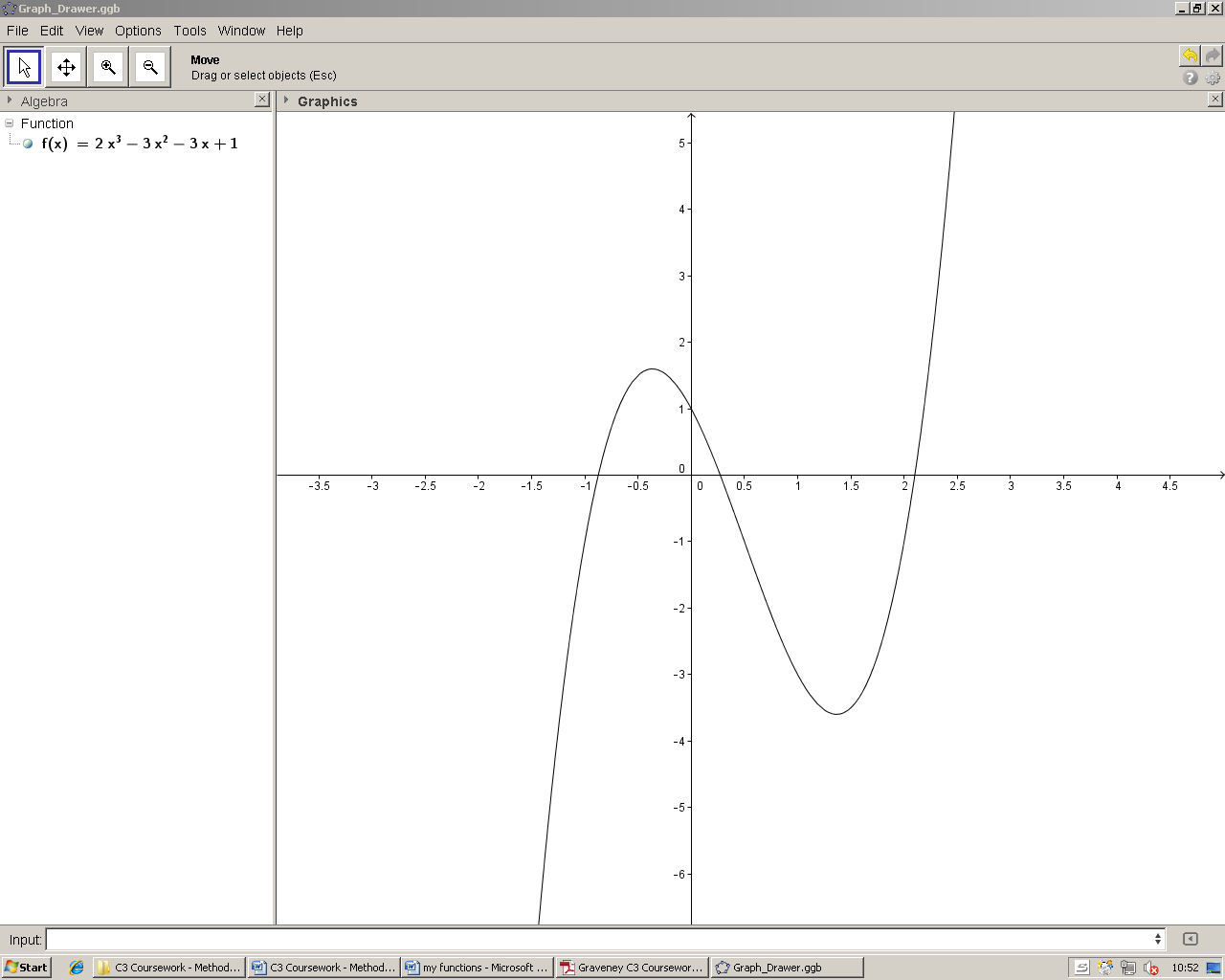
|  |
| --- |
| C3 Coursework |
| Methods for Solving Equations |
| By Norbert Podsadowski (Candidate 8745) |

|  |
| --- |
| 2015  For the OCR Mathematics (MEI) specification |

Method 1: Decimal search (change of sign)

Equation 1: (Success)



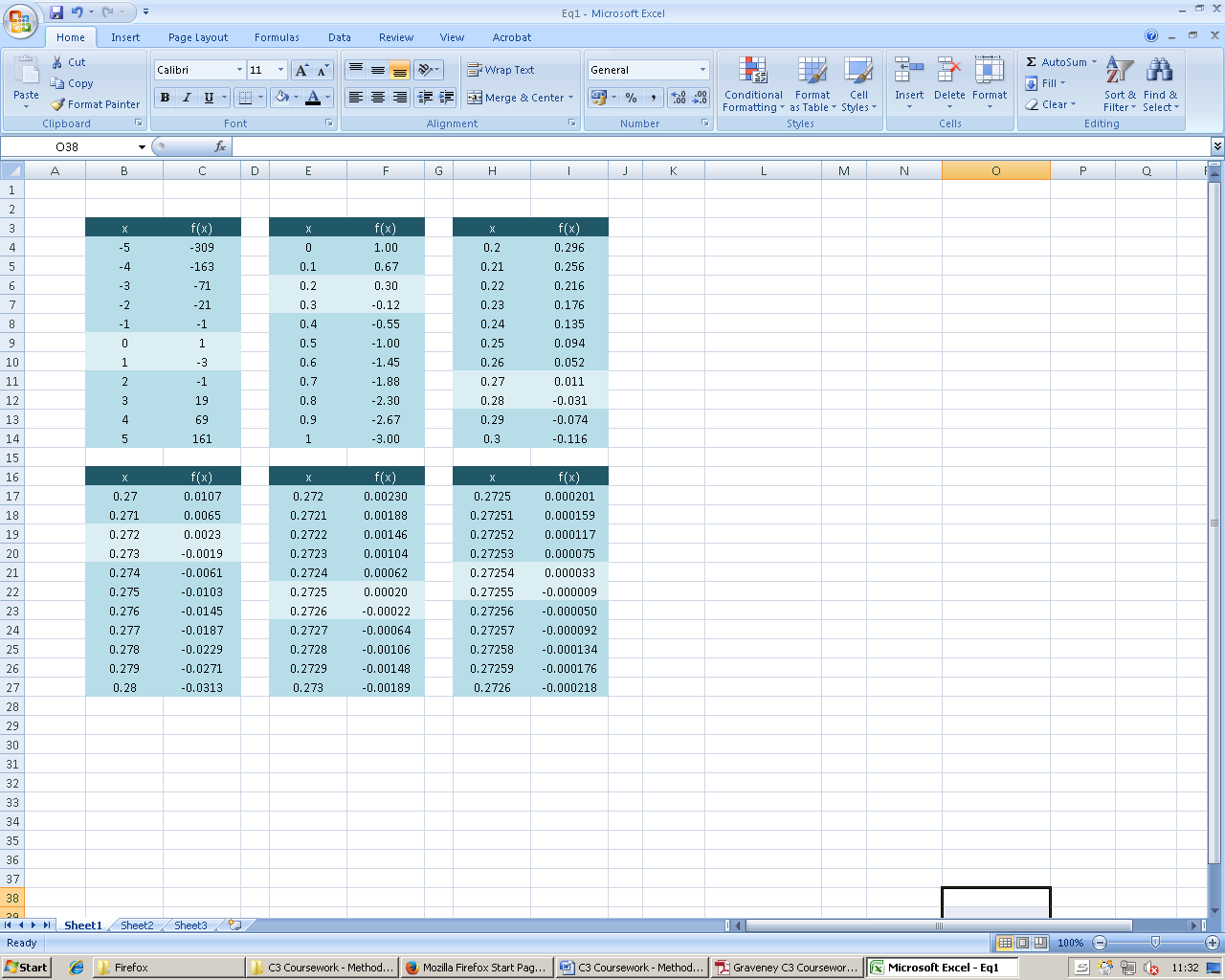
Root 3

y

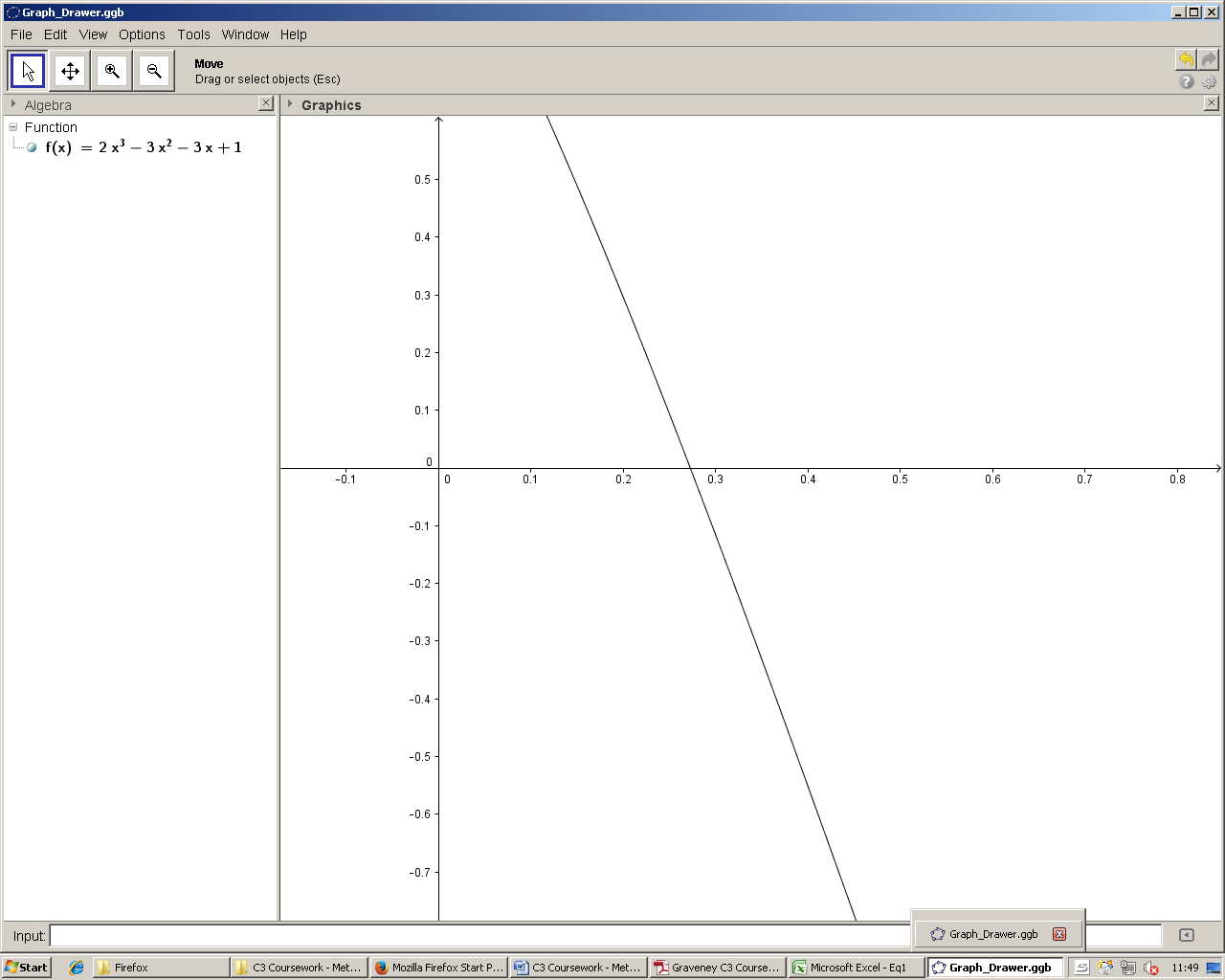
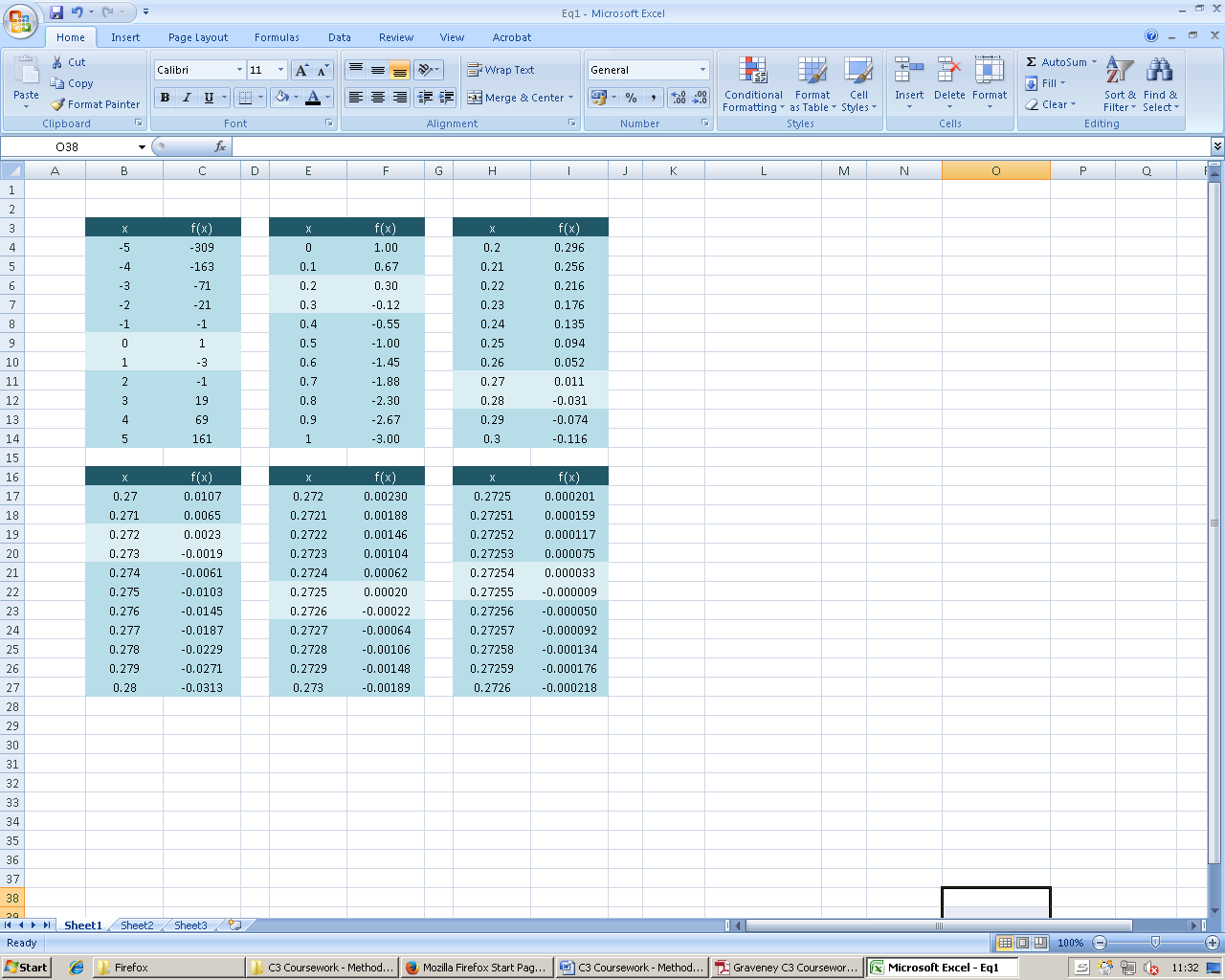
x

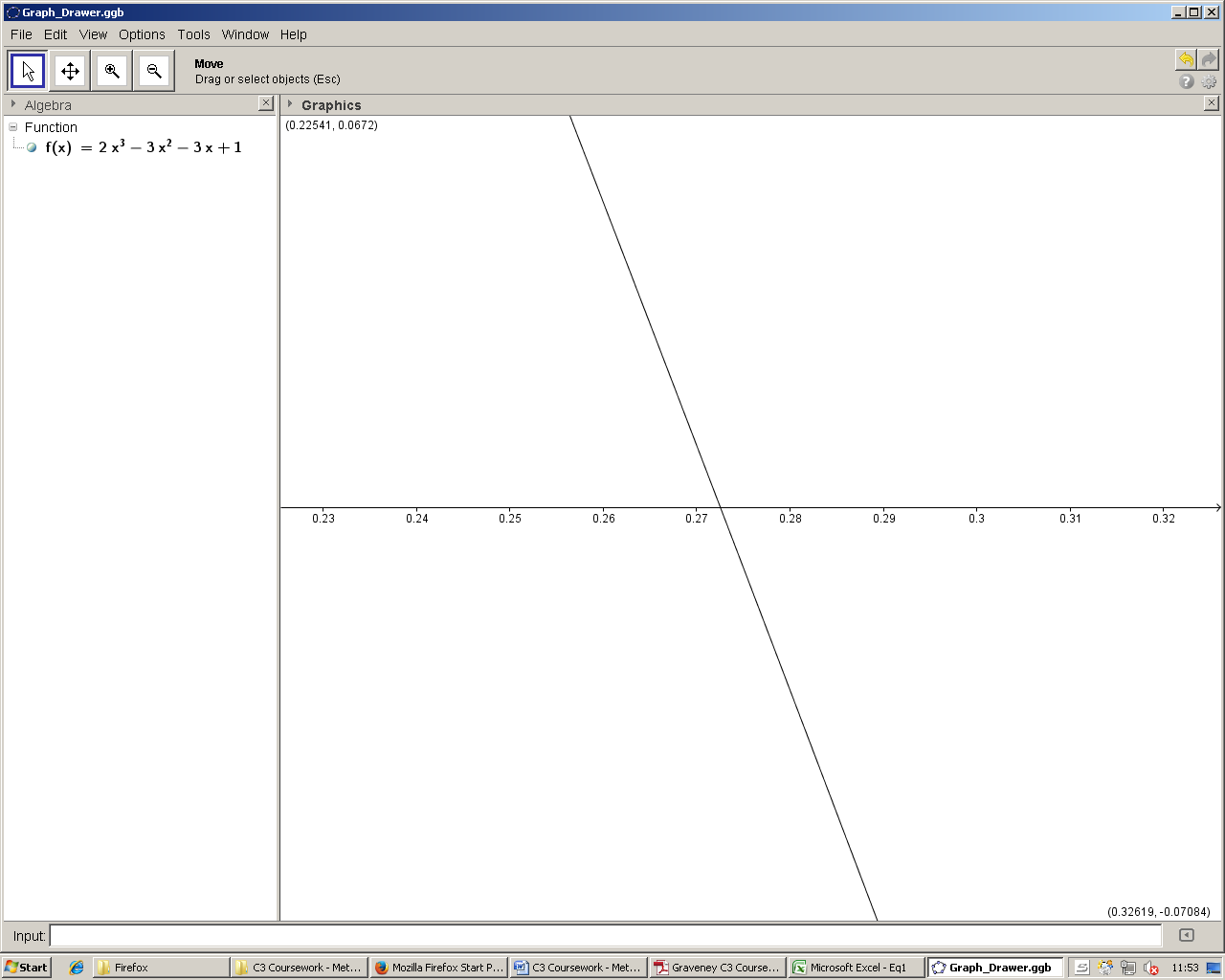
Root 2

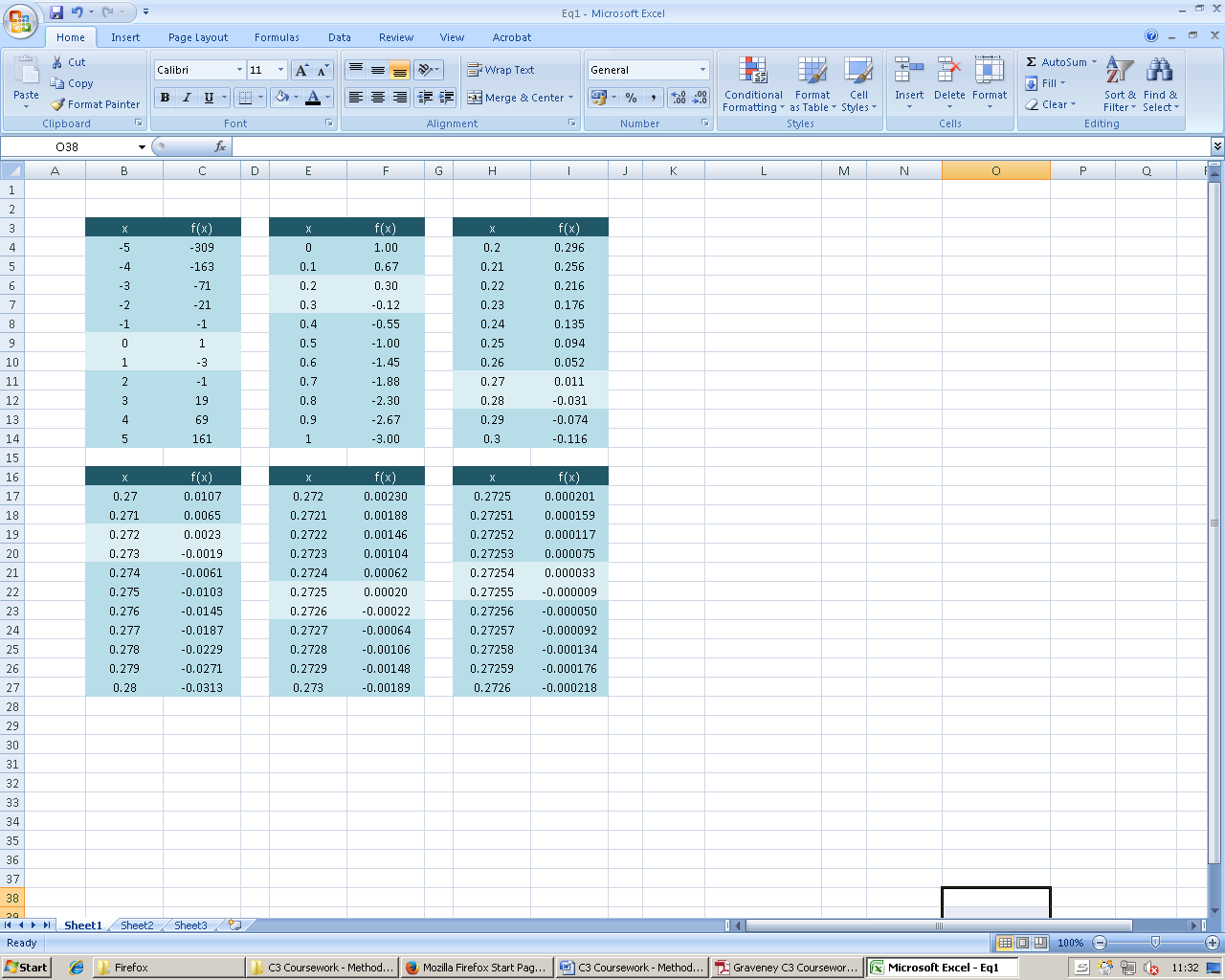
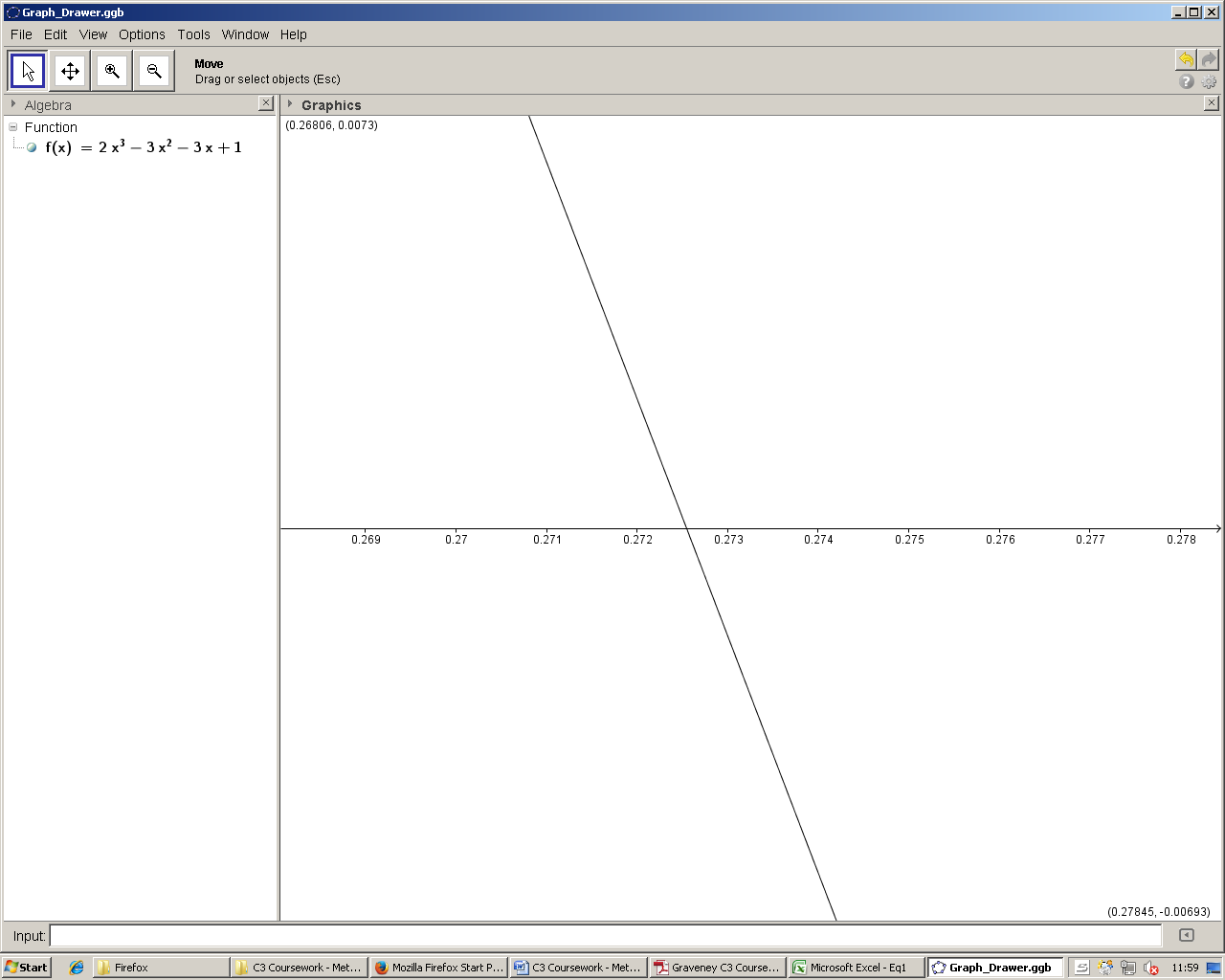
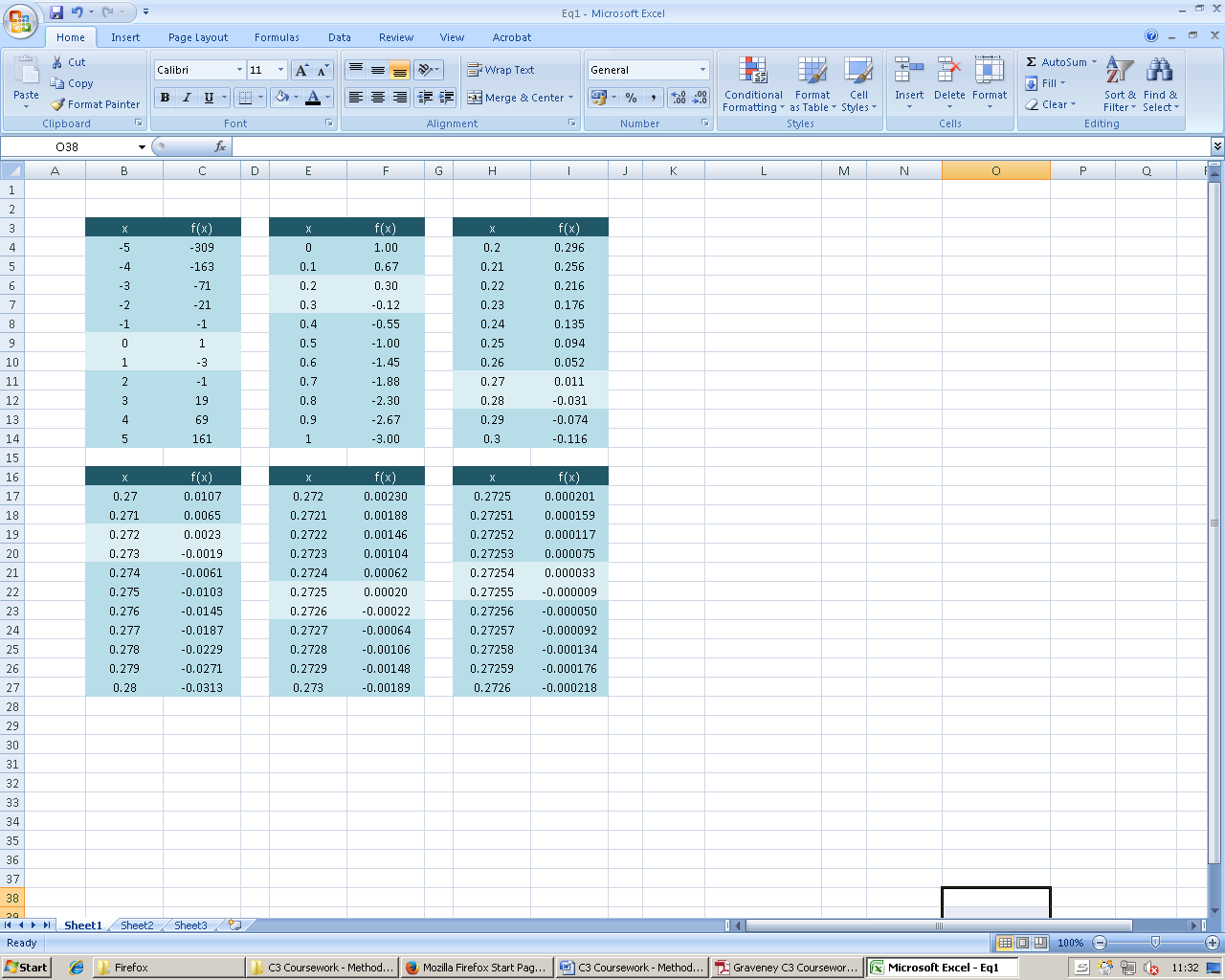
Root 1

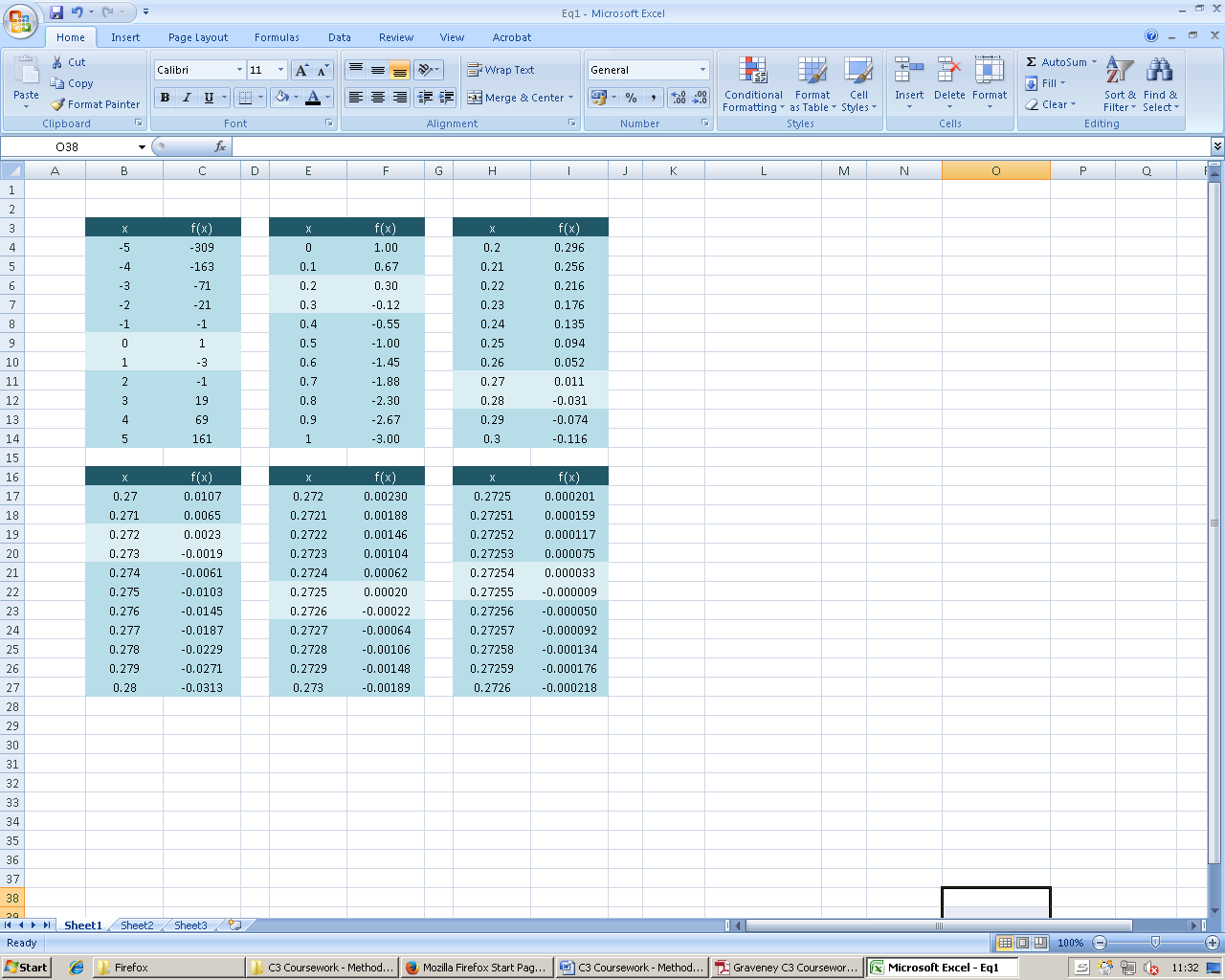
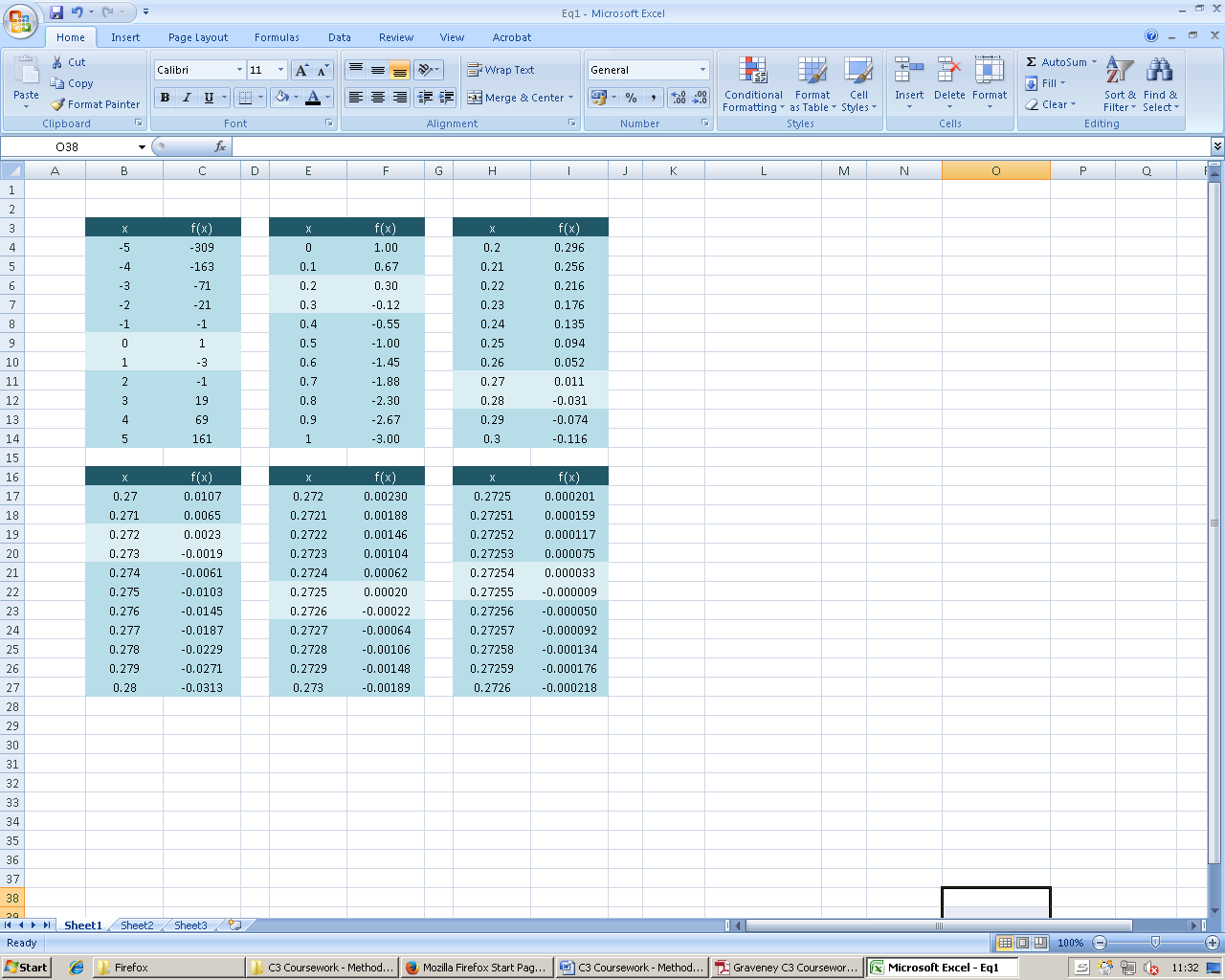
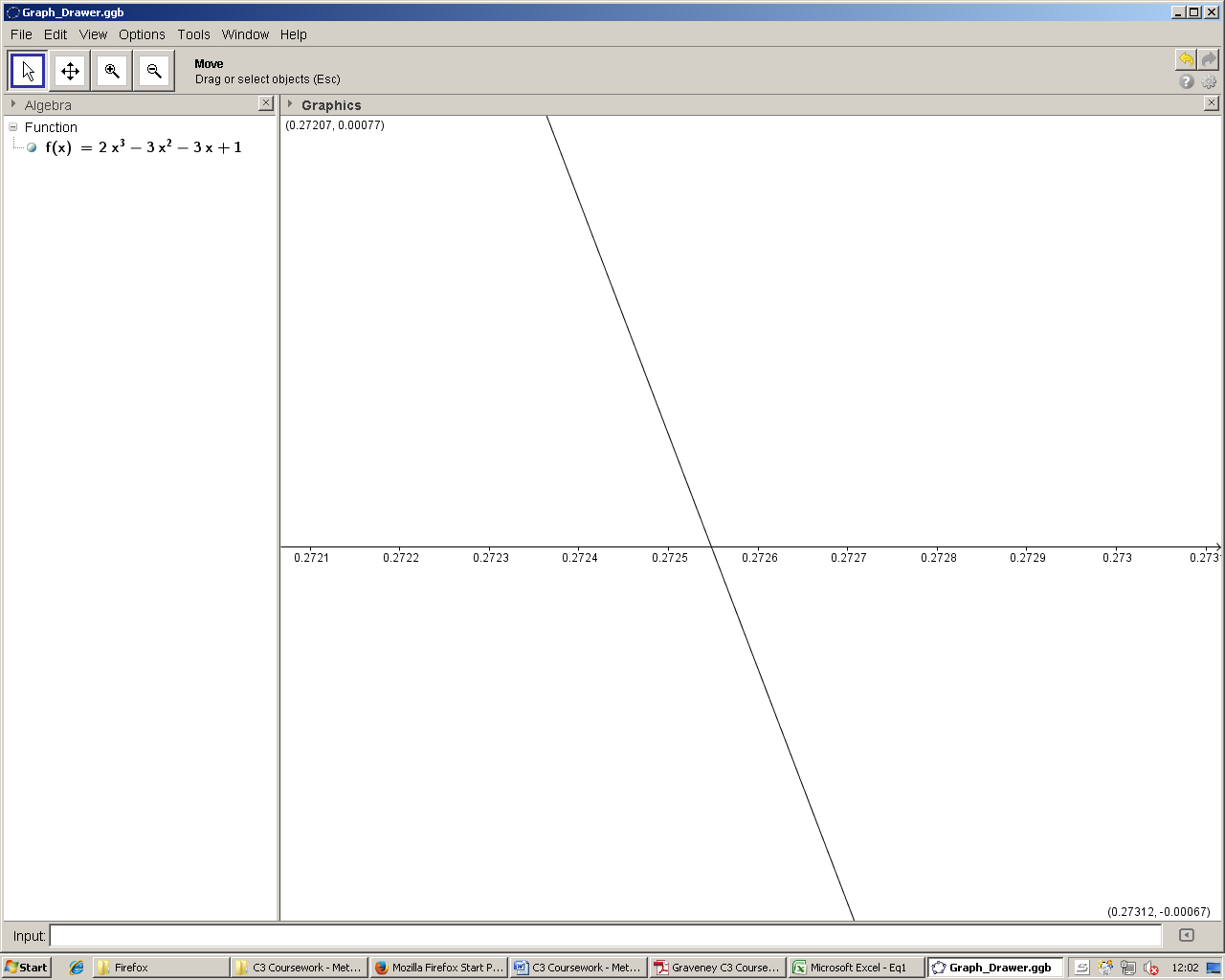
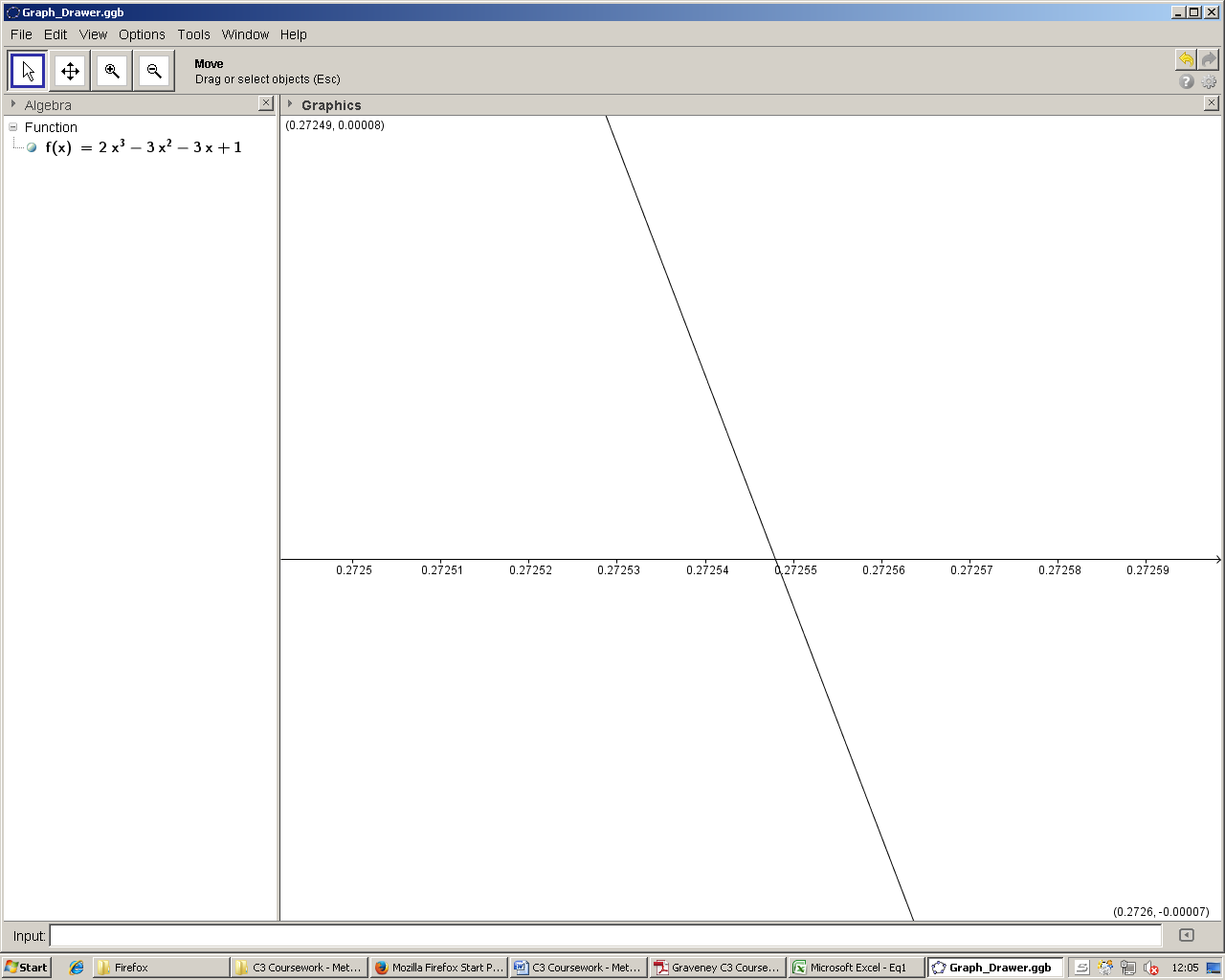
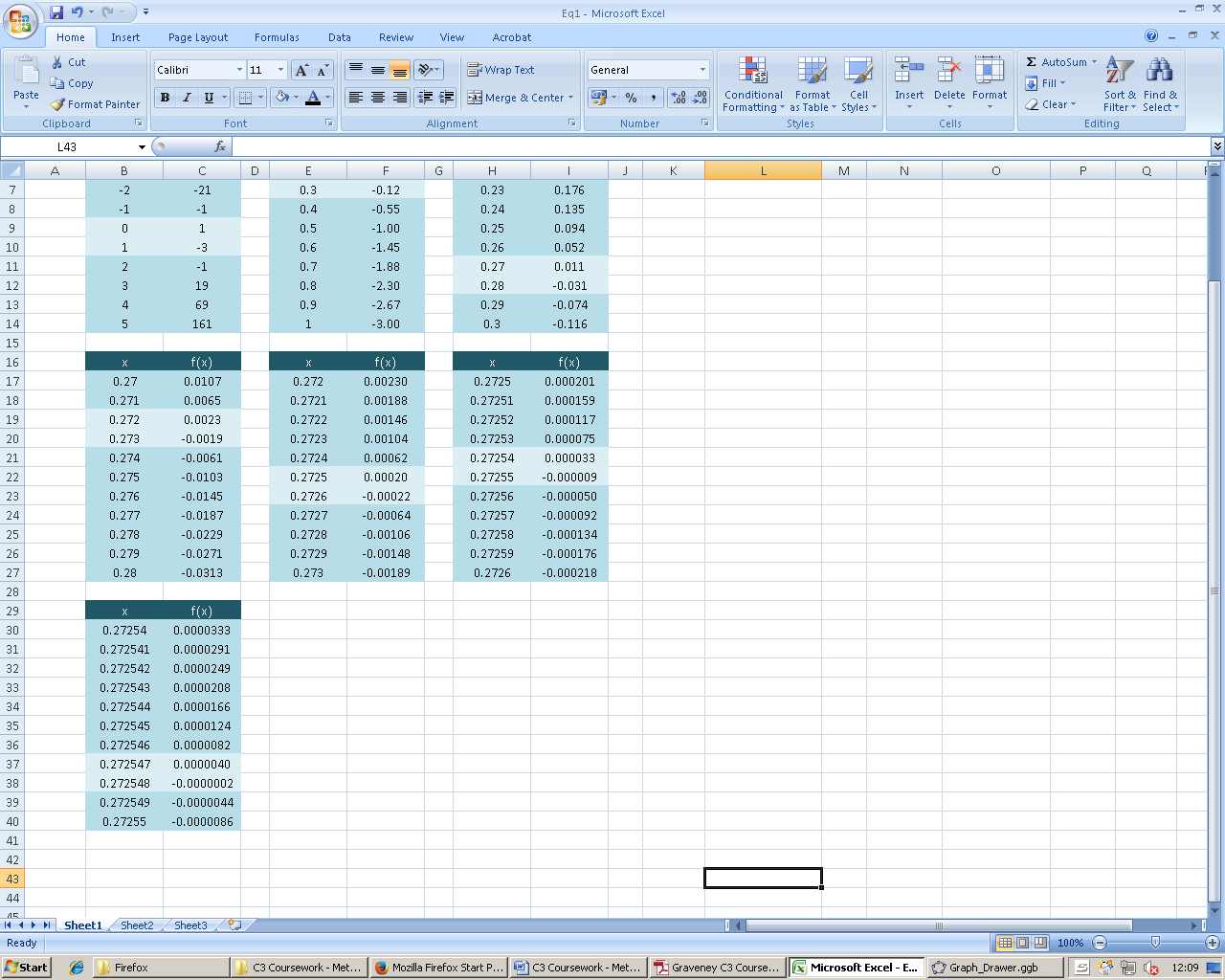
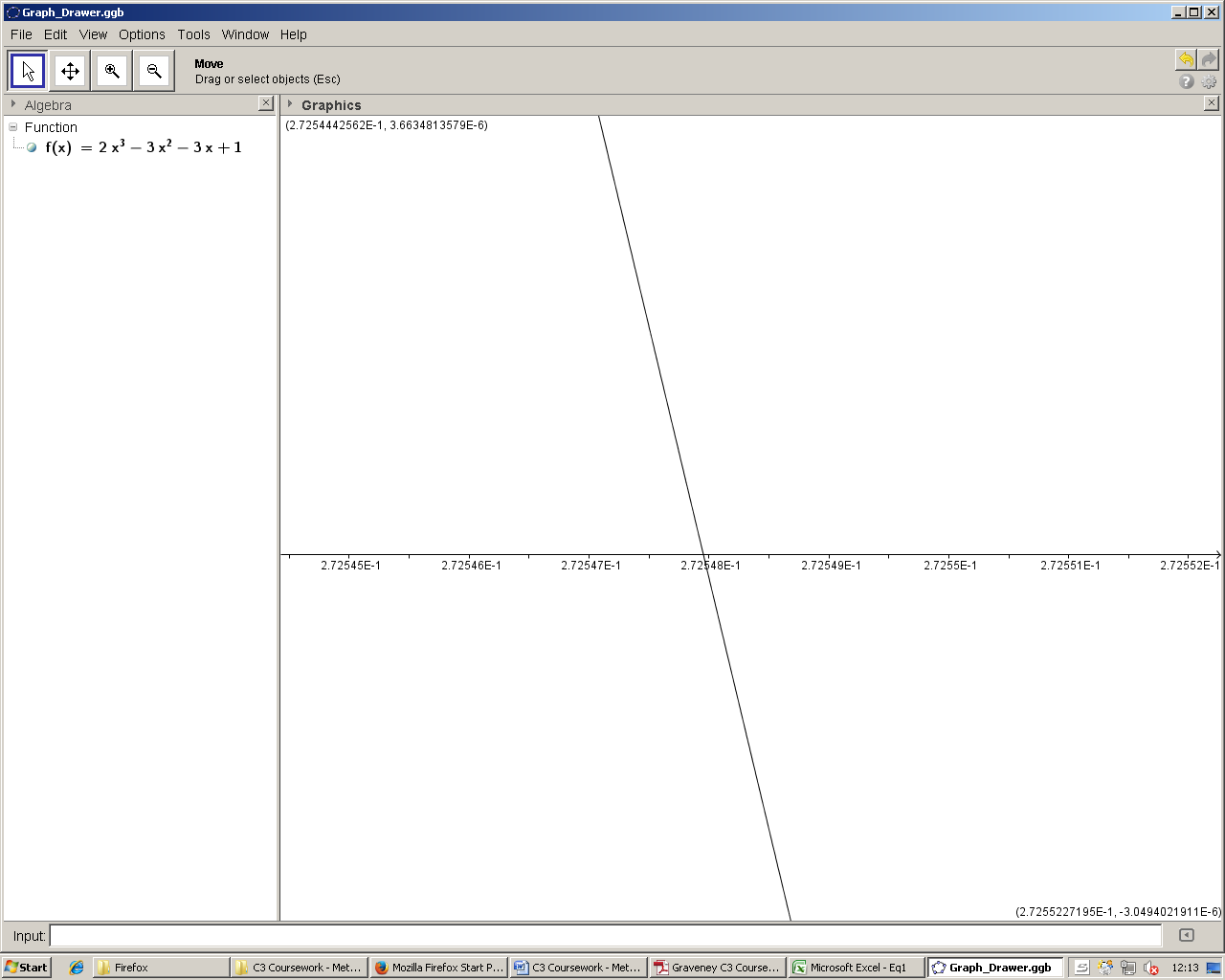
The decimal search method involves looking for a change of sign between intervals which get increasingly smaller. A change of sign signifies that a root must lie within the interval of where the change of sign occurred. For the first table of values, integer increments of 1 are used. If a change of sign occurs during this initial interval, another interval is used between the x-values which caused the sign change, with its increment decreased by a factor of 10 (i.e. 0.1 for the second table, 0.01 for the third, etc.). This process is repeated until the root is found to the desired degree of accuracy.

This is the first table of a decimal search for the function above. An initial interval of [-5, 5] is chosen. Using a spread sheet, the function is evaluated at x = -5, x = -4, x = -3, and so on. The sign is found to change at three different points – between [-1, 0], [0, 1] and [2, 3] for roots 1, 2 and 3 respectively. To demonstrate how to find root 2, the interval for the next table is [0, 1].

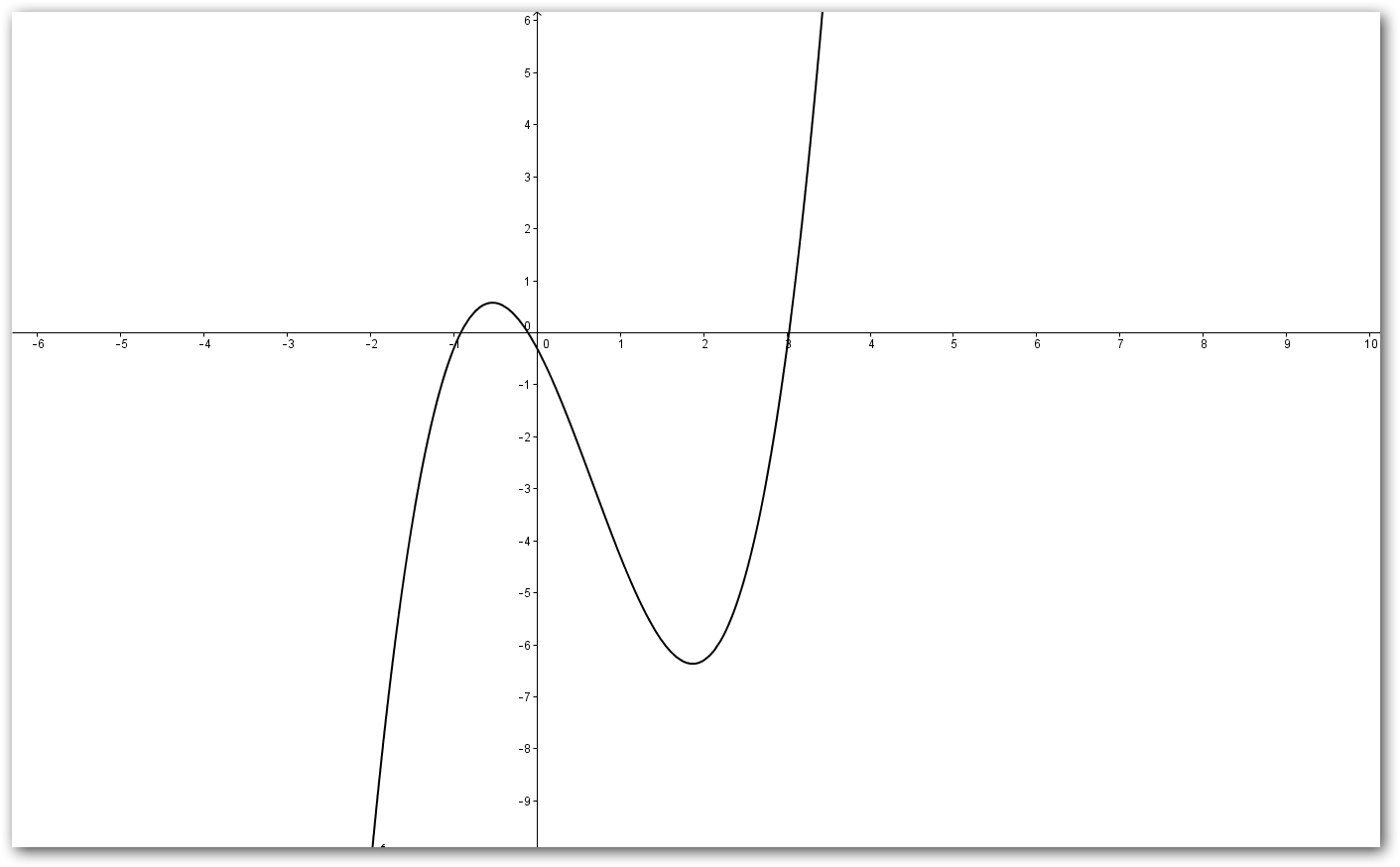
Now the function is evaluated at 0, 0.1, 0.2, etc, until a change of sign is found, and the process is repeated to‘hone in’ on the root. This may be illustrated graphically:

The root lies between [0.2, 0.3] as a change of sign has occurred during that interval. The next table of values evaluates x = 0.21, 0.22, 0.23… and this process is repeated until x reaches 6 significant figures.



After 7 tables of values: (5 significant figures). The maximum error bounds of root 2 are .

Equation 2: (Failure)



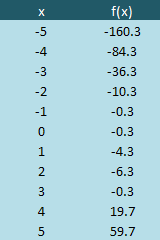
y

Root 1

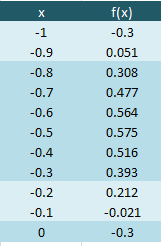
Root 2

x

Root 3

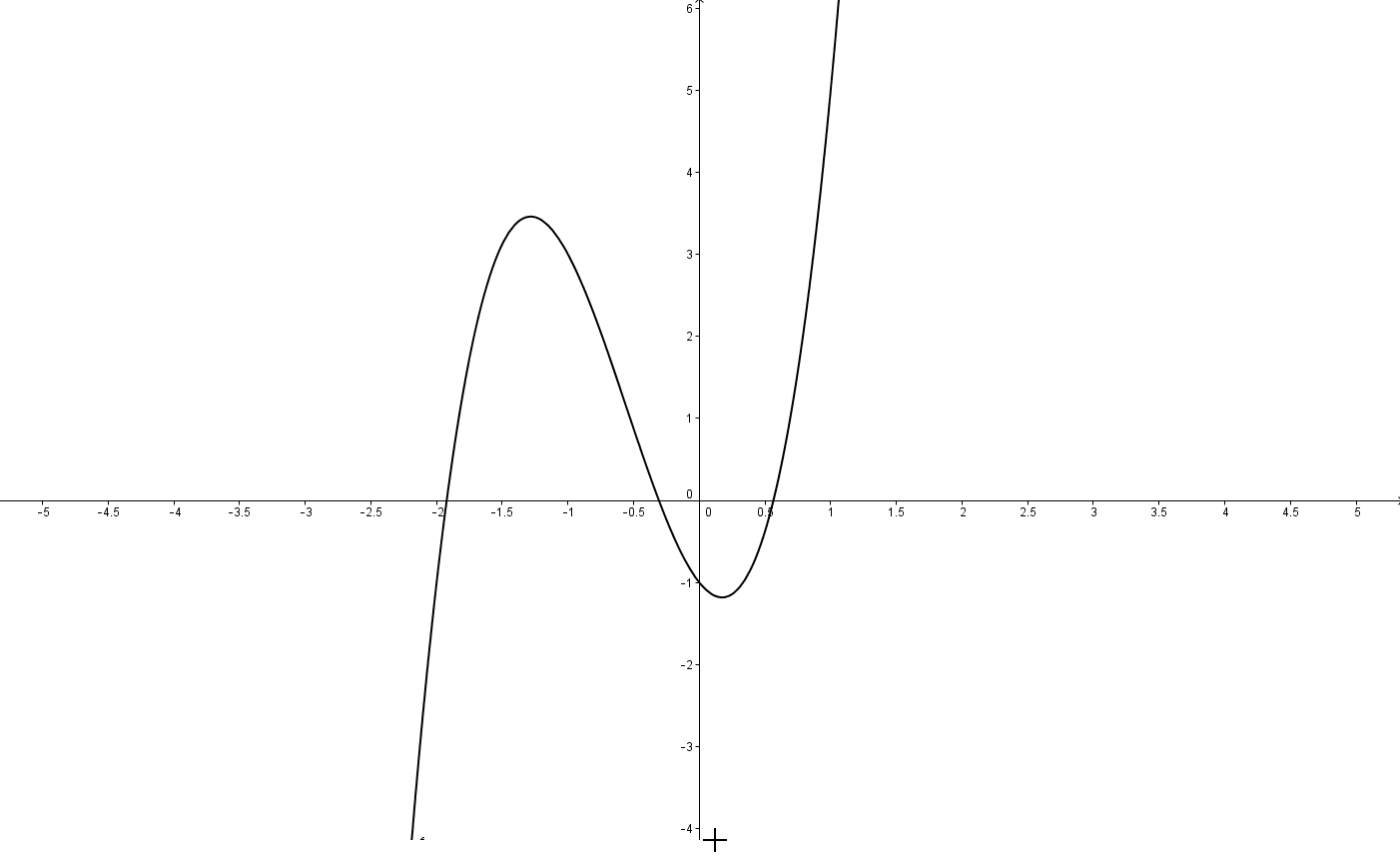


The decimal search method will be used in attempt to find root 1, which lies in the same integer interval as root 2 [-1, 0]. Like with the first equation, an initial table of integer values is used. One change of sign occurs between [3, 4] which shows the presence of root 3. However, as no change of sign is detected between [-1, 0], root 1 and 2 would simply be ignored here. As on the graph, mulitple roots (1 and 2) are in the same integer interval, and a computer programmed to find roots after detecting a change of sign would simply ignore these and proceed to only find root 3.



If the table was done to 1 decimal point (as on the right), then two roots would be detected and found. However, a program using a decimal search method can never be able to find roots of every single equation – no matter how small the initial interval (1dp in this case), there will always be an infinite number of curves which can have multiple roots in between any interval, making their roots undetectable.

Method 2: Newton-Raphson

Equation 3: (Success)

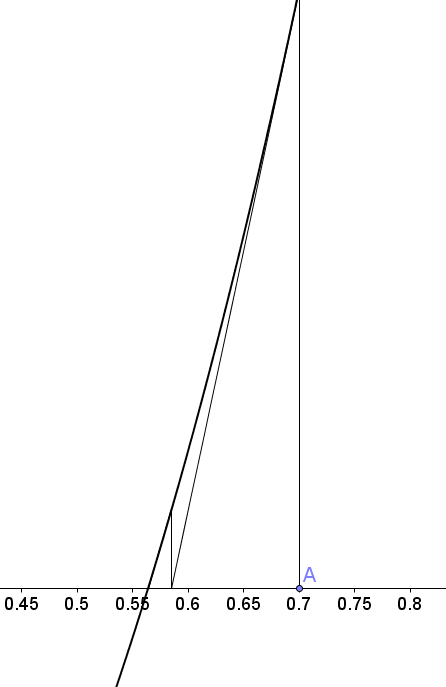
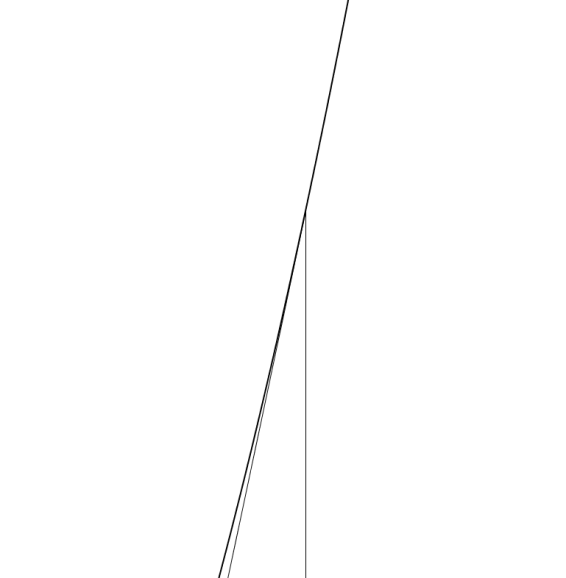
y

x

Root 1

Root 2

Root 3

The Newton-Raphson method of solving equations is a fixed-point iteration method. Firstly an estimate for the root is chosen – in this example, root 3 will be found, and the graph is used to provide an initial estimated value of the root at 0.7. The equation of the tangent to the curve at x = 0.7 is then used to give the next estimate of the root at the point where the tangent crosses the x-axis. This may be illustrated graphically as on the left.

Tangent at x = 0.7

Initial estimate, x0

New estimate, x1

x

The value of the initial estimate is x0 = 0.7. Using the Newton-Raphson iterative formula:

This is the value of x1 and can be used to find x2, which can then be used to find x3 and so on, until the root is found to the desired degree of accuracy.

[illustration]

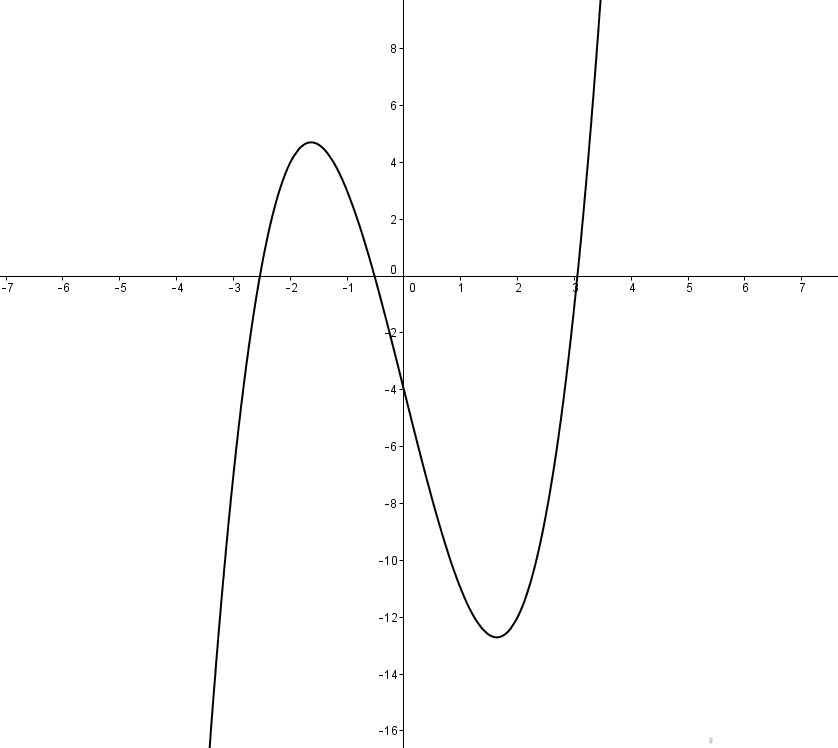
After 4 iterations, the root is found to be to 5 significant figures. In order to establish the error bounds, the function is evaluated at .

A change of sign occurs between these two values. The maximum error bounds are therefore .

Equation 4: (Failure)

Method 3: Rearrangement

Equation 5:



y

Root 1

x

Root 2

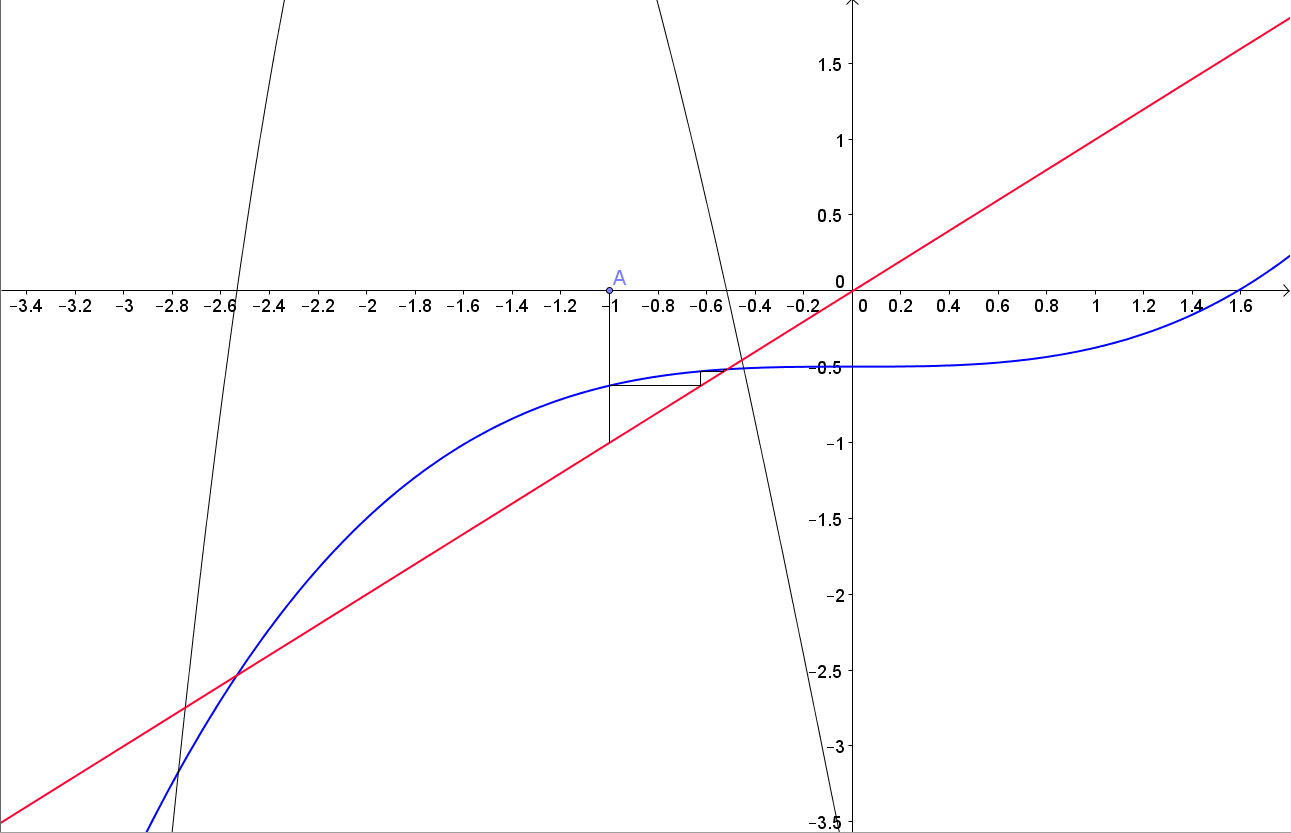
Root 3

The rearrangement method is another fixed-point iteration method and involves rewriting the equation into the form . The exact roots of the equation are at the points of intersection of and . This form allows an initial starting value to be evaluated, and the result of this being used to find the next value, and so on, until the values converge to the root being searched for. This gives rise to the iterative formula:

The equation can be rearranged as follows:

This is now in the form , and substituting this into the iterative formula gives:

This formula may now be used to find the roots of this equation. Root 2 will be found in this example. A starting value close to the root is chosen: . Substituting into the formula:

This is the value of and can be used to find , which can then be used to find and so on, until the root is found to the desired degree of accuracy.

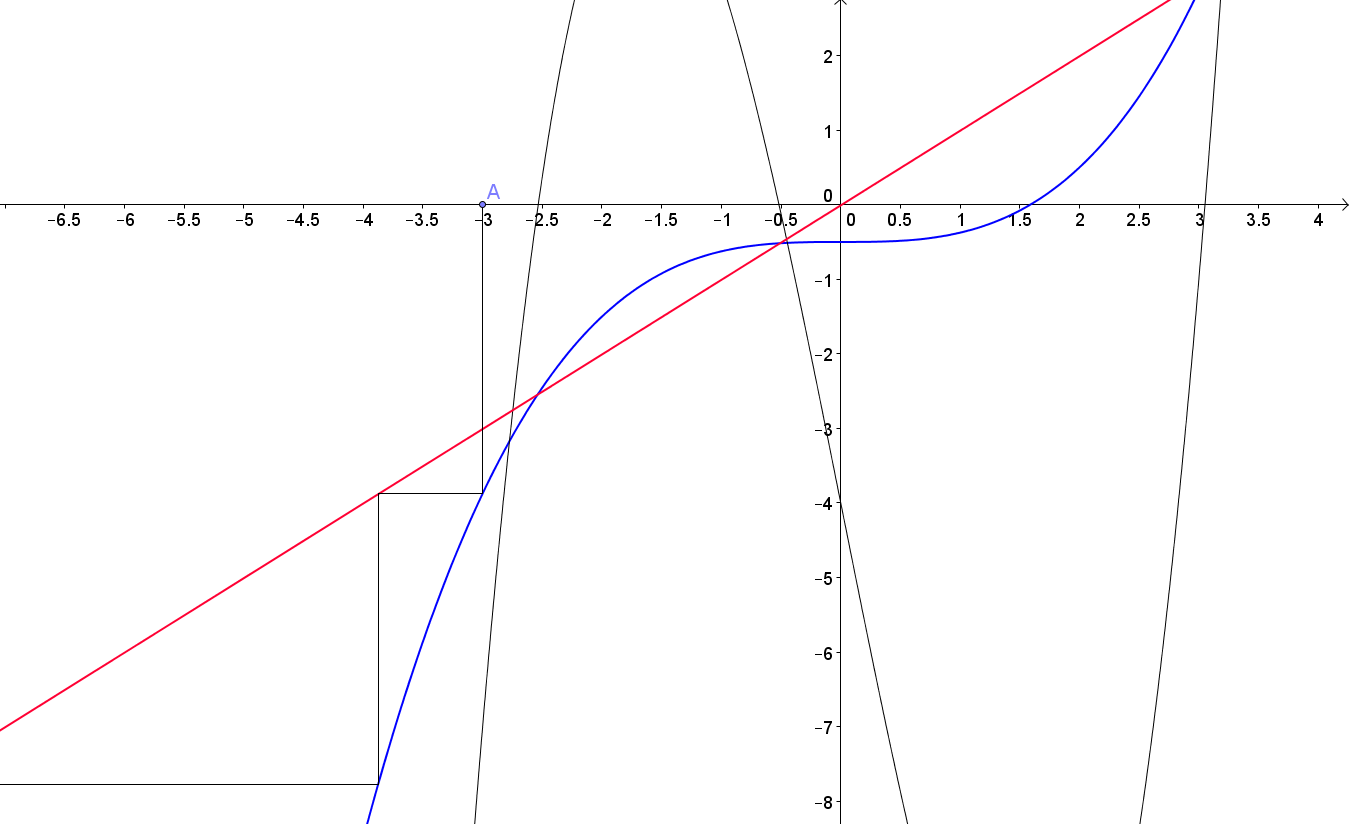
x

y

Root 2

After 10 iterations, the root is 0.51730 to 5 significant figures.  
The gradient of the function is important. Differentiating:

At values of close to the desired root the gradient satisfies the condition :

If this condition was not satisfied for values of close to the root, then the rearrangement method would fail to find root 2. This failure is demonstrated with root 1:

Root 1

y

The values diverge from the desired root as the gradient of near the rot is not between -1 and 1:

This rearrangement therefore works with finding root 2, but not root 1, highlighting the limitations of this method of solving equations.

Comparison of Methods