Numerical Methods

Roots of Non-linear equations

Contents

Roots of Non-linear equations

- Roots of polynomials
  - Polynomials
  - roots
- Roots of systems of non-linear equations
  - Fixed-point iteration
  - Newton-Raphson

- Polynomials: the general form

$$f_n(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

or

$$f_n(x) = \sum_{i=0}^{n} a_i x^i$$

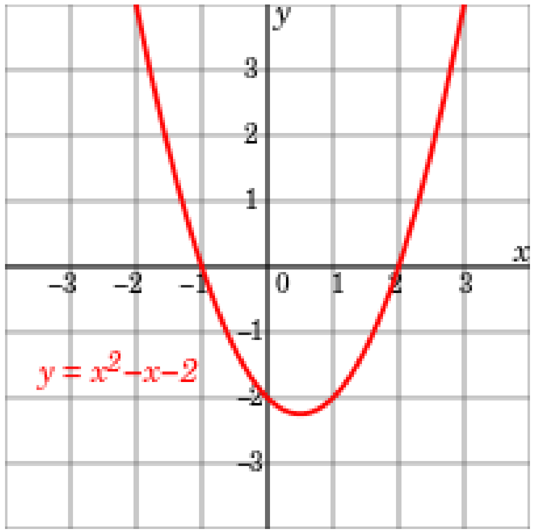
- where *n* is the order of the polynomial, and the *a's* are constant coefficients

- Polynomials are a special type of nonlinear algebraic equation

$$f_n(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0 \quad \text{or} \quad f_n(x) = 0$$

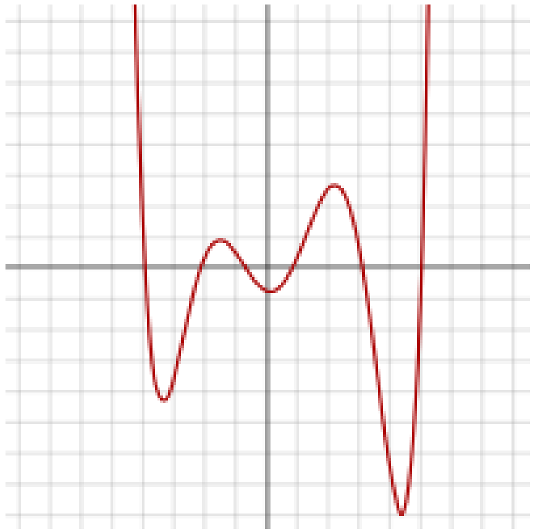
- The solutions of this equation are called the roots of the polynomial, In general, an *nth* order polynomial will have *n* roots
- they are the zeroes of the function f (corresponding to the points where the graph of f meets the x-axis)

- A polynomial function in one real variable can be represented by a graph



The graph of a polynomial function of degree 2

The graph of a polynomial function of degree 3



The graph of a polynomial function of degree 6

Characteristics of polynomials

- The relation between the roots of a polynomial and its coefficients is described by Vieta's formulas

$$f(x) = ax^2 + bx + c$$

Roots x_1, x_2 of the equation $f(x) = 0$ satisfy

$$x_1 + x_2 = -\frac{b}{a}\ ,\ x_1 x_2 = \frac{c}{a}$$

- In general case, Vieta's formulas

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

$$\begin{cases} x_1 + x_2 + \cdots + x_{n-1} + x_n = -\dfrac{a_{n-1}}{a_n} \\ (x_1x_2 + x_1x_3 + \cdots + x_1x_n) + (x_2x_3 + x_2x_4 + \cdots + x_2x_n) + \cdots + x_{n-1}x_n = \dfrac{a_{n-2}}{a_n} \\ \quad \vdots \\ x_1x_2 \ldots x_n = (-1)^n \dfrac{a_0}{a_n}. \end{cases}$$

Companion matrix

In general, the monic polynomial of degree n

$$\boxed{p(x) = x^n + a_1 x^{n-1} + \cdots + a_{n-1}x + a_n}$$

A companion matrix

$$A = \begin{pmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & \\ . & . & \cdots & . & . \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

An n-by-n matrix (square matrix)

Characteristics: x_i is a root of the polynomial $p(x)$

→ x_i is also a eigenvalue of the companion matrix A

Example

- Suppose we have a polynomial

$$a_1x^5 + a_2x^4 + a_3x^3 + a_4x^2 + a_5x + a_6 = 0$$

Dividing by a_1 and rearranging yields

$$x^5 = -\frac{a_2}{a_1}x^4 - \frac{a_3}{a_1}x^3 - \frac{a_4}{a_1}x^2 - \frac{a_5}{a_1}x - \frac{a_6}{a_1}$$

A special matrix

polynomial's companion matrix

$$\begin{bmatrix} -a_2/a_1 & -a_3/a_1 & -a_4/a_1 & -a_5/a_1 & -a_6/a_1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Calculate the eigenvalues?

Eigenvalue

- The eigenvalues of matrix A are values of λ that satisfy the equation

$$|A - \lambda I| = 0$$

- The fundamental theorem of algebra implies that the characteristic polynomial of an *n* by *n* matrix A, being a polynomial of degree *n*, can be factored into the product of *n* linear terms

$$|A - \lambda I| = (\lambda_1 - \lambda)(\lambda_2 - \lambda)\cdots(\lambda_n - \lambda), \quad \Rightarrow \quad \text{Roots}$$

Eigenvalues : characteristic roots, characteristic values, proper values

Case study 1

Find the eigenvalue of a matrix $A = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix}$

Step 1: We need to solve the equation

Step 2: Calculation

Step 3: calculate the determinant

Case study 2

Find the roots of a polynomial

$$f(x) = x^3 - 7x + 6$$

Step 1: the companion matrix

Step 2: The eigenvalues of A

Step 3: The eigenvalues of A are the polynomial roots

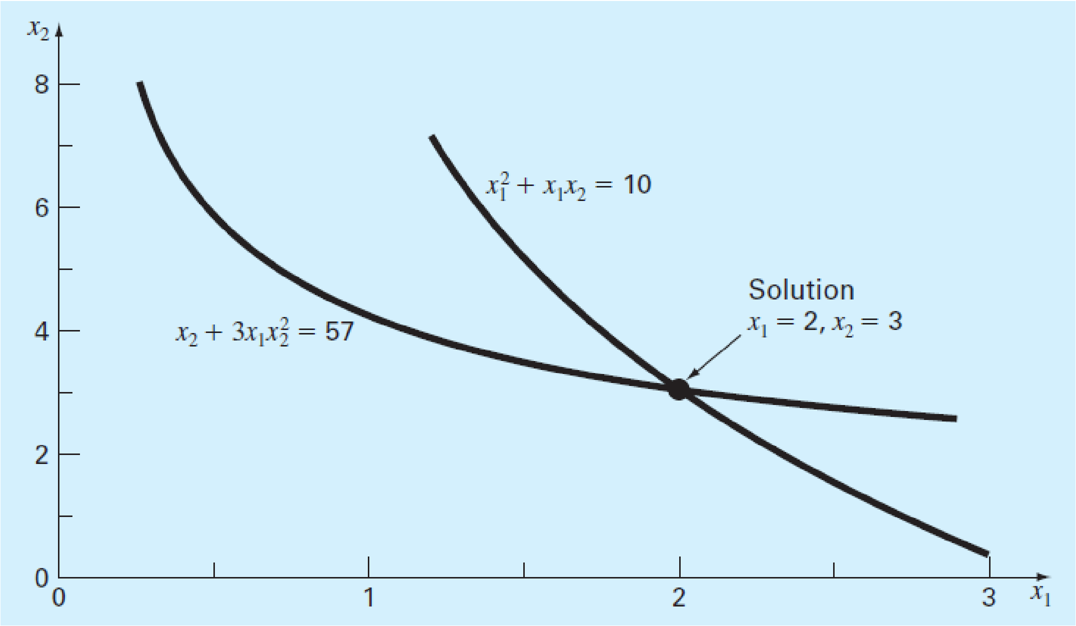
- Roots of polynomials
  - Polynomials
  - roots
- Systems of non-linear equations
  - Fixed-point iteration
  - Newton-Raphson

- Objectives: the primary objective is to find the root of a system of nonlinear equation

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, \ldots, x_n) &= 0 \\
\cdot \qquad\qquad & \quad \cdot \\
\cdot \qquad\qquad & \quad \cdot \\
\cdot \qquad\qquad & \quad \cdot \\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
$$

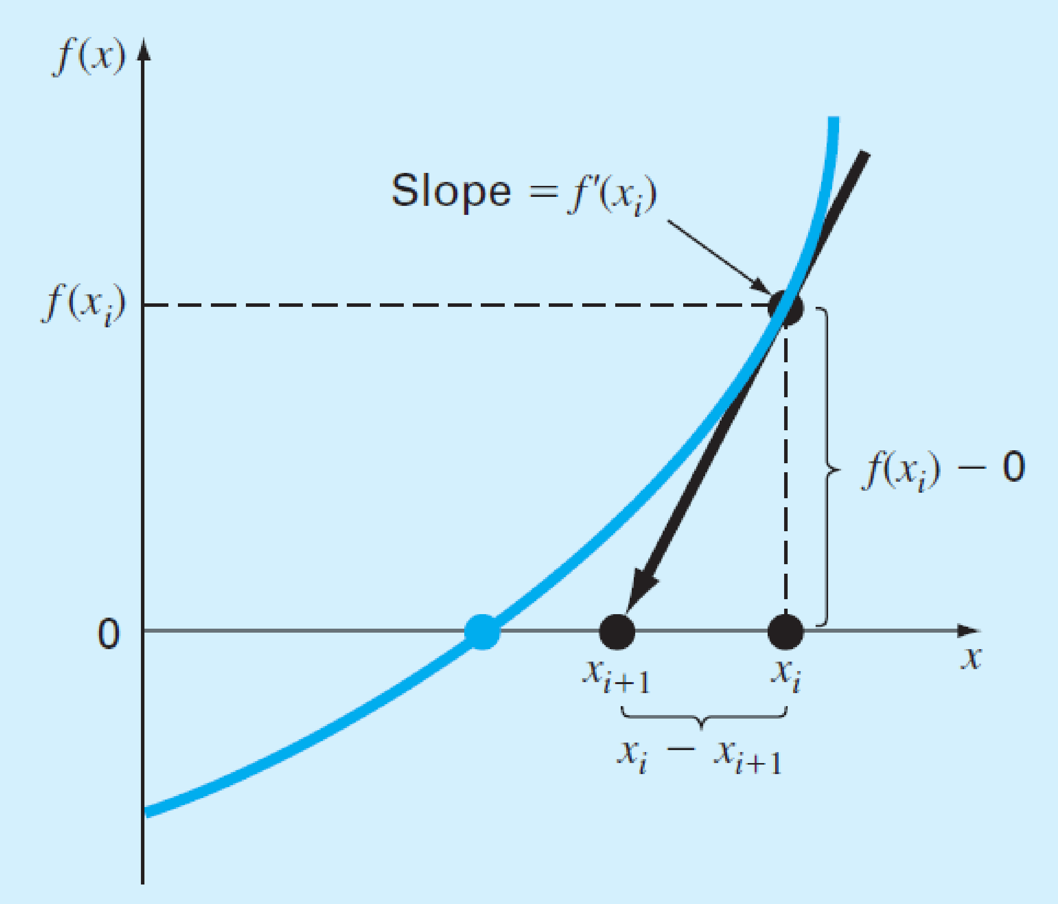
Example

$$
x_1^2 + x_1 x_2 = 10
$$
$$
x_2 + 3x_1 x_2^2 = 57
$$



the solution is the intersection of the curves

Newton-Raphson



$$f(x) = 0$$

the first derivative at x is equivalent to the slope

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

which can be rearranged to yield

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Single-equation forme Newton-Raphson

⇨ Systems of non-linear equations

- Newton-Raphson

the two-equation case

$$\begin{cases} u(x,y)=0 \\ \mathrm{v}(x,y)=0 \end{cases}$$

$$u_{i+1} = u_i + (x_{i+1} - x_i)\frac{\partial u_i}{\partial x} + (y_{i+1} - y_i)\frac{\partial u_i}{\partial y}$$

$$v_{i+1} = v_i + (x_{i+1} - x_i)\frac{\partial v_i}{\partial x} + (y_{i+1} - y_i)\frac{\partial v_i}{\partial y}$$

the two-equation Newton-Raphson

$$x_{i+1} = x_i - \frac{u_i \dfrac{\partial v_i}{\partial y} - v_i \dfrac{\partial u_i}{\partial y}}{\boxed{\dfrac{\partial u_i}{\partial x}\dfrac{\partial v_i}{\partial y} - \dfrac{\partial u_i}{\partial y}\dfrac{\partial v_i}{\partial x}}} \qquad y_{i+1} = y_i - \frac{v_i \dfrac{\partial u_i}{\partial x} - u_i \dfrac{\partial v_i}{\partial x}}{\boxed{\dfrac{\partial u_i}{\partial x}\dfrac{\partial v_i}{\partial y} - \dfrac{\partial u_i}{\partial y}\dfrac{\partial v_i}{\partial x}}}$$

determinant of the *Jacobian* of the system

The two-equation Newton-Raphson approach can be generalized to solve n simultaneous equations

Newton-Raphson

Input: a continuous function u, v

1. Evaluate $u'(x,y)$, $v'(x,y)$ symbolically

2. Use an initial guess of the root x_i, y_i to estimate the new value of the root x_{i+1}, y_{i+1} as

3. Find the relative approximate error

Case study 1

Use the multiple-equation Newton-Raphson method to determine roots of equations

$$u(x, y) = x^2 + xy - 10 = 0$$

$$v(x, y) = y + 3xy^2 - 57 = 0$$

Case study 1

$$u(x, y) = x^2 + xy - 10 = 0$$

$$v(x, y) = y + 3xy^2 - 57 = 0$$

Step 1: Initiate the computation with guesses

Step 2: The first derivatives of the function can be evaluated

Case study 1

$$u(x, y) = x^2 + xy - 10 = 0$$

$$v(x, y) = y + 3xy^2 - 57 = 0$$

Step 3: the determinant of the Jacobian for the first iteration

Step 4: The values of the functions can be evaluated at the initial guesses

Case study 1

$$u(x, y) = x^2 + xy - 10 = 0$$

$$v(x, y) = y + 3xy^2 - 57 = 0$$

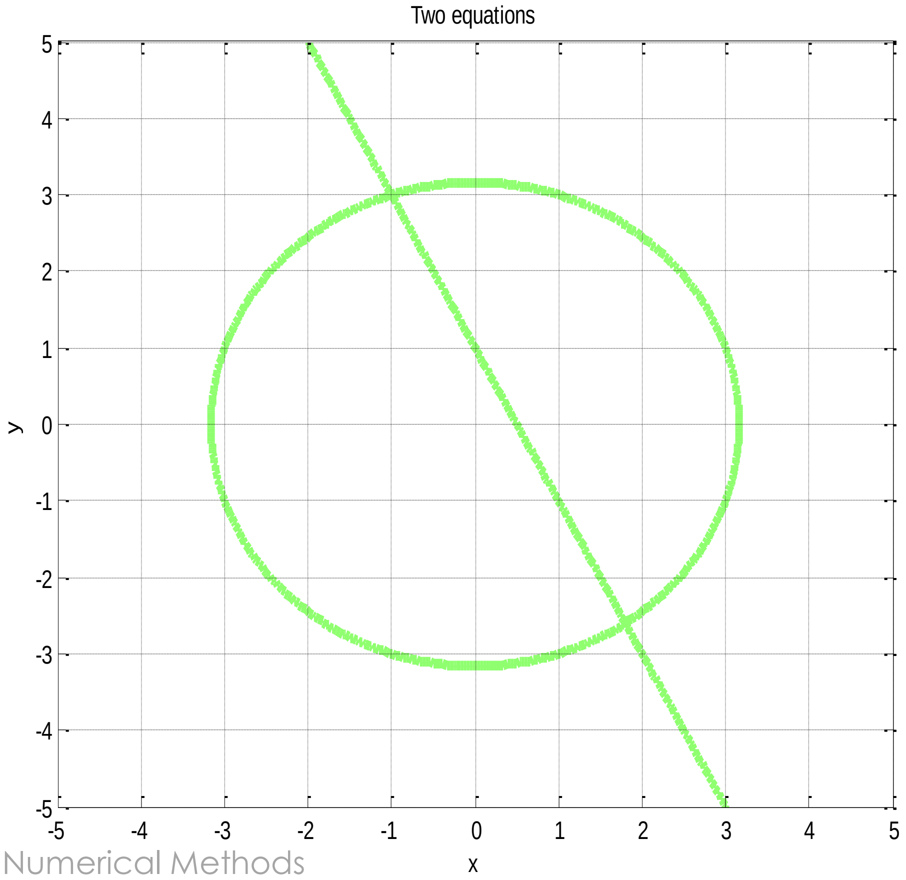
Step 5: These values can be substituted into the two-equation Newton-Raphson

Step 6: the results are converging to the true values of x = 2 and y = 3. The computation can be repeated until an acceptable accuracy is obtained

Case study 2

Use the multiple-equation Newton-Raphson method to determine roots of equations

$$\begin{cases} x^2 + y^2 = 10 \\ 2x + y = 1 \end{cases}$$



Matlab

Exercise

```
%% bisection
f=@(x)(x^3-x-2)
a=1
b=2
[p,i] = bisection(f,a,b)
```

Write the function: bisection(f,a,b) to solve the

- quadratic equation (Energy, Weo, Space)
- cubic equation (ICT and MSN group)

Matlab

```
clc
clear all
close all
%%
```

Matlab

```
clc
clear all
close all
%%
```

```
function [ x ] = bisection(f,x_l,x_u,esp_s)
```

Find the root of a non-linear equation `f=@(x)(x^3-x-2)`

The fzero function is designed to find the real root of a single equation.
A simple representation of its syntax is fzero(function,x_0) or fzero(function,[x_0 x_1])

Step 1: To find a zero of the function $f(x) = x^3 - x - 2$, write an anonymous function

Step 2: Then find the zero near one guess

Case study 1-Matlab

Find the eigenvalue of a matrix $A = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix}$

Step 1: We need to solve the equation

$$\boxed{\det(A - \lambda I) = 0.}$$

Step 2: Calculation

Step 3: calculate the determinant

Case study 2 - Matlab

Find the roots of a polynomial $f(x) = x^3 - 7x + 6$

Step 1: the companion matrix

Step 2: The eigenvalues of A

Step 3: The eigenvalues of A are the polynomial roots

Case study 2 - Matlab

Find the roots of a polynomial $f(x) = x^3 - 7x + 6$

```
%
u = [1  0  -7 6]
n= 3
A = diag(ones(n-1,1),-1);
A(1,:) = -u(2:n+1)/u(1);
x = eig(A)
%
```

A =

0	7	-6
1	0	0
0	1	0

Case study 2 - Matlab

Find the roots of a polynomial

$$f(x) = x^3 - 7x + 6$$

2nd method:

$$c = [1 \quad 0 \quad -7 \quad 6]$$

$$A = compan(c)$$

$$\Rightarrow \quad A = \begin{bmatrix} 0 & 7 & -6 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Case study 2 - Matlab

3rd method:

Step 1: coefficients of polynomial

$$syms\ \ x$$
$$f = x^3 - 7x + 6$$
$$c = \mathrm{sym2poly}(f)$$

Step 2: the companion matrix

$$A = compan(c) \Rightarrow A = \begin{bmatrix} 0 & 7 & -6 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Case study 2 - Matlab

Find the roots of a polynomial $f(x) = x^3 - 7x + 6$

```
clc
clear all
close all
%% CALCULATE THE ROOT OF POLYNOMINALS F = X^3 - 7X + 6
u = [1  0  -7 6]
n = 3
A = diag(ones(n-1,1),-1);
A(1,:)  = -u(2:n+1)/u(1);
x = eig(A)
```

Case study 2 - Matlab

Find the roots of a polynomial $f(x) = x^3 - 7x + 6$

```
clc
clear all
close all
%% CALCULATE THE ROOT OF POLYNOMINALS F = X^3 - 7X + 6

% 2nd method
c = [1  0  -7 6]
A = compan(c)
x = eig(A)
```

Case study 2 - Matlab

Find the roots of a polynomial $f(x) = x^3 - 7x + 6$

```
clc
clear all
close all
%% CALCULATE THE ROOT OF POLYNOMINALS F = X^3 - 7X + 6

% 3rd method
syms x
f = x^3 -7*x + 6
c = sym2poly(f)
A = compan(c)
x = eig(A)
```

Case study 2 - Matlab

Find the roots of a polynomial

$$f(x) = x^3 - 7x + 6$$

```
clc
clear all
close all
%% CALCULATE THE ROOT OF POLYNOMINALS F = X^3 - 7X + 6

% 4th method
syms x
f = x^3 -7*x + 6
c = sym2poly(f)
x = roots(c)

p = poly(x)
```

Case study 3 - Matlab

Find the roots of a system

$$u(x, y) = x^2 + xy - 10 = 0$$

$$v(x, y) = y + 3xy^2 - 57 = 0$$

```
%%   Solver of non linear systems:
x=[1.5  3.5]'; % initial guess
N = 5000;
n = 2;
for k=1:N,
    F=[ x(1)*x(1)+ x(1)*x(2) - 10 ;  x(2) + 3*x(1)*x(2)*x(2)-57];
    J=[2*x(1)+x(2), x(1) ; 3*x(2).^2 , 1 + 6*x(1).*x(2)];
    dx=J\F;
    x=x-dx
    k=k+1
    pause
end
x'
```

Case study 3 - Matlab

Find the roots of a system

$$u(x, y) = x^2 + xy - 10 = 0$$

$$v(x, y) = y + 3xy^2 - 57 = 0$$

```
clc
clear all
close all

fun = @myfun;
x0 = [0,0];
options=optimset('Display','iter');
%Option to display output
[x,fval] = fsolve(fun,x0,options)
```

```
function F = myfun(x)

F(1) = x(1)^2 + x(1)*x(2) - 10;
F(2) = x(2) +3*x(1)*x(2)^2 -57 ;
```