Definition 1 *Let* $f : \mathbb{R}^2 \to \mathbb{R}$ *and* $\Phi : \mathbb{R}^2 \to graph(f) \subseteq \mathbb{R}^3$ *where* $\Phi(x,y) = (x, y, f(x,y)))$ *and both* f *is a smooth functions.* Φ *is invertible where* $\Phi^{-1} : graph(f) \to \mathbb{R}^2$ *and* $\Phi^{-1}(x, y, f(x,y)) = (x,y)$*.*

Proposition 1 *Because* f *is smooth,* Φ *is also smooth. Consequently, they are both infinitely differentiable.*

Definition 2 *Let* $\sigma : [0,1] \to graph(f) \subseteq \mathbb{R}^3$ *with* $\sigma(t) = (\sigma_1(t), \sigma_2(t), \sigma_3(t))$*. The length of this curve* $l(\sigma)$ *in Euclidean space is given by:*

$$l(\sigma) = \int_0^1 \sqrt{\sigma_1^2(t) + \sigma_2^2(t) + \sigma_3^2(t)}\ dt \tag{1}$$

$$= \int_0^1 \sqrt{\sigma'(t) \cdot \sigma'(t)}\ dt \tag{2}$$

Proposition 2 *Let* $\gamma : [0,1] \to \mathbb{R}^2$ *and set* $\sigma = \Phi \circ \gamma$*. From definition 2, the length of* $\sigma(t)$ *under the transformation is:*

$$l_\Phi(\sigma) = \int_0^1 \sqrt{(\Phi \circ \gamma)'(t) \cdot (\Phi \circ \gamma)'(t)}\ dt \tag{3}$$

Definition 3 *For two vectors in* $\mathbb{R}^2$*,* $\boldsymbol{u} = (u_1, u_2)$ *and* $\boldsymbol{v} = (v_1, v_2)$*, the dot product is* $\boldsymbol{u} \cdot \boldsymbol{v} = u_1 v_1 + u_2 v_2$*. However, we will need to use a more general form, the inner product to account for the transformation to the space. The inner product* $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ *must satisfy the following conditions:*

i $\langle\ .\ ,\ .\ \rangle : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$

ii $\langle \boldsymbol{u}, \boldsymbol{u} \rangle \geq 0$ *and* $\langle \boldsymbol{u}, \boldsymbol{u} \rangle = 0 \iff \boldsymbol{u} = 0$

iii $\langle \alpha(\boldsymbol{u} + \boldsymbol{w}), \boldsymbol{w} \rangle = \alpha \langle \boldsymbol{u}, \boldsymbol{w} \rangle + \alpha \langle \boldsymbol{u}, \boldsymbol{w} \rangle$

iv $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{v}, \boldsymbol{u} \rangle$

Let $\boldsymbol{e_1} = (1,0)$ *and* $\boldsymbol{e_2} = (0,1)$ *be our base vectors in* $\mathbb{R}^2$ *and* $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ *is as follows:*

$$\begin{aligned}
\langle \boldsymbol{u}, \boldsymbol{v} \rangle &= u_1 v_1 (\boldsymbol{e_1} \cdot \boldsymbol{e_1}) + u_1 v_2 (\boldsymbol{e_1} \cdot \boldsymbol{e_2}) + u_2 v_1 (\boldsymbol{e_2} \cdot \boldsymbol{e_1}) + u_2 v_2 (\boldsymbol{e_2} \cdot \boldsymbol{e_2}) && (4) \\
&= (u_1, u_2) \begin{pmatrix} \boldsymbol{e_1} \cdot \boldsymbol{e_1} & \boldsymbol{e_1} \cdot \boldsymbol{e_2} \\ \boldsymbol{e_2} \cdot \boldsymbol{e_1} & \boldsymbol{e_2} \cdot \boldsymbol{e_2} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} && (5) \\
&= (u_1, u_2) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} && (6) \\
&= \boldsymbol{u}^{\intercal} \mathbb{I} \boldsymbol{v} && (7)
\end{aligned}$$

Note that in euclidean space, $\boldsymbol{e_1} \cdot \boldsymbol{e_2} = \boldsymbol{e_2} \cdot \boldsymbol{e_1} = 0$ and so $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = u_1 v_1 + u_2 v_2$ which is the dot product. Our inner product is now formed such that we can derive a matrix $\boldsymbol{A}$ from base vectors of our transformed space.

Definition 4 *Let $\boldsymbol{A}$ be a 2x2 real, symmetric and positive definite matrix such that $\boldsymbol{A} : \mathbb{R}^2 \to Matrices(\mathbb{R}^2)$.*

Definition 5 *Our inner product in a non-euclidean space transformed at any point (x, y) by f is given by $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_f (x, y) = \boldsymbol{u}^{\intercal} \boldsymbol{A}_f (x, y) \boldsymbol{v}$.*

Our claim is that the expression used to calculate the length of $\gamma(t)$ in non-euclidean space (equation 3) can be expressed as an inner product (definition 5) containing the matrix $\boldsymbol{A}$. More formally:

$$(\Phi \circ \gamma)'(t) \cdot (\Phi \circ \gamma)'(t) = \langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_f (\dot{\gamma}(t)) = \dot{\gamma}(t) \boldsymbol{A}_f (\dot{\gamma}(t)) \dot{\gamma}(t) \tag{8}$$

Proposition 3 *The matrix $\boldsymbol{A}$ that satisfies equation 8 is the following:*

$$\boldsymbol{A}_f (x, y) = \begin{pmatrix} 1 + \frac{\partial f}{\partial x}(x, y)^2 & \frac{\partial f}{\partial x}(x, y)^2 \frac{\partial f}{\partial y}(x, y) \\ \frac{\partial f}{\partial x}(x, y) \frac{\partial f}{\partial y}(x, y) & 1 + \frac{\partial f}{\partial y}(x, y)^2 \end{pmatrix} \tag{9}$$

Proof: Let $v_1^{(x,y)}(t) = t\boldsymbol{e_1} + (x, y)$ and $v_2^{(x,y)}(t) = t\boldsymbol{e_2} + (x, y)$ be our base vector functions. These functions are chosen such that $\dot{v}_1^{(x,y)}(t) = \boldsymbol{e_1}$ and $\dot{v}_2^{(x,y)}(t) = \boldsymbol{e_2}$. Our formulation of the inner product in equation 5 gives some indication on how our $\boldsymbol{A}$ matrix will be formed. If we compose Φ with our base vector functions then we can extract a function which provides the base vectors of the transformed space at any point (x, y). This composition for $v_1^{(x,y)}(t)$ and $v_2^{(x,y)}(t)$ is $(\Phi \circ v_1^{(x,y)})'(0)$ and $(\Phi \circ v_2^{(x,y)})'(0)$ respectively. And, from this we can define our $\boldsymbol{A}$ matrix:

$$\boldsymbol{A}(x,y) = \begin{pmatrix} (\Phi \circ v_1^{(x,y)})'(0) \cdot (\Phi \circ v_1^{(x,y)})'(0) & (\Phi \circ v_1^{(x,y)})'(0) \cdot (\Phi \circ v_2^{(x,y)})'(0) \\ (\Phi \circ v_2^{(x,y)})'(0) \cdot (\Phi \circ v_1^{(x,y)})'(0) & (\Phi \circ v_2^{(x,y)})'(0) \cdot (\Phi \circ v_2^{(x,y)})'(0) \end{pmatrix} \tag{10}$$

Now we need to evaluate each of these expressions. Firstly, we can simplify v_1 and v_2 as $v_1^{(x,y)}(t) = (t+x, y)$ and $v_2^{(x,y)}(t) = (x, y+t)$ respectively. We will look at v_1 since the derivation is similar to that of v_2.

$$(\Phi \circ v_1^{(x,y)})'(0) = (t+x, y, f(t+x,y))'(0) \tag{11}$$

$$= (1, 0, \frac{d}{dt} f(t+x.y))(0) \tag{12}$$

And given that by the chain rule:

$$\frac{d}{dt} f(x(t), y(t)) = f_x(x(t), y(t))\dot{x}(t) + f_y(x(t), y(t))\dot{y}(t) \tag{13}$$

The expression becomes:

$$(\Phi \circ v_1^{(x,y)})'(0) = (1, 0, \frac{\partial f}{\partial x}(x,y)) \tag{14}$$

And when we evaluate the dot product, our entry for row 1 column 1 of the $\boldsymbol{A}$ matrix becomes:

$$(\Phi \circ v_1^{(x,y)})'(0) \cdot (\Phi \circ v_1^{(x,y)})'(0) = (1, 0, \frac{\partial f}{\partial x}(x,y)) \cdot (1, 0, \frac{\partial f}{\partial x}(x,y)) \tag{15}$$

$$= 1 + \frac{\partial f}{\partial x}(x,y)^2 \tag{16}$$

Similar calculations will yield the other entries in the table, verifying proposition 3.

Lemma 0.1 *The following equation is true:*

$$(\Phi \circ \gamma)'(t) \cdot (\Phi \circ \gamma)'(t) = \dot{\sigma}_1^2(t) + \dot{\sigma}_2^2(t) + (\frac{\partial}{\partial x} f(\sigma(t))\dot{\sigma}_1(t) + \frac{\partial}{\partial y} f(\sigma(t))\dot{\sigma}_2(t))^2 \tag{17}$$

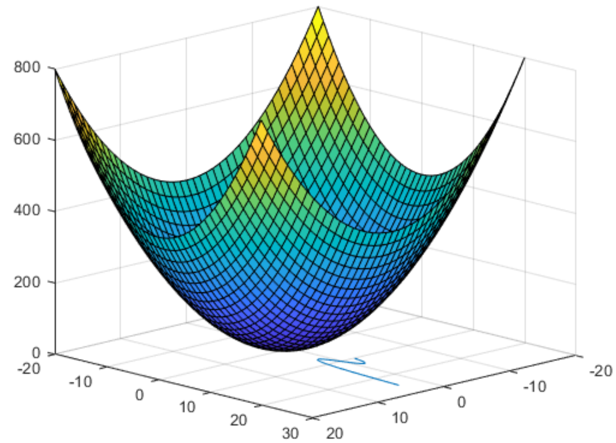

Figure 1: A random curve in blue and the mapping function $f(x, y) = x^2 + y^2$.

Proof:

$$(\Phi \circ \gamma)(t) = (\sigma_1(t), \sigma_2(t), f(\sigma(t))) \tag{18}$$

$$(\Phi \circ \gamma)'(t) = (\dot\sigma_1(t), \dot\sigma_2(t), \frac{d}{dt} f(\sigma(t))) \tag{19}$$

$$= (\dot\sigma_1(t), \dot\sigma_2(t), \frac{\partial}{\partial x} f(\sigma(t))\dot\sigma_1(t) + \frac{\partial}{\partial y} f(\sigma(t))\dot\sigma_2(t)) \tag{20}$$

$$(\Phi \circ \gamma)'(t) \cdot (\Phi \circ \gamma)'(t) = \dot\sigma_1^2(t) + \dot\sigma_2^2(t) + (\frac{\partial}{\partial x} f(\sigma(t))\dot\sigma_1(t) + \frac{\partial}{\partial y} f(\sigma(t))\dot\sigma_2(t))^2 \tag{21}$$

Proposition 4 *We can now prove our claim that* $(\Phi\circ\gamma)'(t)\cdot(\Phi\circ\gamma)'(t) = \dot\gamma(t)\boldsymbol{A}_f(\gamma(t))\dot\gamma(t)$.

RHS:

$$\dot\gamma(t)\boldsymbol{A}(\gamma(t))\dot\gamma(t) = (\dot\gamma_1(t), \dot\gamma_2(t)) \begin{pmatrix} 1 + \frac{\partial f}{\partial x}(x, y)^2 & \frac{\partial f}{\partial x}(x, y)^2 \frac{\partial f}{\partial y}(x, y) \\ \frac{\partial f}{\partial x}(x, y)\frac{\partial f}{\partial y}(x, y) & 1 + \frac{\partial f}{\partial y}(x, y)^2 \end{pmatrix} \begin{pmatrix} \dot\gamma_1(t) \\ \dot\gamma_2(t) \end{pmatrix} \tag{22}$$

$$= (\dot\gamma_1(t) + \dot\gamma_1(t)\frac{\partial}{\partial x} f(\gamma)^2 + \dot\gamma_2(t)\frac{\partial}{\partial x} f(\gamma(t))\frac{\partial}{\partial y} f(\gamma(t)), \tag{23}$$

$$\dot\gamma_2(t) + \dot\gamma_2(t)\frac{\partial}{\partial x} f(\gamma(t))^2 + \dot\gamma_1(t)\frac{\partial}{\partial x} f(\gamma(t))\frac{\partial}{\partial y} f(\gamma(t))) \begin{pmatrix} \dot\gamma_1(t) \\ \dot\gamma_2(t) \end{pmatrix} \tag{24}$$

$$= \dot\gamma_1(t)^2 + \dot\gamma_1(t)^2\frac{\partial}{\partial x} f(\gamma(t))^2 + 2\dot\gamma_1(t)\dot\gamma_2(t)\frac{\partial}{\partial x} f(\gamma(t))\frac{\partial}{\partial y} f(\gamma(t)) \tag{25}$$

$$+ \dot\gamma_1(t)\dot\gamma_2(t) + \dot\gamma_2(t)^2 + \dot\gamma_2(t)^2\frac{\partial}{\partial y} f(\gamma(t))^2 \tag{26}$$

$$= \dot\gamma_1^2(t) + \dot\gamma_2^2(t) + (\frac{\partial}{\partial x} f(\gamma_1(t), \gamma_2(t))\dot\gamma_1(t) + \frac{\partial}{\partial y} f(\gamma_1(t), \gamma_2(t))\dot\gamma_2(t))^2 \tag{27}$$

Which is equivalent to the result derived in lemma 0.1. Therefore, our claim that $(\Phi \circ \gamma)'(t) \cdot (\Phi \circ \gamma)'(t) = \dot{\gamma}(t)\boldsymbol{A}_f(\gamma(t))\dot{\gamma}(t)$ is true. From this, we can redefine the non-euclidean length:

$$l(\sigma) = \int_0^1 \sqrt{\langle \sigma'(t), \sigma'(t) \rangle_f (x, y)}\, dt \tag{28}$$
$$= \int_0^1 \sqrt{\sigma'(t)\boldsymbol{A}_f(\sigma(t))\sigma'(t)^\intercal}\, dt \tag{29}$$

Lets look at the programmatic implementation of this in the function `utility_fitness_non_euclidean_3d`. At the beginning, we used the identity matrix for our `par$non_euclidean_A` parameter. If the curve between two points is a straight line, then this can verify that the implementation is correct. The code below is an interpretation of equation 29 however we are treating time as discrete rather than continuous.

```
# function to calculate non euclidean distance
utility_fitness_non_euclidean_3d <- function(self, genotype, par){
        complete_genotype <- rbind(par$non_euclidean_start, genotype,
                                                           par$non_euclidean_end);

        if (is.null(par$non_euclidean_A)){
                par$non_euclidean_A <- array(c(1, 0, 0, 1), dim = c(2, 2));
        };

        bounds <- par$non_euclidean_bounds;
        curve <- complete_genotype;

        differential <- differentiate(genotype);

        time_interval <- (bounds[2] - bounds[1]) / (dim(curve)[1] - 1);
        time_stamps   <- ((1:dim(curve)[1]) - 1) * time_interval;

        differential <- array(append(time_stamps, differential), dim = dim(curve));

        total_area <- 0;
        for (time_index in 1:(dim(curve)[1] - 1)){
                current_gradient <- differential[time_index, ];
                A_component <- current_gradient %*% par$non_euclidean_A;
                magnitude   <- sqrt(sum(A_component * current_gradient));

                local_area <- magnitude * time_interval;
                total_area <- total_area + local_area;
        }
        return(total_area)
}
```

Using the identity matrix as A will result in the curve shown in figure 2. Because this graph is a straight line, we can now replace our matrix A with a matrix function that changes depending on the location in the space. We will use $f(x, y) = x^2 + y^2$ to

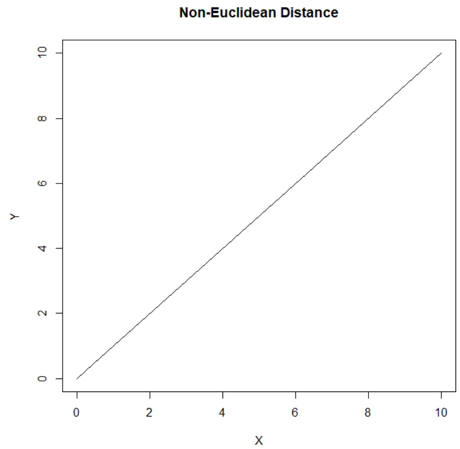

Figure 2: Non-Euclidean Distance: Using an Identity Matrix

define the matrix A:

$$A_f(x,y) = \begin{pmatrix} 1 + \frac{\partial f}{\partial x}(x,y)^2 & \frac{\partial f}{\partial x}(x,y)^2 \frac{\partial f}{\partial y}(x,y) \\ \frac{\partial f}{\partial x}(x,y) \frac{\partial f}{\partial y}(x,y) & 1 + \frac{\partial f}{\partial y}(x,y)^2 \end{pmatrix} = \begin{pmatrix} 1 + 4x^2 & 4xy \\ 4xy & 1 + 4y^2 \end{pmatrix} \tag{30}$$

```
# calculate the relative A matrix for the function f(x, y) = x^2 + y^2
get_A <- function(coordinate){
        x <- coordinate[1];
        y <- coordinate[2];

        return(array(c(
        1 + (4*(x^2)), 4*x*y, 4*x*y, 1 + (4*(y^2))
        ), dim = c(2, 2)))
}
```

We've replaced the parameter `par$non_euclidean_A` with the function `get_A()` inside of the fitness function, which will return the corresponding matrix A for a given coordinate. The coordinate we are using is the average of the coordinates on each side of the segments which make up the curve. The `par$non_euclidean_start` and `par$non_euclidean_end` will be $(0, 10)$ and $(10, 0)$ respectively. You can run this simulation using the `ga_example()` function from the ga-package:

```
# run the non-euclidean-var example
gapackage::ga_example(name = "non-euclidean-var")
```

The simulation will result in the curve shown in figure 3. You can see that it curves around the parabolic shape of the function. I've taken the data for this curve and used MATLAB to plot the curve on the function to make it more clear what is going on. You can see this in figure 4. The blue line represents the shortest Euclidean distance between the two coordinates. The red line shows the shortest distance between the two

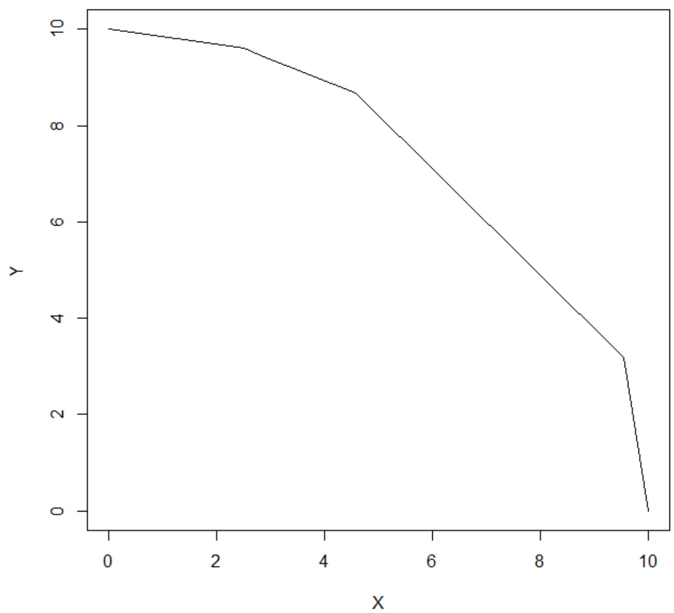

Figure 3: Non-Euclidean Distance: Relative A Matrix Results

coordinates where the space is transformed according the function $f(x, y) = x^2 + y^2$.

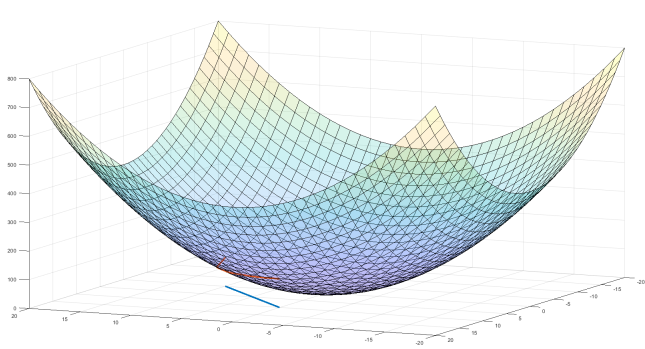

Figure 4: Euclidean Distance (Blue) / Non-Euclidean Distance (Red) on Function $f(x, y) = x^2 + y^2$