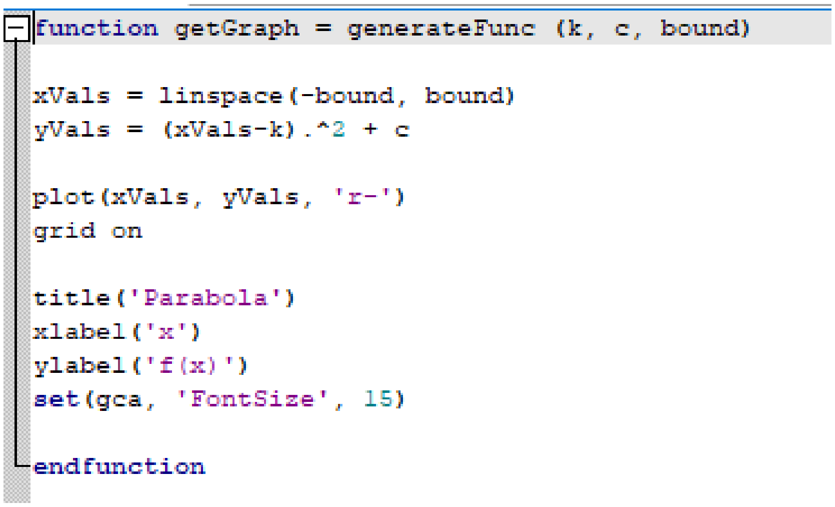
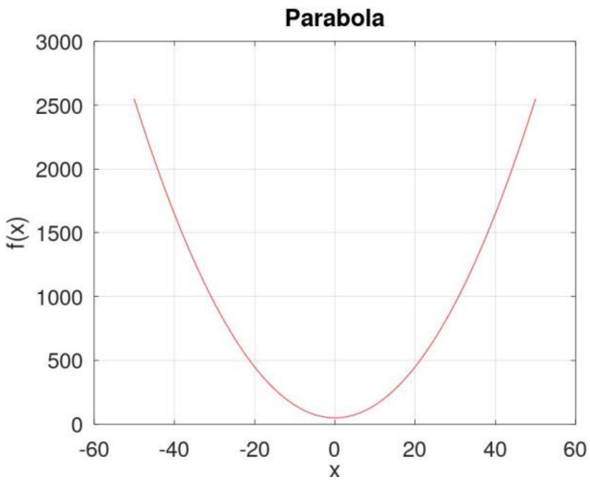
Nick Fichera

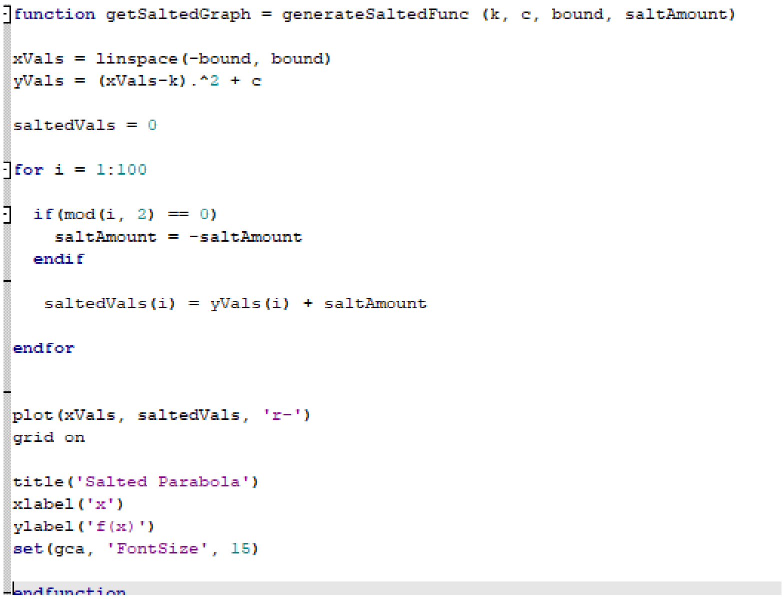
Octave Report



```
function getGraph = generateFunc (k, c, bound)

xVals = linspace(-bound, bound)
yVals = (xVals-k).^2 + c

plot(xVals, yVals, 'r-')
grid on

title('Parabola')
xlabel('x')
ylabel('f(x)')
set(gca, 'FontSize', 15)

endfunction
```

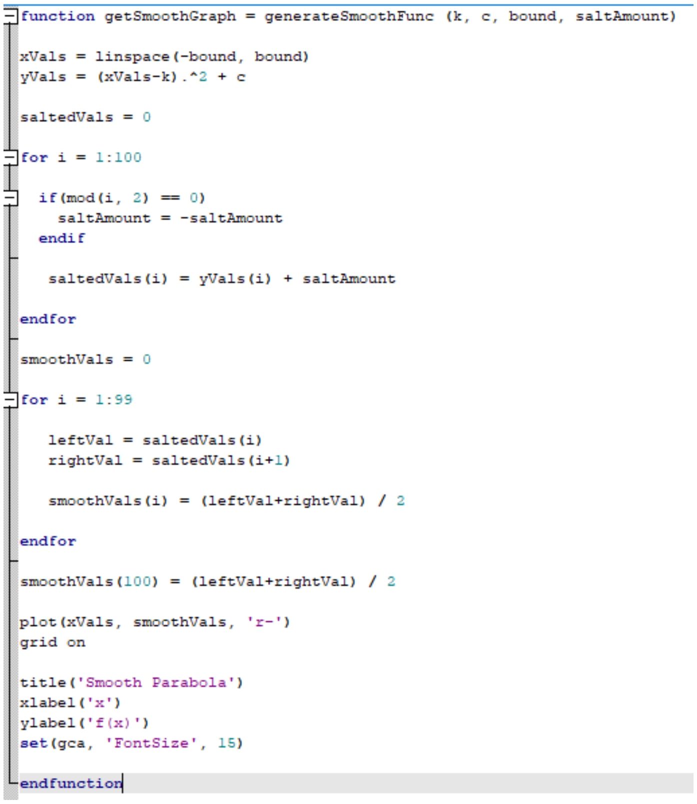


So, initially I didn't know much about Octave. I downloaded it and the first thing I learned was how to make a function. I then setup the initial function. I found online that linspace() essentially creates an array of values between two number parameters, in this case, from the negative bound to the positive bound supplied by the parameter. I chose to create a quadratic function with parameter k (the horizontal shift), c (vertical shift), and bound, which is the range of x values to plug into the function. Then, yVals becomes an array where each x value is computed and then everything is plotted.



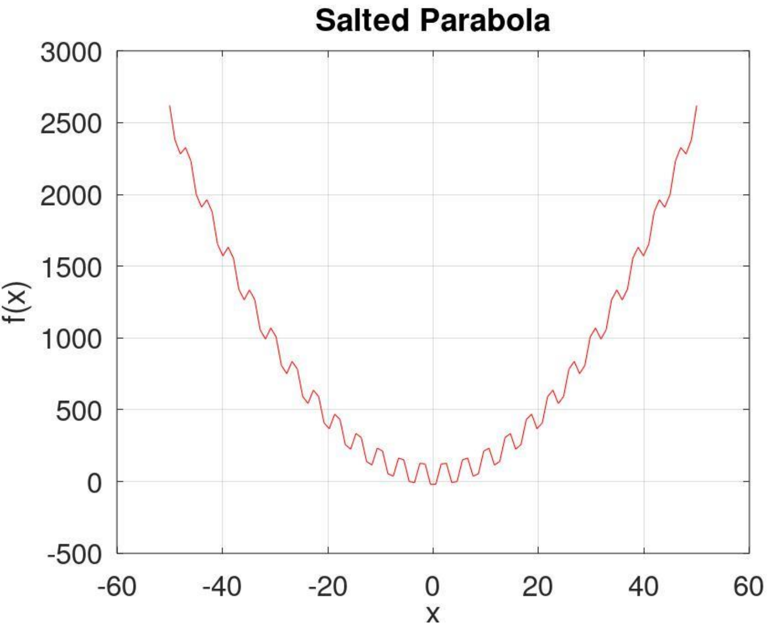
```
function getSaltedGraph = generateSaltedFunc (k, c, bound, saltAmount)

xVals = linspace(-bound, bound)
yVals = (xVals-k).^2 + c

saltedVals = 0

for i = 1:100

  if(mod(i, 2) == 0)
    saltAmount = -saltAmount
  endif

   saltedVals(i) = yVals(i) + saltAmount

endfor


plot(xVals, saltedVals, 'r-')
grid on

title('Salted Parabola')
xlabel('x')
ylabel('f(x)')
set(gca, 'FontSize', 15)

endfunction
```

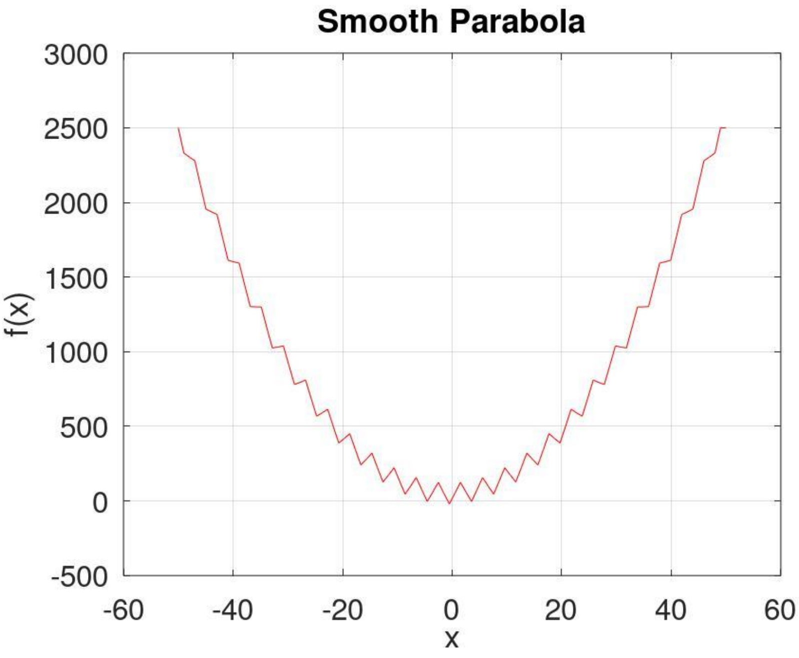
In the next function, getSaltedGraph, it is the same as the last function except we have a for loop here to salt the values. The purpose of the if statement is to negate the value of the saltAmount each iteration, which creates a salted graph.



```
function getSmoothGraph = generateSmoothFunc (k, c, bound, saltAmount)

xVals = linspace(-bound, bound)
yVals = (xVals-k).^2 + c

saltedVals = 0

for i = 1:100

  if(mod(i, 2) == 0)
    saltAmount = -saltAmount
  endif

   saltedVals(i) = yVals(i) + saltAmount

endfor

smoothVals = 0

for i = 1:99

   leftVal = saltedVals(i)
   rightVal = saltedVals(i+1)

   smoothVals(i) = (leftVal+rightVal) / 2

endfor

smoothVals(100) = (leftVal+rightVal) / 2

plot(xVals, smoothVals, 'r-')
grid on

title('Smooth Parabola')
xlabel('x')
ylabel('f(x)')
set(gca, 'FontSize', 15)

endfunction
```

In getSmoothGraph, I wasn't too sure how I could get the saltedVals from the salt function, so here I have it salt and then smooth the data. This result of this code isn't perfect, it doesn't completely smooth the graph but interestingly it creates a sharper edge version of the saltedGraph. I looked online and tried to use the rgdtsmcore and regdatasmooth from the Octave Forge library however I had a little trouble trying to figure it out.


Salted Parabola
f(x)
x


Smooth Parabola
f(x)
x