Original scientific paper

http://doi.org/10.24867/JGED-2020-1-037

A mathematical approach for creative graphics design

Panchanand Jha [1]
Bibhuti Bhusan Biswal [2]

[1] Raghu Engineering College, Department of Mechanical Engineering, Visakhapatnam, India
[2] National Institute of Technology, Department of Industrial Design, Rourkela, India

Corresponding author:
Panchanand Jha
e-mail:
dr.panchanandjha@gmail.com
bbbiswal@nitrkl.ac.in

ABSTRACT

Mathematics is most beautiful thing in our daily life but when it comes to equations and formulae people seems to be unpleasant. This equations and formulae exhibits interactive properties and relation in our daily life. It is sure that the almost all beautiful thing in our nature even Universe unveils mathematical properties. These properties produce many patterns and shape and is directly related to geometric algebra as we can see in nature for example, fluorescent green beetle and colourful galaxies. These gorgeous properties of nature can be transformed into equations and logics. Therefore, the main aim of this paper is to generate creative graphics using mathematical equations such as number theory, position vectors and trigonometry functions. The sectional organization of this paper is as follows: section 1 gives the introduction of interactive graphic design and related work. Mathematical approach for graphic design has been briefed in section 2. In section 2 adopted algorithms and Graphical visualization of design is given. Graphic designs are applied to Indian ethnic wear and T-shirts. Last Section gives the conclusions and further scope of this work.

KEY WORDS
Design, Collatz conjucture, double pendulum

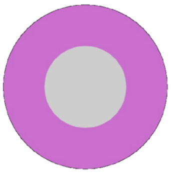
First recieved: 25.12.2019.
Accepted: 2.2.2020.

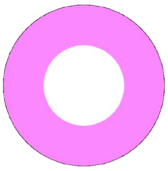
Introduction

Creative graphics using computer program can help designer to have control over the produced image or art i.e. the designer can edit or change the art. This manipulation of design can be achieved either by using control tool/library or by using command through input device. The major advantage of using creative design are: 1) Achieve high quality of art, 2) Precise results and accuracy, 3) Higher productivity and lower design cost, 4) Less computation time etc. You can't step much ahead in computer graphics programming without comprehensive math abilities. Without math, graphical programming stays limited to things, for example, changing shades of objects or making a few items through code for entertainment only. Codes and graphics libraries are only tools; mathematical skills and creative imagination can be transformed into virtual reality using computer programming. Trigonometry, Matrices, Linear Algebra, and Calculus are an unquestionable requirement (Blair-Early & Zender, 2008). The computer created visual design and arts can be looking realistic as we have experiencing in our daily life. There are various software's for rendering, animations, visual arts such as Terragen software used to create landscape, blender, Houdini, Sketchup, etc. and apart from these software's, computer programs can also be used for generating graphics design/art. Matlab, Python, Java, JavaScript etc. are popular programming languages which are widely used by scientific community (Neumann, 2019; Sugimoto & Nagasawa, 2017; Stojanović, 2016). This current research work is dealing with processing.js and p5.js web editor tool based on JavaScript for generating the creative graphics. Processing.js and p5.js is multi-functionality, collective with user friendly interfacing, various graphic libraries which help user to reduce time and enhance creativity (McCarthy, Reas & Fry, 2016; Reche-Martinez, Martin & Drettakis, 2004).

The JavaScript library have various options to play with image, sound, video, text etc. and also great lighting, hue, alpha value which helps designer to showcase their creativity. Table 1 shows example code for creating circle with multiple colours and alpha values and it can be scene that centre of circle is charcoal grey that is extracted from the color(250, 126, 255, 200). Similarly there are various commands for pixel array such as blue(), brightness(), green(), saturation() and hue(). These lines of code are user-friendly and can be edited as per requirement. Similarly for creating various shapes with user dependent size, there are commands known as 2D primitives, attributes, curves, vertex, 3D primitives and 3D models and these library includes different command lines for example 2D primitives have arc(), ellipse(), circle(), line(), rect() etc. More details about processing.js can be found in references (McCarthy, Reas & Fry, 2016).

Table 1
Sample code for generating circle

function setup() {
  let canva= createCanvas(400, 400);
  noStroke();
  let c = color(250, 126, 255, 200);
  fill(c);
  ellipse(200, 200, 200, 200);
  let value = alpha(c); // Sets ‘value’ to 102
  fill(value);
  ellipse(200, 200, 100, 100);
 	
}
	

function setup() {
  let canva= createCanvas(400, 400);
  noStroke();
 let c = color(250, 126, 255, 200);
 fill(c);
 ellipse(200, 200, 200);
 let value = alpha(c); // Sets ‘value’ to 102
 fill(value);
 ellipse(200, 200, 100, 100);
 	
}
	

Mathematical approach for graphic design

Math is useful for graphic design for example design of web page, chocolate wrappers, billboard, packaging, printed T-shirts etc. Mathematics play crucial role for designing things in our daily life products. It can be hard like calculus or may be basic intermediate algebra but the graphics design has to go through it. Golden ratio, Fibonacci number, number theory etc. are the most common mathematical techniques that are used to design webpages, computer graphics etc. Creative and innovative ideas for designing products, industry related modules even graphics are dependent on mathematical theories. Most of the tools or computer programming languages are equipped with basic algebraic equations and many times we don’t even need to write those equations (Pickover, 2001; Silva, 1999; Vince, 2017; Ward, 1991; Mantiuk, & Heidrich, 2009).

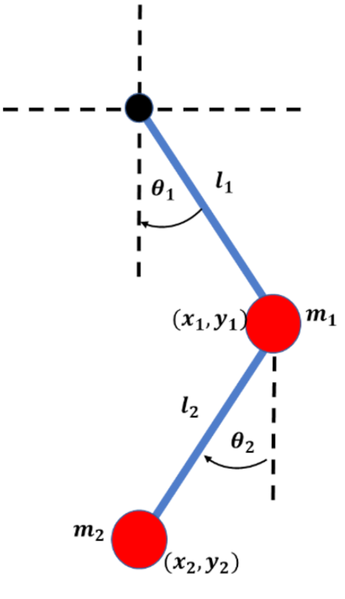
As we have seen in Table 1 that creating a circle need a simple command ellipse(x, y, w, h), but when it comes to parametric design than mathematical equations help a lot for precise outputs. Therefore, in this article two different and known theories of mathematics have been adopted for the graphic design. These theories are based on kinematic equations of double pendulum and recursive number theory (Collatz conjecture).

Double pendulum

Kinematics of double pendulum has been used to create random graphics using processing.js JavaScript language. Position coordinate of second rod has been used to draw random circle with dimension based on the current position and orientation. Brief introduction and kinematics equations have been discussed below.

Kinematics of double pendulum

A double pendulum contains two links with two degrees of freedom, means two joint variables(θ_1 and θ_2) required to describes the position and orientation of the end point (x, y) of pendulum as shown in figure 1. The position and joint variables can be calculated from kinematics equations (Jankowski, 2011). Kinematics of any mechanism can be defined as calculation of position, orientation, velocity, acceleration, displacement etc. without considering the forces or mass. Kinematics of upward rod can be given by trigonometry equation (1) through (10).



» **Figure 1:** *Double pendulum*

x_1, y_1= position of mass m_1
x_2, y_2= position of mass m_2
l_1 and l_2=length of rods
θ_1, θ_2 = joint angle of rod l_1 and l_2

$$x_1 = l_1 sin\theta_1 \qquad (1)$$

$$y_1 = -l_1 cos\theta_1 \qquad (2)$$

$$x_2 = x_1 + l_2 sin\theta_2 \qquad (3)$$

$$y_2 = y_1 - l_2 cos\theta_2 \qquad (4)$$

$$\dot{x}_1 = \dot{\theta}_1 l_1 cos\theta_1 \qquad (5)$$

$$\dot{y}_1 = \dot{\theta}_1 l_1 sin\theta_1 \qquad (6)$$

$$\dot{x}_2 = \dot{x}_1 + \dot{\theta}_2 l_2 cos\theta_2 \qquad (7)$$

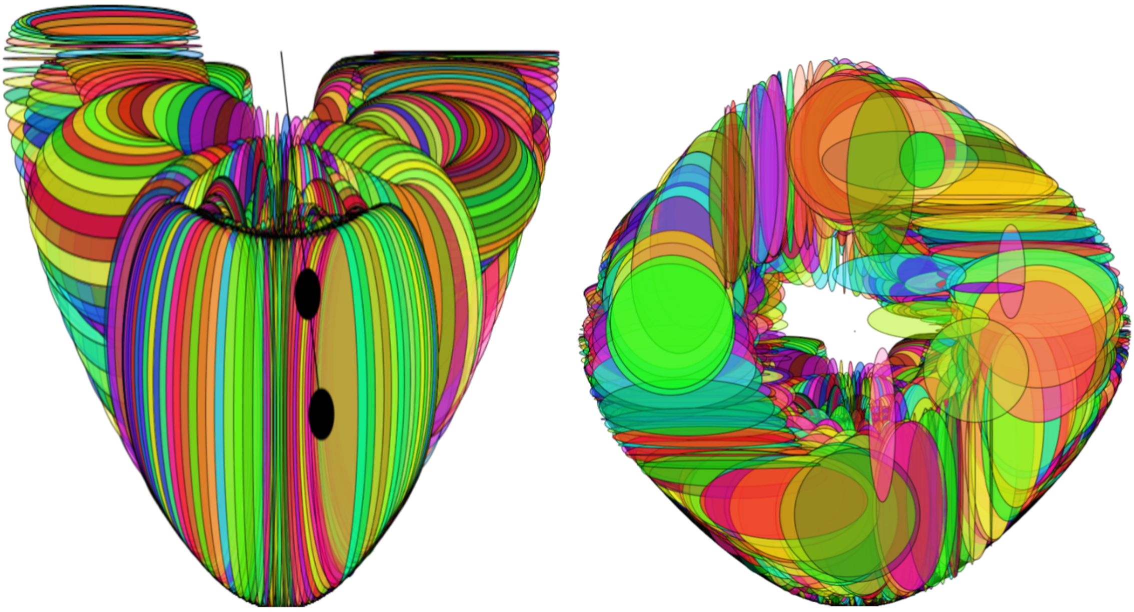
$$\dot{y}_2 = \dot{y}_1 + \dot{\theta}_2 l_2 sin\theta_2 \qquad (8)$$

Equation 1 and 2 are the displacement equations and derivative of it gives velocity with respect to time.

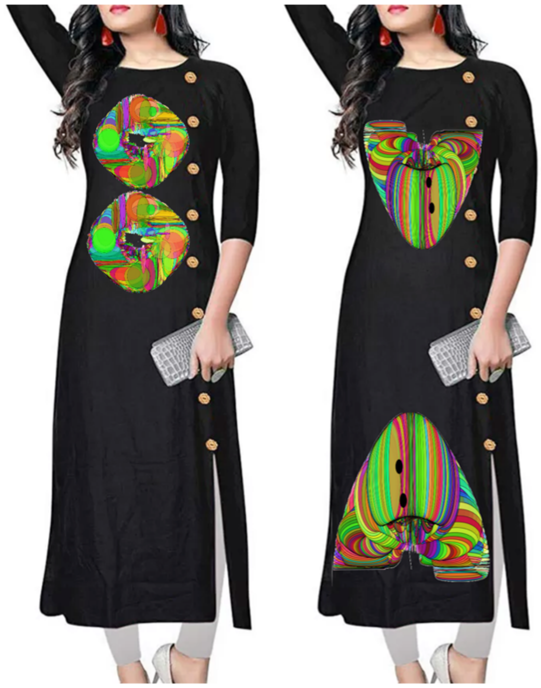
The acceleration can be found by differentiating velocity equation (5) through (8) and equation (9)-(10) can be used in programming to accelerate the pendulum. Figure 2 and 3 is the result of oscillating pendulum with tracing of position (x_2,y_2) of mass m_2. The position of the mass 2 is varying with the velocity and accelerating and which causes to change the dimension of circle. Figure 2 is made using double pendulum code in processing.js JavaScript in which circle at oscillating rod 2 has been attached. The possible application of creative graphics has been used in Indian traditional Kurti as shown in figure 3. The JavaScript library for circle is ellipse **(x, y, w, h); where x** and **y** is pixel location **w, h** are width and height of the ellipse. The width and height is changing with respect to the current position of the mass 2 and that causes to generate the various shape of ellipses. The color of each circle is continually changing with the JavaScript command fill **(R, G, B, Alpha)**, the values of RGB can be randomly given from 0 to 255 and which gives random color in the ellipses.

$$\ddot{\theta}_1 = \frac{-g(2m_1+m_2)sin\theta_1-m_2gsin(\theta_1-2\theta_2)-2\sin(\theta_1-\theta_2)m_2(\dot{\theta}_2^2 l_2+\dot{\theta}_1^2 l_1\cos(\theta_1-\theta_2))}{l_1(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))} \qquad (9)$$

$$\ddot{\theta}_2 = \frac{2\sin(\theta_1-\theta_2)\left(\dot{\theta}_1^2 l_1(m_1+m_2)\right)+g(m_1+m_2)cos\theta_1+\dot{\theta}_2^2 l_2 m_2\cos(\theta_1-\theta_2))}{l_2(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))} \qquad (10)$$



» **Figure 2:** *Shown below is generated through the concept of double pendulum*



» **Figure 3:** *Double pendulum generated graphics imprinted on Indian Kurti*

Collatz conjecture (3*n+1) algorithm

The Collatz conjecture is mathematical estimation or guesswork that proposed in 1937 by Lothar Collatz, (O'Connor & Robertson, 2006). It is otherwise called the (3×n+1) conjecture, the Ulam guess, the Kakutani's concern, the Thwaites conjecture, or the Syracuse issue, (Olloh, 2018). Many mathematical researchers are trying to theorized this problem of conjecture but it is recursive in nature and leads to again an estimation work. The basic pseudocode is given in Table 2.

Table 2

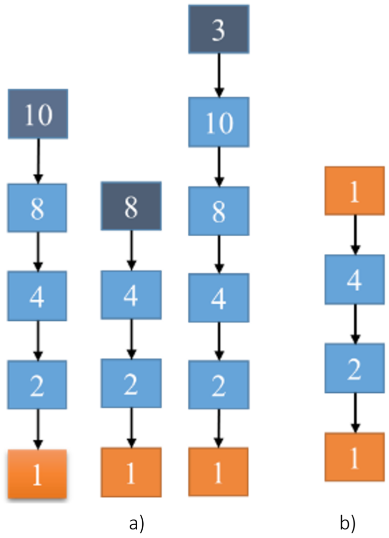
Pseudocode of Collatz Conjecture

Pseudocode

```
Begin Collatz(n) {
// if the number is even
  if (n /2 = = 0) {
    return n / 2;
    // if the number is odd
  } else {
    return (n * 3 + 1)/2;
  }
}
```

The Collatz Conjecture is that regardless of what the number (n) is taken, the procedure will in every case inevitably arrive at 1 see the Figure 2. The large iteration of collatz conjecture from 10 billion to 100 quadrillion are determined by Leavens & Vermeulen (1992) and Crandall (1978) individually. Ren et al. (2018) checked $(2^{100000}-1)$ can come back to 1 after multiple times of (3×n+1) calculation, and multiple times of n/2 calculation, which is the biggest number being confirmed so far (Krasikov & Lagarias, 2003). So far nobody has attempted to make sense of whether the entirety of the positive numbers in the end arrive at one, yet we realize that the majority of them do as such. Specifically, Krasikov & Lagarias (2003) demonstrated that the quantity of numbers at last arriving at one in the interim [1, n] is at any rate relative to x 0.84. Therefore, large number sequence can be generated from this algorithm and can be used to create graphics out of these numbers as a pixel.

Continue applying this rule until you arrive at a revising sequence of numbers. For instance, if we start with both of the numbers 3 or 10 and apply this rule it ends at 1 see figure 4 (a) but again 1 is an odd number so it a repeating sequence see figure 4 (b). Therefore, computer programming is efficient approach to generate number sequence and these number sequence can later be used to generate the graphics and with little modifications on available library it can leads to various graphic designs (Swimmingthestyx, 2019).



» **Figure 4:** *Number sequence of Collatz conjecture*

Once the pseudo code is implemented in programming language such as JavaScript it became more user-friendly to generate various design. The genuine fun of Collatz is mapping it. By expanding the quantity of number sequence and twisting impact, graphics start to get entirely magnificent. It can be seen in Table 3. The designs are made utilizing p5.js JavaScript. Research has been made on the code and tried to adjust the parameters by hit and trial to get better graphics.

The different graphics have been achieved using random command in JavaScript. From Table 4 it can be seen that manipulating the values of random numbers as shown in row number 13, 14, 20, 22 (highlighted) can create different designs. Table 5 gives various possible application of the Collatz conjecture theory and is imprinted on T-shirts.

Table 3 (first part)

Visualization of modified collatz conjecture

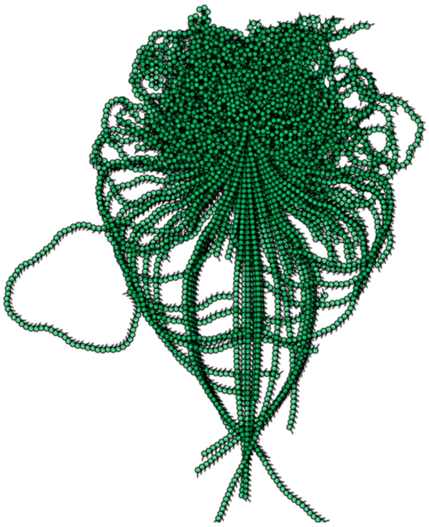
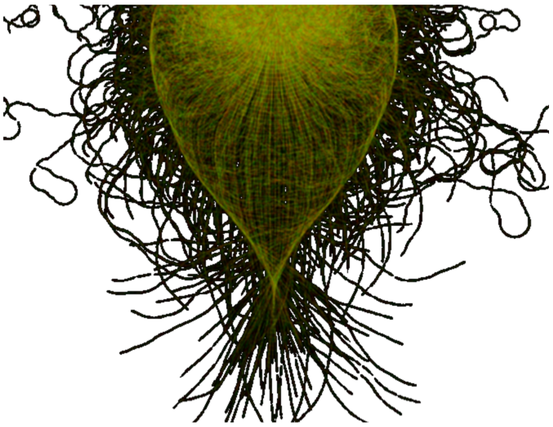
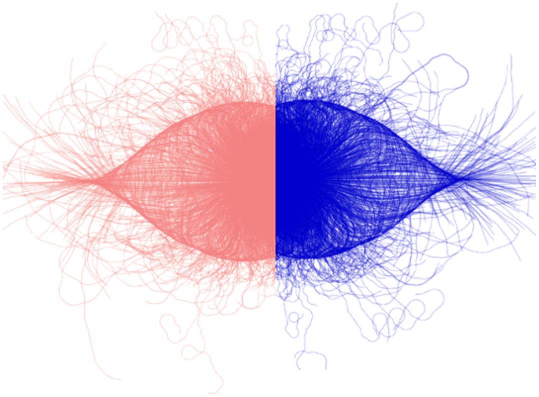
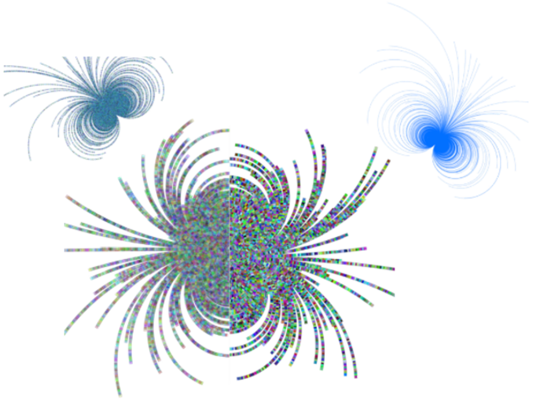
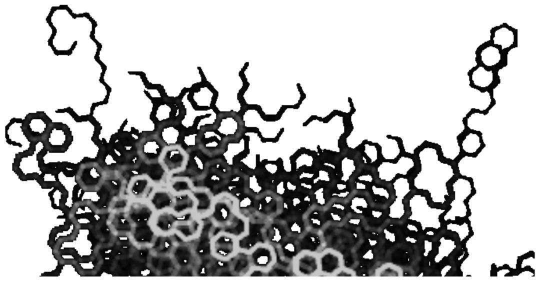
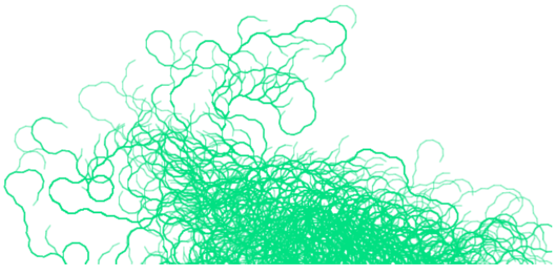
	
	
	

Table 3 (second part)

Visualization of modified collatz conjecture

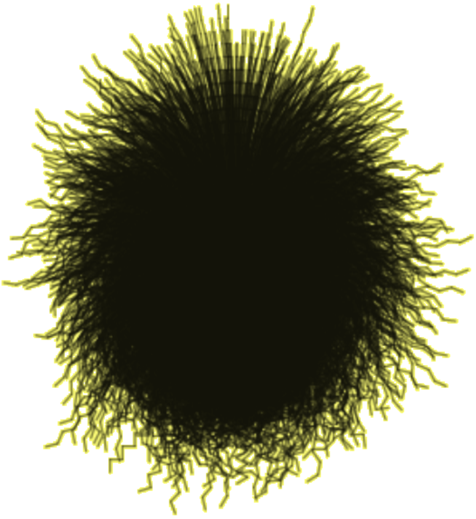
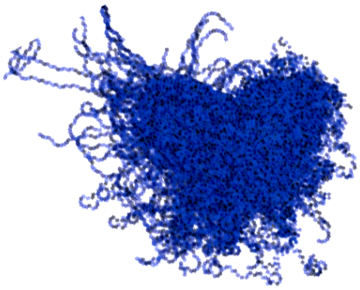
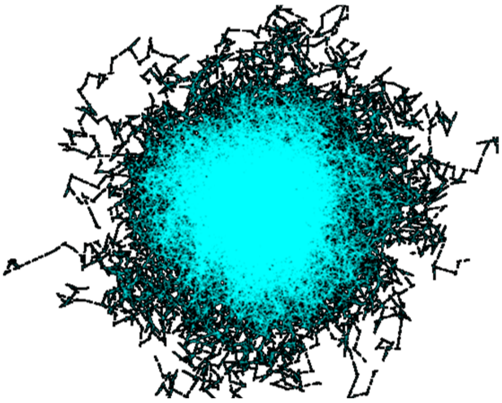
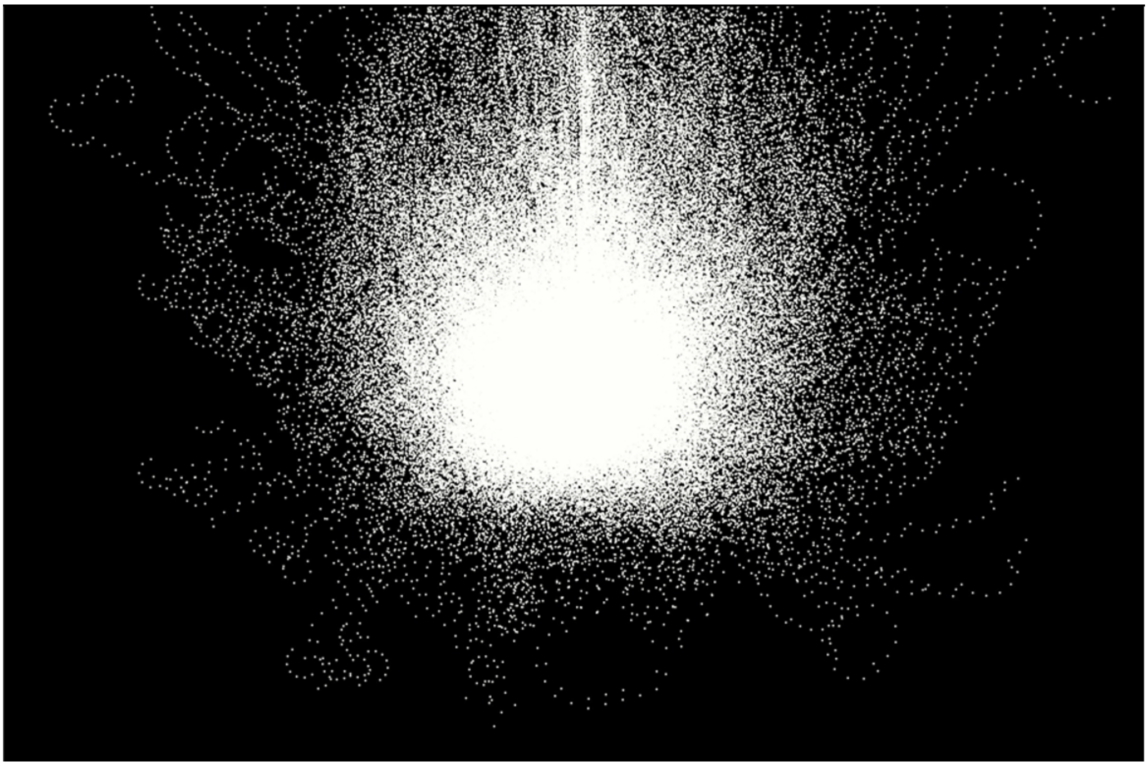
	
	

Table 3 (third part)
Visualization of modified collatz conjecture



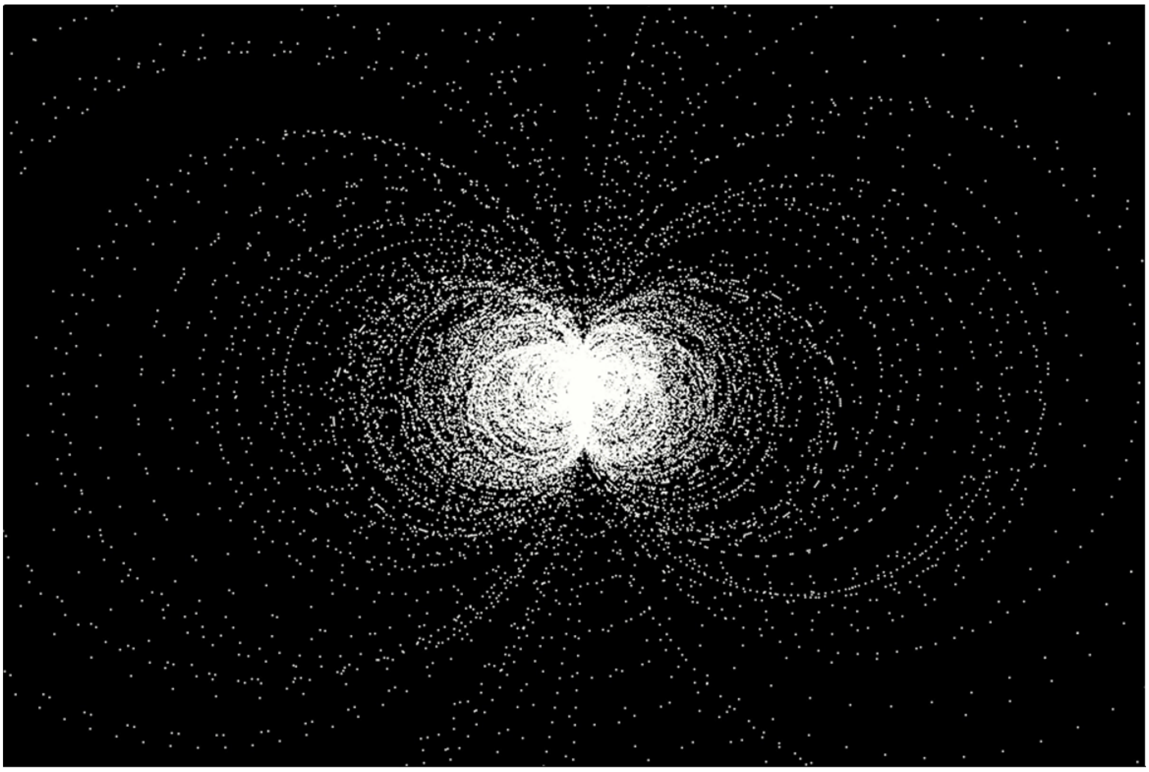
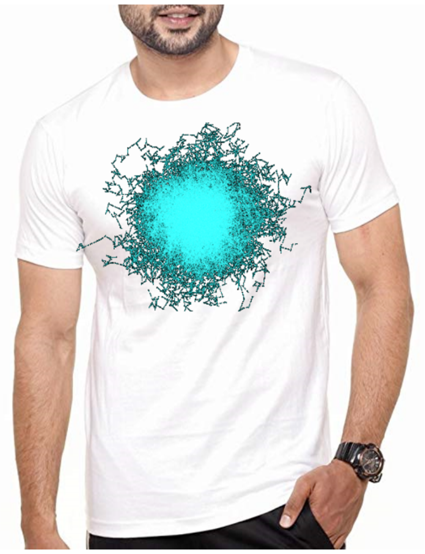
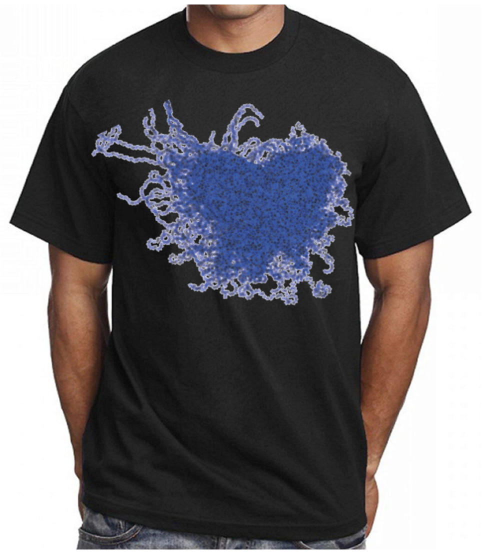
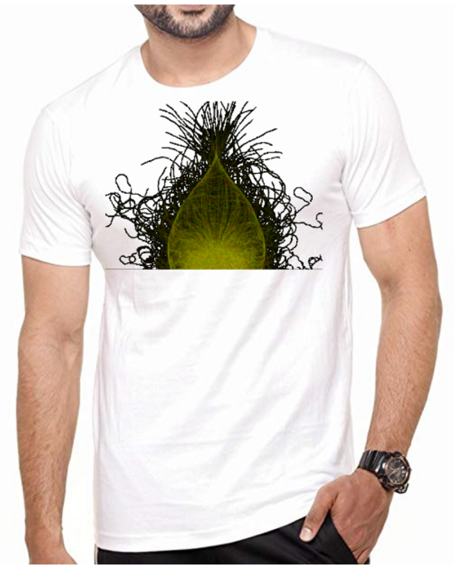
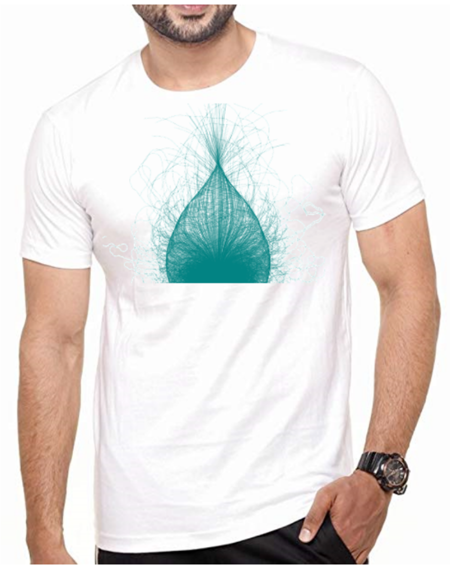
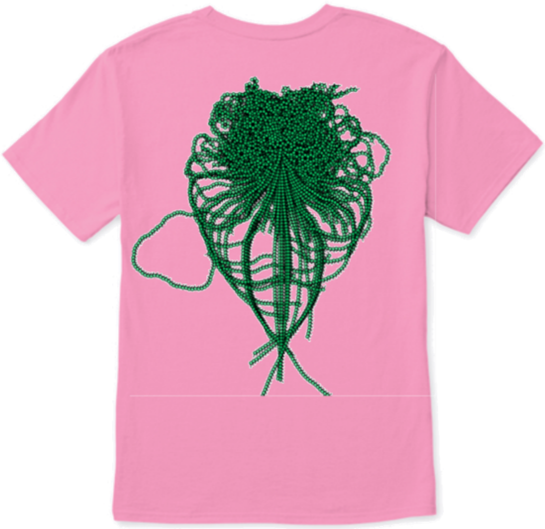
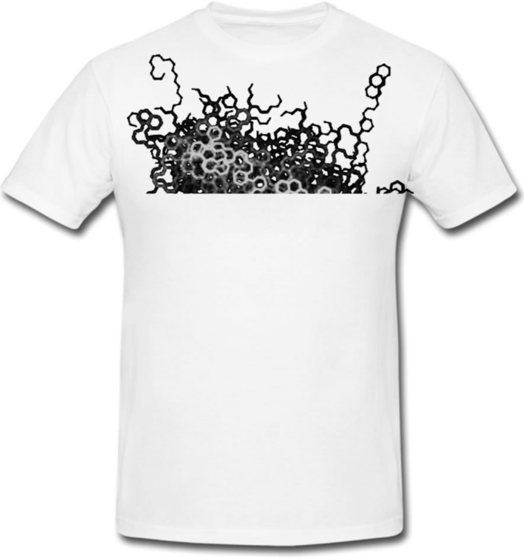


Table 4

Processing.js JavaScript code for Collatz conjecture

```
void setup() {
size(1800, 800);
  background(255, 255, 255);
  for (int x = 1; x < 10000; x++) {
IntList seq = new IntList();
int n = x;
do {
seq.append(n);
n = colla_tz(n);
} while (n != 1);
seq.append(1);
seq.reverse();
float l = random(10);
float theta = random(-100, 100);
resetMatrix();
translate(width/2, height/2);
for (int y = 0; y < seq.size(); y++) {
int val = seq.get(y);
if (val % 2 == 0) {
rotate(tan(10*sin(theta)));
} else {
rotate(tan(cos(-theta)));
   }
strokeWeight(1);
stroke(0, 0, 0, 100);
fill(random(255), random(250), random(250), 150);
point(10, 10);
translate(0,-l);
      }
    }
 }
int colla_tz(int n) {
  // even
  if (n % 2 == 0) {
return n / 2;
// odd
} else {
return (n * 3 + 1)/2;
  }
}
```

Table 5
Possible applications of graphics design

Conclusion and further scope

Many of us never know from where the design comes from in our day to day life. It can be different structures made on clothes, a photo picture on boards, even on magazines there can be plenty of images. These creative designs that are on computer or piece of paper is the basis of visual graphic designs. In this work graphic design is made out of mathematical logic and shown a potential application. The printed T-shirt and Indian Kurti is an example of the use of creative computer graphics design. Behind all these designs mainly two methods which are kinematics of pendulum and recursive number theory. The printed T-shirts and Kurti's seem to be pleasant. These mathematical approach can also be used in various fields such as website design, magazine covers, bags, toys etc. There are various other mathematical algorithms such as golden ratio, Fibonacci number, fractal tree etc. can be adopted for creating graphics designs.

References

Blair-Early, A. & Zender, M. (2008) User Interface Design Principles for Interaction Design. Design Issues. 24 (3), 85-107. Available from: doi: 10.1162/desi.2008.24.3.85

Crandall, R. (1978) On the "3x + 1" Problem. Mathematics of Computation. 32 (144), 1281.

Silva, T. (1999) Maximum excursion and stopping time record-holders for the problem: Computational results. Mathematics of Computation. 68 (225), 371-385. Available from: doi: 10.1090/S0025-5718-99-01031-5

Jankowski, K. (2011) Dynamics of double pendulum with parametric vertical excitation. MSc thesis. Technical University of Lodz.

Krasikov, I. & Lagarias, J. (2003) Bounds for the 3x+1 problem using difference inequalities. Acta Arithmetica. 109 (3), 237-258. Available from: doi: 10.4064/aa109-3-4

Leavens, G. & Vermeulen, M. (1992) 3x+1 search programs. Computers & Mathematics with Applications. 24 (11), 79-99. Available from: doi: 10.1016/0898-1221(92)90034-F

Mantiuk, R. & Heidrich, W. (2009) Visualizing High Dynamic Range Images in a Web Browser. Journal of Graphics, GPU, and Game Tools. 14 (1), 43-53. Available from: doi: 10.1080/2151237X.2009.10129276

McCarthy, L., Reas, C. & Fry, B. (2016) Make: Getting started with p5.js. San Francisco, Maker Media, Inc.

Neumann, E. (2019) myPhysicsLab Double Pendulum. Available from: https://www.myphysicslab.com/pendulum/double-pendulum-en.html [Accessed: 22nd December 2019].

O'Connor, J.J. & Robertson, E.F. (2006) Lothar Collatz. Scotland, St Andrews University School of Mathematics and Statistics.

Olloh, J. (2018) On the Fundamentals of Collatz Conjecture. SSRN Electronic Journal. Available from: doi: 10.2139/ssrn.3302210

Pickover, C. A. (2001) Wonders of Numbers. Oxford, Oxford University Press.

Reche-Martinez, A., Martin, I. & Drettakis, G. (2004) Volumetric reconstruction and interactive rendering of trees from photographs. ACM Transactions on Graphics. 23 (3), 720-727. Available from: doi: 10.1145/1186562.1015785.

Ren, W., Li, S., Xiao, R. & Bi, W. (2018) Collatz Conjecture for 2^100000-1 Is True- Algorithms for Verifying Extremely Large Numbers. In: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, 8-12 October 2018, Guangzhou, China. New Jersey, IEEE. pp. 411-416. Available from: doi: 10.1109/SmartWorld.2018.00099

Stojanović, N. (2016) A method for generating stochastic 3D tree models with Python in Autodesk Maya. Journal of Graphic Engineering and Design. 7 (2), 25-30. Available from: doi: 10.24867/JGED-2016-2-025

Sugimoto, K. & Nagasawa, S. (2017) Cause and Effect of Design Features and Brand Value: Consumer Interpretation of Design and Value of Long- and Short-Term Products. The Design Journal. 20 (sup1), S4213-S4226. Available from: doi: 10.1080/14606925.2017.1352920

Swimmingthestyx.com. (2019) Graphing the Collatz Conjecture | Swimming the Styx. Available from: http://swimmingthestyx.com/?p=447 [Accessed 22nd December 2019].

Vince, J. (2017) Mathematics for computer graphics. London, Springer.

Ward, G. (1991) A recursive implementation of the perlin noise function. In: Glassner, A.S. (ed.) Graphics Gems II. The Graphic Germs Series: A Collection of Practical Techniques for the Computer Graphic Programmer. San Diego, Academic Press, pp. 396-401. Available from: doi: 10.1016/B978-0-08-050754-5.50087-6