

Incorporating  
Scientific Research Methods  
into  
High School Programming Classes

Shawn Liner

RET LSU 2015

## Contents

The Purpose of this Document: .....	3
Skills Requisites for the materials .....	3
Starter Projects: .....	3
Simple Numeric.....	3
List .....	3
Next step, but still introductory.....	3
Science Based Projects:.....	4
Random Path Generator: .....	4
Collatz Conjecture: .....	4
Terminal Velocity: .....	4
Scientific Algorithms .....	5
Least Squares Fit .....	5
Monte-Carlo Integration .....	5
What is the Monte-Carlo Method?.....	5
An Example Monte-Carlo Integration: .....	5
Problem:.....	6
Extensions: .....	7
Monte-Carlo Calculation of Pi ( $\pi$ ) .....	8
What is the Monte-Carlo Method?.....	8
Finding $\pi$ using Monte-Carlo Integration:.....	8
Problem:.....	8
Extensions: .....	9
Newton-Raphson Method: .....	10
What is the Newton-Raphson Method? .....	10
An Example Newton-Raphson Method: .....	10
Problem:.....	10
Extensions: .....	12
Warnings: .....	12

## The Purpose of this Document:

Perhaps the best start is to tell what this documents is **not**. It is not intended to teach programming. It is instead strictly a resource for teachers looking for that kick-start or for interesting projects for their programming classes that have scientific bases and logic.

## Skills Requisites for the materials

Students are expected to have (or be in the process of learning) a base set of skills including.

- Variables
- Get input
- Print to screen
- Loop
- Functions
  - Inside current file
  - From separate file
- Math functions
  - Power
  - Modulus
- Graphics
- Read/Write to File

## Starter Projects:

These projects are short and simple, but are good introductory ideas:

### Simple Numeric

- Check to see if a number is even or odd  
This is a good place to introduce modulus.
- Check to see if a number is prime.  
Also a good place to use modulus
- Is the number a power of two?
- Calculate the factorial of a number
- Check to see if an entry is a palindrome
- Add two fractions and return the answer written as a fraction

### List

These are good ideas for working with lists of numbers. Something that scientists often find themselves doing.

- Find Sum
- Find Average
- Find the standard deviation and standard error in the mean
- Sort an entry of numbers

### Next step, but still introductory

- A program that drops the vowels from a sentence.  
(Great for working with lists, and conditional branching)
- Create a cipher program  
A cipher program is a simple replacement encryption. The most common is an alphabetical shift.

That is, A is replaced with B, and so on until we get to Z is replaced with A. A slightly more advanced version could take the ascii code of a number and multiply it by successive digits of  $\pi$ . So the first letters ascii code would be multiplied by 3, the next by 1, then 4, etc. However, there's no rule that you have to start on the first digit. The chosen start digit could be the secret that is needed to decipher the code.

- Dice Roll / Craps:

The first step is quite simple and a good introduction to random numbers. However, the steps get much harder when you start to introduce the rules of craps.

### Science Based Projects:

#### Random Path Generator:

In this project students will have the code pick a random direction and random distance to walk a given number of times, say 100. In order to do this students will need to learn about adding vectors by components.

This project is interesting because of the questions it can bring. Where can the object never end up? Where is it most likely to end up? These questions bring us to the idea of atomic orbitals.

#### Collatz Conjecture:

The Collatz conjecture says that if you start with an integer and follow two simple rules you will always end up with 1. 1) If the number is even, you divide by 2 2) if the number is odd, you triple it and add 1. Have them check to see if it's true. How many tries do they need to do before they 'prove it'?

#### Terminal Velocity:

Terminal velocity, the maximum speed an object will obtain while falling, is difficult to calculate directly. There are two forces competing with each other, the weight of the object and gravity. This project is not for the faint of heart. The programming is relatively easy, but you might want to find a physics teacher friend to get the physics right.

The force of gravity is simply the weight of the object so if a ball weighs 4N, that force is pulling it to the ground.

The drag force is more complicated and is a function of several things. A simple version might be:

$$D = \frac{1}{2} C \rho A v^2$$

$D = \text{drag force}$

$C = \text{drag coefficient}$

$\rho = \text{density of air, roughly } 1 \frac{\text{kg}}{\text{m}^3}$

$A = \text{cross sectional area} = \pi R^2 \text{ (for a ball)}$

$v = \text{speed that the ball is falling}$

Have them find the speed of a falling object at a given time. You will have to give them (make up) a drag coefficient.

A good list is found here. [http://www.engineeringtoolbox.com/drag-coefficient-d\\_627.html](http://www.engineeringtoolbox.com/drag-coefficient-d_627.html)

You must use Newton's second law.

$$\sum F = ma = m \frac{\Delta v}{\Delta t}$$
$$\frac{(Weight - Drag)}{m} \times \Delta t = \Delta v$$
$$v_f = \frac{Weight - \frac{1}{2} C \rho A v_i^2}{m} \times \Delta t + v_i$$

So you calculate a new speed ( $v_i$ ) by using the old speed ( $v_i$ ) and a selected time interval ( $\Delta t$ ). The time interval is the trick here. The smaller it is, the more accurate your answer. If we plug in 10s, we will get an estimate of the speed at 10s. However, it will not be very accurate. A better method would be to do 10 iterations each with a  $\Delta t$  of 1s.

The graph to the right shows a run from the example program. You can see that the speed approaches 45m/s. This would be the terminal velocity for the object.

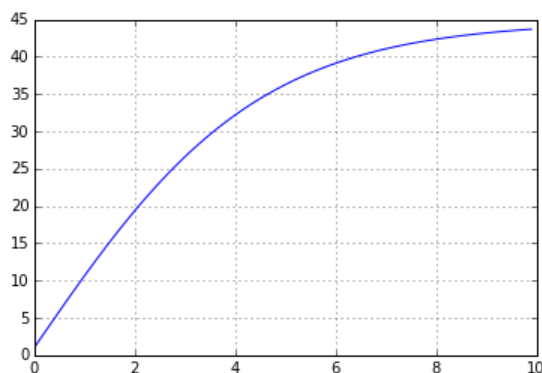


Figure 1 Graph of velocity for a falling object from example program

## Scientific Algorithms

The following are methods of taking difficult math problems and solving them, even if we don't know how to do the algebra. These methods are frequently used in science, but seldom heard of at the high school level. As such, I've written much longer descriptions and included sample code in the zip file.

### Least Squares Fit

Least squares fit is a common practice in science. It is the method of finding a linear equation to relate to variables to each other. It is included in Excel, and on most programmable calculators. There are many sites dedicated to the least squares fit. You could give them data and ask them to find a best fit equation. They will, of course have to find the algorithm.

### Monte-Carlo Integration

#### What is the Monte-Carlo Method?

Like its name implies the Monte-Carlo method implies randomness or chance. (Monte-Carlo is a casino in the Country of Monaco.) Generally speaking they are performed by taking large numbers of random samples. Thus they lend themselves quite well to computer programming project.

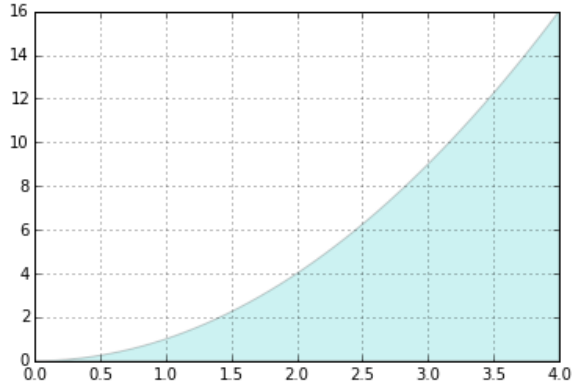
#### An Example Monte-Carlo Integration:

In this project we will ask students to find the area under a curve, (Integration). Calculus is not required for this exercise. However, if some of your students have had calculus, you may have to ask them to hold their tongue as you discuss it with other students. Alternately, you can ask

them to us the calculus to 'check' your answers. This is a common calculation performed in science.

**Problem:**

What is the area under the curve  $f(x)=x^2$  from  $x=0$  to  $x=4$ ? (See the shaded area on the graph to the right).



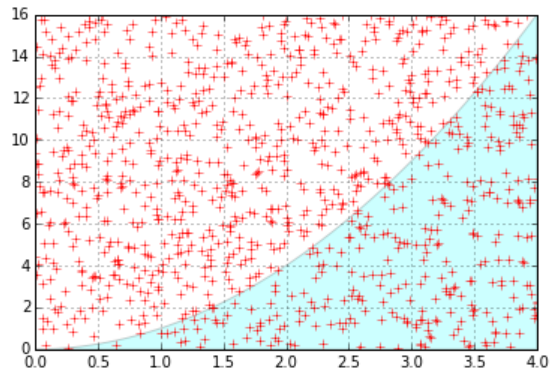
**Monte-Carlo Solution:**

- Let's define an area that includes the area we're searching for, but is easy to calculate. In this case the rectangle created by the four points (0,0), (4,0), (4,16), and (0,16). (Note that the point (4,16) is on the function we are examining). This area is easy to calculate.

$$A_{tot} = b * h = x_{max} * f(x_{max}) = 4 * 16 \text{ (in this case)}$$

It is important to see here that  $b = x_{max}$  and  $h=f(x_{max})$ . In other words, the function defines the boundaries we use to create our rectangle.

- Now we pretend to shoot darts at the graph with some assumptions.
  - All darts will hit the graph
  - There is complete randomness to the shots (There is no skill that would increase the chance of hitting the middle, or in the shaded area)
  - The odds of landing in a shaded region or non-shaded region is directly related to the mathematical area of the region.



- The trick here is to recognize that the ratio(R) of areas is related to the ratio of the number of shots that hit in the shaded area.

$$R = \frac{n_{shots \text{ in shaded area}}}{n_{total}} = \frac{a_{shaded}}{a_{total}}$$

Which leads to

$$a_{shaded} = R * a_{tot} = R * x_{max} * f(x_{max})$$

**Programming Methodology:**

- Take a shot (Pick a random point on the graph area)

- Test to see if it's in the shaded area.  
(Easiest way is to compare your random y value to the function value of your x value.)
- Repeat many times
- Use the ratio of shots in the shaded area to total shots to find the area of the shaded area.

*Mathematical Solution:*

It turns out that the mathematical version of this specific example is actually quite simple. However, it involves calculus. The integral of  $y=x^2$  is  $(1/3)x^3$ .

$$Area = \int x^2 dx = \frac{1}{3}x^3 = \frac{1}{3}4^3 = 21.33$$

So the area under  $y=x^2$  from zero to  $x=4$ , is 21.33. I really only offer this so that you can check your results.

*Extensions:*

- Ask for input from the user as to the range to find the area under. (instead of 4 in our example)
- Ask students to modify the program so that any function can easily be entered. (ie  $y=x^3$ ,  $y=\cos(x)$ , etc)
- Calculate  $\pi$  from the area under  $y=\sqrt{x^2 + r^2}$ . This extension is covered later as a separate activity.

## Monte-Carlo Calculation of Pi ( $\pi$ )

### What is the Monte-Carlo Method?

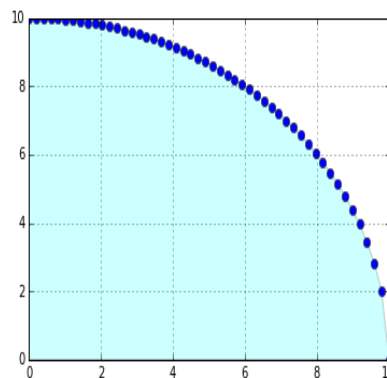
Like its name implies the Monte-Carlo method implies randomness or chance. (Monte-Carlo is a casino in the Country of Monaco.) Generally speaking they are performed by taking large numbers of random samples. Thus they lend themselves quite well to computer programming projects.

### Finding $\pi$ using Monte-Carlo Integration:

This exercise is an extension of Monte-Carlo Integration covered in a previous project. In this project we will ask students to find the area under a portion of a circle. Because we know the area of a circle is related to pi, we can use this method to find pi. (My students have found this to be a very cool exercise because pi is such a mysterious number.)

### Problem:

What is the value of pi ( $\pi$ )?



### Monte-Carlo Solution:

4. First let's recognize that the portion of the circle we're looking at is a function.

$$y = \sqrt{x^2 + r^2}$$

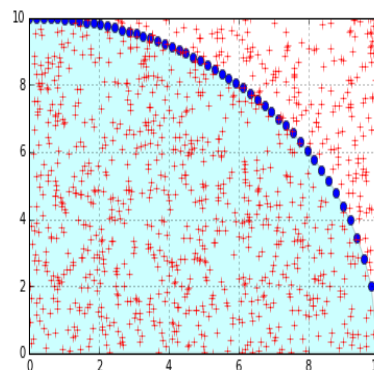
(In this case we've chosen a radius of 10)

5. Let's define an area that includes the area we're searching for, but is easy to calculate. In this case the rectangle created by the four points (0,0), (10,0), (10,10), and (0,10).

$$A_{tot} = b * h = x_{max} * y_{max} = 100$$

It is important to see here that the area is 100 because we chose a radius of 10.

6. Now we pretend to shoot darts at the graph with some assumptions.
  - All darts will hit the graph
  - There is complete randomness to the shots (There is no skill that would increase the chance of hitting the middle, or in the shaded area)
  - The odds of landing in a shaded region or non-shaded region is directly related to the mathematical area of the region.



The trick here is to recognize that the ratio(R) of areas is related to the ratio of the number of shots that hit in the shaded area.

$$R = \frac{n_{shots \text{ in shaded area}}}{n_{total}} = \frac{a_{shaded}}{a_{total}}$$

$$a_{shaded} = \frac{1}{4}a_{circle} = \frac{1}{4}\pi r^2$$



$$a_{total} = x^2$$

Which by substitution and cross multiplying leads to

$$x^2 R = \frac{1}{4} \pi x^2$$

$$\pi = 4R$$

So, we find out that we can calculate  $\pi$  as four times the ratio of those shots that land in the quarter circle to the total number of shots we've taken.

#### *Programming Methodology:*

- Take a shot  
(Pick a random point on the graph area)
- Test to see if it's in the shaded area.  
(Easiest way is to compare your random y value to the function value of your x value.)
- Repeat many times
- Use the ratio of shots in the shaded area to total shots to find a value of  $\pi$ .

#### *Mathematical Solution:*

$\pi$  has an accepted value of 3.141592653589793. Use this to compare with the students answers.

#### *Extensions:*

- Does changing the radius change the answer?
- Does it matter how many shots you take?  
Do you get a good answer with 10 shots, 100, 1000?
- Have students alter the program to calculate  $\pi$  with 100 shots, 100 times. Then find the average, standard deviation, and standard error.
- Have students modify the program to use a half-circle, a whole circle, a sphere.

## Newton-Raphson Method:

### What is the Newton-Raphson Method?

Frequently we find ourselves in need of finding the zeroes of a function. (The places where the value of a graph go to zero.) The steps of this method are relatively straight forward. We make a guess. Find the slope at that location. Create a linear equation through that point with that slope. The zero of that equation gives us our next guess. Theoretically, each step brings us closer and closer to the right answer.

### An Example Newton-Raphson Method:

In this example we are going to use a simple equation to show a complicated method. Most computers and calculators have built in square root functions. But, at one point in time, these were not common. So, we use NR to find the zeros of a modified square equation. (Interesting some calculators use this method to find roots.)

### Problem:

What is the square root of 5?

### Newton-Raphson Solution:

7. First we must convert this into an equation that has a zero.

$$x^2 = 5$$

(but this has no zero, so we modify it)

$$y = x^2 - 5$$

When 'x' equals the square-root of 5, y will equal zero.

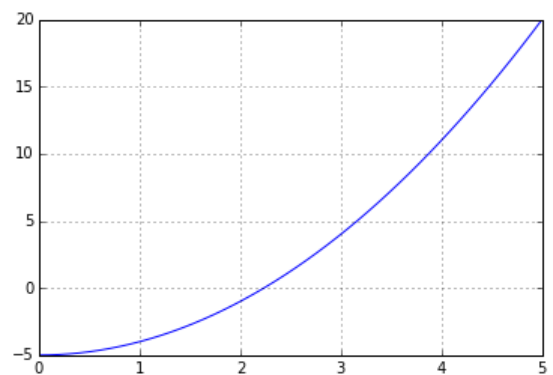


Figure 2 Graph of Equation

8. Let's pretend we haven't seen that graph and try to guess what the square root is. We know that  $2^2=4$ , and  $3^2=9$ . So, our answer must lie between 2 and 3. Let's Guess 2.5.

So, our first x value is 2.5. Our first y value is  $(2.5^2 - 5) = 1.25$

9. We need to find the equation of a line that runs through our function. The first step of this is to find the slope of our function ( $y=x^2-5$ ). There are two ways to do this.
  - a. The first involves calculus and we are going to skip it for two reasons. Calculus is beyond the scope of this material. And, that method is only available when the function is simple enough to find the derivative.
  - b. The second method is the standard slope formula.

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

but the trick here is, we want a slope at a point. So we use our guess as the first point and a tiny bit to the right of our point as the second point. So,  $x_1$  is our guess (2.5 in this case.) And  $x_2$  is our guess plus a tiny amount. (let's add 0.01)

$$m = \frac{f(2.51) - f(2.5)}{2.51 - 2.5} = 5.01$$

10. Now that we know the slope we need the y intercept.

$$y = mx + b$$

$$b = y - mx = 1.25 - 5.01 * 2.5 = -11.25$$

So our equation is set and it is

$$y = 5.01 * x - 11.25$$

11. Now we solve for when our simple linear equation is zero.

$$0 = 5.01 * x - 11.25$$

$$x = \frac{11.25}{5.01} = 2.455$$

This is our first answer. My calculator says the answer is 2.236. So, our answer is pretty close. If it's not close enough what do we do?

12. Repeat this whole process again with this answer (2.455) as our guess.

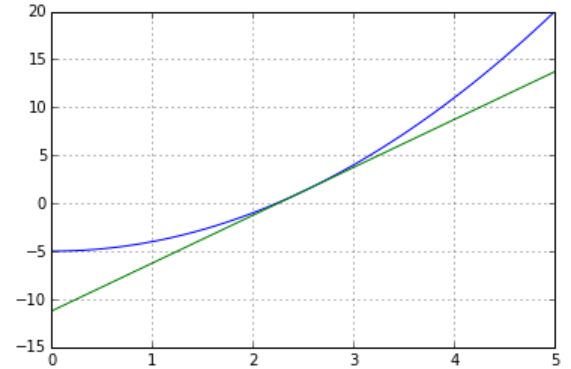


Figure 3 The zero of the tangent line is our approximation of the zero.

#### Programming Methodology:

The steps for programming follow directly from the algorithm. I would recommend breaking this apart and pre-teaching parts. For example. Give them a curve and ask them to find the slope at any given point before starting this assignment.

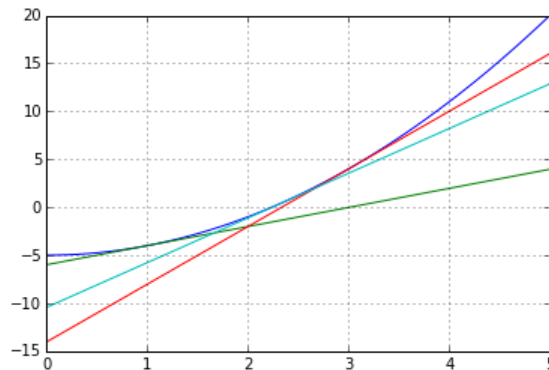


Figure 4 Triple iteration of  $y=x^2-5$

The graph above shows three iterations of the equation ( $y=x^2-5$ ) with a relatively bad initial guess of 1. The iterations are in order of Green, Red, cyan. The values given were converged nicely towards the accepted value of 2.36.

zero 1 at  $x= 3.01005025126$

zero 2 at  $x= 2.33445380933$

zero 3 at  $x= 2.23793449092$

#### Extensions:

- Some obvious extensions would be to find cube roots instead.
- Find the intersection of two equations. ( $y_1 - y_2 = 0$ )
- Take two iterations or more
- Have the program take iterations until the change in the answer is less than a certain percentage.

#### Warnings:

- Note, this example could find the negative answer if you pick a negative first guess.
- There could be a problem if you pick a complicated equation. There could be more than one zero
- Another problem would be if you start with a bad first guess. Your math could end up never finding the answer.