SOFTWARE 'CINDERELLA' AND ITS APPLICATION IN VISUALIZATION OF PHYSIC AND MATHEMATICS

MARINA M. ZIROJEVIĆ[*], DUŠAN S. JOKANOVIĆ[*] AND ĐORĐE BARALIĆ[†]

[*] Production and Management Faculty Trebinje
University of East Sarajevo
Stepe Stepanovica bb, 89 101 Trebinje, BiH
e-mail: marina.zirojevic@live.com, dusanjok@yahoo.com, web page: http://www.fpmtrebinje.com

[†]Mathematical Institute SASA
Kneza Miloša 36, 11 001 Belgrade, Serbia
e-mail: djbaralic@turing.mi.sanu.ac.rs, web page: http://www.mi.sanu.ac.rs/

Summary. In this paper we present posibilites to use the 'Cinderella' software in theory of curves and vector fields visualizations as well as in understanding their mechanical and geometric characteristics. These objects are extensively studied in physics, medicine and other sciences. The use of 'Cinderella' is exemplified in animation of cardioid, astroid, cubic curves, quadratic curves and their singularities. We emphasize the educational role of this software and application of its language 'SindyScript' and illustrate the way in which physical processes are modelled by the means of "Cinderella' software.

1 INTRODUCTION

Cinderella is a free software package for doing dynamic geometry on computer initialy developed by Jürgen Richter-Gebert. In a very first stage of development software was created in Objective-C on NeXT platform. Later, it was rewritten in Java by Jürgen Richter-Gebert and Ulrich Kortenkamp.

The application areas of 'Cinderella' reach from pure Euclidean geometry via physics to kinematics and computer-aided design. The important distinction of 'Cinderella' to other software is that it keeps track of your construction steps and is able to re-do them, even for a different placement of the base elements. It is possible that, after you have made a sketch, pick a point and drag it to some other position and the rest of construction is updated during the move which enables the valid instance of geometric construction. Beside alowing us to make drawing which look nice and to have access to elements that lie outside the drawing region, this feature is even more useful when we finished construction and want to explore geometric properties of it.

The benefits using 'Cinderella' are much more visible in it's version 'Cinderella.2' which offers a completely new suite of application. In this 'Cindrella' version powerful engine for physical simulations called 'CindyLab' is provided. 'CindyLab' is the part of Cinderella which enables us to create simulations of physical experiments with a few mouse clicks, no matter if it is simple or very complex physical scenarios. The basic model behind this simulations in 'Cinderella' is a particle/forces model.There are particles with certain masses that behave according to Newtons laws of motion. At the same time there is a variety

2010 Mathematics Subject Classification: 11H56, 18A10, 68R10.
Key words and Phrases: Cinderella, CindyLab, CindyScript, Dynamic Geometry

of forces such as springs, gravity, magnetic fields, walls, etc. that influence the movement of the particles. A wide range of experiments can be simulated using this model.

If we going further, 'Cinderella' isn't just software for dynamic geometry and physical simulation. A programming enviroment named 'CindyScript' is significant enhancement to Cinderella which represent very powerful and easy to learn functional language. It is designed primarily with the intention to allow high-level interaction with geometric constructions or physical simulations created in 'Cinderella.2'. Nevertheless, it can also be used as a standalone language for performing mathematical calculations.

In the following section we will emphasize and give a brief overview of the main 'Cinderella' applications.

2 MAIN 'CINDERELLA' CHARACTERISTICS

A unique trait which is possible due to mathematical foundation of 'Cinderella' is capability to view a single construction in different geometrical interpretations at the same time. This feature maintained an abstract model of the construction which can be displayed through one or more windows.

2.1 Euclidean plane and spherical projection

Tipes of geometry which are present in latest 'Cinderella' version are Euclidean geometry, hyperbolic geometry, and elliptic geometry. User can switch among these three geometries by pressing the button Euc for Euclidean geometry, Hyp for hyperbolic geometry, or Ell for elliptic geometry.

Since the Euclidean view is the usual drawing surface, it is default view at startup 'Cindrella'. It is the natural window for doing Euclidean geometry.

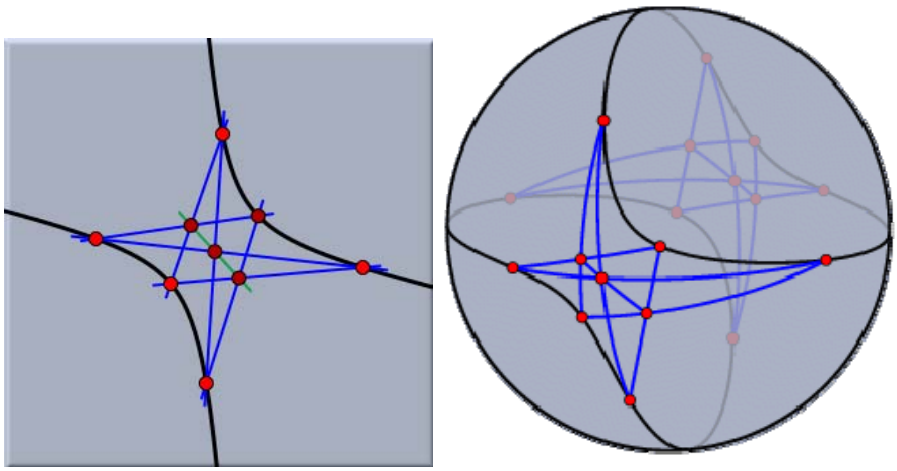


Figure 1: Pascal's theorem in a Euclidean and Spherical view [3]

The spherical view arises from a projection of the Euclidean plane onto the surface of a ball. The center of the projection is the center of the sphere. The plane does not pass through the center.

This projection maps each point to an antipodal pair of points on the sphere. Each line is mapped to a great circle (an equator) on the sphere. The incidence structure is preserved. Working with the spherical view allows elements at infinity to be manipulated. They lie on the boundary of the image of the unrotated sphere. The spherical view is a natural way to represent elliptic geometry. Measurement of angles between lines corresponds to measuring angles on the sphere. Measuring distances corresponds to the usual geodesic measurement of distances on the sphere, keeping in mind that antipodal points are identified with each other.

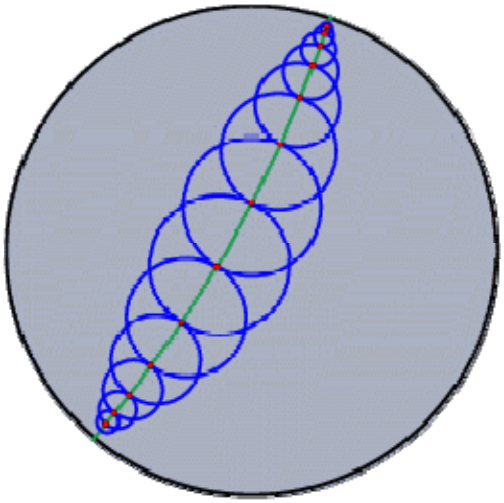


Figure 2: Hyperbolic circles of equal size [3]

The hyperbolic view is the natural view for hyperbolic geometry. In fact, doing hyperbolic geometry is the main reason for opening a hyperbolic view. The hyperbolic view represents an implementation of the Poincaré disk model of hyperbolic geometry. In this model the (finite part of the) hyperbolic plane is represented by a disk. Each line is represented by a circular arc that is orthogonal to the boundary of this disk. The measurement of angles between lines is conformal. This means that you can read off angles by measuring Euclidean angles between the circular arcs. The measurement of distances is such that the elements on the boundary are "infinitely far away" from any other point on the disk. If you "walk" in hyperbolic unit steps in one direction, you will never reach the boundary. In the disk, the steps seem to become smaller and smaller (in Euclidean measurement).

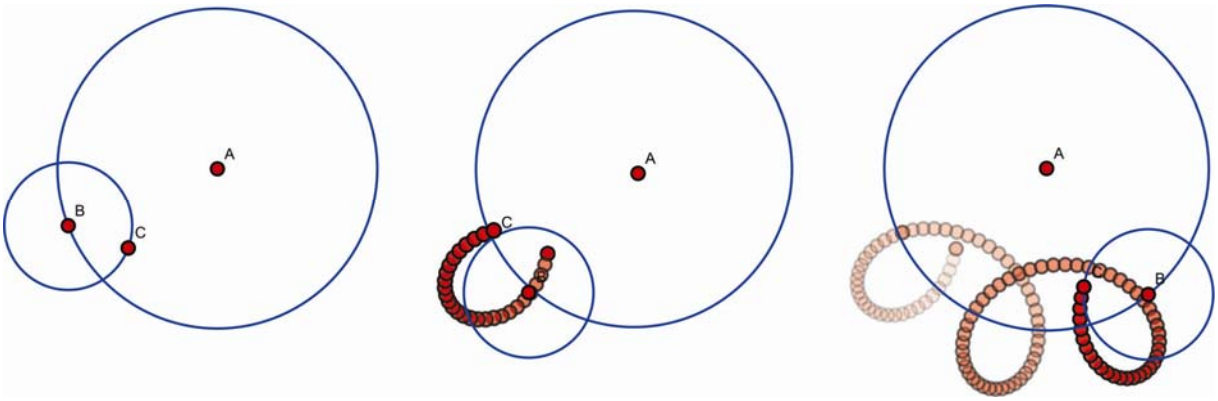


Figure 3: The animation of epicycles in 'Cinderella' [2]

3 PROGRAMMING WITH 'CindyScript'

The following section summarizes the most fundamental concepts of 'CindyScript'. 'CindyScript' make all calculations by executing functions. Every function takes the argument and return output value. In order to define function in 'CindyScript' user have to specify the name of a function, provide a parameter list and write down the body of a function. For example, function which calculates the sum of the first `n` squares we define as follows: `f(n):=sum(1..n,i,i^2)`.

Mathematical operators in 'CindyScript' can be applied to numbers and lists. Integer, real and complex numbers can be added, subtracted, multiplied, divided and taken to the power with the operators `+`, `-`, `*`, `/`, `^`. Elementary mathematical functions are also avaliable. For example, square root, exponential function and natural logarithm can be obtained by functions:`sqrt(<expr>)`, `exp(<expr>)`, `log(<expr>)`. The standard trigonometric functions are available through the following operators: `sin(<expr1>)`, `cos(<expr1>)`, `tan(<expr1>)`, `arcsin(<expr1>)`, ...

Writing a sequential code in 'CindyScript' is not so different from writing code in sequential programming languages such as C or Java. In what follows, we will exemplify how conditional branching and various kinds of loops can be generated in 'CindyScript'. The conditional operator with sintax **`if(<bool>,<expr>)`** act in a way that the expression `<expr>` is evaluated if the Boolean condition `<bool>` evaluates to true. Otherwise, ____ is returned. In following paragraph we will describe while loop which evaluates the expression `<expr>` as long as the condition `<bool>` is true [3].

```
x=0; sum=0;
Expression=while(x<4,
    x=x+1;
    sum=sum+x;
    println(x+" --> "+sum);
    sum
  );
println(Expression);
```

3.1 Linear Algebra with 'CindyScript'

As one of most important branch of mathematics, Linear Algebra concerne study of linear sets of equations and allows the analysis of rotation in space, least square fitting as well as many other problems in mathematics, physics and engineering. Since matrix and determinant are extremely useful tools of linear algebra, there are the most fundamental and elementary concepts of 'CindyScript'.

For vectors and matrix representation in 'CindyScript' we use lists of elements which can be created by placing the elements in square brackets, separatedby commas. For example,`[1,2,5,0,-2]` is a list of numbers and `["a","b","3"]` is a list of strings. The

list whose all elements are numbers we call a "*number vector*". If the elements of a list are again lists, and if all these lists have the same length, then such a list is called a *matrix*. Whether a certain list is a number vector or matrix can be tested with the operator `isnumbervector(<expr>)` or `ismatrix(<expr>)`.

Besides addition and multiplication, as described earlier in this section, there are also several arithmetic operations from linear algebra that are responsible for vector and matrix administration. In what follows, we provide some examples of this operators.

Dimensions of a matrix we can easily calculate using function `matrixrowcolum (<matrix>)`. Operator which returns transpose of a matrix is `transpose(<matrix>)`. Functions `row(<matrix>,<int>)` and `column(<matrix>,<int>)` gives rows and columns of a matrix at position `<int>` as a vector. Furthermore, extracting a submatrix of a matrix is available using function `submatrix(<matrix>,<int1>,<int2>)`. Resulting submatrix is obtained by deleting the column with index `<int1>` and the row with index `<int2>`. These functions are illustrated in a next example.

Expression	Result
`matrixrowcolum([[8,7,6],[1,2,3],[9,12,15]])`	`[3,3]`
`transpose([[8,7,6],[1,2,3],[9,12,15]])`	`[[8,1,9],[7,2,12],[6,3,15]]`
`row([[8,7,6],[1,2,3],[9,12,15]],2)`	`[[1,2,3]]`
`column([[8,7,6],[1,2,3],[9,12,15]],3)`	`[[6,3,15]]`
`submatrix([[1,2,4],[3,2,3],[1,3,6],[5,4,7]],2,3)`	`[[1,4],[3,3],[5,7]]`

As an extremely important function, determinant can be used for many purposes. Operator that calculates the determinant of a square matrix is `det(<matrix>)`. If we observe the square invertible matrix we can calculate the inverse of that matrix using function `inverse(<matrix>)`. Adjunct and eigenvalues of a square matrix are calculable with opertators `adj(<matrix>)`and `eigenvalues(<matrix>)`. We will use the next example to show application of these functions in 'CindyScript'.

Expression	Result
`m=[[1,1,1],[2,3,-2],[1,1,2]];` **`println(eigenvalues(m));`** **`println(det(m));`** **`println(inverse(m));`**	**`[0.1067,2.9466 - i*0.8297,2.9466 + i*0.8297]`** **`1`** **`[[8,-1,-5],[-6,1,4],[-1,0,1]]`**

The following example is given to show how we can use 'CindyScript' for solving a linear equation. The 'Cinderella' operator `linearsolve(<matrix>,<matrix>)`calculates a solution x of the systemof equations $Ax=b$. The matrix A must be square (n times n) and invertible and b can either be an n dimensional vector, it can be a matrix with n rows.

Expression	Result
`m=[[1,1,0],[0,1,0],[0,1,1]];` **`x=linearsolve(m,[2,3,4]);`** **`println(x);`** **`println(m*x);`**	**`[-1,3,1]`** **`[2,3,4]`**

Next pharagraph shows the way in which 'CindyScript' can be used for numerical calculation such as the derivative of a function and construction a tangent to a given function. The picture below demonstrate the 'CindyScript' code and the resulting image created by the program [2].

```
f(x):=(x+3)*(x+1)*(x-5)*0.1;
g(x):=d(f(#),x);
plot(f(x));
plot(g(x),color->[1,0,0]);
t=tangent(f(#),A.x);
draw(t,color->[0,0.51,1]);
```

Figure 4a: Code which demonstrates the use of the 'CindyScript' operators

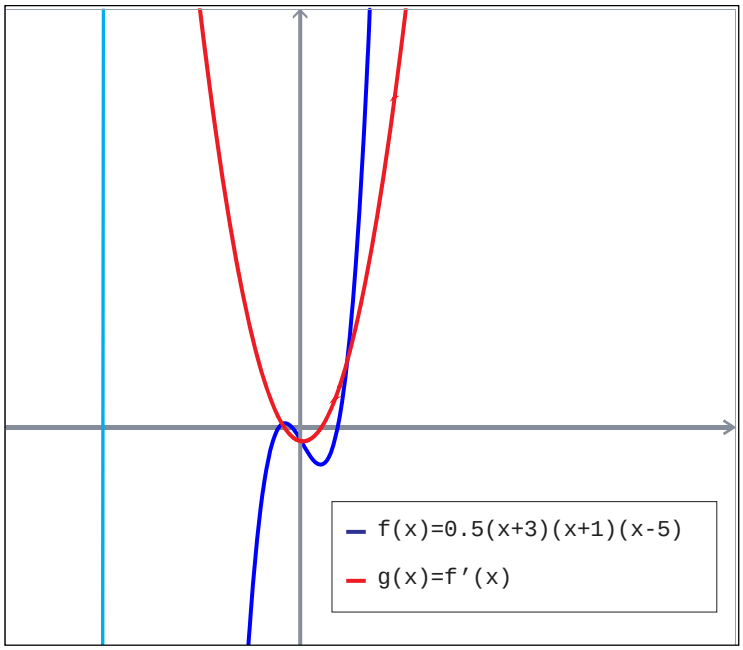


Figure 4b: A graph of functions f(x) and g(x)

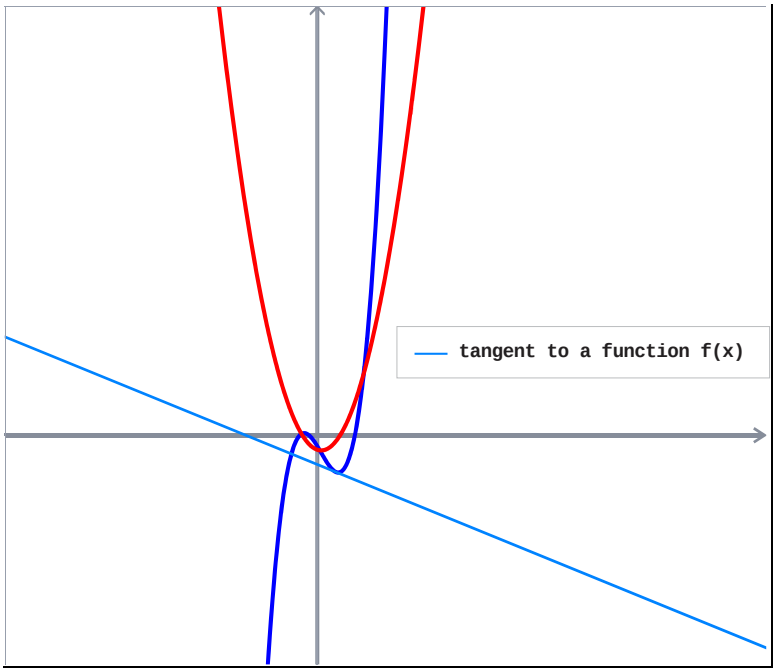


Figure 4c: A tangent to a function f(x)

4 CindyLab

As we mentioned in the Introduction, 'CindyLab' is a part of 'Cinderella' which enable us to create and simulate physical experiments.

Similar to geometric part of 'Cinderella', drawing a physical experiments means adding a objects such as mass, free points, gravity, walls, sun, fixed planets, etc. As soon as a physical object is added, an animation control panel appears in the geometric view. Pressing the play button starts the physics simulation. If a velocity was assigned to the mass, the mass will start moving automatically. If the mass is influenced by forces or from Gravity, the velocity of the mass will change according to the force while the mass is moving.

The basic properties of all physics objects can be set and changed using the physic tab of the Inspector. In 'Cinderella' software Insector presents a central controlling unit for all properties of a configuration and its objects, and physics insector for generic a mass looks as on a next Figure. The following table is given to show the main object for doing physics experiments in 'CindyLab'.

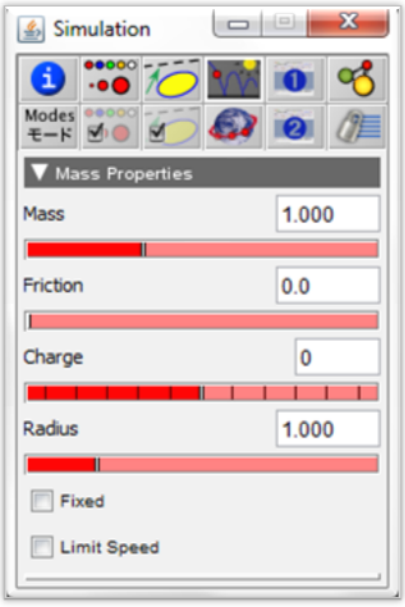


Figure 5: Physics Inspector in 'CindyLab'

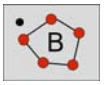
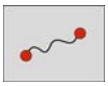
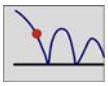
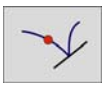
	Free Mass
	Velocity
	Gravity
	Sun
	Magnetic Field
	Spring
	Floor
	Bouncer

Table 1: Physics objects in 'CindyLab'

In the following picture we give an example of a physics simulation with ‘CindyLab’**.** We are modeled a planet orbiting a sun, i.e. Kepler low of planetary motion.

Figure 6: Overview of animation of Kepler law of planetary motion in ‘CindyLab’

Acknowledgements:

This research was supported by the Grant 19/6-020/961-120/14 of the Ministry for Science and Technologies of the Republic of Srpska.

REFERENCES

[1] J. Richter-Gebert, “Perspectives on Projective Geometry”, *Springer-Verlag Berlin Heidelberg*, (2011).

[2] Veljko Vranić, Đorđe Baralić, “Cinderella – način da vidimo apstraktnu matematiku”, *Zbornik radova sa Treće matematičke konferencije Republike Srpske*, **2**, 69-77 (2014).

[3] Jürgen Richter-Gebert, Ulrich H. Kortenkamp "The Cinderella.2 Manual Working with the Interactive Geometry Software", *Springer-Verlag Berlin Heidelberg*, (2012)

Received July, 30 2015.