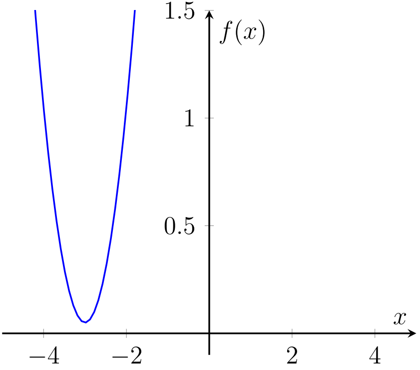
- Complications.
- Roots of polynomials.
- Non-linear systems of equations with multiple variables.
- Summary.

Complications

- There are no roots at all!

The hardest thing of all is to find a black cat in a dark room, especially if there is no cat.

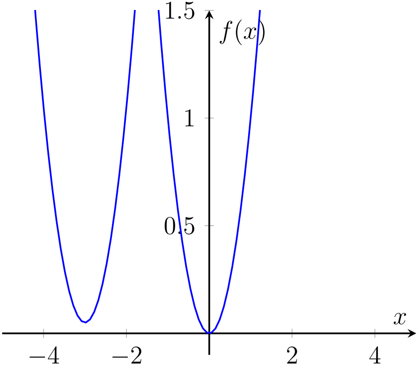
Confucius.



- There are no roots at all!

> The hardest thing of all is to find a black cat in a dark room, especially if there is no cat.
>
> Confucius.

- There is one root but the function does not change sign.

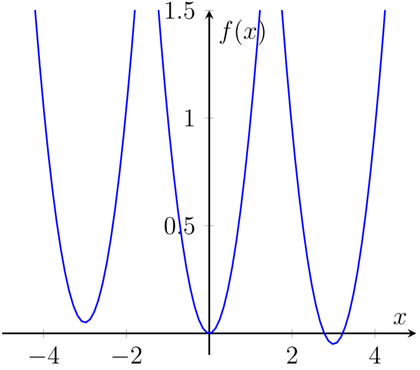


Complications

- There are no roots at all!

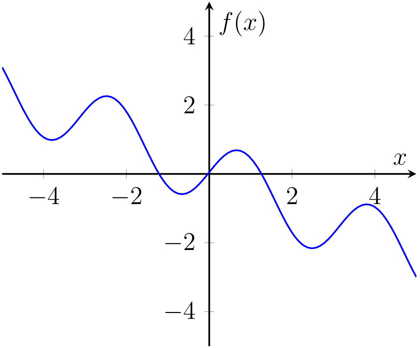
> The hardest thing of all is to find a black cat in a dark room, especially if there is no cat.
>
> Confucius.

- There is one root but the function does not change sign.
- There are two or more roots in the interaval $[a, b]$.



More complications

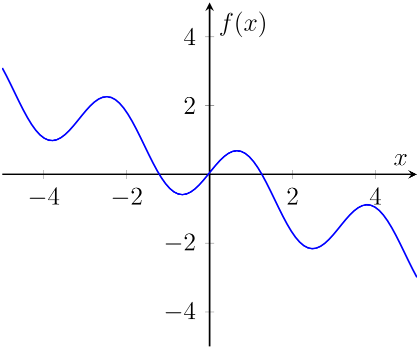
There are many roots.



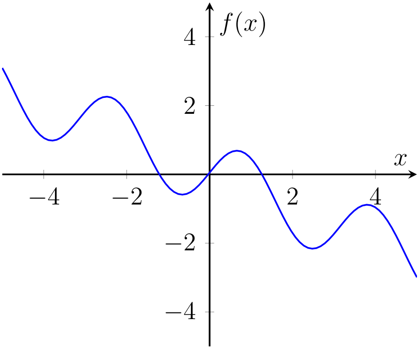
What will happen if you use bracketing methods for this?

Multiple roots: Brute force method

- The brute force method is a good method for dealing with multiple roots.



- The brute force method is a good method for dealing with multiple roots.
- One splits the original interval $[a, b]$ into smaller intervals with some step size (say h) and then applies the previously mentioned methods in each of the sub-intervals.



Step size in Brute force method

- If the step size is too large – one may miss multiple roots.

Step size in Brute force method

- If the step size is too large – one may miss multiple roots.
- Choosing too small a step size will result in too many computations.

Step size in Brute force method

- If the step size is too large – one may miss multiple roots.
- Choosing too small a step size will result in too many computations.
- A graphical analysis is very helpful in deciding the step size h.

Step size in Brute force method

- If the step size is too large – one may miss multiple roots.
- Choosing too small a step size will result in too many computations.
- A graphical analysis is very helpful in deciding the step size h.
- *It may be good to evaluate roots with h and then with $h/10$ to confirm that the number of roots remains unchanged.*

Some general comments on roots of polynomials

- The fundamental theorem of algebra states that n^{th} degree polynomial has exactly n zeros or roots.

Some general comments on roots of polynomials

- The fundamental theorem of algebra states that n^{th} degree polynomial has exactly n zeros or roots.
- These roots can be real or complex.

Some general comments on roots of polynomials

- The fundamental theorem of algebra states that n^{th} degree polynomial has exactly n zeros or roots.
- These roots can be real or complex.
- If the coefficients are all real, then the complex roots always occur in conjugate pairs.

Some general comments on roots of polynomials

- The fundamental theorem of algebra states that n^{th} degree polynomial has exactly n zeros or roots.
- These roots can be real or complex.
- If the coefficients are all real, then the complex roots always occur in conjugate pairs.
- The roots may be simple (ie. single) or repeated (ie multiple).

Bracketing methods

- Bracketing methods cannot be used to find repeated roots with even multiplicity because the polynomial function does not change sign at such roots.

Bracketing methods

- Bracketing methods cannot be used to find repeated roots with even multiplicity because the polynomial function does not change sign at such roots.
- They can however be used to find repeated roots with odd multiplicity.

Bracketing methods

- Bracketing methods cannot be used to find repeated roots with even multiplicity because the polynomial function does not change sign at such roots.
- They can however be used to find repeated roots with odd multiplicity.
- Open domain methods are often more efficient than closed domain methods.

Open domain methods

- These methods can be used to find roots – both complex roots as well as real roots.

Open domain methods

- These methods can be used to find roots – both complex roots as well as real roots.
- For complex roots – one just needs to use complex arithemetic – the algorithms remain the same.

Open domain methods

- These methods can be used to find roots – both complex roots as well as real roots.
- For complex roots – one just needs to use complex arithemetic – the algorithms remain the same.
- There are various modifications of the Newton's method which are often used to find roots. For eg. Bairstow's method.

Non linear system of equations: two variables

- A non linear system of equations of two variables:

$$f(x, y) = 0$$
$$g(x, y) = 0$$

Non linear system of equations: two variables

- A non linear system of equations of two variables:

$$f(x, y) = 0$$
$$g(x, y) = 0$$

- There are no good general methods for solving multiple non linear equations.

Non linear system of equations: two variables

- A non linear system of equations of two variables:

$$f(x, y) = 0$$
$$g(x, y) = 0$$

- There are no good general methods for solving multiple non linear equations.
- Extension of bracketing methods is not obvious to systems of non linear equations.

Non linear system of equations: two variables

- A non linear system of equations of two variables:

$$f(x, y) = 0$$
$$g(x, y) = 0$$

- There are no good general methods for solving multiple non linear equations.
- Extension of bracketing methods is not obvious to systems of non linear equations.
- Open domain methods (especially Newton's method) are easily extended to solve such systems.

Non linear system of equations: two variables

- A non linear system of equations of two variables:

$$f(x, y) = 0$$
$$g(x, y) = 0$$

- There are no good general methods for solving multiple non linear equations.
- Extension of bracketing methods is not obvious to systems of non linear equations.
- Open domain methods (especially Newton's method) are easily extended to solve such systems.
- One always needs a good guess!!!

Extension of Newton's method to two variables

We are given two equations:

$$f(x, y) = 0$$
$$g(x, y) = 0$$

We have to find the solution (x^*, y^*) such that:

$$f(x^*, y^*) = 0$$
$$g(x^*, y^*) = 0$$

Taylor expanding about (x^*, y^*):

$$f(x, y) = f(x^*, y^*) + (x - x^*)f'_x + (y - y^*)f'_y + \dots$$
$$g(x, y) = g(x^*, y^*) + (x - x^*)g'_x + (y - y^*)g'_y + \dots$$

Extension of Newton's method to two variables

Keeping only first-order terms and given $f(x^*, y^*) = 0$ and $g(x^*, y^*) = 0$, one has a system of linear equations for x^* and y^* such that:

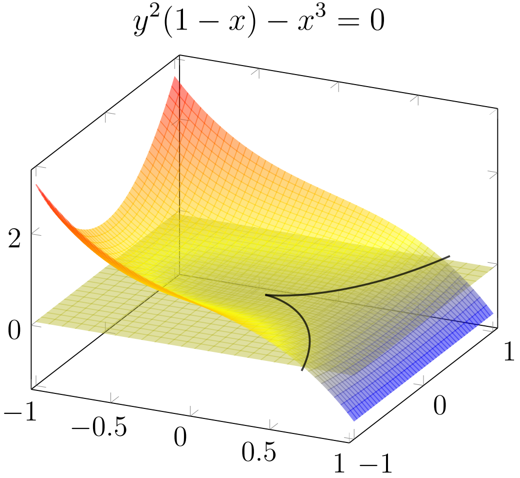
$$x^* = x + \frac{f'_y g(x,y) - g'_y f(x,y)}{f'_x g'_y - f'_y g'_x}$$
$$y^* = y + \frac{g'_x f(x,y) - f'_x g(x,y)}{f'_x g'_y - f'_y g'_x}$$

This method can be easily generalized to solving n non linear equations.

Non linear systems of equations: Example

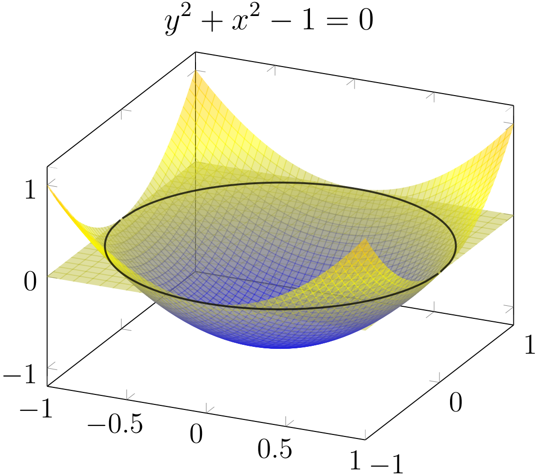
$$f(x,y) = y^2(1-x) - x^3 = 0$$
$$g(x,y) = y^2 + x^2 - 1 = 0$$

Non linear systems of equations: Example



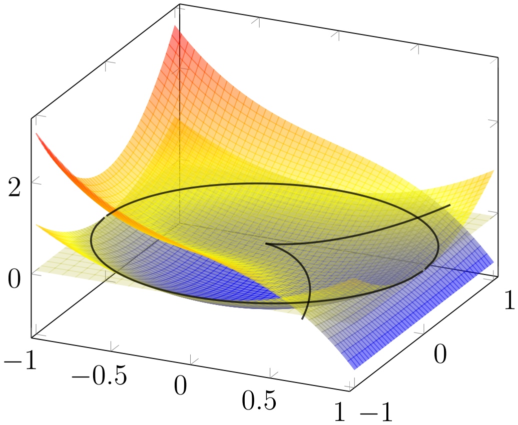
$h(x,y)=0$ is also shown. The black line shows the intersection of the surface with $h(x,y)=0$.

Non linear systems of equations: Example



$h(x, y) = 0$ is also shown. The black line shows the intersection of the surface with $h(x, y) = 0$.

Non linear systems of equations: Example



$h(x, y) = 0$ is also shown.

Newton's method: example

i	x	y	f(x,y)	g(x,y)	dx	dy
1	1.000000	1.000000	-1.000000	1.000000	-0.250000	-0.250000
2	0.750000	0.750000	-0.281250	0.125000	-0.119048	0.035714
3	0.630952	0.785714	-0.023352	0.015448	-0.012757	0.000414
4	0.618195	0.786128	-0.000298	0.000163	-0.000161	0.000023
5	0.618034	0.786151	-0.000000	0.000000		

Root is: 0.6180340120481262, 0.7861513762373674

i	x	y	f(x,y)	g(x,y)	dx	dy
1	1.000000	-1.000000	-1.000000	1.000000	-0.250000	0.250000
2	0.750000	-0.750000	-0.281250	0.125000	-0.119048	-0.035714
3	0.630952	-0.785714	-0.023352	0.015448	-0.012757	-0.000414
4	0.618195	-0.786128	-0.000298	0.000163	-0.000161	-0.000023
5	0.618034	-0.786151	-0.000000	0.000000		

Root is: 0.6180340120481262, -0.7861513762373674

Pitfalls of root finding

- Lack of good initial guess.

Pitfalls of root finding

- Lack of good initial guess.
- Convergence to the wrong root.

Pitfalls of root finding

- Lack of good initial guess.
- Convergence to the wrong root.
- Closely spaced roots/ multiple roots/ Inflection points.

Pitfalls of root finding

- Lack of good initial guess.
- Convergence to the wrong root.
- Closely spaced roots/ multiple roots/ Inflection points.
- Complex roots.

Pitfalls of root finding

- Lack of good initial guess.
- Convergence to the wrong root.
- Closely spaced roots/ multiple roots/ Inflection points.
- Complex roots.
- Ill conditioning of the non linear equations

Pitfalls of root finding

- Lack of good initial guess.
- Convergence to the wrong root.
- Closely spaced roots/ multiple roots/ Inflection points.
- Complex roots.
- Ill conditioning of the non linear equations
- Slow convergence

Some thoughts

- All the methods work for simple/smoothly varying problems if:

Some thoughts

- All the methods work for simple/smoothly varying problems if:
- One starts from a good initial guess.

Some thoughts

- All the methods work for simple/smoothly varying problems if:
- One starts from a good initial guess.
- Efficiency is important if the problem has to be solved multiple times.

Some thoughts

- All the methods work for simple/smoothly varying problems if:
- One starts from a good initial guess.
- Efficiency is important if the problem has to be solved multiple times.
- Anticipate complex roots.

Some thoughts

- All the methods work for simple/smoothly varying problems if:
- One starts from a good initial guess.
- Efficiency is important if the problem has to be solved multiple times.
- Anticipate complex roots.
- Trade off between your time and computer's time...

Some thoughts

- All the methods work for simple/smoothly varying problems if:
- One starts from a good initial guess.
- Efficiency is important if the problem has to be solved multiple times.
- Anticipate complex roots.
- Trade off between your time and computer's time...
- Choosing the right method is difficult – something that works in one equation can miserably fail in another.

Some thoughts

- All the methods work for simple/smoothly varying problems if:
- One starts from a good initial guess.
- Efficiency is important if the problem has to be solved multiple times.
- Anticipate complex roots.
- Trade off between your time and computer's time...
- Choosing the right method is difficult – something that works in one equation can miserably fail in another.
- If possible, visualize the data – this helps a lot.