



Discrete Structures: CMPSC 102

Oliver BONHAM-CARTER

Fall 2019
Week 3

Newton's Method

Application In Mathematics

Newton's Method

Guess the root

Derivatives

General Equation

Simple Example: x^2

Automate with Python

Working with Numbers

Type conversions

Fractions

Exceptions

Formatting

Suppose we have a very complicated equation to solve and we need to find some mathematical way to solve for x .

We are given an equation to solve: Find the Roots

$$f(x) = 48x(1+x)^{60}(1+x)^{60} + 1 = 0$$

- An approximate solution
- Let's plot the equation to see where it crosses the x axis
- Ask: for what value of x does this x -axis intersection happen?
- In general, how do we find values of x ?

Plot the Equation

Approaching solutions by approximation ...

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

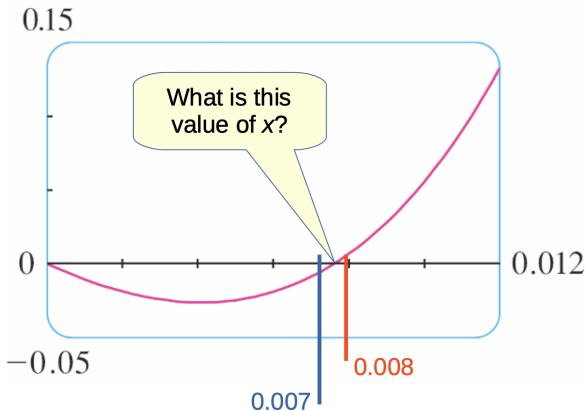
Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting



- We know that there are two roots.
- One solution is zero (or near) and the other is between 0.007 and 0.008
- We can estimate the value of one of the points (sort of) ...

Plot the Equation

Newton's Method

Guess the root

Derivatives

General Equation

Simple Example: x^2

Automate with Python

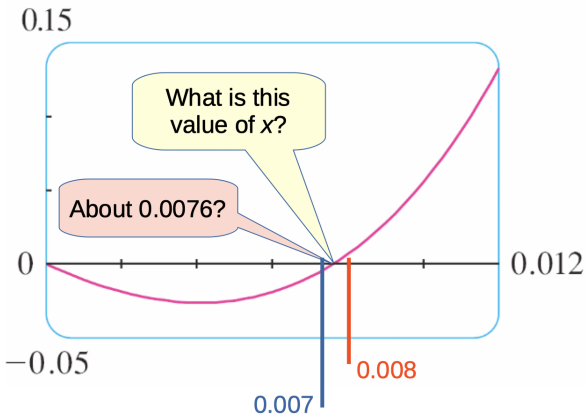
Working with Numbers

Type conversions

Fractions

Exceptions

Formatting



- Estimate a solution between 0.007 and 0.008 (approx 0.0076)
- Want to be able to calculate this value to any level of accuracy
- How to find these *roots* mathematically for any *zoomed-in* value?



Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

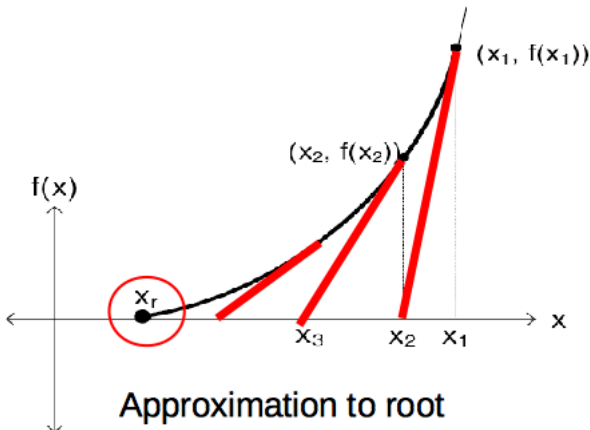
Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting



- Approximate the root (x_r) using *Newton's Method*

Isaac Newton

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

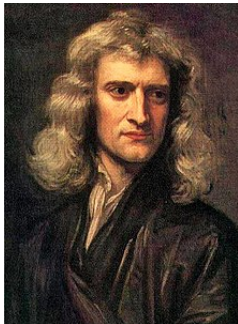
Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting



- Time line: 25 December 1642 20 March 1726 or 1727)
- English mathematician, astronomer, theologian, author and physicist
- One of the most influential scientists of all time
- A key figure in the scientific revolution.

Guess a root

Linear approximation

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

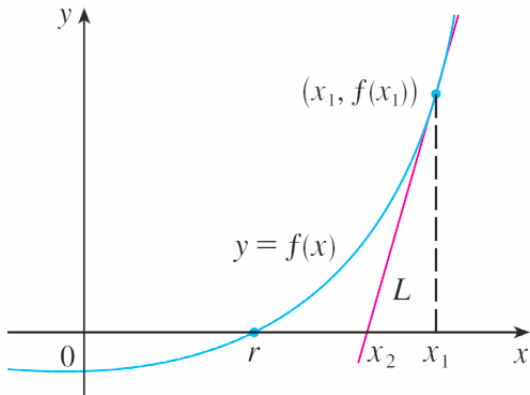
Working with
Numbers

Type
conversions

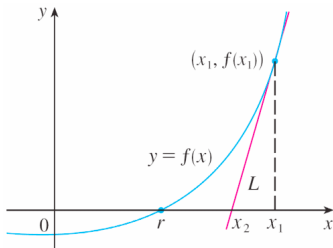
Fractions

Exceptions

Formatting



- The relevant root is labeled r
- First approximation for x_1 is a simple *guess* made by understanding the plot



- Consider the tangent line, L to the curve $y = f(x)$ at the point $(x_1, f(x_1))$ and look at the x -intercept of L , labeled x_2 .
- Main idea: the tangent line is close to the curve and its x -intercept (an intersection point at x_2), is close to the x -intercept of the curve (the root r).
- This point root r that want to find!

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

The Derivative

1. Choose an interval

2. Find the raw change

$$f'(x) = \lim_{dx \rightarrow 0} \frac{f(x+dx) - f(x)}{dx}$$

4. Make your model perfect

3. Find the rate of change

Roughly speaking...

- The derivative is an equation extracted from the original $f(x)$ used to find the x values of where the $y = 0$.

Newton's
MethodGuess the
root

Derivatives

General
EquationSimple
Example: x^2 Automate
with PythonWorking with
NumbersType
conversions

Fractions

Exceptions

Formatting

Roughly speaking...

- We want to find where line L passes x -intercept
- Slope of line L : $f'(x)$ (the derivative is tangent to curve)
- Line formula (from algebra): $y = m * x + b$
- To find a formula for x_2 in terms of x_1

$$y - f(x_1) = f'(x_1)(x_2 - x_1) \quad (1)$$

Thinking Recursively

Use derivatives to find lines crossing x -axis, converging on root

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

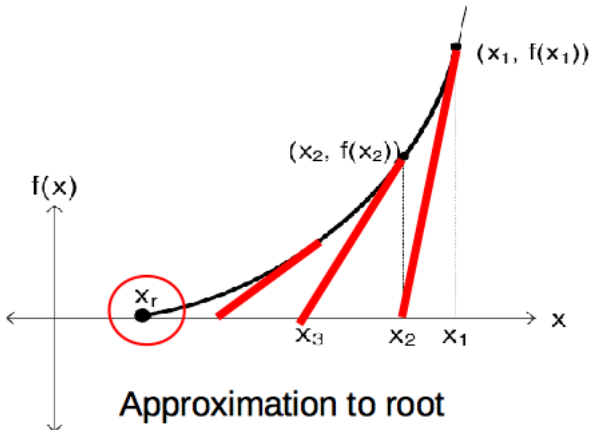
Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting



We want to insert initial approximation values of x back into the line equation (recursively) to find the next approximation (and converge on the root, x_r). The slope of the tangent line is $f'(x)$, known as the derivative.



Find a General Equation for Finding Roots

Newton's
MethodGuess the
root

Derivatives

General
EquationSimple
Example: x^2 Automate
with PythonWorking with
NumbersType
conversions

Fractions

Exceptions

Formatting

We are solving for x_2 , or a closer approx. of root!

$$y - f(x_1) = f'(x_1)(x_2 - x_1)$$

$$0 - f(x_1) = f'(x_1)(x_2 - x_1)$$

$$f(x_1) = -f'(x_1)(x_2 - x_1)$$

$$f(x_1) = x_1 * f'(x_1) - x_2 * f'(x_1)$$

$$f(x_1) - x_1 * f'(x_1) = x_2 * f'(x_1)$$

$$x_2 * f'(x_1) = x_1 * f'(x_1) - f(x_1)$$

$$\frac{x_2 * f'(x_1)}{f'(x_1)} = \frac{x_1 * f'(x_1)}{f'(x_1)} - \frac{f(x_1)}{f'(x_1)}$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Now what?!

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

First approximation of root values

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Second approximation of root values

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Third approximation of root values

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

General approx. of root vales; x_{n_1} from previous x_n

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(n)}$$

Let's find the **square** root for a number

Recall that Newton's method finds an approximate root of $f(x) = 0$

Define $f(x)$

$$f(x) = x^2 - a$$

$$x^2 = a$$

$$x = \sqrt{a} \quad (\text{find positive root, } a)$$

$$x = -\sqrt{a}$$

Define the derivative of $f(x)$, $f'(x)$, using calculus

$$f'(x) = 2x$$

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Definitions

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Establish the Approximation Equation

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Definitions

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Note: a in $f(x)$ is the initial guess!

$$f(x) = x^2 - a$$

$$f'(x) = 2x$$

The root to find

$$a = x_r$$

The initial guess of root (to start the method)

$$x_1 = 1.0$$

General approx. for root x_{n+1} from approx. root x_n

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(n)}$$

Finding Square Root of a

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Definitions

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Knowns

$a = 2$ (want to find \sqrt{a})	$f(x) = x^2 - 2$ (function)
$x_1 = 1.0$ (initial guess)	$f'(x) = 2x$ (derivative)

$$\begin{aligned}x_1 &= 1.0 - \frac{f(1.0)}{f'(1.0)} \\&= 1.0 - \frac{(1.0)^2 - 2}{2 * (1.0)} \\&= 1.0 - \frac{1.0 - 2}{2} \\&= 1.0 - \frac{-1.0}{2} \\&= \frac{3.0}{2} \\&= 1.5\end{aligned}$$

Table of Iterations

Finding square root of a

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Definitions

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Calculations

Guess			Approx. root
x_n	$f(x) = x_n^2 - 2$	$f'(x_n) = 2x$	$x_n - \frac{f(x_n)}{f'(x_n)}$
1	-1	2	$1 - \frac{-1}{2} = \frac{3}{2} = 1.5$
$\frac{3}{2}$	$\frac{1}{4} = 0.25$	3.0	$\frac{3}{2} - \frac{(\frac{1}{4})}{3} = \frac{17}{12} = 1.4167$
$\frac{17}{12}$	$\frac{1}{144}$	$\frac{17}{6}$	$\frac{17}{6} - \frac{\frac{1}{144}}{\frac{17}{6}} = \frac{577}{408} = 1.4142$

Python to the rescue

```
>>> math.sqrt(2)
1.4142135623730951
```

Automate with Python

Finding square root of a

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Atom newtonsMethod.py

```
n = 2.0 # the number from which to find square root.
guess = 1.0 # initial value for approx

print("  Initial values:  n = ",n, "guess = ",guess)

while abs(n - guess*guess) > .0001:
    #find  $x_n - \frac{f(x_n)}{f'(x_n)}$ 
    guess = guess - (guess*guess - n)/(2*guess)
    print("    *Current guess:  ",guess)
root = guess

print("    Result :",root)
```

Put This Script Into a Function

Finding square root of a

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Atom newtonsMethodFunction.py

```
def NM(n, guess):  
    print("    Initial values:  n = ",n, "guess = ",guess)  
  
    while abs(n - guess*guess) > .0001:  
        #find  $x_n - \frac{f(x_n)}{f'(x_n)}$   
        guess = guess - (guess*guess - n)/(2*guess)  
        print("    *Current guess:  ",guess)  
        root = guess  
    return root  
#end of NM()  
  
#get parameters to call function NM()  
n = 2 # the number from which to find square root.  
guess = 1.0 # initial value for approx  
print(" Finding root : ",n)  
print(" Approx guess : ", guess)  
print(" Result : ",NM(n, guess))
```

Working with numbers

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

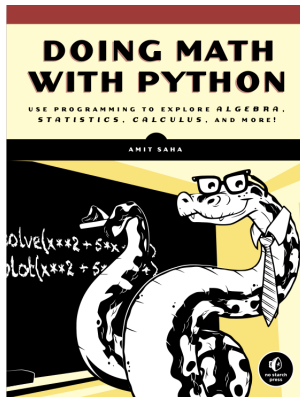
Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting



Chapter 2: Working with Numbers

You Can Use Python as a Calculator

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Integers

- $1 + 2$
- $200 + 4$
- $x_int = 1 + 2$
- `type(x_int) #ls: <class 'int'>`

Floats

- $1.0 + 2.2$
- $200.001 + 56.05$
- $x_flt = 123.007 + 0.002$
- $x_flt = 100 / 4$ #ls: `<class 'float'>!`

Multiplication and Division

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Floor Division

- Rounding down to nearest integer
- $3 / 4 = 0.75$
- $3 // 4 = 0$
- $50 / 6 = 8.33$
- $50 // 6 = 8$
- No *Ceiling* operator with a single character like this... :-)

Ceiling With Python's Math Library

- `import math`
- `math.floor(5/6)`
- `math.ceil(5/6)`

Powers and Types

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Floor Division

- $2^3 = 2 * *3 = \text{math.pow}(2,3)$
- $(5 + 5)**5$
- $2^{(1/2)} = ??$

Variable Names

- $a1 = 2$
- `type(a1) #ls: <class 'int'>!`
- $a2 = 2.0$
- `type(a2) #ls: <class 'float'>!`
- $a3 = 3/4$
- `type(a3) #ls: <class 'float'>!`

Type Conversions

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Start with an integer, end with a float

- `a_int = 3`
- `bflt = 0.1415`
- `print(a_int + bflt)`
- `type(a_int + bflt) #ls: <class 'float'>!`

Start with a float, end with an integer

- `aflt = 3.1415`
- `b_int = int(aflt) # conversion`
- `type(b_int) #ls: <class 'int'>!`

Start with a string, end with an integer

- `a_str = "3.1415"`
- `bflt = float(a_str) #ls: <class 'float'>!`
- `c_int = int(bflt) #ls: <class 'int'>!`

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Working With Fractions

- *from fractions import Fraction*
- `f = Fraction(3,4)`
- `f`
- `print(f)`

Complex/Imaginary Numbers

$$i = \sqrt{-1}$$

- $i = \text{sqrt}(-1)$
- $i^2 = -1$

$$a + bi$$

Real Part \rightarrow a b \rightarrow Imaginary Part \rightarrow i \rightarrow $\sqrt{-1}$

$$1 + i$$

$$39 + 3i$$

$$0.8 - 2.2i$$

$$-2 + \pi i$$

$$\sqrt{2} + i/2$$

Complex/ Imaginary Numbers

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

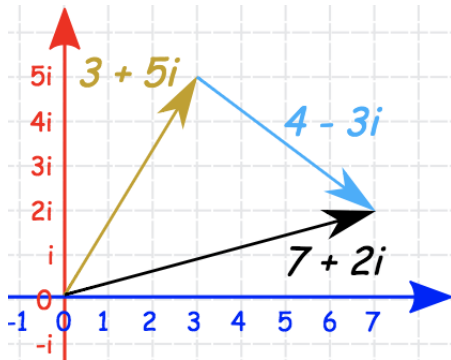
Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting



$$1 + i$$

$$39 + 3i$$

$$0.8 - 2.2i$$

$$-2 + \pi i$$

$$\sqrt{2} + i/2$$

- `a_cp = 2 + 3j` #ls: `<class 'complex'>`!

Fractions

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

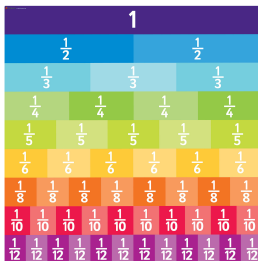
Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting



```
from fractions import Fraction
a = Fraction(3,4)
type(a) #<class 'fractions.Fraction'>
```

```
print(Fraction(3,4)) #3/4
```

```
Fraction(3,4) - Fraction(1,1) # Fraction(-1, 4)
print(Fraction(3,4) - Fraction(1,1)) #-1/4
```

Exceptions

When working with data-types, use exception handling

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Wrong data-type for python keyword input()

- Invalid inputs lead to errors:
 - `a = float(input())` #enter "Hello"
 - `ValueError: could not convert string to float`
 - `float(input())` # was not possible
 - `float("hi")` # also not possible

Use Exceptions

- `try: ... except: ...`
- Used to detect and prevent errors dealing with data types from crashing code.

Try and Except

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Wrong data-type for input()

```
try:
    a = float(input(` Enter a float :  `))
except ValueError:
    print(` Entry invalid...`)
```

Use Exceptions

- Used to detect and prevent errors dealing with data types from crashing code.
- Note that this exception handling will not crash the program.
- Can you build another exception handling block to catch strings being converted to integers?
 - Catch `int(input('hello'))`

Accept Integers or Floats, not Strings

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

sandbox/exceptionHandling

```
print("Exception handling...")
try:
    a_int = int(input("  Enter an integer, not a string  :"))
except ValueError:
    print("  Cannot convert string to ints or floats...")
```

Catch zeros in denominator of fractions

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Wrong data-type for input()

```
from fractions import Fraction # load library
print("Exception handling...")
a_fraction = Fraction(input('Enter a fraction: '))
```

Catch the exception

```
print("Exception handling...")
from fractions import Fraction # load library
try:
    a_fraction = Fraction(input('Enter a fraction: '))
except ZeroDivisionError:
    print(" Cannot divide by zero...")
```


Catch bad complex numbers

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

Wrong data-type for input()

```
from fractions import Fraction # load library
print("Exception handling...")
z_complex = complex(input('Enter a complex number: '))
```

Catch the exception

```
print("Exception handling...")
from fractions import Fraction # load library
try:
    z = complex(input('Enter a string as a complex number: '))
except ValueError:
    print("  This is not a complex number...")
```

Formatting strings

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

```
item1 = 'apples'
```

```
item2 = 'bananas'
```

```
item3 = 'grapes'
```

```
print('I have: {0} and {1} and {2}').  
format(item1, item2, item3))
```

```
#note: all on same line
```

```
print('I have: {0} and {1} and {2} and {3}').  
format(item1, item2, item3))
```

```
#fix:
```

```
print('I have: {0} and {1} and {2} and {3}').  
format(item1, item2, item3, "PINEAPPLES"))
```

Formatting Numbers

Make a formatted multiplication table

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

sandbox/multiplicationTable.py

```
def multi_table(a):  
    for i in range(1, 11):  
        print('{0} x {1} = {2}'.format(a, i, a*i))  
  
multi_table(4) # begin program by calling function
```

Miles to KM Converter with Formatting, part 1

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

sandbox/kmConverter.py

```
'''
Unit converter: Miles and Kilometers
'''
def print_menu():
    print('1. Kilometers to Miles')
    print('2. Miles to Kilometers')
#end of print_menu()

def km_miles():
    km = float(input('Enter distance in km: '))
    miles = km / 1.609
    print('Distance in miles: {0}'.format(miles))
#end of km_miles()
```

Miles to KM Converter with Formatting, part 2

Newton's
Method

Guess the
root

Derivatives

General
Equation

Simple
Example: x^2

Automate
with Python

Working with
Numbers

Type
conversions

Fractions

Exceptions

Formatting

sandbox/kmConverter.py

```
def miles_km():
    miles = float(input('Enter distance in miles: '))
    km = miles * 1.609
    print('Distance in kilometers: {0}'.format(km))
#end of miles_km()

if __name__ == '__main__':
    # Note: execute this program when run
    # importing this code into another
    # script will not prompt a menu

    print_menu()
    choice = input('Choose a conversion : ')
    if choice == '1':
        km_miles()
    if choice == '2':
        miles_km()
```