boxed{z + \frac{d}{c}} \qquad\qquad t_1 : z \to z + m$$

$$f_2(z) = \frac{1}{\boxed{f_1(z)}} = \frac{1}{z + \frac{d}{c}} \qquad\qquad t_2 : z \to \frac{1}{z}$$

$$f_3(z) = \boxed{\frac{bc - ad}{c^2}} f_2(z) = \frac{bc - ad}{c^2} \cdot \frac{1}{z + \frac{d}{c}} \qquad\qquad t_3 : z \to kz$$

$$f_4(z) = f_3(z) \boxed{+ \frac{a}{c}} = \frac{bc - ad}{c^2} \cdot \frac{1}{z + \frac{d}{c}} + \frac{a}{c} \qquad\qquad t_4 : z \to z + n$$

Here $t_1(z)$ and $t_4(z)$ are translation transformations, $t_2(z)$ is a reciprocal transformation, and $t_3(z)$ is a composition of rotation and scaling transformations.

Every Möbius transformation can be performed in the following three steps [1], [2]:

1) Projection of the complex plane onto a sphere,
2) Rigid motion of the sphere,
3) Projection back from the sphere onto the plane.

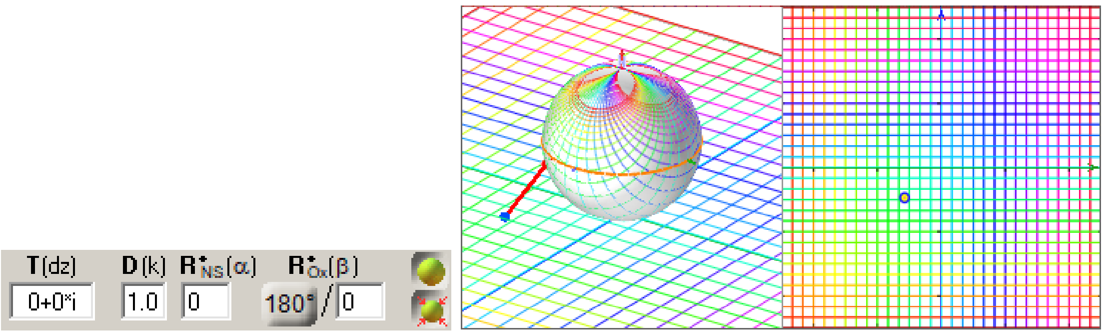
In particular, this motion of the sphere is a composition of the following transformations:

- Translation: $M(z) = z + b$, when $a = 1$, $c = 0$, $d = 1$
- Dilation (scaling): $M(z) = az$, when $a \in$ R and $b = 0$, $c = 0$, $d = 1$
- Rotation: $M(z) = az$, when $|a| = 1$ and $b = 0$, $c = 0$, $d = 1$
- Reciprocal transformation: $M(z) = \frac{1}{z}$, when $a = 1$, $b = 0$, $c = 1$, $d = 0$.

We are already familiar with the first three of them.

To see these transformations as a result of sphere transformations, we will use the sphere transformation mechanism of *VisuMatica*.

To start the exploration, define any complex function, say, $w = z$. Open the "3D *view*" in the "*Mapping* 2D" dialog and select the "*Stereographic Projection*". Click on the "*Simulation*" tab. Pressing the "*Riemann sphere transformation*" button  results appearance of the interface for sphere transformation with proper two bottom views.



a) b)

Figure 16

Fig.16 a) displays the interface for controlling the sphere movement. Initially, the sphere is located at the complex zero. The text boxes allow to enter definite values of T – translation by some complex number *dz*, D – dilation with the coefficient *k*, interpreted as the sphere lifting or dropping by *k*, R_{NS} – the rotation angle in degrees around the sphere's *z*-axis (North-South axis), and R_{Ox} the rotation angle in degrees around the sphere's *x*(*real*)-axis[5].
Fig.16 b) shows the initial state of the scene: the left half - the domain's regular colored grid from the upper left quarter with its projection onto the sphere, and the right half - the colored grid on the plane under the sphere as a result of a projection of the sphere's grid[6] according to the default parameters values in Fig.16 a). Nothing happened.
It is important to note that both of these images in Fig.16 b) are in no way related to the function *w=f*(*z*) itself, and thus, the top-right quarter of the screen that depicts its Range.
Change the values in the left three input boxes of the interface and observe the changes in the sphere's grid projection onto the plane in the right half of Fig.16 b). Use parametric definitions, say, *a*+*i***b*, *c*, and *d,* and play with the values of these parameters.

- Are the results of these changes consistent with the proposed transformations of the complex plane?

Add two complex variables, say, $aa := a + i \cdot b$ and $bb := c + i \cdot d$.

Redefine expressions: T as *bb*, D as |*aa*|, R as argd(*aa*)[7], and the function as $w = \dfrac{aa \cdot z + bb}{0 \cdot z + 1}$.

Change the *a*, *b*, *c*, and *d* values (use the slider and animation buttons). Compare the Range (the upper-right view) and the result of the projection of the transformed sphere onto the complex plane (the lower right).

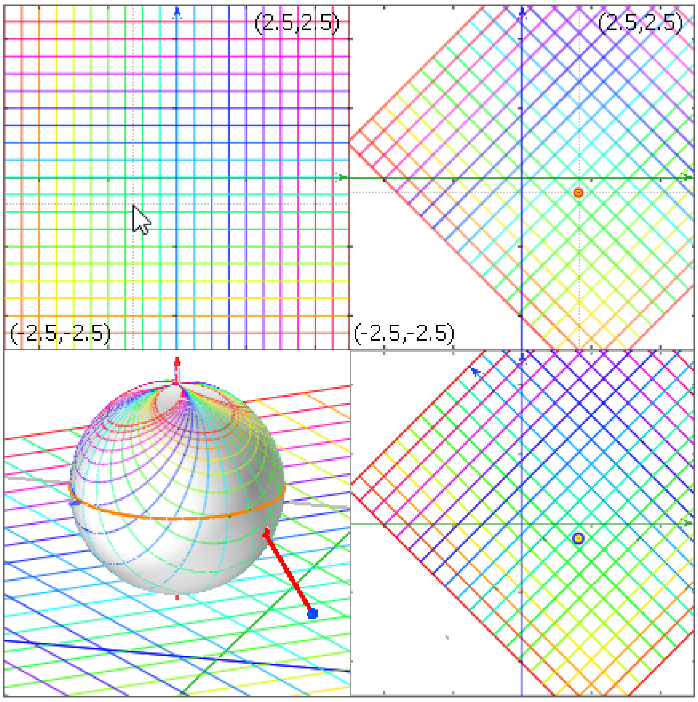
- Conclude.

Fig.17 a) presents the *VisuMatica* display that corresponds to the concrete values of parameters (Fig.17 b). The 3D view here was zoomed in to make clear the correspondence of grids' on the sphere and its projection onto the plane.
As always, with the help of the keyboard keys ←, ↑, ↓, →, we can rotate the three-dimensional scene, and with the / button - select the most convenient view. Fig.18 shows the situation when the button is pressed (left) and released (right). Accordingly, when the scene rotates, the center of the sphere or the center of the grey boundary rectangle will remain stationary.
Of course, in all four views you can see the synchronized position of the moving mouse: as always, the mouse pointer (arrow) in the Domain view, the red-yellow-red point in the Range

[5] The button determines the position of the sphere concerning the plane of projection. By default, this button is pressed - the sphere's center lies on the vertical center line of the view, and the projection plane is moving. In the released state of this button , the plane is motionless, and the sphere itself moves. Of course, the result is the same. The presence of these options allows us to choose the most convenient visualization of the result. To achieve this goal, the ability to zoom in / out of the scene (*Ctrl* + *V* / *Ctrl* + *Shift* + *V*) helps a lot.

[6] The right image presents the result of this projection on the plane within the Domain's boundaries.
[7] The function argd (*z*) returns the argument value of the complex number *z* in degrees.

view, the big red point on the sphere with its projection – the blue point on the projection plane, and the blue-yellow-red point in the 2D view of the projection's result. Both right mouse images are in the same position.



a)

name	expres...	value
a		1
b		0.5
c	1/sqr(2)	0.70711
d	1/sqr(2)	0.70711
cb	a+i*b	1+0.5*i
ca	c+i*d	0.707107+0.707107*i

b)

Figure 17

One can add points, lines, and circles to the model, move them and become sure that their images in all four views remain consistent: lines are displayed as lines (they are circles on the sphere), and circles as circles. Herewith, the angles between the lines and circles are preserved.

We got a general model of Möbius transformation in the case of $c = 0$ [8] – the linear transformation $M(z) = aa \cdot z + bb$ and confirmed that it is equivalent to a proper composition of translation, dilation, and rotation of the sphere around its North-South (z-axis), followed by the projection onto the complex plane.

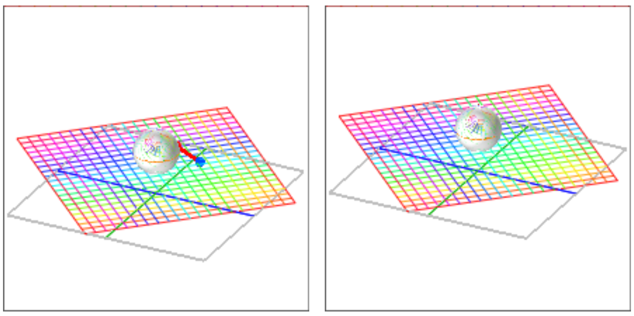


Figure 18

Consider the remaining transformation that is part of the composition of the mentioned motions of the sphere.

[8] For convenience, without loss of generality, $d = 1$.

3.1. The Reciprocal transformation

Reciprocal transformation is defined by the complex function $\boldsymbol{w} = \dfrac{1}{\boldsymbol{z}}$.

Considering that $\dfrac{1}{z} = \overline{\left(\dfrac{1}{\bar{z}}\right)}$ and $\dfrac{1}{\bar{z}} = \overline{\left(\dfrac{1}{z}\right)}$, and the fact that the complex numbers z and z^* are represented by points symmetric about the real axis, we can guess that many of the known properties of the inversion are also valid in the case of reciprocal transformation.

- Which of the pictures in Fig.19 depicts the inversion transformation, and which is the reciprocal one?

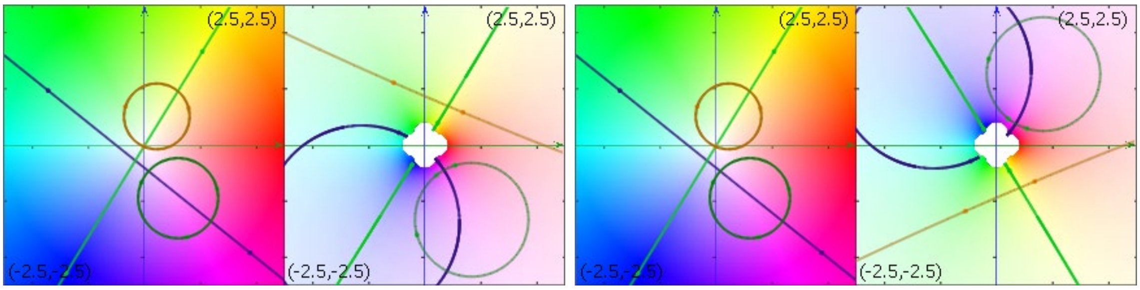


Figure 19

- What properties of the inversion seem to be confirmed in the case of the reciprocal transformation, judging by this picture?
- Let us find out which motion of the sphere, if any, provides this transformation.
- Set "*Mapping style*" to "*empty*", enable "*show zone mapping*", and in the "*zone boundaries*" tab, set the "*circular*" zone to "*circles*" only.
- Set the "3D *view*" to "*stereographic projection*". Press the "*z & f(z)*" button to enable the common stereographic view.
- Select the "3D *right view*" option in the "3D *views…*" menu.

As a result of this tuning the *x*(*real*)-axis (green dot in the center) "looks" at us (Fig.20).
Drag the zone in the Domain view and observe the Domain grid's projection onto the left sphere, its Range image projection onto the right sphere, and both grid projections onto the central sphere.

Replace the zone show from "*circular*" to "*rectangular*" and repeat dragging.

Summarizing these observations, we conclude that *the spherical mechanism of the reciprocal function is the rotation around the x-axis by 180° of the projection of the domain onto the sphere*.

Create a new model of the reciprocal function to check this conclusion visually.
The model includes line QO and a circle with a center in point P that passes via point O. First, note that the image sphere looks inverted relating to the original one!
Move points P and/or Q and ensure that images of line and circle remain lines (Fig.21 a).

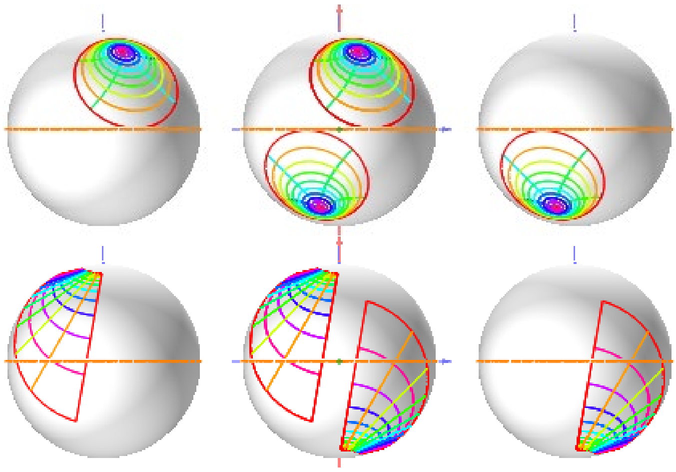
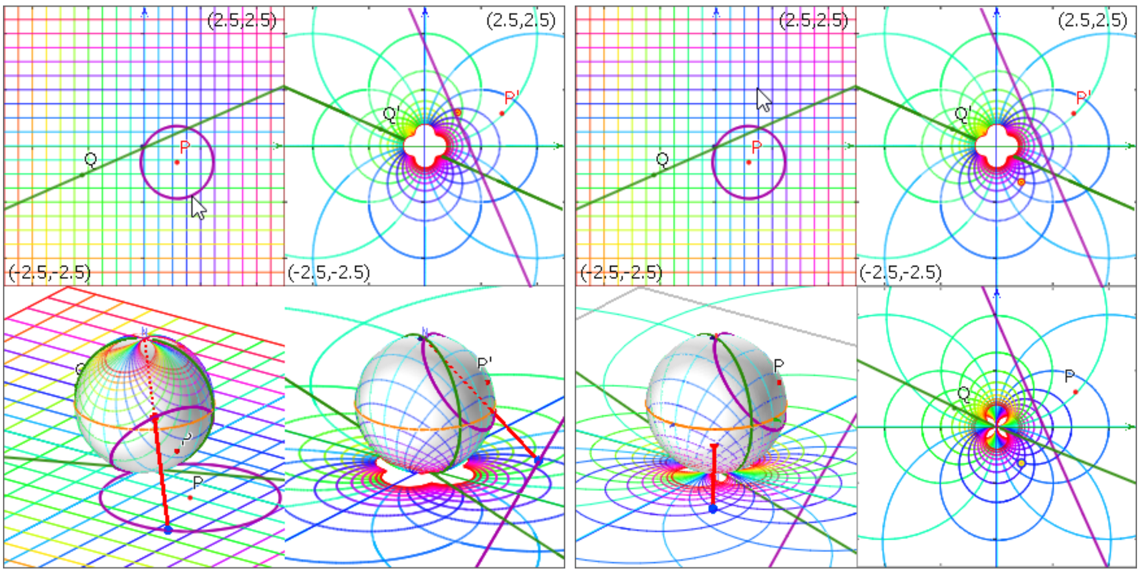


Figure 20

We know that the circle passing through the north pole corresponds to a straight line on the plane. The circles passing through the south pole of the sphere did not differ in anything special: they correspond to images-circles on the plane (the left half of Fig. 101 a).

Let us use the sphere transformation mechanism (Fig.16 a) to find out why their images in the Range view and on the projection onto the plane from the right sphere are straight lines. Press the "180°" button. As result, the value of the rotation angle of the sphere around the *x*(*real*)-axis next to the button will be replaced by 180° and the sphere will "turn over": its north pole will become south, and the south, on the contrary, will become the north pole (Fig. 101 b).

- Compare images in all quarters of the images a) and b) in Fig.21 and confirm our conclusion about the relation between the sphere rotation and reciprocal transformation.



a) b)

Figure 21

- Explain, why the line and circle images are straight lines.

Let us return to the Möbius transformation and consider the case where $c = 1$ and $d = 0$.

Open the model used in studies of *Mobius linear* $w = \dfrac{aa \cdot z + bb}{0 \cdot z + 1}$ and add a new function $w = \dfrac{aa \cdot z + bb}{\mathbf{1 \cdot z + 0}}$. The image in the Range view changes drastically.

Turn on the mechanism for modeling the movement of the sphere, change expressions of the first three parameters expressions to T as *aa*, D as |*bb*|, and R as argd(*bb*) accordingly to the discovered four-steps sequence of sphere movements' parameters with $c = 1$ and $d = 0$. Add a rotation around the *x*(*real*)-axis by 180°.

It worked! The <u>*composition of the transformations of the sphere*</u> with the given parameters leads to <u>*the same result as the Möbius function, given by the new expression*</u> (Fig.22).

- Change the values of parameters to confirm the coincidence of the right-hand images.

Unfortunately, our mechanism for modeling the sphere movement has a serious limitation. It does not support the possibility of repeated shifting, as is required in the general Möbius transformation, and even more so, the use of other types of motions and their compositions to study a wider range of complex plane transformations.

The built-in functions of *VisuMatica* help to solve this problem. They allow us to construct any composition of the following transformations:

- *RotateXd180*(*z*) – performs the rotation transformation of complex *z* by an angle of 180° around the *x*(*real*)-axis (*RXd180* (*z*) – a shorter version) [9],
- *RotateXd*(*z, angle*) – performs a rotation transformation of complex *z* by *angle* degrees around the *x*(*real*)-axis (*RXd* (*z, angle*) – a shorter version),
- *RotateXr*(*z, angle*) – performs a rotation transformation of complex *z* by *angle* radian around the *x*(*real*)-axis (*RXr* (*z, angle*) – a shorter version),
- *RotateYd*(*z, angle*) – performs a rotation transformation of complex *z* by *angle* degrees around the *y*(*imaginary*)-axis (*RYd* (*z, angle*) – a shorter version),
- *RotateYr*(*z, angle*) – performs a rotation transformation of complex *z* by *angle* radian around the *y*(*imaginary*)-axis (*RYr* (*z, angle*) – a shorter version),
- *RotateZd*(*z, angle*) – performs a rotation transformation of complex *z* by *angle* degrees around the *z*-axis (*RZd* (*z, angle*) – a shorter version),
- *RotateZr*(*z, angle*) – performs a rotation transformation of complex *z* by *angle* radian around the *z*-axis (*RZr* (*z, angle*) – a shorter version),
- *SymmetryZ0*(*z*) – performs the mirror symmetry transformation of complex *z* in the plane O*xy* (*SZ0*(*z*), *MirrorZ0*(*z*), *MZ0*(*z*) – other versions),
- *Dilate*(*z, k*) – performs the dilate transformation of complex *z* with a ratio of real *k* (*Dl*(*z, k*) – a shorter version),
- *Translate*(*z, c*) – performs the translation transformation of complex *z* by complex *c* (*Tr*(*z, c*) – a shorter version).

[9] All rotations in the list are counterclockwise when viewed from the positive direction of the axis.

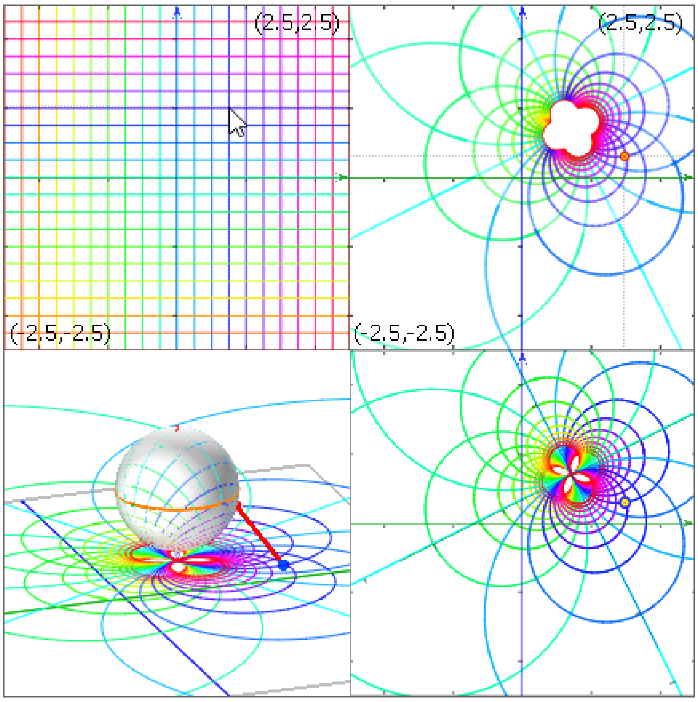


Figure 22

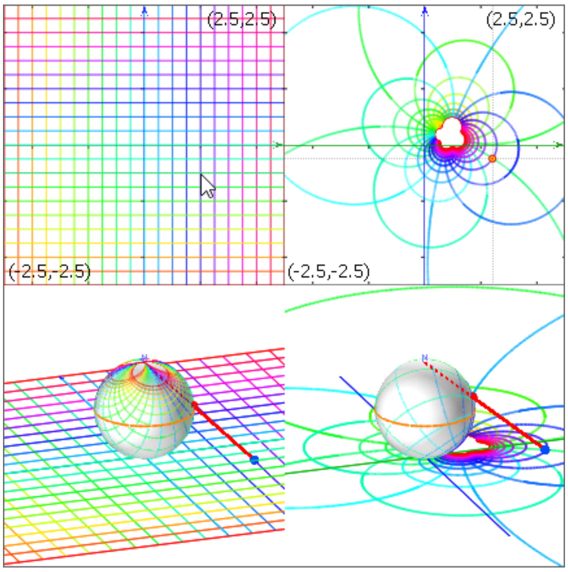
Let us use them to construct a general Möbius transformation model. To this end: add four real variables k, l, m, n, and two complex variables cc and dd ($cc:=k+i*l$, $dd:=m+i*n$) to the previous model and redefine the function using the expression

$$\boldsymbol{w}=Tr(Dl(RZd(RXd(Tr(\boldsymbol{z},dd/cc),180),argd(mult)),|mult|),aa/cc).$$

Add a new function by the general Möbius transformation expression

$$w=\frac{aa\cdot z+bb}{cc\cdot z+dd}.$$

As was expected, you got *the same result* (Fig.23).



name	expression	value
a	1/sqr(2)	0.70711
b	1/sqr(2)	0.70711
c		1
d		0.5
aa	a+i*b	0.707107+0.707107*i
bb	c+i*d	1+0.5*i
k		2
l		1
m		-1
n		1
cc	k+i*l	2+1*i
dd	m+i*n	-1+1*i
mult	(bb*cc-aa*dd)/cc^2	0.669706-0.226274*i

Figure 23

- Play around with parameters' values, keeping $cc \neq 0$, switch between showing f_1 and f_2 using the legend icons, and thus make sure the two functions are identical.

Add five additional functions to the model to present the separated "steps" of the composition $f_3, \ldots, f_7$ as follows:

Step	Function-Transformation	Legend name
1	***w***=*Tr*(***z***,*dd/cc*)	f_3
2	***w***=*RXd*(***f₃***(***z***),*180*)	f_4
3	***w***=*RZd*(***f₄***(***z***),*argd*(*mult*))	f_5
4	***w***=*Dl*(***f₅***(***z***),|*mult*|)	f_6
5	***w***=*Tr*(***f₆***(***z***),*aa/cc*)	f_7

Press sequentially on the legend's icons $f_3 \to f_4 \to f_5 \to f_6 \to f_7$ to see the "process" of transformations with the outcome – the general Möbius transformation

$$w = f_7(f_6(f_5(f_4(f_3(z))))) = \frac{aa \cdot z + bb}{cc \cdot z + dd}.$$

Change the values of concrete parameters to see their implication to the definite step of composition and the common result. Ensure that the outcomes of f_7 and f_2 are the same.

- Can it be argued that the Möbius transformation is conformal, preserves angles, and maps lines and circles to lines and/or circles?
  Justify your answer if it is "*Yes*" or give a counterexample if the answer is "*No*".

Note that with the help of the mentioned transformations and their compositions, it is possible to define a much larger range of plane transformations. The analytical representation of such transformations is often a complicated mathematical problem.

- Name the function $w = SymmetryZ0(z)$. Express it analytically.

- What transformation presents the function: *w=Tr*(*Dl*(*SZ0*(*Tr*(*z,-*(*1+2*i*)),*4*),*1+2*i*)? Express it analytically.

4. Remarks and Conclusions

A well-known example of modeling the Möbius transformation [1] in the video created by the authors [3] has several disadvantages:

1. It does not allow you to "manage" the demonstration due to the very nature of the films. In this case, the show can only be stopped, slowed down, or accelerated.

2. Does not reveal the analytical connection between the movements of the sphere and the function itself.

3. The Ray Tracing algorithm for rendering a 3D scene used to create the film allows us to get perspective images of real objects of extremely high quality, but a straight line with changing "thickness" is not entirely correct.

Other models, such as [4], also do not present any analytics and have limited modeling capabilities.

The proposed support for the study of plane transformations with the help of *VisuMatica* is free from the mentioned shortcomings and allows the construction of the entire range of necessary models and their explorations in the educational process. There are two short videos to illustrate these transformations.

[1] http://atcm.mathandtech.org/EP2022/invited/21955/Riemann.mp4.

[2] http://atcm.mathandtech.org/EP2022/invited/21955/Transform.mp4.

5. References

[1] Arnold Douglas N., Rogness Jonathan. Möbius Transformations Revealed. In: Notices Of The American Mathematical Society, 2008, Volume 55, Number 10

[2] Siliciano Rob. Constructing Mobius Transformations with Spheres. In: Rose-Hulman Undergraduate Mathematics Journal, 2012, Vol. 13, Iss. 2, Article 8

[3] https://www.youtube.com/watch?v=0z1fIsUNhO4

[4] https://www.geogebra.org/m/GhaSJw3t

[5] Needham Tristan. Visual Complex Analysis. 1999, Oxford University Press

[6] Nodelman Vladimir. Nonlinear Space Transformations and Educational Software. In: Research Jornal of Mathematics & Technology, 2015, Vol. 4, №1