

A GRAPHICAL INTRODUCTION TO TRANSFORMATIONS IN THE REAL PLANE

NICHOLAS BOYDSTON

1. INTRODUCTION

The purpose of this paper is to introduce the reader to the topic of transformations via a companion Java application, called the Java Transformation Viewer (JTV). In general a transformation is a mapping from one set to another. This paper specifically focuses on transformations from \mathbb{R}^2 to \mathbb{R}^2 , that is, within the Euclidean plane. As we proceed we will use the JTV to illustrate transformations and topics related to them, specifically linear, affine, coordinate system, and conformal transformations and their properties. While our examples are very specific, the reader is encouraged to play with the JTV.

Figure 1 is an example of what the JTV looks like. There are a number of examples discussed in this paper. They are meant to be read while the reader has access to a running version of the JTV. They should be entered into the JTV as they are read. The main JTV window has several portions, which we now describe.

1.1. The Drawing Area. The most significant feature of the JTV is the pair of areas with x and y axes. In Figure 1 these areas have ellipses drawn in them. These are the portions of \mathbb{R}^2 that represent the domain and codomain of the transformation. The area on the left is directly editable, that is, the user can draw on it. It is called the drawing area. The controls above the drawing area are used to help draw figures. The five round check boxes at the left of the controls are used to specify what sort of drawing the mouse will do. The operations are described here in terms of dragging on the drawing area. In order to drag; click and hold the left mouse button while moving the mouse over the editable area. For most of these options only the start and end points of the

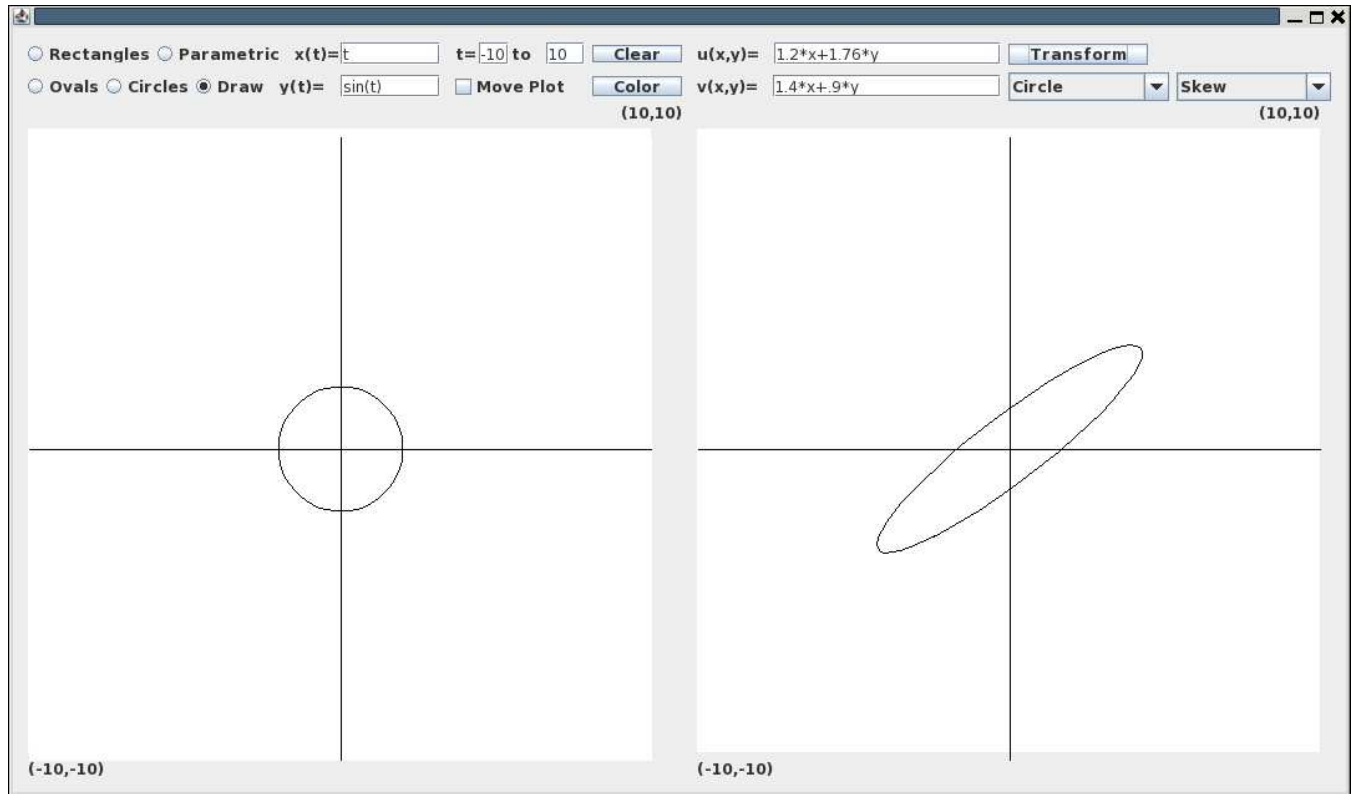


FIGURE 1. A screen-shot of the Java Translation Viewer showing a the linear transformation $u=1.2*x+1.76*y$, $v=1.4*x+.9*y$ applied to a circle and resulting in an ellipse.

drag are important. The right hand area displays the result of the transformation on the contents of the drawing area.

The **Draw** option allows you to draw freely with the mouse pointer. When you drag across the editable area it leaves behind a line. This is the default type of drawing.

Rectangles will draw rectangles with corners at the start and end points.

The **Ovals** option will draw an oval. The start and end points of the drag define a rectangle. The oval is the largest oval that can be inscribed in that rectangle.

To use the **Circles** option, you click at the center of the circle and then drags the radius out to where you want it.

The final option, **Parametric**, is slightly more complicated. To use it, you fill the $x(t)=$ and $y(t)=$ boxes with the parametric form of the curve you wish to draw. Fill in the $t=_{_}to_{_}$ fields with the range of t .

The JTV uses an external library, specifically the JEPlite parser[7], to evaluate the user-defined functions. Standard operators like addition, subtraction and multiplication all work. Several of other functions are also available. Exponentiation is indicated with the carrot symbol (^). The list of available functions is available in Appendix A. This list is the same as the list of available functions for the transformations.

To draw the specified parametric curve on the area simply click anywhere on the area, after entering the defining expressions. When the `Move Plot` box is checked you can offset the plot of the curve by dragging it.

Notice that there are ordered pairs of numbers at two corners of each area. These pairs of numbers represent the coordinates of the corners. Knowing where the corners of the visible area are should help you to fit the parametric curve to the available space.

Exercise 1.1. *Use the parametric method to draw a spiral around the origin. Hint: the sine and cosine functions will be useful.*

The **Clear** button clears the drawing area. If you have drawn something that you no longer want, then the clear button will remove it, along with everything else in the drawing area. The **Color** button allows you to select the color with which you wish to draw.

Exercise 1.2. *Draw something with different colors. Can you get a red spiral going one direction and a green one going the other?*

1.2. The Transformation Area. The area on the right is the transformation area. This area is not directly editable. The transformation area shows the result of the current transformation applied to the figure in the drawing area.

The controls for the transformation are rather simple. The transformation is entered as two coordinate-functions, $u(x, y)$ and $v(x, y)$. For example to shift the figure in the drawing area 1 unit to the right you would set $u = x + 1$ and $v = y$. Try some simple transformations on a figure.

When you click the **transform** button the program will calculate the transformation of the figure in the drawing area, and display it in the transformation area.

The final two controls are collectively called “the zoo”. They contain a number of preset examples. The one on the left is a collection of preset images. Selecting one of these will cause the contents of the drawing area to be replaced by the preset image you selected. The one on the right is a collection of example transformations. Selecting one of them will change the transformation functions. The list of these options and descriptions of each is available in Appendix B.

Another feature of the JTV is cursor tracking. There is a red circle that indicates where the cursor’s location in the drawing area is mapped to by the current transformation.

Note that, as the mouse traces a figure in the drawing area, the tracking red circle will trace the transformation of that figure in the transformation area. This illustrates which part of the figure in the transformation corresponds to the various parts of the figure in the drawing area.

Exercise 1.3. *Run the program and try several of the preset transformations on the preset figures. Create a custom transformation. Observe the behavior of the red circle under various transformations.*

Now that we have introduced the JTV we will use it to examine transformations more thoroughly. We examine four particular types/uses of transformations: linear transformations, affine transformations, coordinate system changes, and conformal mappings. We will begin by defining transformations.

2. TRANSFORMATIONS

We will be visualizing transformations as changes in the shape of a plane figure. The JTV will help us to visualize and understand these plane transformations.

We begin with a definition of planar transformations.

Definition 2.1. *A transformation from \mathbb{R}^2 to \mathbb{R}^2 is simply a function, denoted T that maps a subset D of \mathbb{R}^2 to \mathbb{R}^2 . That is $T : D \rightarrow \mathbb{R}^2$, where $D \subseteq \mathbb{R}^2$*

Everything in this paper will be in reference to planar transformations, but remember that transformations in different spaces exist.

We will be using function notation for transformations. We have two ways to do this: coordinate notation and vector notation. In coordinate notation the function $u(x, y)$ is the x coordinate of the transformed point, and $v(x, y)$ is the y coordinate. When using vector notation, we say $T(\vec{x}) = \vec{b}$ where $\vec{x} \in D$, $\vec{b} \in \mathbb{R}^2$. This translates to coordinate notation as $\vec{x} = (x, y)$ and $\vec{b} = T(\vec{x}) = (u(x, y), v(x, y))$. Thus the untransformed point is \vec{x} and the transformed point is $T(\vec{x})$.

As you can see from the definition, the class of general transformations is very broad. We will be working with transformations that are fairly “nice,” i.e. that can be represented by algebraic expressions. This is partially because they are easier to conceptualize and mostly because the JTV software can only handle algebraic expressions.

One particularly important, though trivial, transformation is the identity transformation. The identity transformation maps every point to itself, that is $u(x, y) = x$, and $v(x, y) = y$. In vector notation it is represented by $I(\vec{x}) = \vec{x}$.

Most of the transformations that we examine will be piecewise differentiable algebraic expressions. It is important to note that, because computers are only capable of discrete calculations, the program assumes that the figures it is drawing can be approximated locally by straight lines. In fact, behind the scenes it approximates everything as straight lines. Although for some of the drawing area

shapes, the approximation is at a library level rather than an application level. The rectangles, ovals and circles are stored as position data in the drawing area and rendered by library routines that draw the appropriate figure. The freehand drawing is stored as a large number of small line segments. You can see this if you use the freehand drawing tool, hold down the mouse button and drag the cursor around the drawing area as quickly as you can. Some of the resulting figure is made up of fairly long line segments rather than a smooth curve where your cursor passed. The parametric portion of a figure is stored as a text function and a position offset. The transformed figure is always stored as a collection of short line segments. To generate this collection of segments the program first samples points from figures in the drawing area and then transforms them.

This local linearity assumption is not true when the transformation function is not differentiable, and thus we use functions that are differentiable almost everywhere.

One property that continuity, and thus local linearity, gives us is related to the connectedness of a region.

A connected region is any set A , such that if B is a subset of A and B is both open and closed in A then either $B = A$ or B is the null set.

A path connected region is a region E such that for any two points a, b in E there exists a continuous function $f : [0, 1] \rightarrow E$ such that $f(0) = a$ and $f(1) = b$. [8] Any path connected region is connected.

With the continuity and differentiability constraints we know

Theorem 2.2. *When limited to continuous transformations, a path connected region in \mathbb{R}^2 will be mapped to another path connected region in \mathbb{R}^2 .*

Proof: To show this consider any two points that are in a path connected region, then there exists some continuous curve, $\vec{x} = \vec{f}(t), t \in [0, 1]$, in the region connecting those two points. If the transformed region is not connected then there will exist at least one pair of points such that the

transformation of every curve between them is discontinuous. If the transformation of some curve, $\vec{T}(\vec{x}) = \vec{T}(\vec{f}(t))$ for $t \in [0, 1]$, is discontinuous then there exists a point t' such that $\lim_{t \rightarrow t'} \vec{T}(\vec{f}(t)) \neq \vec{T}(\lim_{t \rightarrow t'} \vec{f}(t)) = \vec{T}(\vec{f}(t'))$. Thus T is not continuous. If the two sided limit does not exist then the function is also discontinuous. By contrapositive we can see that if T is continuous then connected regions will be mapped to connected regions.

There is a similar result for connected regions. Any continuous function maps connected sets to connected sets[9].

Thus if you have some simple closed figure in the plane and a continuous transformation, the figure may be deformed, stretched and twisted by the transformation, but it will remain just one shape. For example a circle will change to a single shape no matter how deformed.

Some transformations are defined in terms of what aspects of a figure are invariant when they are applied and how those aspects that are not invariant are changed. For example, a rotation preserves distance between points, as does a translation. As we just saw, transformations from continuous functions preserve path connectedness.

Discontinuous transformations usually give us two types of discontinuities: *jump* and *singularity*. Jump discontinuities create places where a figure is sheared into two parts. For example the transformation $u=(\text{abs}(x)+2)(x/\text{abs}(x))$, $v=y$ splits any figure that crosses the y axis into two parts. This function is discontinuous at $x = 0$. One of the difficulties with the JTV is that it cannot detect jump discontinuities, and will thus fail to disconnect lines that cross a jump discontinuity.

Another other type of discontinuity is a singularity. With this type, there exists a point where the transformation is not continuous, but the transformation is continuous all around the point. An example of this is $u=x/(x^2+y^2)$, $v=y/(x^2+y^2)$. This transformation is discontinuous at $(x, y) = (0, 0)$, but continuous everywhere around it. Unlike a jump discontinuity this does not divide the plane into sections. That is, if this is the only discontinuity then, if D is any arbitrarily small disk around the point of discontinuity then there exists a continuous path $f(t) : [0, 1] \rightarrow D$

between any two other points in any two points in D such that $T(f(t))$ is also a continuous path.

More plainly, you can always draw a line around it that is unbroken in the transformation.

It is useful to define the inverse of a transformation when it exists. A transformation will have an inverse if and only if it is one-to-one and onto.

Definition 2.3. *Let $T : D \rightarrow \mathbb{R}^2$ be a transformation. If there exists a function $T' : D \rightarrow \mathbb{R}^2$ such that $T(T^{-1}(\vec{x})) = T^{-1}(T(\vec{x})) = \vec{x}$ for all \vec{x} in D , then T^{-1} is called the inverse of T .*

That is the composition of a transformation and its inverse, in any order, is the identity transformation. Not all transformations have inverses. For example, the transformation $u(x, y) = x^2$, $v(x, y) = y$ in the real plane has no inverse because it is not one-to-one, and it is not onto.

It is difficult to make specific statements about general transformations. For this reason we will be examining several families of transformations: linear transformations, affine transformations, coordinate system changes, and conformal mappings.

You may also wish to use the JTV to get an intuitive feel for a transformation. A good drawing area figure for this purpose is often a grid. This is available from the collection of figures under the label **Grid 2**. Pull this figure up and be ready to use it with the examples in the following paragraphs.

2.1. Linear Transformations. Linear transformations include stretches, scaling and rotations. They do not include translation, that is, shifting a figure in one direction by a constant amount. Under a linear transformation the origin of the plane always maps to the origin. The transformation collection contains an example linear transformation called **Skew**.

2.2. Affine Transformations. Affine transformations are very similar to linear transformations, but affine transformations allow translations. Also, the representation of affine transformations is more complicated than that of linear transformations. The transformation collection does not have

an affine transformation, but you can get one by adding “+10” to the end of the $v(x, y) =$ function in the **skew** transformation.

2.3. Coordinate System Changes. Coordinate system changes represent a very broad group of transformations. They are functions for taking a point in one coordinate system and finding its representation in another. There are few general results to be stated regarding them. Instead we will be looking at a variety of examples of coordinate changes. Two types are particularly interesting, invertible changes of coordinate system and projections. The transformation collection has two change of coordinates examples, **Polar**, and **Inverse Polar**. Despite their names, they are not quite true inverses of each other.

2.4. Conformal Mappings. Conformal mappings are a class of transformations that have an unusual invariant. Some transformations preserve geometric properties such as length or area. Conformal mappings preserve angles of intersection of lines. Straight lines are not necessarily mapped to straight lines, but wherever two lines intersect they intersect at the same angle in the original figure and the transformed figure. The collection has an example of a conformal mapping, labeled **Conformal**. As an example you should try the **Conformal** transformation with the **Grid 2** figure.

Having finished this brief overview of transformations we will now examine these specific families of transformations in more detail. The reader is encouraged to have the JTV available.

3. LINEAR TRANSFORMATIONS

Linear Transformations are the simplest type of transformation.

Definition 3.1. *A linear transformation is a transformation $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ such that $T(\vec{u} + \vec{v}) = T(\vec{u}) + T(\vec{v})$ and $T(c\vec{u}) = cT(\vec{u})$ where \vec{u} and \vec{v} are in the domain of T , and c is a scalar.*

All linear transformations may be represented by matrix multiplication of the vectors in the domain. Proof of this can be found in Lay's Linear Algebra book[2]. Consider a matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

The associated transformation is $A([x, y]^T)$, or $u(x, y) = a \cdot x + b \cdot y$, $v(x, y) = c \cdot x + d \cdot y$, from the definition of matrix vector multiplication.

Example 3.2. Find the transformation associated with a matrix

$$A = \begin{bmatrix} 4 & 3 \\ 2 & 7 \end{bmatrix}.$$

From our previous observation we enter $u=4*x+3*y$, and $v=2*x+7*y$.

3.1. Eigenvectors and Eigenvalues.

The aspects of the matrix representation of a linear transformation can give us useful information about the properties of the transformation itself. Foremost among these aspects are the eigenvalues and eigenvectors of the matrix.

Definition 3.3. An **eigenvector** of an $n \times n$ matrix A is a non-zero vector \vec{x} such that $A\vec{x} = \lambda \cdot \vec{x}$ for some scalar (i.e. real or complex number) λ . The scalar λ is then called an **eigenvalue** of A and \vec{x} is called an eigenvector corresponding to λ . The eigenvector corresponding to an eigenvalue is not unique.

An important property of the determinant of a matrix is that, if the determinant of a matrix B is zero, then $B\vec{x} = \vec{0}$ has a solution. If $\det(A - \lambda I_n) = 0$ then λ is an eigenvalue. This equation is called the characteristic equation, and it is an n^{th} degree polynomial in λ .

Given some non-zero \vec{x} in \mathbb{R}^n and an $n \times n$ matrix A , the vector \vec{x} is an eigenvector of A if and only if there exists some scalar λ such that $A\vec{x} = \lambda\vec{x} = \lambda I_n \vec{x}$, where I_n is the $n \times n$ identity matrix. This condition can be rearranged to $(A - \lambda I_n)\vec{x} = \vec{0}$. Thus we can also say that λ is an eigenvalue of A if and only if $(A - \lambda I_n)\vec{x} = \vec{0}$ has at least one nontrivial solution.

A useful result from Lay[2] is that the eigenvalues of a triangular matrix, that is a matrix with all zero entries on one side of the main diagonal, are the entries on its main diagonal.¹ This fact combined with rules for how the determinant changes under row operations make finding eigenvalues much easier than using the determinant directly to find the characteristic equation.

Example 3.4. Let $A = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix}$. With this A , we can show that $(A - \lambda I_n)\vec{x} = \vec{0}$ has nontrivial solutions for $\lambda = 1$ and $\lambda = 2$. Two independent eigenvectors are $[1, 0]^T$, and $[1, \frac{1}{3}]^T$ respectively.

The solutions to the example can be found from the characteristic polynomial. The characteristic polynomial for a matrix can be found by setting $\det(A - \lambda I_n) = 0$, and solving for λ . The characteristic polynomial for this example is $(1 - \lambda)(2 - \lambda) = 0$

Exercise 3.5. Try finding the eigenvalues of several 2×2 matrices. Also, try transforming figures with the above linear transformation A , as well as some of your own. Pay close attention to how the different transformations alter figures. Some example 2×2 matrices are

$$\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 4 \end{bmatrix}, \begin{bmatrix} 6 & 3 \\ -2 & 4 \end{bmatrix}.$$

3.2. Symmetric Matrices.

There is a connection between the eigenvectors and eigenvalues of a matrix and the geometry of the transformation it represents.

¹The proof of this is in Lay page 306.

Definition 3.6. Let A be an $m \times n$ matrix. Then A^T is the $n \times m$ matrix whose rows are the columns of A . This is called the **transpose** of A .

The transpose of a square matrix is a reflection of that matrix about its main diagonal.

Definition 3.7. Let A be a square $n \times n$ matrix. Let a_{ij} refer to the element in the i^{th} row and the j^{th} column of A .

A matrix A is symmetric if and only if, for all i and j in $1, 2, \dots, n$, it is true that $a_{ij} = a_{ji}$.

Thus a matrix is symmetric if and only if it is equal to its transpose. An example of a symmetric 4×4 matrix is:

$$\begin{bmatrix} 1 & 3 & 6 & 5 \\ 3 & 3 & 4 & 2 \\ 6 & 4 & 7 & 4 \\ 5 & 2 & 4 & 5 \end{bmatrix}$$

If A is a symmetric matrix and T is a linear transformation based on A , then the action of T is to stretch figures along the eigenvectors of the matrix by a factor of the associated eigenvalue. Also, symmetric matrices have orthogonal eigenvectors. That is, the eigenvectors of symmetric matrices are at right angles to each other.

We can use the orthogonality and multiplicative properties of the eigenvectors of symmetric matrices to demonstrate the fact that transforming a figure by a symmetric matrix stretches the figure along the eigenvectors of the matrix by an amount proportional to the associated eigenvalues. Given any symmetric 2×2 matrix A , let λ_1 and λ_2 be its eigenvalues and \vec{v}_1 and \vec{v}_2 be associated orthogonal, normalized (of length 1) eigenvectors. Further, let \vec{x} be any vector in \mathbb{R}^2 . Since \vec{v}_1 and \vec{v}_2 are orthogonal and normal, \vec{x} can be written as $a \cdot \vec{v}_1 + b \cdot \vec{v}_2$ where $a^2 + b^2 = \|\vec{x}\|^2$. Since \vec{v}_1 and \vec{v}_2 are the eigenvectors of A , we can see that $A\vec{x} = A(a \cdot \vec{v}_1 + b \cdot \vec{v}_2) = aAv_1 + bAv_2 = \lambda_1 \cdot a\vec{v}_1 + \lambda_2 \cdot b\vec{v}_2$. Thus

x is merely stretched along the eigenvectors of A by an amount proportional to the corresponding eigenvalues.

Exercise 3.8. *To see the stretching effect use the `circle` figure preset with a transformation from a symmetric matrix.*

3.3. Complex Eigenvalues. Matrices with only real components can have complex eigenvalues and eigenvectors.

Example 3.9. *The Matrix $\begin{bmatrix} 3 & 1 \\ -1 & 3 \end{bmatrix}$ has complex eigenvalues $3 - i$ and $3 + i$, and eigenvectors $[1, -i]$, and $[1, i]$. These can be found by looking at the characteristic equation of the matrix, $(3 - \lambda)(3 - \lambda) + 1 = (3 - \lambda)^2 + 1 = 0$. Thus $3 - \lambda = \pm\sqrt{-1}$, which gives $\lambda = 3 \pm i$.*

When this occurs the two eigenvalues of a 2×2 matrix are always complex conjugates of each other, since if a complex number is a solution to the characteristic equation, its conjugate is also a solution and there are only two eigenvalues available. Since eigenvectors can be calculated from the eigenvalues and the elements of the matrix itself, the components of the eigenvectors of a real valued 2×2 matrix are complex conjugates of each other. Matrices with complex elements require significantly more computational steps than matrices with only real elements. Note that the JTV cannot handle complex arithmetic.

Theorem 5.9 in Lay [2] says that if A is a real 2×2 matrix with complex eigenvalue $\lambda = a - bi$ and $b \neq 0$ and associated eigenvector \vec{v} in \mathbb{C}^2 , then $A = PCP^{-1}$, where $P = [\text{Re}(\vec{v}) \text{ Im}(\vec{v})]$ and

$$C = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}.$$

Note that this is the product of some rotation matrix and some scaling matrix. Since \vec{v} is an eigenvector, it can be of any magnitude, so \vec{v} may be expressed as

$$\vec{v} = \begin{bmatrix} 1 \\ c + di \end{bmatrix}.$$

P can be a real triangular matrix since we can express \vec{v} in a form with one component with a zero imaginary part. Thus P^{-1} is also triangular, and so they both have real eigenvalues. So any matrix A can be represented as a rotation and a scaling combined with some non-rotation transformation. So a linear transformation contains a rotation (other than 0 or 180 degrees) if and only if it has complex eigenvalues.

Rotation matrices produce linear transformations with only a rotational component. The rotational matrix used to rotate points by ϕ radians counterclockwise about the origin is $A = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}$. Note that since this matrix is only symmetric at angles $\phi = 0^r$ and $\phi = \pi^r$ for some integer r , we can see that the only occasions on which a rotation merely stretches a figure is when it does nothing to it and stretches it by a factor of 1, (rotation by 0 radians) and when it flips it around both axes and thus stretches it by a factor of -1 , (rotation by π radians).

Exercise 3.10. *Try several different rotations with the JTV. Use the freehand drawing tool and rotate the results.*

Example 3.11. *Let us determine the matrix, and transformation for a 27 degree counterclockwise rotation. Since the trigonometric functions work in the counterclockwise direction we don't have to worry about that. Our trigonometric functions do require their angles in radians though. 27° is*

$27\frac{\pi}{180} \approx 0.4712^r$. The matrix is then

$$A = \begin{bmatrix} \cos(0.4712) & -\sin(0.4712) \\ \sin(0.4712) & \cos(0.4712) \end{bmatrix} = \begin{bmatrix} .891 & -0.454 \\ .454 & .891 \end{bmatrix}.$$

Thus the transformation should be $u = .891*x - .454*y$, $v = .454*x + .891*y$.

3.4. Combination of Linear Transformations. An advantage of using matrices to represent linear transformations comes when we wish to compose transformations. Suppose that T_1 and T_2 are two linear transformations with matrices A_1 and A_2 respectively. For any vector \vec{x} , it is the case that $T_1(T_2(\vec{x})) = T_1(A_2\vec{x}) = A_1(A_2\vec{x}) = (A_1A_2)\vec{x}$. Thus we have a simple matrix multiplication to calculate the composition of these two functions.

Example 3.12. Let us combine a stretch and a rotation. First we will use a linear transformation

to stretch the figure along the x axis. Then we will use another linear transformation to rotate the

figure by $\frac{\pi}{6}$ radians counter clockwise. The matrices are $A_1 = \begin{bmatrix} \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) \\ \sin(\frac{\pi}{6}) & \cos(\frac{\pi}{6}) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -.5 \\ .5 & \frac{\sqrt{3}}{2} \end{bmatrix}$

and $A_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$. Then $A_1A_2 = \begin{bmatrix} \sqrt{3} & -.5 \\ 1 & \frac{\sqrt{3}}{2} \end{bmatrix}$. Thus this is a matrix that will transform a point

as though it had had its x coordinate doubled and then been rotated by 30 degrees. Try using this transformation with some rectangles in the JTV.

The order of matrix multiplication is important, since matrix multiplication is generally not commutative. The matrices should be arranged in the same order as the operations that they represent. With the last operation leftmost and the first operation on the right.

3.5. Other Types of Matrices. These are not the only classes of linear transformations. There exist non-symmetric matrices with real eigenvalues. These will stretch figures in the plane, but not necessarily along the matrices eigenvectors.

4. AFFINE TRANSFORMATIONS

Affine transformations are similar to linear transformations. In \mathbb{R}^2 , an affine transformation can be expressed as $u(x, y) = a_1x + b_1y + c_1$ and $v(x, y) = a_2x + b_2y + c_2$. Note that this can be represented as a linear transformation in \mathbb{R}^3 . We would use the matrix equation

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

This applies the linear transformation of $\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}$ to the figure and then translates it by c_1 in the x direction and c_2 in the y direction. Linear transformations in \mathbb{R}^2 are obviously a subset of affine transformations in \mathbb{R}^2 , since we need only let $c_1 = c_2 = 0$. Some other subsets of affine transformations are dilations about a specific point, translations, reflections about arbitrary lines, and rotation about an arbitrary point.

For translation we use the matrix

$$\begin{bmatrix} 1 & 0 & c_1 \\ 0 & 1 & c_2 \\ 0 & 0 & 1 \end{bmatrix}.$$

This translates all points by c_1 in the x direction and c_2 in the y direction. While the matrix

$$D = \begin{bmatrix} a & 0 & c_1 \\ 0 & a & c_2 \\ 0 & 0 & 1 \end{bmatrix}$$

produces a dilation by a factor of a around $[\frac{c_1}{1-a}, \frac{c_2}{1-a}]^T$. We can see that the center of the dilation is $[\frac{c_1}{1-a}, \frac{c_2}{1-a}]^T$ because $D[\frac{c_1}{1-a}, \frac{c_2}{1-a}, j]^T = [\frac{c_1}{1-a}, \frac{c_2}{1-a}, j]^T$.

We can represent the composition of affine transformations as linear transformations in one higher dimension. For example, we could find the composition of two affine transformations in \mathbb{R}^2 by multiplying their 3×3 matrices together.

Exercise 4.1. *Let f and g be affine transformations given by $f_u(x, y) = 3x + 7y + 1$, $f_v(x, y) = 2x - 5y - 2$, $g_u(x, y) = y - 2$ and $g_v(x, y) = 6x + y + 3$.*

(a) *Convert both of these affine functions to matrix form and multiply them to find the matrix form of $f \circ g$. Then convert this matrix back to a pair of real-valued functions and enter them into the JTV. Use this to transform some figure.*

(b) *Enter the composition $f \circ g$ directly into the JTV, and transform the same figure. Compare the results with those of part a.*

The functions that produce affine transformations are those that most people think of as line functions. For example, $y = m \cdot x + b$ is an affine transformation from the line $y = 0$ in \mathbb{R}^2 to \mathbb{R}^2 . The equation for a plane is an affine transformation from the plane $z = 0$ in \mathbb{R}^3 to \mathbb{R}^3 .

Affine transformations preserve straight lines in a geometric sense. They do not preserve angles or distances, just as linear transformations do not preserve angles or distances. We might say that affine transformations preserve parallelness, or relative equality of slope, of lines. Linear transformations preserve this as well, but none of the other types of transformations we will be looking at will preserve parallelness unless they are a subset of the affine transformations.

Since rotations and translations are a subset of the affine transformations, we can use affine transformations to manipulate figures as we would manipulate rigid bodies in space. Rigid bodies do not change shape. All we can do with rigid bodies is translate and rotate them. In other words, if we limit ourselves to these two types of operations, then our transformation will produce figures that are congruent to the original figures. This only applies to a specific subset of affine transformations. Specifically rotations and translations.

For this reason affine transformations in three dimensions are often used for graphics applications. Point of view changes and object position changes can be represented as affine transformations, which can in turn be represented by 4×4 matrices. These matrices can then be multiplied together to get a single matrix that will, in one matrix multiplication, perform all the the various camera and object manipulations. This method helps reduce the number of required calculations necessary for real time 3-D graphics, such as are used in video games.

5. COORDINATE SYSTEM CHANGE

The most familiar coordinate system to most people is the Cartesian coordinate system. This system of n dimensions consists of n perpendicular number-lines. Points are specified by a series of n real coordinates, where the i^{th} coordinate is the location of the point along the i^{th} number line. The most commonly considered example is an xy -plane, such as the `Grid 2` example in the JTV.

All coordinate system changes are onto transformations since any point must be representable in the new coordinate system. For example linear transformations can be considered as shifts of coordinate system that only skew and stretch the axes.

5.1. Polar Coordinates. Consider the polar coordinate system. The transformation is $u = \cos(x) \cdot y$, $v = \sin(x) \cdot y$; a slightly modified version of this is called `Polar` in the set of predefined transformations. Using `Polar` to transform rectangles in the drawing space you get the truncated wedges that you might expect (as long as the rectangles are not too wide of course).

The set of predefined figures contains an entry called `Box Stack` which you can use to test the `Polar` transformation. It will draw a set of similar squares, one above the other, near the y axis. Transform it with the `Polar` transformation and you will see the identical boxes transformed into sections of a wedge.

The inverse of this transformation also works, it is $u = (x^2 + y^2) \cdot 5$, $v = \text{angle}(x, y)$. If you could draw nice truncated wedges in the drawing space they would translate to rectangles with this

transformation. Note that this is not quite the inverse. Since the polar coordinate transformation is not one-to-one, it does not have a true inverse. No non-one-to-one transformation has a true inverse because that inverse would be required to map each point in the codomain of the transformation onto more than one point in the domain, and would thus not be a function.

To see an example of this choose the **Radial** figure from the figure examples and the **Inverse Polar** from the set of example transformations. The result should look something like a grid in half of the plane. The fact that it is in only half of the plane tells us that the **Polar** transformation is not one-to-one. That is, some images are not unique. In fact every image is mapped to by an infinite number of preimages.

You should try both these transformations as an exercise. Note that the first transformation here gives overlapping outputs in some cases. You should also attempt to modify the first transformation so that it never gives an overlapping output with figures that can be drawn in the JTV. Note how one possible set of overlapping images has been precluded by mapping the whole of the visible x axis to $[-\pi, \pi]$.

One can see from the examples or in Figure 2 that each box is transformed to a piece of wedge. Looking at the areas of the rectangles and how they relate to wedge pieces can help us to determine how to compute area integrals in these coordinate systems, since during integration we are essentially adding infinitesimal areas. The fact that the boxes have constant area, but the wedges have larger area as function of r , motivates the factor of r in the $rdrd\theta$ that is the infinitesimal term in the integration.

We can tell from the specifics of the transformation that the wedges are cut off at particular radii; that is, each wedge piece is a large wedge with a smaller wedge removed from it. Since a wedge is a portion of a circle we may calculate its area. Let the wedge be defined as having radius r and angular width θ . The area of the circle from which the wedge is taken is $A = \pi \cdot r^2$. Since a circle has 2π radians of angle in its whole area and is symmetric with regards to angle, the area of the

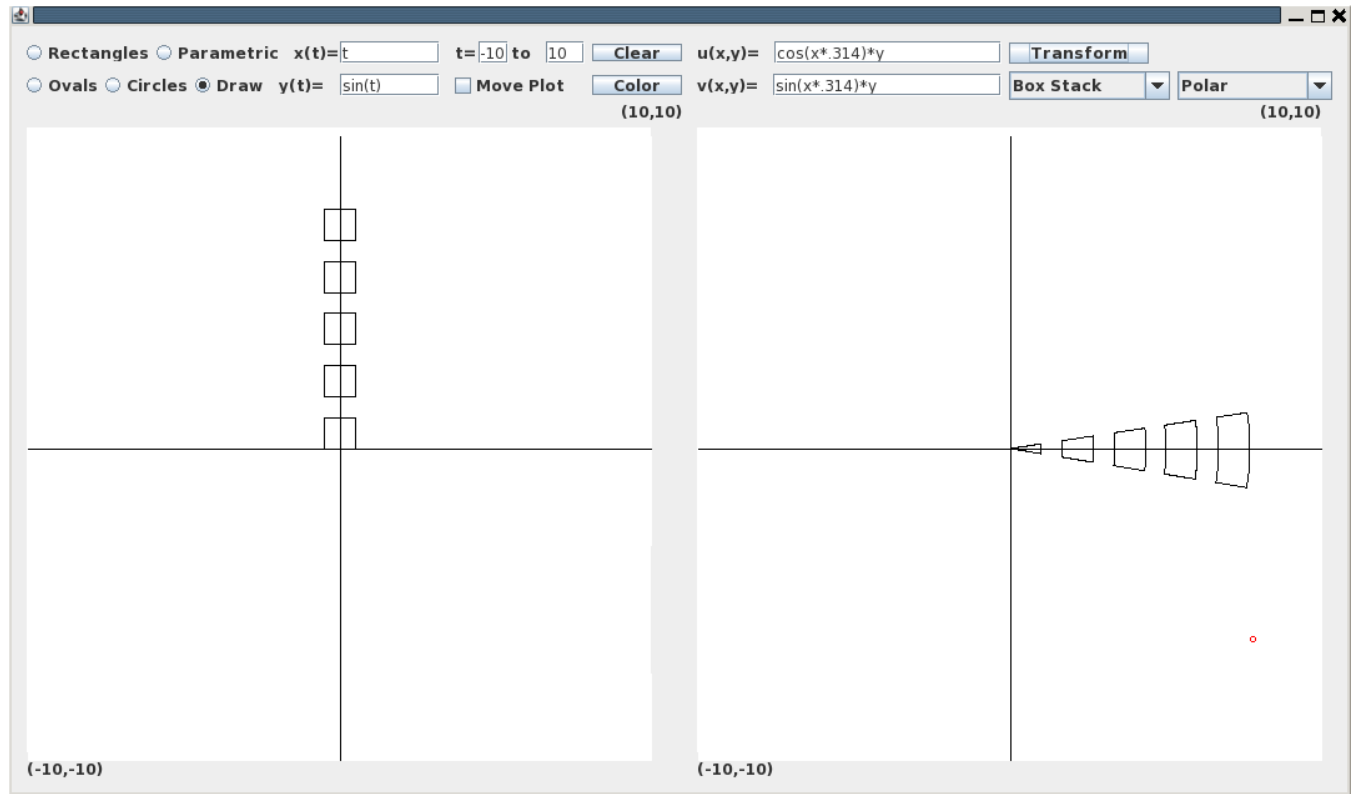


FIGURE 2. An example of the transformation between rectangular and polar coordinate systems.

wedge is $\frac{\theta}{2\pi} \cdot A = \frac{1}{2}\theta \cdot r^2$. Say then that our chunk of wedge has an interior radius r and an exterior radius R , and contains θ radians of angle. Its area is $\frac{1}{2}\theta \cdot R^2 - \frac{1}{2}\theta \cdot r^2 = \frac{\theta}{2} \cdot (R^2 - r^2)$.

5.2. Other Coordinate Systems. We often characterize coordinate systems by the properties of the lines produced by holding one coordinate constant and varying the other. In the standard Cartesian coordinate system this gives us lines parallel to the x and y axes. In the polar coordinate system these give us concentric circles and radial lines.

Note that in both of these systems the intersections between lines for different coordinates are all right angles. Many coordinate systems have this property. Coordinate systems that have this property are called *orthogonal coordinate systems*. [5] The conformal maps, as noted in the next section, have this property when treated as coordinate system changes from the Cartesian plane. As mappings they preserve all angles of intersection though not only those between grid lines.

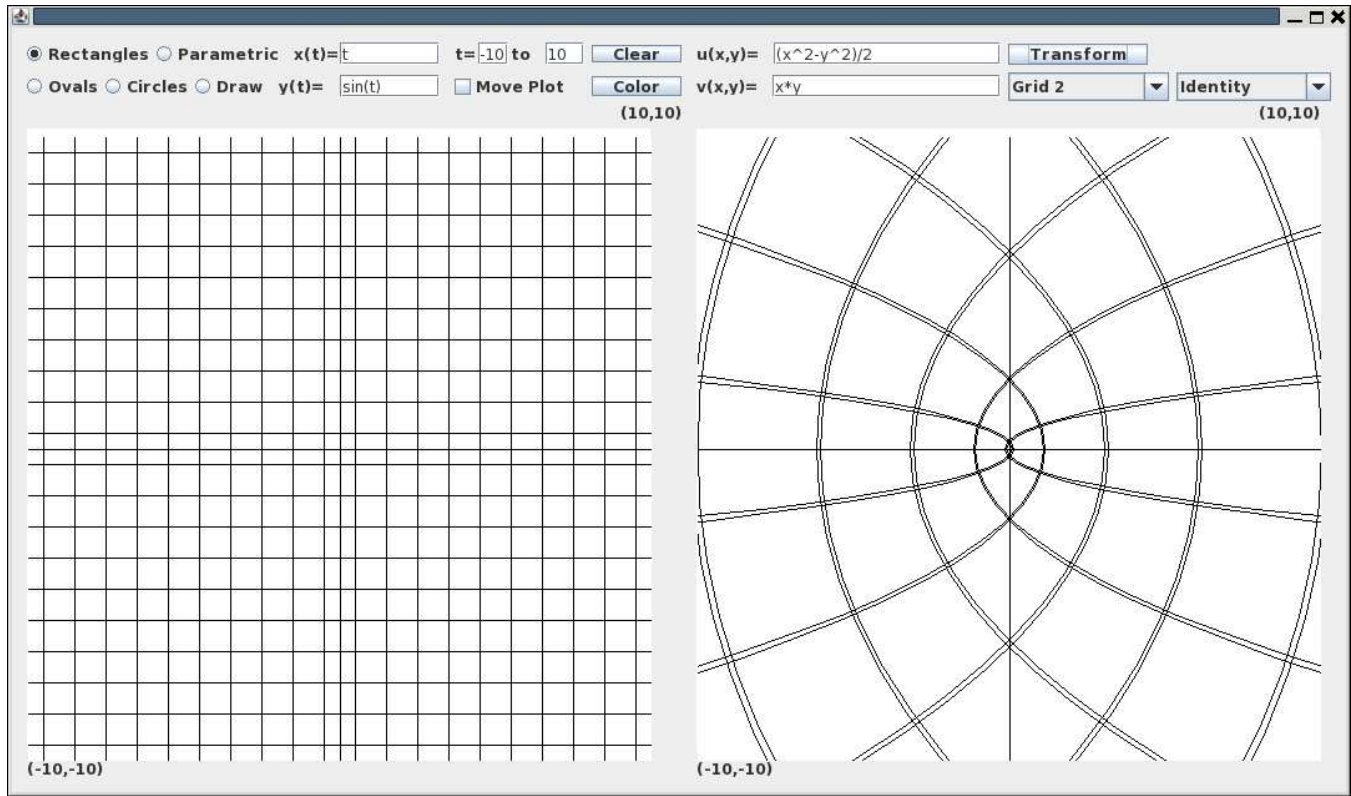


FIGURE 3. A screen-shot of the program with an elliptical coordinate change transformation.

An example of another orthogonal coordinate system is the parabolic coordinate system.[6] The origin is the focal point for an infinite number of pairs of parabolas. These parabolas are the coordinate lines. Lines parallel to the x -axis form parabolas opening in one direction and lines parallel to the y -axis for parabolas opening in the other.

The transformation for the parabolic coordinate system is $u = (x^2 - y^2)/2$, $v = x*y$. The parabolas of the coordinate lines are clearly visible in Figure 3. Many other orthogonal coordinate systems exist as well such as hyperbolic and elliptic.

There are useful non-orthogonal coordinate systems. For example most of the coordinate system produced by the simplest transformations, linear transformations, are not orthogonal.

6. CONFORMAL MAPS

6.1. The Complex Plane. We can map the complex numbers onto the plane. This is usually done by assigning the real part to the x axis and the imaginary part to the y axis. This is called

the complex plane. Any complex number is a point in the complex plane. Any complex function $f : \mathbb{C} \rightarrow \mathbb{C}$ can be viewed as a transformation from the plane to the plane.

Since the JTV cannot handle complex numbers directly, we must separate a complex function into its real and imaginary parts before we enter it. The real component of the function is entered in $u(\mathbf{x}, \mathbf{y})$, and the imaginary component is entered in $v(\mathbf{x}, \mathbf{y})$. This separation is not necessarily easy, however for simple algebraic functions it is possible. We also denote the real component of the complex function's complex argument as x , and the imaginary component as y . That is, $C(x + yi)$ is the complex form of the transformation that we must separate.

Since the figures that we are working with are composed of line segments, we may represent them as sets of curves in the complex plane. The image of the figure under some transformation is another set of curves. We can use this idea of curves in the complex plane to examine properties of certain transformations.

6.2. Perpendicular Lines. Let some curve in our figure be represented by a complex parametric function $p = p(t)$ where t is some real parameter. For example, in Figure 4, $p(t)$ could be one of the circles or one of the radial lines. Further suppose that $\frac{d}{dt}p(t)$ exists and is non-zero. Note that the derivative of $p(t)$ with respect to t is the direction of a tangent line to $p(t)$, where the derivative exists.

Let the transformation, $T(x + yi)$, be a continuous and everywhere differentiable function with a non-zero derivative mapping the complex numbers to the complex numbers. We will be considering mappings that are conformal in the neighborhood of our figure, thus the function need only be differentiable in the neighborhood our figure.

First note that the image of this curve in our figure can be represented by a complex function of the real parameter t , denoted $q(t) = T(p(t))$. This is just function composition; the composition of

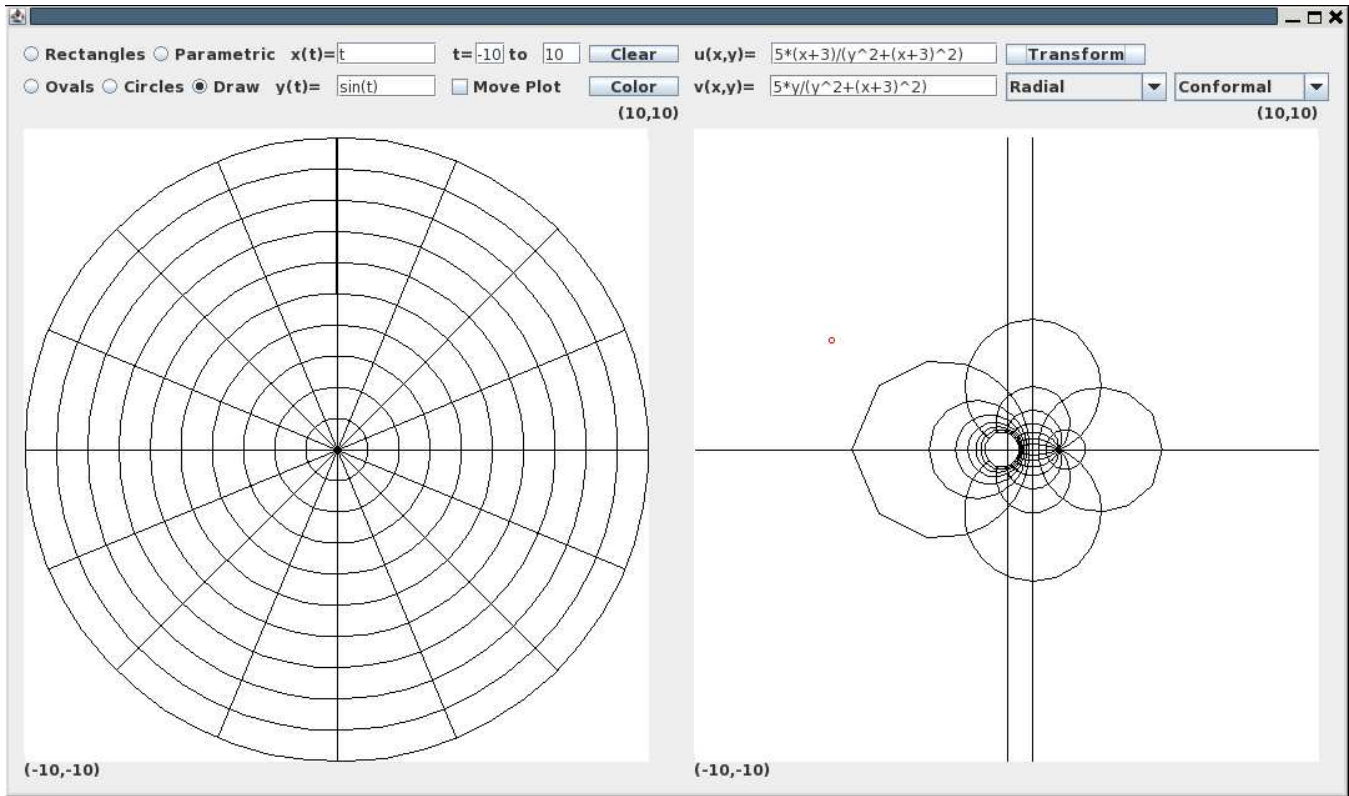


FIGURE 4. A screen-shot of the program with a transformation that is conformal at all points except $(-3, 0)$.

the mapping from \mathbb{R} to \mathbb{C} and a mapping from \mathbb{C} to \mathbb{C} . By the chain rule,

$$\frac{d}{dt}q(t) = \frac{d}{dp(t)}T(p(t)) \cdot \frac{d}{dt}p(t).$$

If $p(t_0)$ has a tangent line then so will $q(t_0)$ by the chain rule, since $q(t)$ is the composition of $p(t)$ with a differentiable function. We can use some algebra with complex variables to determine that the change in the angle of the tangent of $q(t_0)$ is equal to the sum of the tangent of $p(t_0)$ and a complex constant that depends only on $p(t_0)$ and not its tangent.[4] Thus any other function would have the same angle change at that point, since the angle in the plane of the tangent of the image any curve, that has a tangent at $p(t_0)$, at $p(t_0)$ is equal to the angle in the plane of the tangent of the curve at $p(t_0)$ plus a constant that depends only on the point $p(t_0)$; the angle of intersection of two curves at $p(t_0)$ is the angle of intersection of their tangents at $p(t_0)$, and this angle is equal to the angle of their intersection plus and minus the constant for the point $p(t_0)$. [4]

We see that if two curves intersect at an angle θ in the original figure, then their images will intersect at the same angle θ in the image of the figure. This mapping is called conformal on the set of all points at which this property holds.

Thus the only property that a function must have to be a conformal mapping is to be everywhere differentiable with a derivative that is nowhere zero.

Example 6.1. *Let us look at an example. Let a, b, c , and d be real numbers such that $ad - bc \neq 0$. The mapping $T(p) = \frac{ap+b}{cp+d}$ is conformal for $p \neq \frac{-d}{c}$ because it is differentiable and defined with non-zero derivative for all complex numbers satisfying those criteria. Since our transformational program does not accept complex numbers directly, we will have to separate the real and imaginary parts by hand. If $p = x + yi$ then*

$$\begin{aligned} T(p) &= \frac{a(x + yi) + b}{c(x + yi) + d} \\ &= \frac{ax + b + ayi}{cx + d + cyi} \\ &= \frac{ax + b + ayi}{cx + d + cyi} \cdot \frac{cx + d - cyi}{cx + d - cyi} \\ &= \frac{acy^2 + (ax + b)(cx + d) + (a(cx + d)y - c(ax + b)y)i}{c^2y^2 + (cx + d)^2} \\ &= \frac{acy^2 + (ax + b)(cx + d)}{c^2y^2 + (cx + d)^2} + \frac{a(cx + d)y - c(ax + b)y}{c^2y^2 + (cx + d)^2}i \end{aligned}$$

Let us choose some values for the constants; $a = 0, b = 1, c = 1, d = 3$. Then $T(x + yi) = \frac{acy^2 + (ax + b)(cx + d)}{c^2y^2 + (cx + d)^2} + \frac{a(cx + d)y - c(ax + b)y}{c^2y^2 + (cx + d)^2}i = \frac{0 \cdot 1y^2 + (0x + 1)(1x + 3)}{1^2y^2 + (1x + 3)^2} + \frac{0(1x + 3)y - 1(0x + 1)y}{1^2y^2 + (1x + 3)^2}i = \frac{x + 3}{y^2 + (x + 3)^2} - \frac{y}{y^2 + (x + 3)^2}i$.

The real part is the $u(x, y)$ function and the imaginary part is the $v(x, y)$ function. A conveniently magnified version of this function is available in the set of example transformations. Try it now, especially with the Grid and Grid2 example figures. Figure 4 shows what it should look like with the radial zoo option.

7. CONCLUDING REMARKS

Transformations are difficult to visualize, particularly transformations whose domain is of more than one dimension. It is impossible to visualize mappings from $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ and above.

We have seen that there are many ways to think about transformations. You can think of them as rules for deforming shapes, or as changes into different coordinate systems. Each approach reveals different things about a transformation. Whether or not a particular transformation is onto or one-to-one or not may be difficult to see when you look at it as a rule for deforming shapes, but when you look at it as a coordinate change it might be easier.

Although the approaches vary the idea behind them remains the same. It is thus when we consider various ways of characterizing a transformation, and when we choose to look at a function as a transformation rather than something else. While a rule for deforming shapes and an arbitrary continuous mapping from \mathbb{R}^2 to \mathbb{R}^2 may seem unrelated, the concepts behind them are very similar.

APPENDIX A: JEPLITE FUNCTIONS

Table 1.

Function Name	Function
<code>sin(x)</code>	Sine Function of x
<code>cos(x)</code>	Cosine Function of x
<code>angle(y,x)</code>	The angle between the point (x, y) and the positive x -axis.
<code>atan(x)</code>	The arctangent function of x
<code>acos(x)</code>	The arccosine function of x
<code>asin(x)</code>	The arcsine function of x
<code>sqrt(x)</code>	The square root of x
<code>abs(x)</code>	The absolute value of x
<code>log(x)</code>	The base ten logarithm of x
<code>ln(x)</code>	The base e logarithm of x
<code>mod(x,t)</code>	$x \bmod t$. The floating point remainder of x/t .
<code>x%t</code>	Is the same as <code>mod(x,t)</code> .
<code>x+y</code>	Addition of x and y .
<code>x-y</code>	Subtraction y from x .
<code>x*y</code>	Multiplication of x and y .
<code>x/y</code>	Division of x over y .
<code>x^y</code>	Exponentiation of x to the y power.

APPENDIX B: THE ZOO

Figures: Table 2.

Name	description
No Change	Does not alter the figure that is displayed
Circle	A unit circle centered on the origin
Offset Circle	A unit circle offset from the origin by $+(1,1)$
Grid	A grid of Xs
Box Stack	Several squares stacked from the origin in the plus y direction
Grid 2	A rectangular grid
Grid /w Circles	A rectangular grid with a set of concentric circles centered on the origin
Circles	A set of concentric circles centered on the origin
Radial	A set of concentric circles centered on the origin with lines radiating from the origin outward

Transformation Descriptions: Table 3.

Name	description
Identity	The identity transformation
Skew	an arbitrary linear transformation
Rotation 45	A 45 degree rotation
Polar	Displays points as though x and y were polar coordinates
Inverse Polar	The near inverse of the polar transformation
Swirl	Rotates points through an angle proportional to their distance from the origin
Number 6	An example of an arbitrary transformation
Number 7	Waves and pinches a figure
Conformal	An example conformal mapping

Transformations Equations: Table 4.

Name	$u(x, y)$	$v(x, y)$
Identity	x	y
Skew	$1.2*x+1.76*y$	$1.4*x+.9*y$
Rotation 45	$\text{sqrt}(.5)*x-\text{sqrt}(.5)*y$	$\text{sqrt}(.5)*x+\text{sqrt}(.5)*y$
Polar	$\text{sqrt}(.5)*x+\text{sqrt}(.5)*y$	$\sin(x*.314)*y$
Inverse Polar	$\text{angle}(y,x)/.314$	$\text{sqrt}(x^2+y^2)$
Swirl	$\cos(\text{sqrt}(x^2+y^2))*x-\sin(\text{sqrt}(x^2+y^2))*y$	$\sin(\text{sqrt}(x^2+y^2))*x+\cos(\text{sqrt}(x^2+y^2))*y$
Number 6	$\cos(x)*5+\sin(y)*5$	$\sin(x)*5+\cos(y)*5$
Number 7	$x+\sin(y)$	$\sin(x)+y$
Conformal	$5*(x+3)/(y^2+(x+3)^2)$	$5*y/(y^2+(x+3)^2)$

APPENDIX C: DESIGN CONSIDERATIONS OF THE JTV

One of the major factors in the design of the JTV was the need to deal with functions by sampling them rather than continuously. This is the reason that it can only draw line figures in the drawing area. Many of the shapes in the drawing area are stored as a shape data, that is a a number that signifies what type of shape it is and some other data that specifies it more exactly, like a center and a radius.

In the transformation area all of the shape data is stored as line segments. I do this mostly for speed, but also to simplify the task of calculating the deformation. When I apply the transformation $u=x, v=y+\sin(x)$ to a circle it should no longer be a circle. Since the possible transformations I can apply to a shape are unlimited, I need a representation that responds nearly appropriately to all of them. Collections of line segments fit this requirement.

The reason for the more complicated storage method in the drawing area is also twofold. It simplifies the translation between mouse clicks and shape data, and it is more nearly accurate. Due to the discrete nature of the computations available to the program, perfect accuracy is impossible. Even so, I want the program to be as accurate as is reasonable. Every approximation applied to the user's data, such as converting an oval into a collection of short line segments, reduces the accuracy of the approximation.

Much of the program is built around the JEPlite library. This library evaluates strings as expressions. The library combined with some elementary string manipulation is what allows the program to calculate the change in a point when it is transformed by a function that the user has entered as a string. Without this library the program would not have been completed in a timely manner. The use of this library does cause some limitations however. I cannot easily add functions that are not native to the library, such as \sinh , \cosh , or a factorial function, and malformed inputs are more likely to crash the program.

REFERENCES

- [1] Computer Graphics using Java 2 and 3D, by Hong Zhang, Y. Daniel Liang, (C) 2007, Pearson- Prentice Hall, Upper Saddle River, NJ 07458.
- [2] Linear Algebra and Its Applications, David C. Lay, (C) 2003, Addison Wesley.
- [3] Fundamental Concepts of Geometry, Bruce E. Meserve, (C) 1955, Addison Wesley.
- [4] Complex Analysis, Lars V. Ahlfors, (C) 1953, McGraw-Hill Book Company.
- [5] Mathematical Methods for Physicists, George Arfken, (C) 1985, Academic Press.
- [6] Mathematical Methods in the Physical Sciences, Mary L. Boas, (C) 2006, John Wiley & Sons
- [7] The JEPlite expression parser for Java, <http://jeplite.sourceforge.net/>
- [8] Introduction to Topology, Bert Mendelson, (C)1962-1990, Dover Publications, Inc.
- [9] Closer and Closer, Introducing Real Analysis, Carol Schumacher, (C) 2008, Jones and Bartlett Publishers, Inc.