

Notes on Chapter 3 - Some Simple Numerical Programs

Swarup Tripathy *

January 2022

A curated list of important points for my reference.

1. One thing that distinguishes Python from other programming languages is that it is interpreted rather than compiled. This means that it is executed line by line, which allows programming to be interactive in a way that is not directly possible with compiled languages like Fortran, C, or Java.
2. Multiline comments
 - Triple quoted text should not be considered multi-line comments; by convention, they are docstrings
 - Additional comment styles add unnecessary complications and could decrease readability
 - Guido Tweet: You can use multi-line strings as multi-line comments. Unless used as docstrings, they generate no code.
 - Triple quotes are treated as regular strings with the exception that they can span multiple lines. By regular strings I mean that if they are not assigned to a variable they will be immediately garbage collected as soon as that code executes. Hence are not ignored by the interpreter in the same way that a comment is.
3. Ctrl + C terminates the infinite looping
4. Modern computers are amazingly fast. It takes on the order of one nanosecond-one billionth of a second-to execute an instruction.
5. The range function takes three integer arguments: start, stop, step

*John V Guttag

6. Exhaustive Enumeration Algorithm = Variant of guess and check. We enumerate all possibilities until we get to the right answer or exhaust the space of possibilities.
7. The **for** statement can be used in conjunction with the **in operator** to conveniently iterate over characters of a string.
8. If problem statement given *finding square root of non-negative number* or *finding square root of 2* would be initially answered as **cannot be solved**. The right thing to have asked for is a program that *finds an approximation to the square root* i.e., an answer that is close enough to the actual square root answer.
9. Exhaustive enumeration is a search technique that works only if the set of values being searched includes the answer.
10. ***Bi-Section Search Algorithm***
 - Taking the example of finding a square root of a given number
 - Suppose we know that a good approximation to the square root of x lies somewhere between 0 and \max .
 - We can exploit the fact that numbers are totally ordered.
 - For any pair of distinct numbers, n_1 and n_2 , either $n_1 \leq n_2$ or $n_1 > n_2$. So, we can think of square root of x as lying somewhere on the line and start searching for that interval.
 - Since, we don't necessarily know where to start searching, we begin from the middle. If that is not the right answer (and it won't be most of the time), ask whether it is too big or too small.
 - If it is too big, we know that the answer lies to the left.
 - If it is too small, we know that the answer lies to the right.
 - We then repeat the process on a smaller interval.

```

x=9
epsilon = 0.01
numGuesses = 0
low = 0.0
high = max(1.0,x)
ans = (high+low)/2.0
while abs(ans**2 -x) >= epsilon:
    print('low =',low,'high =',high,'ans =',ans)

```

```

numGuesses += 1
if ans**2 < x:
    low = ans
else:
    high = ans
ans = (high + low)/2.0

print('numGuesses =',numGuesses)
print(ans,'is close to square root of',x)

```

11. In almost all modern programming languages non-integer numbers are implemented using a representation called *Floating Point*.

12. ***Newton Rhapson Method for finding Square root***

Given question to us is that we need to find x such that $x^2 - 24$ is within the range of epsilon value i.e., 0.01

Code

```

epsilon = 0.01
k = 24
guess = k/2.0
while abs(guess*guess - k) >= epsilon:
    guess = guess - (((guess**2)-k)/(2*guess))
    print(guess)
print('Square root of',k,'is about',guess)

```

Output

```

7.0
5.214285714285714
4.908512720156556
4.8989887432139305
Square root of 24 is about 4.8989887432139305

```

We also finally observe that NR method is much more efficient than Bisection algorithm since the total number of Guesses taken by NR method were much lesser than guesses taken by Bisection algo