Paul Xhori

Dr. Gilmore

Mathematical Physics II

PS5

1). The easiest is to enforce $¿x>¿\acute{x}=0$. We just need to make all our functions even so this happens without needing to do anything to the functions. Now we want to make the variance equal to 1. The variance of a function is

$$\sigma^2=\int_{-\infty}^{\infty}(x-\acute{x})^2 f(x)\,dx$$

Let us begin with the Gaussian. Wolfram alpha gives the integral

$$\int_{-\infty}^{\infty} x^2 e^{-\alpha x^2}\,dx=\frac{\sqrt{\pi}}{2\alpha^{\frac{3}{2}}}$$

This means we need to divide by that to get the variance to be one. To normalize the function, we perform the following integral:

$$\int_{-\infty}^{\infty}\frac{2a^{\frac{3}{2}}}{\sqrt{pi}}e^{-\alpha x^2}\,dx=2\alpha$$

This means that $\alpha=\frac{1}{2}$ for our function to be normalized. Therefore, the normalized Gaussian with mean equal to 0 and variance equal to 1 is

$$\frac{e^{\frac{-x^2}{2}}}{\sqrt{2\pi}}=PDF$$

Now let us move on to the double sided exponential. First we find the variance parameter.

$$x^2 e^{-\alpha \vee x \vee ¿\, dx}$$

$$(x)^2 e^{-\alpha \vee x \vee ¿ dx} + \int_0^{\infty} ¿$$

$$x^2 e^{-\alpha \vee x \vee ¿ dx} = \int_{-\infty}^{0} ¿$$

$$\int_{-\infty}^{\infty} ¿$$

Note that $(x)^2 e^{-\alpha \vee x \vee ¿ dx} = \int_0^{\infty} ¿$ because of the absolute value and the exponent of x. Therefore,

$$\int_{-\infty}^{0} ¿$$

this function is

$$x^2 e^{-\alpha \vee x \vee ¿ dx} = \frac{4}{\alpha^3}$$

$$x^2 e^{-\alpha \vee x \vee ¿ dx} = 2 \int_0^{\infty} ¿$$

$$\int_{-\infty}^{\infty} ¿$$

Now we normalize this as follows, again taking advantage of symmetry

$$\frac{2 * \alpha^3}{4} \int_0^{\infty} e^{-\alpha |x|} dx = \frac{a^2}{2} = 1$$

So this means that $a = \sqrt{2}$ for this to be normalized, and the normalized PDF with mean equal to 0 and variance equal to 1 becomes

$$\frac{\sqrt{2}^3}{2} e^{-a|x|} = \frac{e^{-\sqrt{2}|x|}}{\sqrt{2}} = PDF$$

Now we will do the boxcar distribution. We call the constant c initially, and since this function is 0 outside of the range –a to a, to get the variance we perform

$$\int_{-a}^{a} x^2 * c dx = \frac{2a^3 c}{3}$$

So now we normalize the function:

$$\int_{-a}^{a} \frac{3}{2a^3} dx = 2\frac{a*3}{2a^3} = \frac{3}{a^2} = 1$$

So we find that $a = \sqrt{3}$. Therefore, the function is

$$\frac{3}{2\sqrt{3}^3} = \frac{1}{2\sqrt{3}} ¿ x = -\sqrt{3} ¿ \sqrt{3}, 0 \; elsewhere$$

To have a variance of 1, be normalized, and have mean equal to 0.

Now we do this for the parabolic distribution. The parabolic distribution $a - b x^2$ has zeroes

at $\sqrt{\frac{a}{b}}$ and $\sqrt{\frac{a}{b}}$. It is zero outside this range. Therefore, to have unit variance, we perform

$$\int_{-\sqrt{\frac{a}{b}}}^{\sqrt{\frac{a}{b}}} x^2 \left(a - b x^2\right) dx = \frac{4}{15} a * \left(\frac{a}{b}\right)^{\frac{3}{2}}$$

To normalize this, we perform

$$\int_{-\sqrt{\frac{a}{b}}}^{\sqrt{\frac{a}{b}}} \frac{15}{4a\left(\frac{a}{b}\right)^{\frac{3}{2}}}\left(a-bx^2\right)dx=\frac{5b}{a}=1$$

This means that $a=5b$. Therefore, the parabolic distribution with mean equal to 0, variance equal to 1, and normalized is

$$\frac{15}{20b(5)^{\frac{3}{2}}}\left(5b-bx^2\right)=\frac{3}{4*5^{1.5}}*\left(5-x^2\right)¿\, x=-\sqrt{5}¿\sqrt{5},0\, elsewhere=PDF$$

Finally, the triangular distribution. This function could actually be many things, but we will say it is the one that has equal slopes on either side. This has the form $ax+b$ on the left, and $-ax+b$ on the left. This function has zeroes at $-b/a$ and b/a so it has the form

$$f(x)=ax+b\, for\, x=-b/a¿0,-ax+b\, for\, x=0¿b/a,0\, elsewhere$$

The function is even, so we can do the following integral to find the variance

$$2*\int_0^{\frac{a}{b}} x^2(-ax+b)\,dx=\frac{b^4}{6a^3}$$

Then to make normalize the function, we perform the following integral taking advantage of symmetry again:

$$2*\int_0^1 \frac{6a^3}{b^4}(-ax+b)\,dx=\frac{6a^2}{b^2}=1$$

So $6a^2=b^2$ => $\sqrt{6}a=b$. The zeroes now become $-\sqrt{6}$ and $\sqrt{6}$. The functional

form for the right is now $\begin{matrix} a \\ \sqrt{6}¿ \\ ¿ \\ (¿4¿) \\ ¿ \\ ¿ \\ \frac{6a^3}{¿} \end{matrix}$ and the left becomes $\frac{x}{6}+\frac{1}{\sqrt{6}}$. Therefore, the triangular

distribution with mean equal to 0, variance equal to 1, and that is normalized is

$\frac{x}{6}+\frac{1}{\sqrt{6}} for\ x=-\sqrt{6}¿0$

$\frac{-x}{6}+\frac{1}{\sqrt{6}} for\ x=0¿\sqrt{6}$

$0\ elsewhere$

All integrations were performed in Wolfram alpha, with code of the form:

Integrate x*x*PDF dx from (lower limit) to (upper limit)

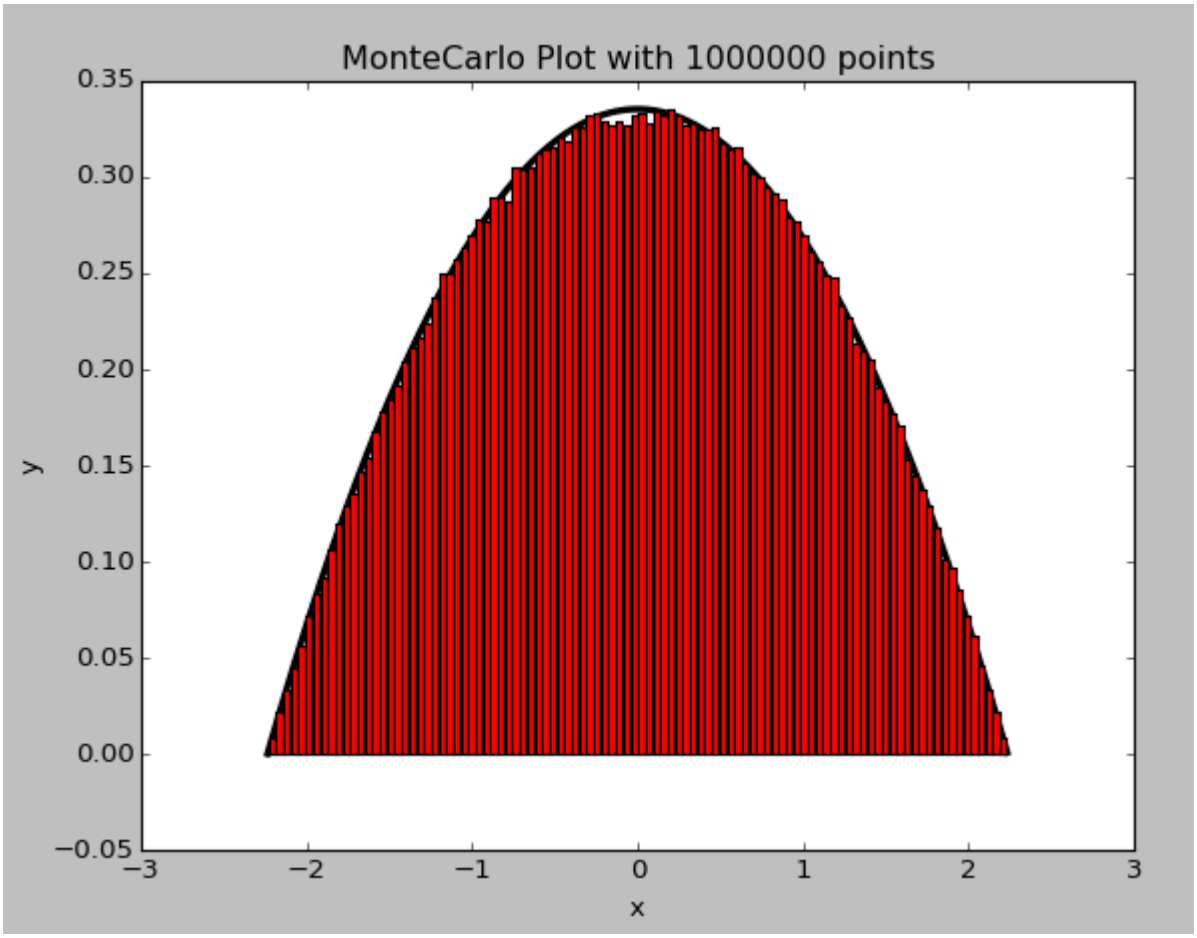
Or

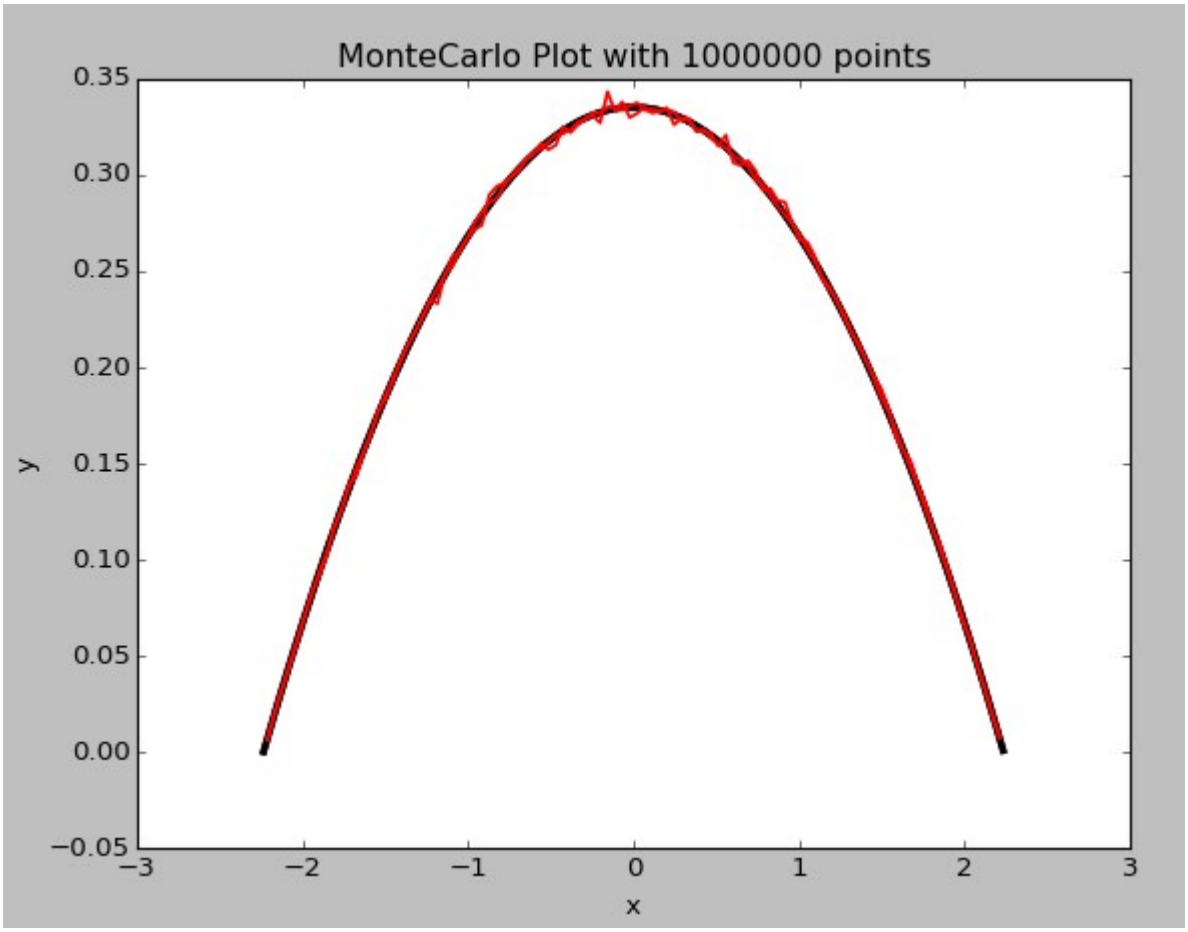
Integrate PDF dx from (lower limit) to (upper limit)

It can handle variable limits as well as constants in the integrand so it was ideal for this problem.

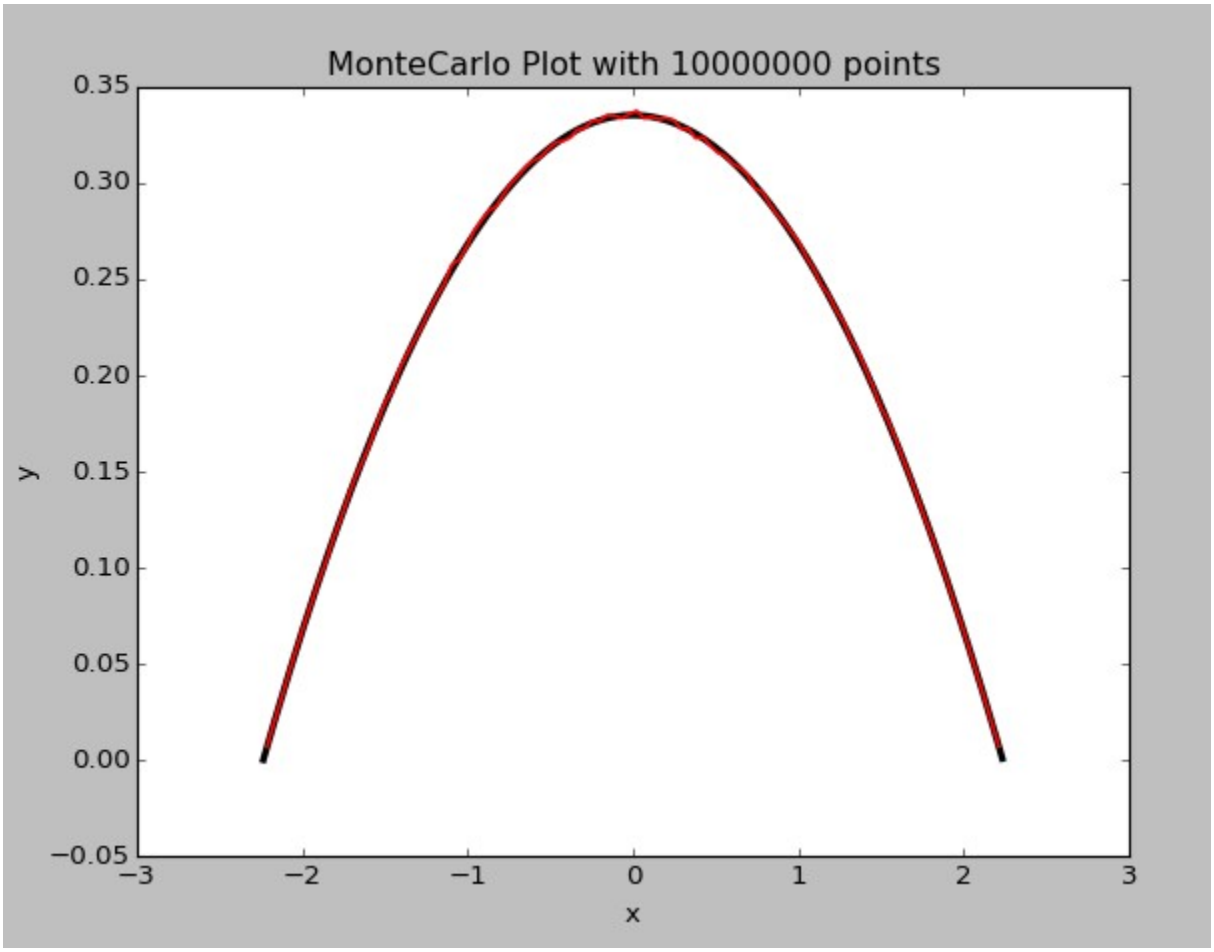
2). The parabolic distribution has x values from $\pm\sqrt{5}$, and y goes from 0 to the max which is

$\frac{3}{4*5^{1.5}}*5 \cong .33541$. The random number generator I am using is called “random.uniform” and is available in Python 3.2 if you import the “random” library. It takes in two arguments and generates random numbers between those two arguments, excluding the upper one. Although, since the numbers are 32 bit, the upper one could eventually be produced as it will have to round. Needless to say, it is so rare for a particular number to be generated that this doesn’t really effect anything. The program generates two random lists, we call one x and one y, and we zip the two so that we have a tuple of values that is the point to test. It performs the test, and if a point happens to pass, a list that is the same length as the x list has a 1 put in in the same index of the passing x-value in the list of x values. Then this pass/fail list is used to filter out x values that passed by putting only the x values that have a 1 in the pass/fail list into a new list that contains only the x values that passed. The passed x values are then put into a histogram function. The resulting histogram array is normalized and plotted against the analytic function for comparison, where the analytic function is plotted with thick line width and is behind the histogram. I also removed the bars and plotted the histogram as a line so we could compare the two functions easier (I actually prefer it this way).


MonteCarlo Plot with 1000000 points
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
y
−3
−2
−1
0
1
2
3
x



The plots compare very well for a generated list of 1,000,000 points, which takes about 6 seconds to run on my computer. I can try 10,000,000 points to get a much better convergence, however, it takes 65 seconds. This suggests that the program runs on the order of N, which is better than N^2 but still pretty slow.



3) We need to construct the CDF for the triangular distribution. We can do this by performing

$$\int_{-\infty}^{x} PDF * dx$$

The lower limit automatically becomes $-\sqrt{6}$ because the PDF is if x is below that range. For x up until zero, the CDF is $\frac{x^2}{12}+\frac{\sqrt{6}}{6}x+\frac{1}{2}$. Then after zero, the integral has as a constant

value .5, which is the part up to zero, in addition to the integral from 0 to x, which is

$\frac{-x^2}{12}+\frac{\sqrt{6}}{6}x$. Therefore, the CDF is

$\frac{x^2}{12}+\frac{\sqrt{6}}{6}x+\frac{1}{2}\,for\,x ¿ -\sqrt{6} ¿ 0$

$\frac{-x^2}{12}+\frac{\sqrt{6}}{6}x+\frac{1}{2}\,for\,x ¿ 0 ¿ \sqrt{6}$

$0\,for\,x \le -\sqrt{6}$

$1\,for\,x \ge \sqrt{6}$

So the domain is $-\sqrt{6} \le x \le \sqrt{6}$, and the range is $0 \le y \le 1$ for the entire function.

Now in order to reconstruct the triangular distribution, we need to get an inverse of this function.

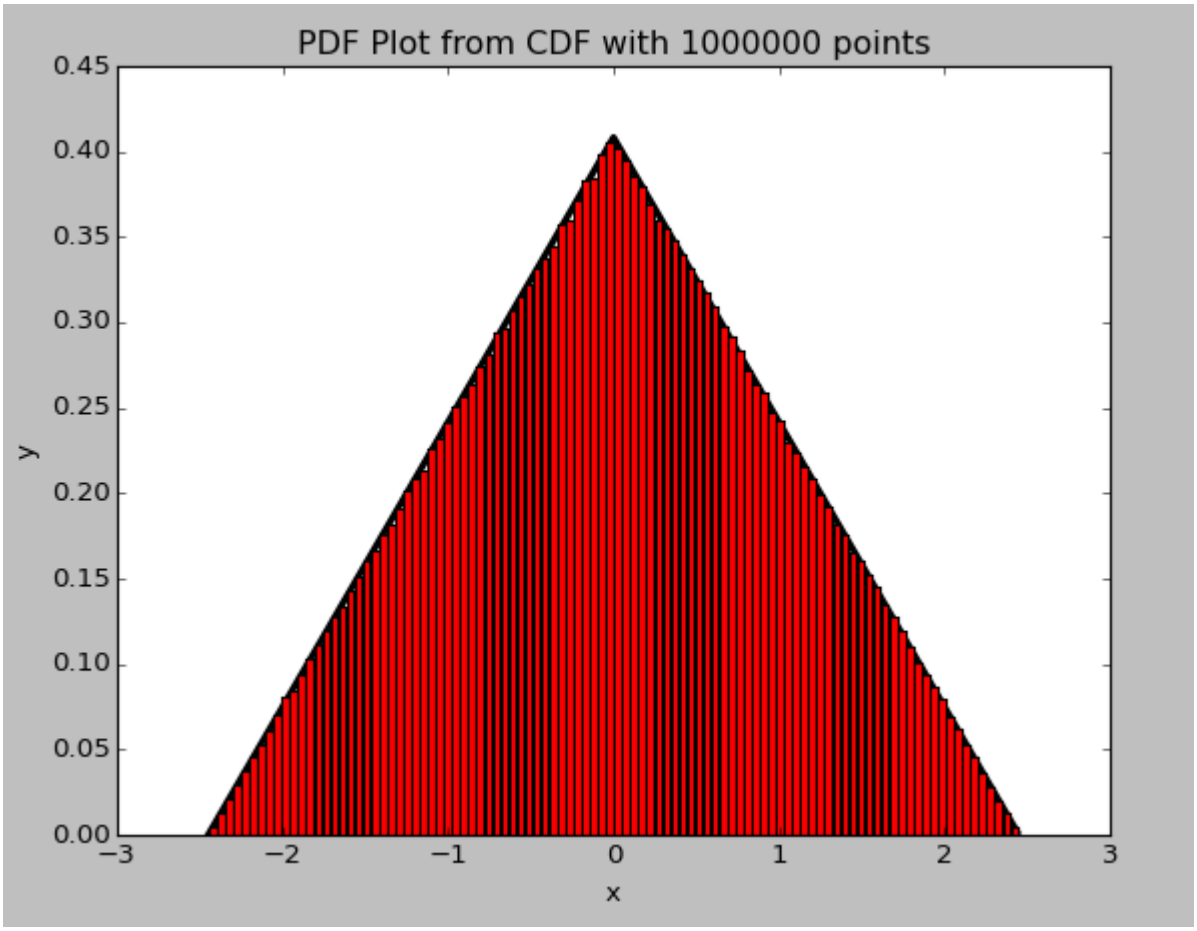
$y = x^2+\frac{\sqrt{6}}{6}x+\frac{1}{2} = ¿\, x = -\sqrt{6} \pm 2\sqrt{3\,y}$

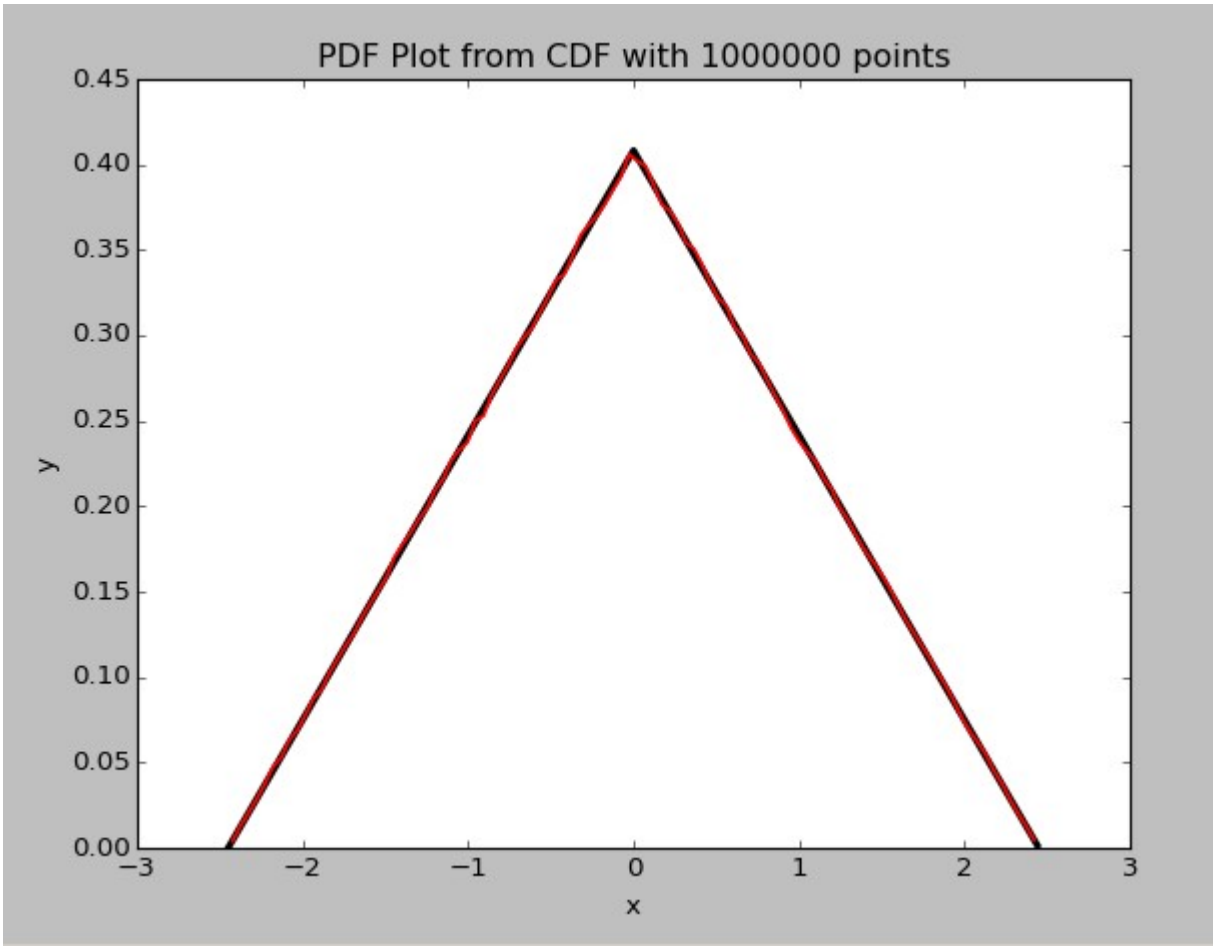
The domain and range switch. So we need to generate random numbers between 0 and .5 for this part of the CDF, and we should only consider the negative x. The second part has inverse

$x = \sqrt{6} \pm 2 * \sqrt{3(1-y)}$

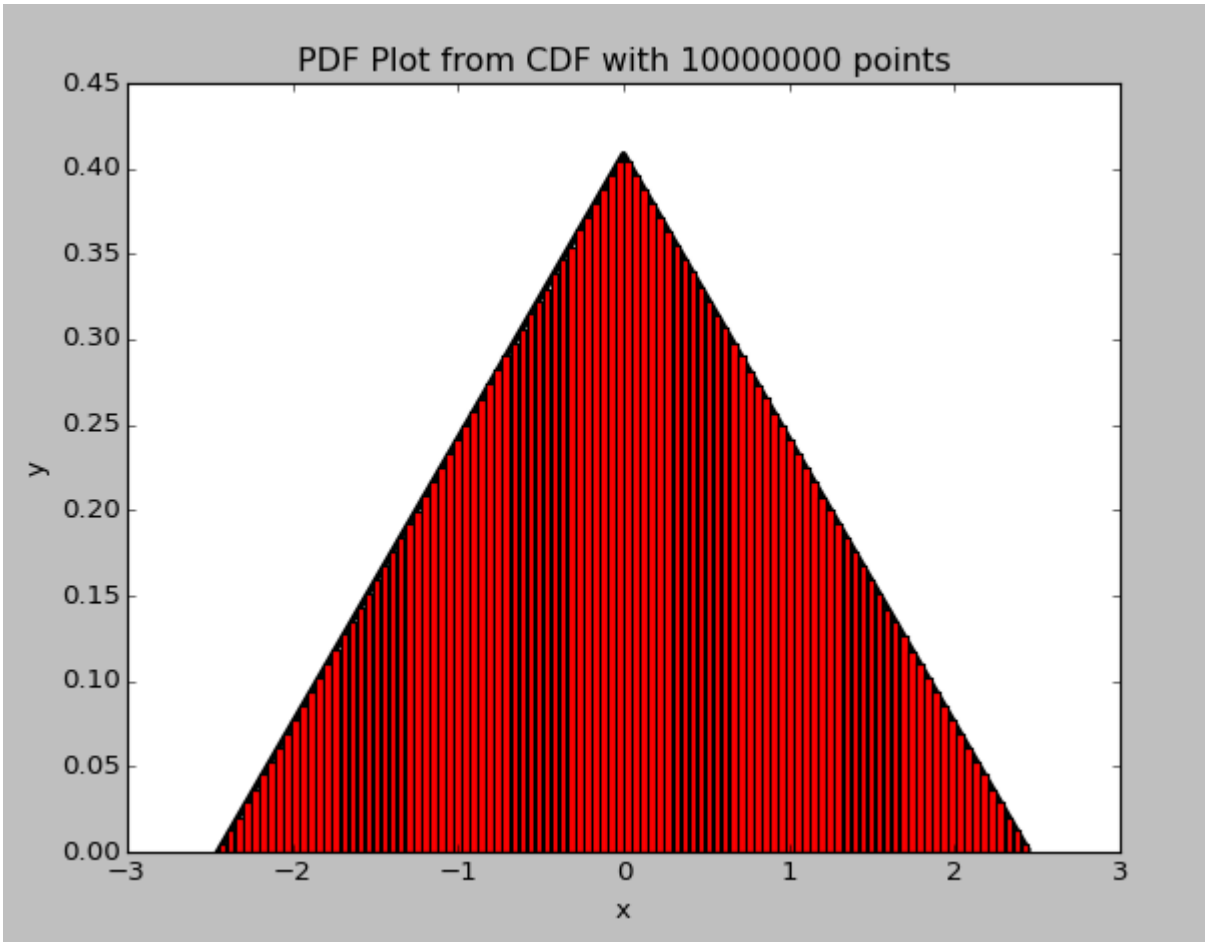
We need to generate values between .5 and 1 for this part of the CDF, and we should only consider positive x. Either one can work for .5, so we'll just use the second one since that is where it is included by default in the random number generator I am using (but again, due to the

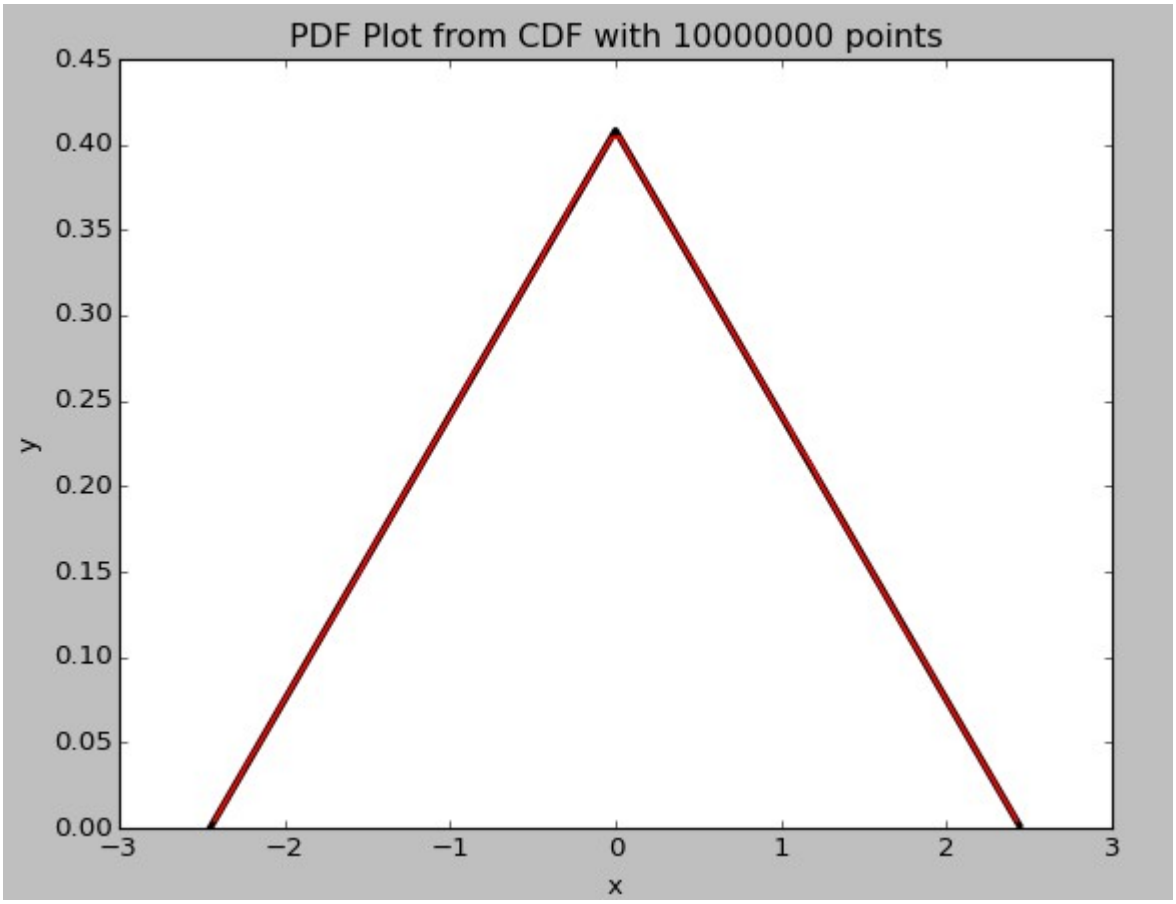
same reason that excluding 1 doesn't matter, which is the ridiculous probability that any one number is selected, this basically doesn't matter). So now the program generates two lists, one between 0 and .5, and one between .5 and 1. The first list is sent to the first inverse function, with the + sign selected to ensure the correct domain of x numbers come out, and the second is sent to the second inverse function, with the minus sign selected to ensure the correct domain of x numbers comes out. The two lists are then combined and a histogram is made from them. I did the same plotting style as in problem 2, with the analytic function in the background and plotted as bars as well as a line. This time, no points are "wasted" as every y value has a corresponding x in the CDF. This recreates the PDF it was created from. Here are the plots below.





With 1,000,000 points, the plot already converges very well. This is because we are guaranteed that all 1,000,000 are included in the results rather than Monte-Carlo in which we don't know beforehand how many will be included. With 10,000,000 points, the results are even better:


PDF Plot from CDF with 10000000 points
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−3
−2
−1
0
1
2
3
x

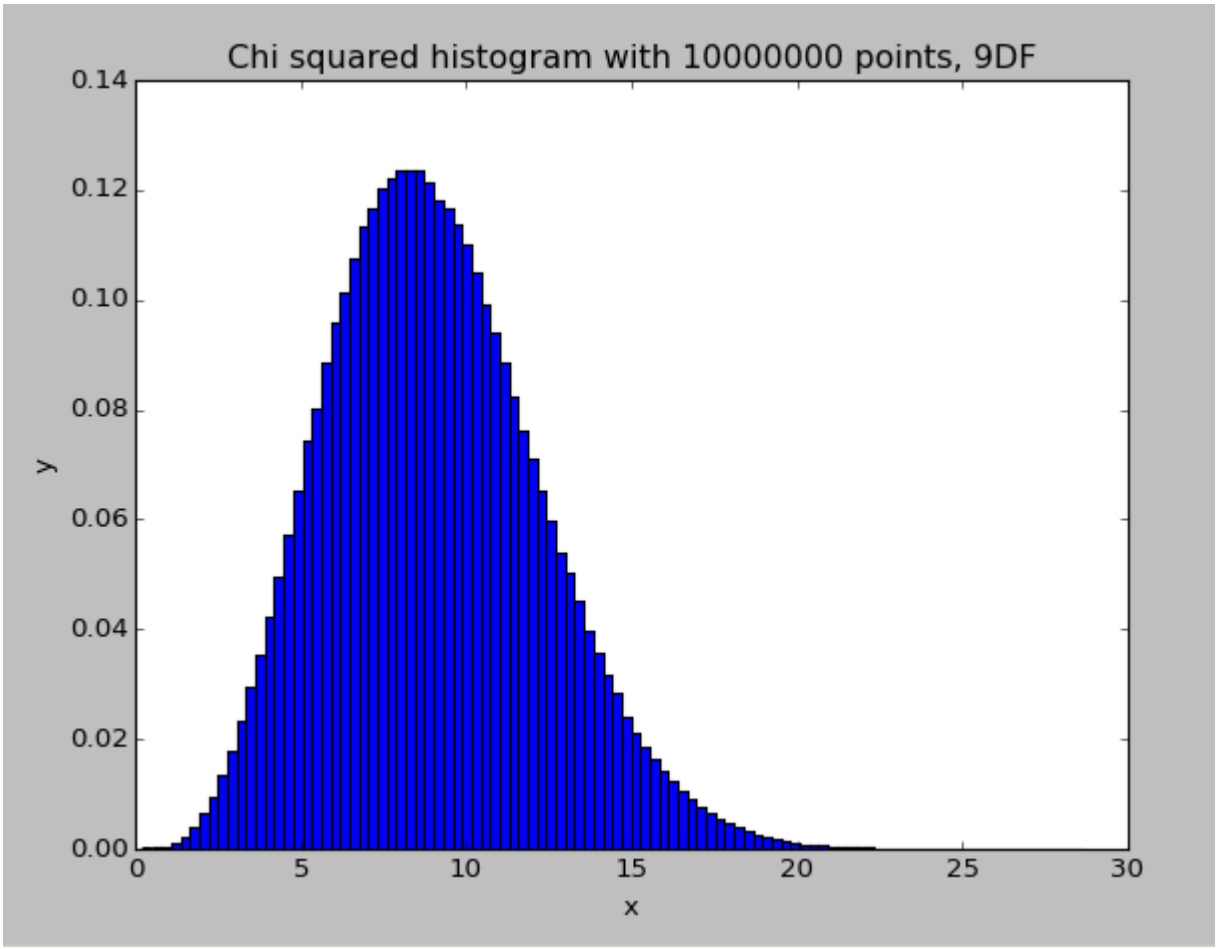


(the squareness at the top is the result of the binning – see the histogram)

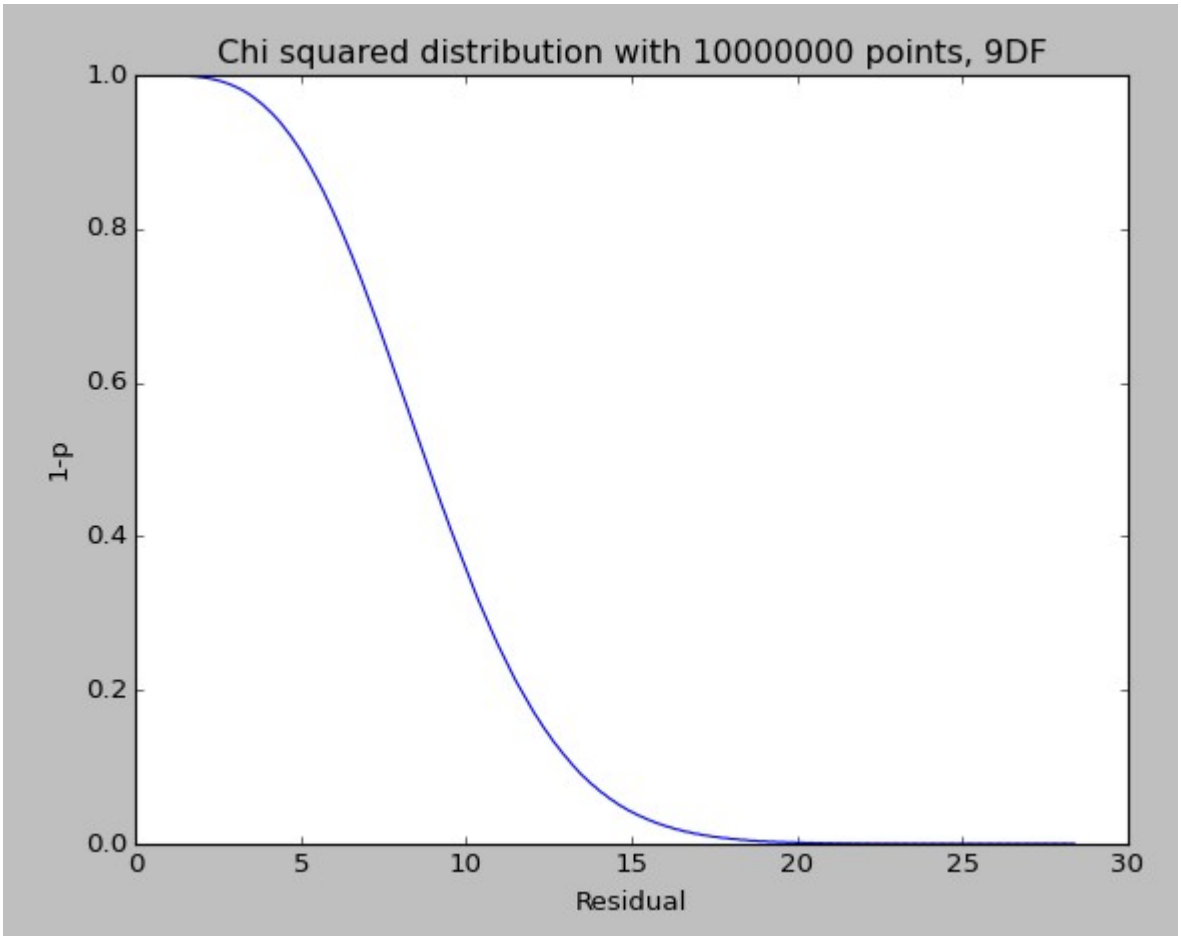
Note that both functions actually are about the same difficulty to "fit" in this way – the parabola and triangle take up about equal area of their square that is defined by their domain and range (of course the area it takes up doesn't matter for what we are doing with the triangle). So it is truly from the wasted points of the first method that makes this method converge with fewer points. Both took approximately 6 seconds for 1,000,000 points and 65 for 10,000,000 points.

4). Now we build up a χ^2 for the parabolic plot. To re-iterate how the x-squared values are obtained, I repeat what I have said in problem 2. The program generates two random lists, we call one x and one y, and we zip the two so that we have a tuple of values that is the point to test. It performs the test, and if a point happens to pass, a list that is the same length as the x list has a

1 put in in the same index of the passing x-value in the list of x values. Then this pass/fail list is used to filter out x values that passed by putting only the x values that have a 1 in the pass/fail list into a new list that contains only the x values that passed. This list is then squared, and we take groups of 9 from this list and sum them until there are no more groups of 9 left to sum. The remaining x values (of which there will be less than 9) are not used. We put these groups of 9 squared sums in a list, and make a histogram out of this list. I have normalized the histogram and plotted it below. I used 10,000,000 points, and about 6.6 million of them passed (incidentally, this means that the parabola takes up 66% of the space of the box defined by its domain and range). It took about 70 seconds, which makes me think the bottleneck in all of these programs is the random number generating function.



Finally, I wrote a list comprehension that calculate $1-\int_0^x \chi^2 dx$ in order to convert the chi-squared histogram into the chi squared distribution, where the value of y for a given x, which is a residual, gives you the p value in statistics. It works by making each element in the "yint" list the integral over a subset of the list that corresponds to a value in xhist. The p value is the probability that our model fits the data by chance. So, if we get a small p value, less than .05, we should not reject the model.



5). I used Excel's solver script to perform this task. You put the data in with x and y as columns, and set cells to your parameters for the fit, in our case, m and b. You take an initial guess at these parameters based on the data (I chose m=.5 and b=2) and then make a column of fit results using

excel functions incorporating your x values and parameters. Then you calculate the error from each point of your initial guess fit from the real y values and square this, and divide it by σ^2. Then you sum all the error. The total formula for the residual looks like

$$r=\frac{1}{\sigma^2}\sum_{i=1}^{11}\left(y_i-\left[m x_i+b\right]\right)^2$$

Then, you tell the Solver plug in to pick the cell with your residual as its minimizing target, and tell it to vary the m and b parameters until it is minimized. Solver then gives you the values it finds that minimize the residuals. I also compared this with Excel's built in linear fit capability and the parameters I got agreed to all but the last two out of 15 decimal places. I will send you the excel file itself since there is no other way to include excel "source code". I got a residual value of 27.52538. Using our χ^2 distribution for 9 degrees of freedom, this result yields a p=.00044. This means that 99.956% of the (6.6 million)/9 χ^2 values I tested had lower residuals than this data set. Therefore, I do not reject the parabolic distribution model for this data.