
NEWTON' S METHOD

One of the most basic recursive numerical techniques is Newton' s Method (also known as the Newton - Raphson method). In this classnote, I will demonstrate a specific form of Newton' s method that we can use to determine the square root of a number. You will write a *Mathematica* based algorithm using this method on an upcoming homework assignment.

Consider a number N , we can find the square root of this number by applying iteratively :

new estimate = $0.5 ((N/\text{previous estimate}) + \text{previous estimate})$

Suppose we want to find the square root of 81 (a number we already know), and we make our initial estimate for the square root 1; Newton' s method gives us :

1 st estimate = 1

2 nd estimate = $0.5 (81 / 1 + 1) = 41$

3 rd estimate = $0.5 (81 / 41 + 41) = 21.488$

4 th estimate = $0.5 (81 / 21.488 + 21.488) = 12.629$

5 th estimate = $0.5 (81 / 12.629 + 12.629) = 9.522$ and you can see that we are getting close

How many iterations do we compute before we say we have an answer? A typical diagnostic is to truncate the iterations when the next estimate varies from the previous estimate by less than some precision; for this case, I will use a precision of 0.001 (meaning when the new estimate differs by less than 0.001 from the previous estimate, we will declare victory and say we have our value for the square root of 81).

6 th estimate = $0.5 (81 / 9.522 + 9.522) = 9.015$

7 th estimate = $0.5 (81 / 9.015 + 9.015) = 9.002$

8 th estimate = $0.5 (81 / 9.002 + 9.002) = 9.001$

(If I were showing 5 sig figs, we would see that this estimate is just slightly off by more than 0.001 from the 7 th estimate, but you should get the idea of how this process works.)