Numerical Methods in Physics and Astrophysics

Kostas Kokkotas

October 28, 2019

Eberhard Karls Univerity of Tübingen

TOPICS

1. Solving nonlinear equations
2. Solving linear systems of equations
3. Interpolation, approximation & curve fitting
4. Numerical differentiation and numerical integration
5. Numerical Solution of ODEs
6. Boundary and Characteristic Value Problems
7. Theoretical Introduction to PDEs
8. Numerical Solution of PDEs
9. Finite Elements Methods

TEXTBOOKS

1. Applied Numerical Analysis C.F. Gerald & P.O.Wheatley, Addison-Wesley 7th Edition (2004)
2. Numerical Recipes. The Art of Scientific Computing W.H. Press, S.A.Teukolsky, & W. T. Vetterling, Cambridge University Press (2007)
3. Numerical Partial Differential Equations J. W. Thomas, Springer (1998)

I. Solving nonlinear equations

A single non-linear equation:

$$0 = 2\sin(x) - e^{-x} - x^2$$

A system of non-linear equations:

$$\begin{aligned} 0 &= e^x - 3y - 1 \\ 0 &= x^2 + y^2 - 4 \end{aligned}$$

- Bisection method
- Linear interpolation
- $x_{n+1} = g(x_n)$
- Newton's method

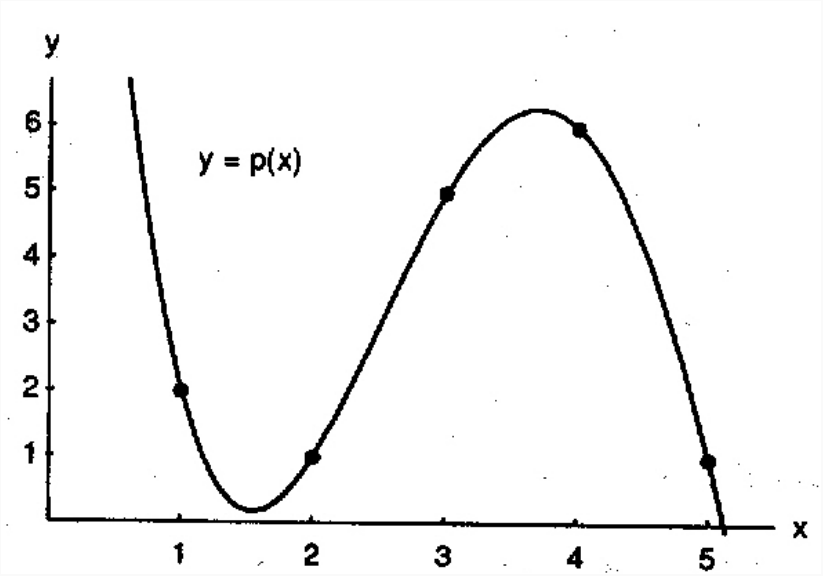
II. Solving systems of linear & non-linear equations

Solve the system: $\mathbf{A} \cdot \mathbf{x} = \mathbf{B}$, where

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ a_{12} & & & a_{2N} \\ \vdots & & & \vdots \\ a_{1N} & \cdots & \cdots & a_{NN} \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$$

- Gaussian elimination
- Iterative methods
- Matrix inverse
- Eigenvalues & Eigenvectors
- Nonlinear systems

III. Interpolation, approximation & curve fitting

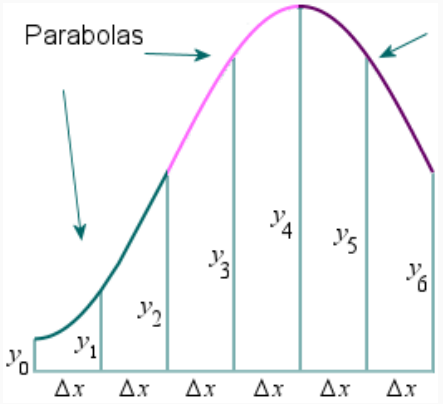


- Interpolating polynomials
- Spline Curves
- Rational function approximations

IV. Numerical Differentiation and Integration

$$I = \int_{x_0}^{x_n} y(x)dx$$

$$\int_{x_0}^{x_n} y(x)dx = \frac{h}{2}\left(y_0 + 2y_1 + 2y_2 + \ldots + 2y_{n-1} + y_n\right) - \frac{b-a}{12}\Delta x^2 y^{(2)}(\xi_1)$$



- Numerical differentiation
- Trapezoidal rule
- Simpson rules
- Gaussian quadrature
- Multiple integrals

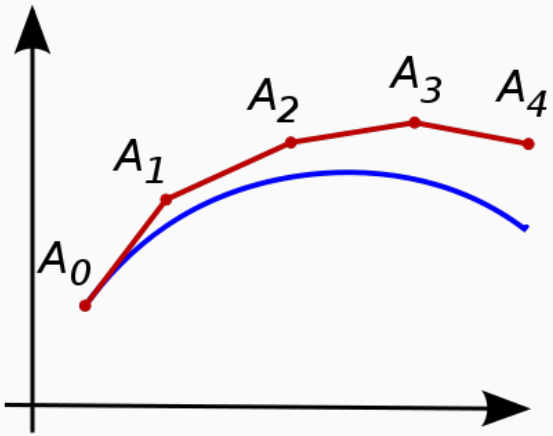
V. Numerical Solution of ODEs

$$y' = f(x, y)$$

with

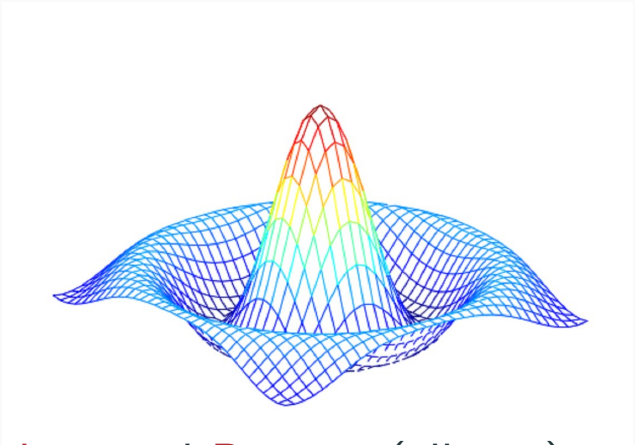
$$y(x_0) = y_0$$

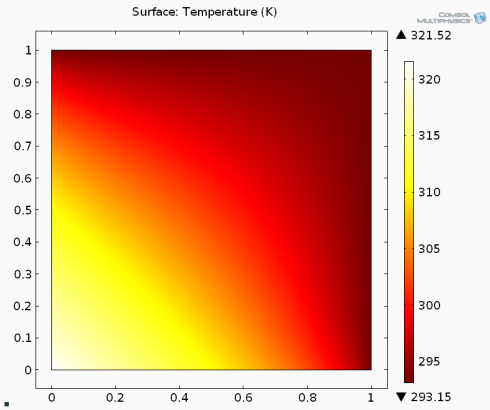
$$y(x) = y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{h^2}{2} y''(x_0) + \cdots$$



- Euler's method
- Runge-Kutta methods
- Adam's method
- Prediction-Correction methods

VI. Numerical Solution of PDEs





- 2-D Laplace and Poisson (elliptic)eqns:

$$\nabla^2 u = 0\,, \quad \nabla^2 u = g(x,y) \quad \text{for} \quad 0 < x < 1 \quad \text{and} \quad 0 < y < 1$$

- The **wave equation**

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0 \quad \text{for} \quad 0 < x < L \quad \text{and} \quad 0 < t < \infty$$

- The **heat equation**

$$\alpha^2 \frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} = 0 \quad \text{for} \quad 0 < x < 1 \quad \text{and} \quad 0 < y < 1$$

Solving nonlinear equations

Solving nonlinear equations

The classical problem is to find a value r such that for a function $f(x)$ where $x \in (a, b)$

$$f(r) = 0 \tag{1}$$

The typical procedure is to find a recurrence relation of the form:

$$x_{n+1} = \sigma(x_n) \tag{2}$$

which will provide a sequence of values $x_0, x_1, \ldots, x_\kappa, \ldots$ and in the limit $\kappa \to \infty$ to lead in a root of equation (1).

For every method we need to address the following questions:

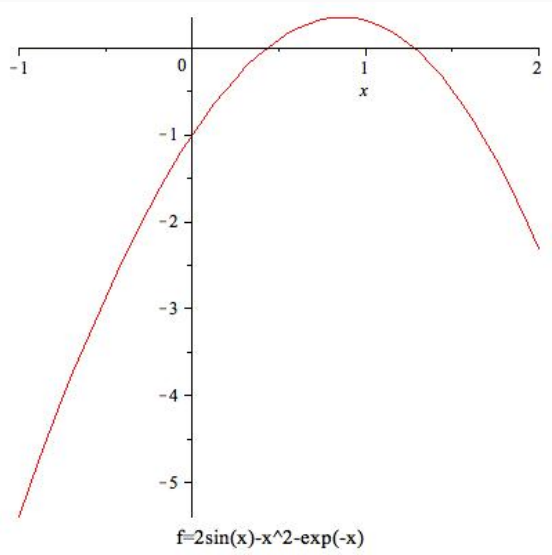
- Under *what conditions* a method will converge.
- If it converges, then *how fast*.
- What is the best choice for the initial guess x_0.

Bisection method (Bolzano) i

Let's assume that we are looking for the roots of the following equation:

$$f(x) = 2\sin(x) - x^2 - e^{-x} = 0 \tag{3}$$

then $x_1 = 0$ and $x_2 = 1$ which give $f(0) = -1$ and $f(1) = 0.31506$



- $x_3 = (x_0 + x_1)/2 = 0.5 \rightarrow f(0.5) = 0.1023$
- $x_4 = (x_0 + x_3)/2 = 0.25 \rightarrow f(0.25) = -0.1732$
- $x_5 = (x_4 + x_3)/2 = 0.375 \rightarrow f(0.375) = -0.0954$
- $x_6 = (x_5 + x_3)/2 = 0.4375 \rightarrow f(0.4375) = 0.0103$
- $x_7 = (x_5 + x_6)/2 = 0.40625 \rightarrow f(0.40625) = -0.0408$
- ...
- $x_n = r_1 \approx 0.4310378790$

the other root is $r_2 = 1.279762546$.

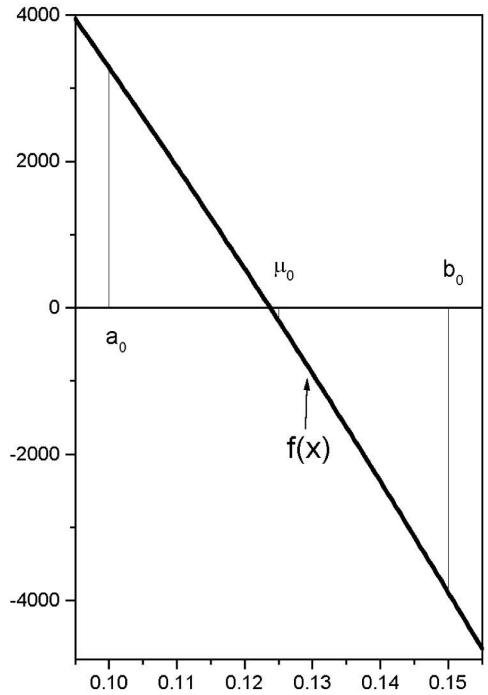
Bisection method (Bolzano) ii

If there exists a root in the interval $[a_0, b_0]$, then $f(a_0) \cdot f(b_0) < 0$. If $\mu_0 = (a_0 + b_0)/2$, then:

- (I) either $f(\mu_0) \cdot f(a_0) < 0$
- (II) either $f(\mu_0) \cdot f(b_0) < 0$
- (III) either $f(\mu_0) = 0$.

If (III), then the root has been found or else we set a new interval

$$[a_1, b_1] = \begin{cases} [\mu_0, b_0] & \text{if} \quad (II) \\ \\ [a_0, \mu_0] & \text{if} \quad (I) \end{cases} \tag{4}$$



Bisection method (Bolzano) iii

```
REPEAT
        SET x3 = (x1 + x2)/2
        IF f(x3) · f(x1) < 0
                SET x2 = x3
        ELSE
                SET x1 = x3
        ENDIF
UNTIL (|x1 − x2| < E) OR f(x3) = 0
```

Table 1: Procedure to find the root of a nonlinear equations using bisection method

Bisection method (Bolzano) iv

ERROR : The error defined as: $\varepsilon_n = |r - x_n|$. For this method

$$\varepsilon_n \leq \frac{1}{2} |a_n - b_n| \tag{5}$$

and in every step the error is half of the previous step

$$\varepsilon_{n+1} = \frac{\varepsilon_n}{2} = \frac{\varepsilon_{n-1}}{2^2} = \cdots = \frac{\varepsilon_0}{2^{n+1}} \tag{6}$$

and if we demand an error $\mathcal{E}$ we can calculate the number of steps needed:

$$n = \log_2 \frac{\varepsilon_0}{\mathcal{E}} \tag{7}$$

CRITISISM

- Slow convergence
- Problem with discontinuities

Linear interpolation i

Assume that the function is linear in the interval (x_1, x_2) where $f(x_1)$ and $f(x_2)$ are of opposite sign. Then there will be a straight line connecting the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ described by the equation:

$$y(x) = f(x_1) + \frac{f(x_1) - f(x_2)}{x_1 - x_2}(x - x_1) \tag{8}$$

which crosses the axis Ox let say in a point x_3 i.e. $f(x_3) = 0$ which will be given by the following relation

$$x_3 = \frac{x_2 f(x_1) - x_1 f(x_2)}{f(x_1) - f(x_2)} = x_2 - \frac{f(x_2)}{f(x_2) - f(x_1)}(x_2 - x_1) \tag{9}$$

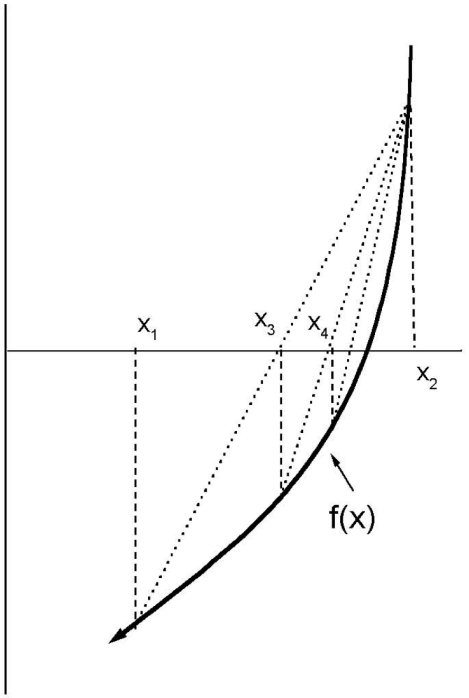
That is we substitute the actual function with a straight line and we assume that the point where this line crosses the axis Ox is a first approximation to the root of the equation.

This new point x_3 will be used to find a better approximation to the actual root r.

Linear interpolation ii

A possible choice of the new points:

- (I) If $f(x_1) \cdot f(x_3) < 0 \Rightarrow x_2 = x_3$
- (II) If $f(x_2) \cdot f(x_3) < 0 \Rightarrow x_1 = x_3$
- (III) If $f(x_3) = 0$



Recurrence relation:

$$x_{n+2} = x_{n+1} - \frac{f(x_{n+1})}{f(x_{n+1}) - f(x_n)} (x_{n+1} - x_n) \, . \qquad (10)$$

Linear interpolation iii

Assume that the root is in the interval (x_1, x_2)

REPEAT

SET $x_3 = x_2 - f(x_2) \cdot \frac{x_2 - x_1}{f(x_2) - f(x_1)}$

IF $f(x_3) \cdot f(x_1) < 0$

SET $x_2 = x_3$

ELSE

SET $x_1 = x_3$

ENDIF

UNTIL $|f(x_3)| < E$

Table 2: 1st algorithm

PROBLEM: Examine after how many iterations you will find the root of equation (3) with an error of the order of $\mathcal{E} \approx 10^{-6}$.

Linear interpolation iv

The root is not included in the initial choice of the interval (x_1, x_2)

REPEAT

SET $x_3 = x_2 - f(x_2) \cdot \frac{x_2 - x_1}{f(x_2) - f(x_1)}$

IF $|f(x_1)| < |f(x_2)|$

SET $x_2 = x_3$

ELSE

SET $x_1 = x_3$

ENDIF

UNTIL $|f(x_3)| < E$

Table 3: 1st algorithm

Convergence

If r is a root of the equation $f(x) = 0$ and $\varepsilon_n = |r - x_n|$ is the error for $x = x_n$, the the convergence of the method of linear interpolation is

$$\varepsilon_{n+1} = k \cdot \varepsilon_n^{1.618} \tag{11}$$

PROOF : If we assume that $x_n = r + \varepsilon_n$ and

$$f(x_n) = f(r + \varepsilon_n) = f(r) + \varepsilon_n f'(r) + \frac{\varepsilon_n^2}{2} f''(r) \tag{12}$$

then the relation

$$x_{n+2} = x_{n+1} - \frac{f(x_{n+1})}{f(x_{n+1}) - f(x_n)} (x_{n+1} - x_n) \tag{13}$$

becomes

$$r + \varepsilon_{n+2} = r + \varepsilon_{n+1} - \frac{\varepsilon_{n+1} f'(r) + \varepsilon_{n+1}^2 f''(r)/2}{f'(r)(\varepsilon_{n+1} - \varepsilon_n) + \frac{1}{2} f''(r) \left(\varepsilon_{n+1}^2 - \varepsilon_n^2\right)} \cdot (\varepsilon_{n+1} - \varepsilon_n)$$

Linear interpolation vi

which leads to:

$$\varepsilon_{n+2} = \varepsilon_{n+1}\left[1 - \left(1 + \frac{\varepsilon_n}{2}\frac{f''(r)}{f'(r)}\right)\right] = -\varepsilon_{n+1}\varepsilon_n \frac{f''(r)}{2f'(r)}$$

but we are aiming for a relation of the form $\varepsilon_{n+1} = k \cdot \varepsilon_n^m$

$$\left.\begin{aligned}\varepsilon_{n+1} &= k \cdot \varepsilon_n^m \\ \varepsilon_{n+2} &= k \cdot \varepsilon_{n+1}^m\end{aligned}\right\} \Rightarrow k \cdot \varepsilon_{n+1}^m = \varepsilon_{n+1}\varepsilon_n\left(-\frac{f''(r)}{2f'(r)}\right)$$

$$\varepsilon_{n+1} = \left(\frac{1}{k}\right)^{\frac{1}{m-1}} \varepsilon_n^{\frac{1}{m-1}} A^{\frac{1}{m-1}} \quad \text{where} \quad A = -\frac{f''(r)}{2f'(r)} \tag{14}$$

$$k \cdot \varepsilon_n^m = \left(\frac{A}{k}\right)^{\frac{1}{m-1}} \varepsilon_n^{\frac{1}{m-1}} \quad \Bigg| \quad \Rightarrow \quad \begin{aligned} k &= \left(\tfrac{A}{k}\right)^{1/(m-1)} \\ m &= \tfrac{1}{m-1} \Rightarrow m^2 - m - 1 = 0 \end{aligned}$$

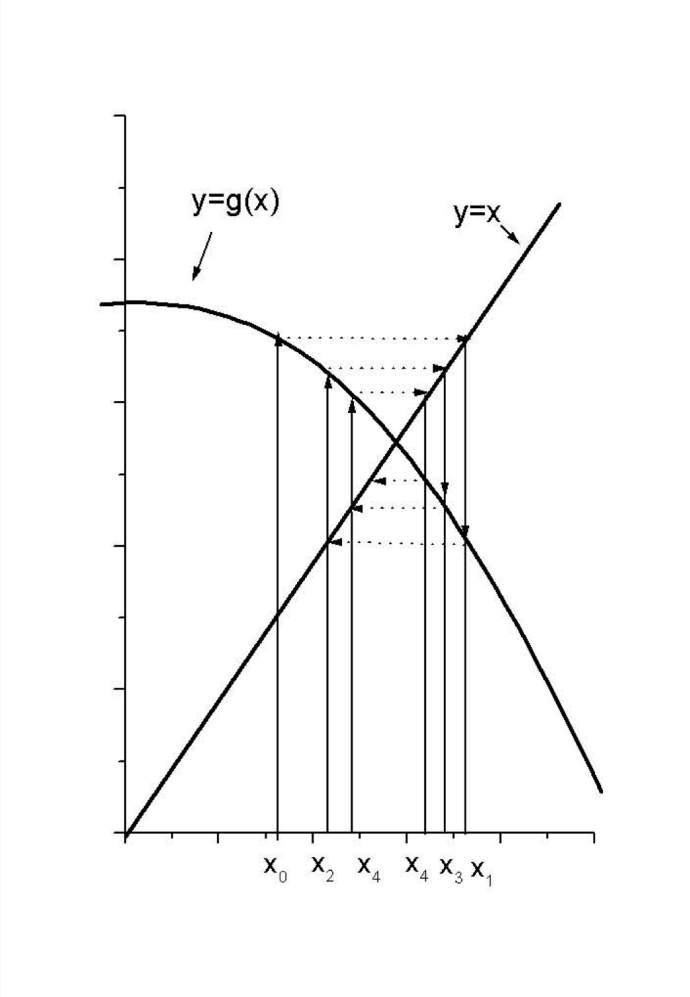
$$m = \frac{1 \pm \sqrt{5}}{2} = 1.618 \quad \text{and} \quad k^m = A = -\frac{f''(r)}{2f'(r)}$$

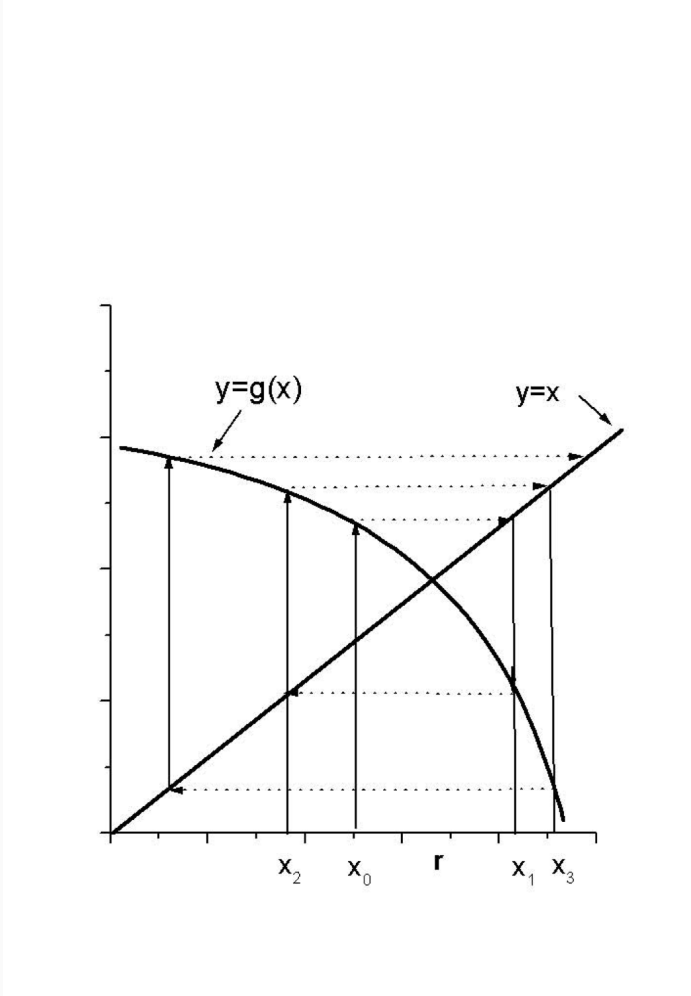
$$\varepsilon_{n+1} = k \cdot \varepsilon_n^{1.618} \tag{15}$$

Method $x_{n+1} = g(x_n)$ i

Write the nonlinear equation in a form $x_{n+1} = g(x_n)$ such as

$\lim_{n\to\infty} x_n = r \quad (n = 1, 2, 3, \dots)$





Method $x_{n+1} = g(x_n)$ ii

THEOREM: If $g(x)$ and $g'(x)$ are continuous on an interval about a root r of the equation $x = g(x)$, and if $|g'(x)| < 1$ for all x in the interval, then the recurrence $x_{n+1} = g(x_n)$ $(n = 1, 2, 3, \dots)$ will converge to the root $x = r$, provided that x_1 is chosen in the interval.

- Note that this is a sufficient condition only, since for some equations convergence is secured even though not all the conditions hold.
- In a computer program it is worth checking whether $|x_3 - x_2| < |x_2 - x_1|$

ERROR After n steps the error will be $\varepsilon_n = r - x_n$

$$\varepsilon_{n+1} = r - x_{n+1} = g(r) - g(x_n) = g'(r)(r - x_n) = g'(r)\,\varepsilon_n$$

I.e. linear convergence.

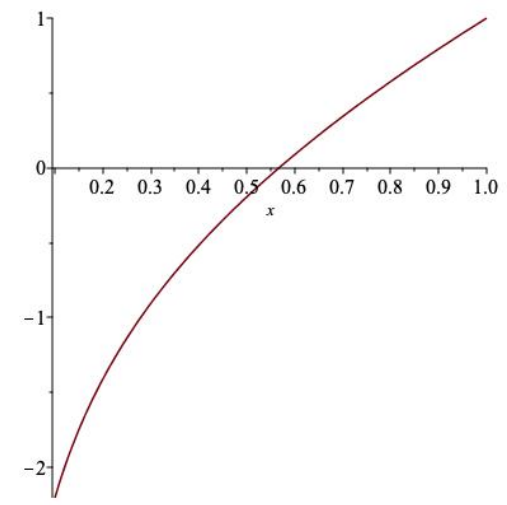
The rate of convergence is rapid if $|g'(x)|$ is small in the interval. If the derivative is negative the errors alternate in sign giving oscillatory convergence.

Method $x_{n+1} = g(x_n)$ iii

Example

Find the root of equation $f(x) = x + \ln(x)$ in the interval $[0.1, 1]$ (r=0.567143).

We will try various forms of rewriting the equation:



- $x_{n+1} = -\ln(x_n)$ but $|g'(x)| = \left|\frac{1}{x}\right| \geq 1$ on $[0.1, 1]$: no convergence.
- $x_{n+1} = e^{-x_n}$ then $|g'(x)| = \left|e^{-x}\right| \leq e^{-0.1} \approx 0.9 < 1$: convergence.
- another writing : $x_{n+1} = \frac{x_n + e^{-x_n}}{2}$ then $|g'(x)| = \frac{1}{2}|1 - e^{-x}| \leq \frac{1}{2}|1 - e^{-1}| = 0.316$: better convergence.
- finally $x_{n+1} = \frac{x_n + 2e^{-x_n}}{3}$ thus $|g'(x)| = \frac{1}{3}|1 - 2e^{-x}| \leq \frac{1}{3}|1 - 2e^{-1}| = 0.03$: even better convergence.

Method $x_{n+1} = g(x_n)$ iv

Thus the obvious choice is:

$$x_{n+1} = \frac{x_n + 2e^{-x_n}}{3} \ .$$

Results: $x_0 = 1$, $x_1 = 0.578586$, $x_2 = 0.566656$, $x_3 = 0.567165$, $x_4 = 0.567142$, $x_5 = 0.567143$

Aitken's improvement

If a method is converging linearly (e.g. $\varepsilon_{n+1} = g'(r)\varepsilon_n$) then it can be "improved" to produce even more accurate results without extra iterations. The error the method $x_{n+1} = g'(x_n)$ after n iterations is:

$$r - x_{n+1} \approx g'(r)(r - x_n)$$

after $n + 1$ is:

$$r - x_{n+2} \approx g'(r)(r - x_{n+1})$$

and by division we get

$$\frac{r - x_{n+1}}{r - x_{n+2}} = \frac{g'(r)(r - x_n)}{g'(r)(r - x_{n+1})}$$

Then by solving for r we get:

$$r = x_n - \frac{(x_n - x_{n-1})^2}{x_n - 2x_{n-1} + x_{n-2}} \tag{16}$$

This relation improves considerably the accuracy of the final result, especially in the case of slow convergence.

Newton's method i

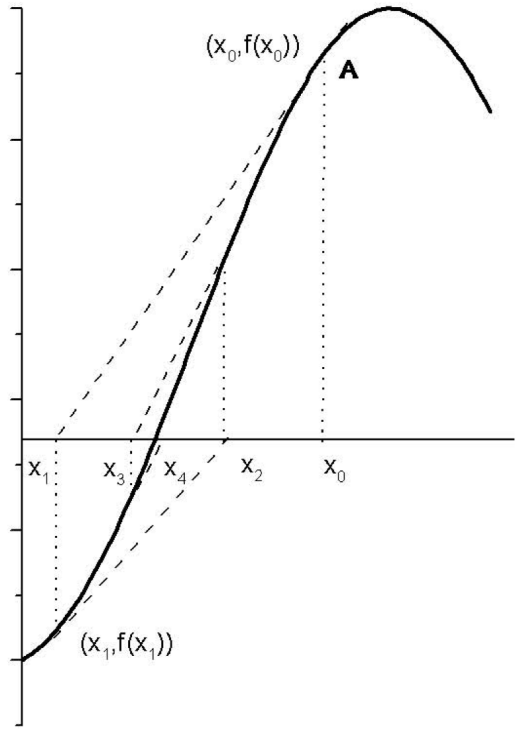
If near the root r of an equation $f(x) = 0$ the 1st and 2nd derivatives of $f(x)$ are continuous we can develop root finding techniques which converge faster than any method presented till now.

Let's assume that after n iterations we reached a value x_{n+1} which is the root of the equation and x_n is a nearby value such as: $x_{n+1} = x_n + \varepsilon_n$.
Then:

$$\begin{aligned} f(x_{n+1}) &= f(x_n + \varepsilon_n) \\ &= f(x_n) + \varepsilon_n f'(x_n) + \frac{\varepsilon_n^2}{2} f''(x_n) + \cdots \end{aligned}$$

but since $f(x_{n+1}) = 0 \rightarrow 0 = f(x_n) + \varepsilon_n f'(x_n)$ i.e.

$$\varepsilon_n = -\frac{f(x_n)}{f'(x_n)} \quad \rightarrow \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (17)$$



.

Convergence I

If r is a root of $f(x) = 0$. Then $x_n = r + \varepsilon_n$ and $x_{n+1} = r + \varepsilon_{n+1}$, thus:

$$r + \varepsilon_{n+1} = r + \varepsilon_n - \frac{f(r + \varepsilon_n)}{f'(r + \varepsilon_n)} = r + \varepsilon_n - \frac{f(r) + \varepsilon_n f'(r) + \frac{1}{2}\varepsilon_n^2 f''(r)}{f'(r) + \varepsilon_n f''(r)}$$

but since $f(r) = 0$ and

$$\frac{1}{1 + \varepsilon f''(r)/f'(r)} \approx 1 - \varepsilon_n \frac{f''(r)}{f'(r)}$$

we get the following relation:

$$\varepsilon_{n+1} = \frac{f''(r)}{2f'(r)} \cdot \varepsilon_n^2 \tag{18}$$

Notice that the convergence of the method is quadratic

.

Convergence II

An alternative way for studying the convergence of Newton's method (17) is based on the method $x_{n+1} = g(x_n)$ i.e. successive iterations will converge if $|g'(x)| < 1$. Here:

$$g(x) = x - \frac{f(x)}{f'(x)} \Rightarrow g'(x) = \frac{f(x)f''(x)}{[f'(x)]^2} \tag{19}$$

since $\epsilon_n = x_n - r = g(x_n) - g(r)$ we get

$$g(x_n) = g(r + \epsilon_n) = g(r) + \cancel{g'(r)\epsilon_n} + \frac{g''(\xi)}{2}\epsilon_n^2 \tag{20}$$

$$= g(r) + \frac{g''(\xi)}{2}\epsilon_n^2, \quad \xi \in [r, x_n] \tag{21}$$

This leads to:

$$g(x_n) - g(r) = \frac{g''(\xi)}{2}\epsilon_n^2 \quad \Rightarrow \quad x_{n+1} - r \equiv \epsilon_{n+1} = \frac{g''(\xi)}{2}\epsilon_n^2 \tag{22}$$

Newton's method iv

Algorithm

COMPUTE $f(x_1)$, $f'(x_1)$
SET $x_2 = x_1$
IF $(f(x_1) \neq 0)$ END $(f'(x_1) \neq 0)$
REPEAT
SET $x_2 = x_1$
SET $x_1 = x_1 - f(x_1)/f'(x_1)$
UNTIL $(|x_1 - x_2| < E)$ OR $(|f(x_2)| < E')$
ENDIF

EXAMPLE : Find the square root of a number a.
We will solve the equation $f(x) = x^2 - a$, obviously $f'(x) = 2x$. Then

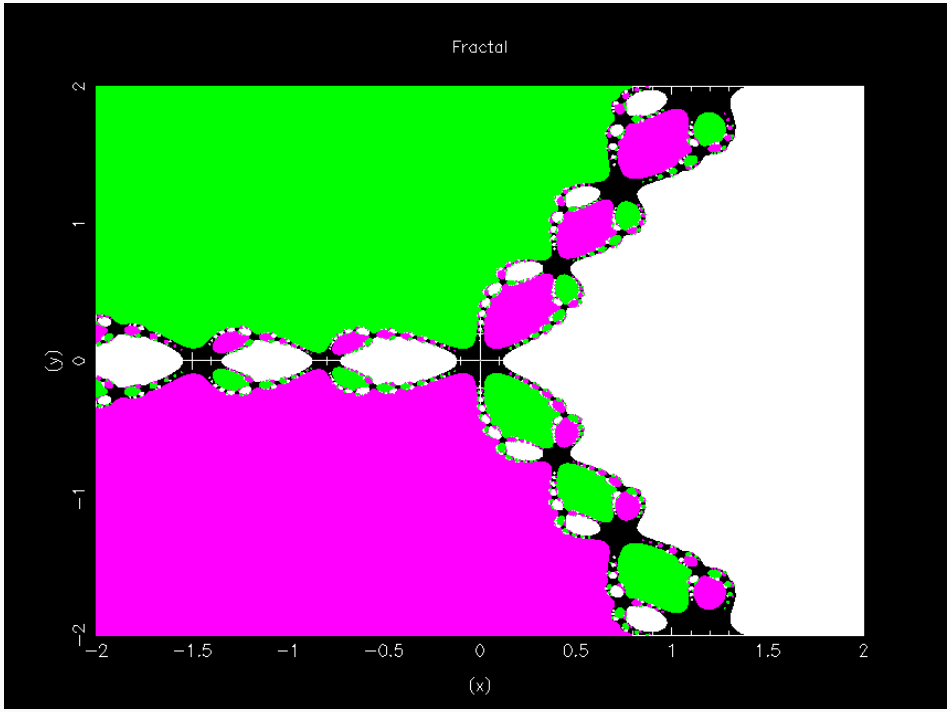
$$x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n} \quad \text{or in a better form} \quad x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right). \tag{23}$$

For $a = 9$ and $x_0 = 4.5$, $x_1 = 3.25$, $x_2 = 3.009615384$, $x_3 = 3.000015360$, $x_4 = 3.00000000004$

EXAMPLE: Fractals

The complex roots $1, (-0.5, \pm 0.866)$ of equation $z^3 - 1 = 0$ can be a very good example of fractal structure.
The root finding algorithm (Newton) is:

$$z_{n+1} = z_n - \frac{z_n^3 - 1}{3z_n^2} \qquad (24)$$



Newton's method : Halley i

Remember that $\varepsilon_n = x_{n+1} - x_n$, then

$$f(x_{n+1}) = f(x_n + \varepsilon_n) = f(x_n) + \varepsilon_n f'(x_n) + \frac{\varepsilon_n^2}{2} f''(x_n) + \dots$$

$$f(x_n) + \varepsilon_n \left[f'(x_n) + \frac{\varepsilon_n}{2} f''(x_n) \right] = 0$$

$$\varepsilon_n = -\frac{f(x_n)}{f'(x_n) + \frac{\varepsilon_n}{2} f''(x_n)}$$

from the 1st order result we substitute

$$\varepsilon_n \approx -\frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) - \frac{f''(x_n) \cdot f(x_n)}{2f'(x_n)}} = x_n - \frac{2f(x_n) f'(x_n)}{2f'^2(x_n) - f''(x_n)\, f(x_n)} \tag{25}$$

obviously for $f''(x_n) \to 0$ we get the Newton-Raphson method.

Convergence

Halley's methods achieves cubic convergence:

$$\varepsilon_{n+1} = -\left[\frac{1}{6}\frac{f'''(\xi)}{f'(\xi)} - \frac{1}{4}\left(\frac{f''(\xi)}{f'(\xi)}\right)^2\right]\cdot\varepsilon_n^3 \tag{26}$$

EXAMPLE: The square root of a number Q (here $Q = 9$) with Halley's method is given by (compare with eq. (23):

$$x_{n+1} = \frac{x_n^3 + 3x_n Q}{3x_n^2 + Q} \tag{27}$$

Newton	Error	Halley	Error
x_0=15	$\varepsilon_0 = 12$	x_0=15	$\varepsilon_0 = 12$
x_1=7.8	$\varepsilon_1 = 4.8$	x_1=5.526	$\varepsilon_1 = 2.5$
x_2=4.477	$\varepsilon_2 = 1.477$	x_2=3.16024	$\varepsilon_2 = 0.16$
x_3=3.2436	$\varepsilon_3 = 0.243$	x_3=3.00011	$\varepsilon_3 = 1.05 \times 10^{-4}$
x_4=3.0092	$\varepsilon_4 = 9.15 \times 10^{-3}$	x_4=3.0000000...	$\varepsilon_4 = 3.24 \times 10^{-14}$

Newton's method : Multipole roots i

If $f(x) = (x-r)^m q(x)$ where m is the multiplicity of root r then

$$f'(x) = (x-r)^{m-1}\left[mq(x) + (x-r)q'(x)\right]$$

i.e. both $f(r) = 0$ and $f'(r) = 0$. Thus the ratio $[f(x)/f'(x)]_{x\to r}$ will diverge. To avoid the problem we construct a new function

$$\phi(x) = \frac{f(x)}{f'(x)} = \frac{(x-r)q(x)}{mq(x) + (x-r)q'(x)}$$

which obviously has r as root with multiplicity $m = 1$ and the recurrence relation is:

$$\begin{aligned} x_{n+1} &= x_n - \frac{\phi(x_n)}{\phi'(x_n)} = x_n - \frac{f(x_n)/f'(x_n)}{\{[f'(x_n)]^2 - f(x_n)f''(x_n)\}/[f'(x_n)]^2} \\ &= x_n - \frac{f(x_n)\,f'(x_n)}{[f'(x_n)]^2 - f(x_n)f''(x_n)} \end{aligned} \tag{28}$$

The convergence is quadratic since we applied Newton-Raphson method for finding the roots of $\phi(x) = 0$ for which r is a simple root.

EXAMPLE

Find the multipole root of $f(x) = x^4 - 4x^2 + 4 = 0$ ($r = \sqrt{2} = 1.414213...$).

Standard Newton-Raphson

$$x_{n+1} = x_n - \frac{x_n^2 - 2}{4x_n} \qquad (A)$$

Modified Newton-Raphson

$$x_{n+1} = x_n - \frac{(x_n^2 - 2)x_n}{x_n^2 + 2} \qquad (B).$$

x	(A)	Error (A)	(B)	Error (B)
x_0	1.5	8.6×10^{-2}	1.5	8.6×10^{-2}
x_1	1.458333333	4.4×10^{-2}	1.411764706	-2.4×10^{-3}
x_2	1.436607143	2.2×10^{-2}	1.414211439	-2.1×10^{-6}
x_3	1.425497619	1.1×10^{-2}	1.414213562	-1.6×10^{-12}

Convergence tests i

We will compare the convergence rate of a method with linear and one with quadratic convergence (Bisection and Newton)
For **linear convergence we** have::

$$\lim_{n\to\infty} \frac{|\varepsilon_{n+1}|}{|\varepsilon_n|} = a \Rightarrow |\varepsilon_n| \approx a|\varepsilon_{n-1}| \approx a^2|\varepsilon_{n-2}| \approx \ldots \approx a^n|\varepsilon_0|$$

then by solving the above equation for n we come to a relation which gives the number of iterations, n, needed to achieve a given accuracy $|\varepsilon_n|$:

$$n \approx \frac{\log_{10}|\varepsilon_n| - \log_{10}|\varepsilon_0|}{\log_{10}|a|} \tag{29}$$

In the same way for **quadratic convergence** we get:

$$\lim_{n\to\infty} \frac{|\tilde{\varepsilon}_{n+1}|}{|\tilde{\varepsilon}_n|^2} = b \Rightarrow |\tilde{\varepsilon}_n| \approx b|\tilde{\varepsilon}_{n-1}|^2 \approx b^3|\tilde{\varepsilon}_{n-2}|^4 \approx \ldots \approx b^{2^{n+1}-1}|\tilde{\varepsilon}_0|^{2^{n+1}}$$

Convergence tests ii

and by solving the above relation for n in order to achieve a given accuracy $|\varepsilon_n|$ we get:

$$2^{n+1} \approx \frac{\log_{10}|\tilde{\varepsilon}_n| + \log_{10}|b|}{\log_{10}|\tilde{\varepsilon}_0| + \log_{10}|b|} \tag{30}$$

If for a given problem we ask that the error after n iterations to be smaller than 10^{-6} i.e. $\varepsilon_n \leq 10^{-6}$ with an initial error $\varepsilon_0 = 0.5$ and for $a = b = 0.7$ the relation (29) gives:

$$n \approx \frac{\log_{10}|10^{-6}| - \log_{10}|0.5|}{\log_{10}|0.7|} \approx 37 \quad \text{iterations}$$

The same assumptions for quadratic convergence will give:

$$2^{n+1} \approx \frac{\log_{10}|10^{-6}| + \log_{10}|0.7|}{\log_{10}|0.5| + \log_{10}|0.7|} \approx 13.5$$

i.e. the number of iterations is only 3!

The difference becomes more pronounce if we demand even higher accuracy e.g. $\varepsilon_n \leq 10^{-14}$ then bisection will need 89 iterations while Newton's method will need only 4!

Selected problems I

1. Find the eigenvalues λ of the differential equation $y'' + \lambda^2 y = 0$ with the following boundary conditions $y(0) = 0$ and $y(1) = y'(1)$.
2. Find the inverse of a number without using division
3. If a is a given number find the value of $1/\sqrt{a}$.
4. Find the root of equation $e^x - \sin(x) = 0$ ($r = -3.183063012$). If the initial interval is $[-4, -3]$ estimate the number of iteration needed to reach an accuracy of 4 digits i.e. $|x_n - r| < 0.00005$.
5. Repeat the previous estimation for using the methods of linear interpolation and
6. Find a few of the roots of $F = \sin^3(2x)\left(1 - 2x^2\right)e^{-2x^2}$.

Solving nonlinear equations

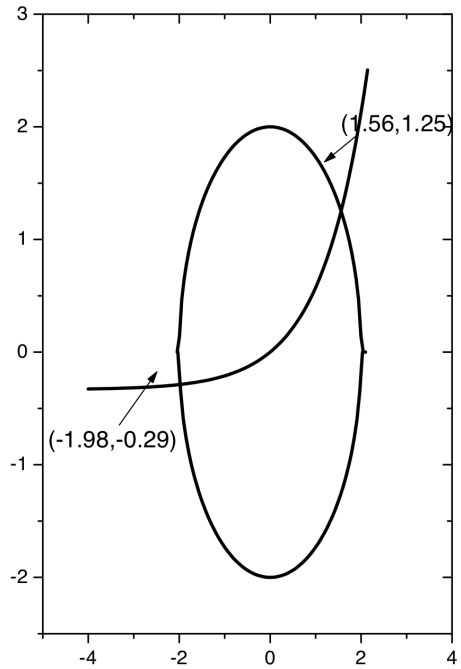
Non-linear systems of equations

We will present two methods based on the techniques of the previous section in solving systems of non-linear equations. The two methods are based on Newton's method and in the $x_{n+1} = g(x_n)$ method.

An example of two non-linear equations is the following:

$$\begin{aligned} f(x,y) &= e^x - 3y - 1 \\ g(x,y) &= x^2 + y^2 - 4 \end{aligned} \tag{31}$$

it is obvious that $f(x,y) = 0$ and $g(x,y) = 0$ are two curves on the xy plane as in the Figure on the right.



Newton's method for 2 equations i

We will develop the method for a system of two non-linear equations while the extension to n nonlinear equations will become obvious.

Let's assume:

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

Let's assume that after $n+1$ iteration the method converged to the solution (x_{n+1},y_{n+1}) i.e. $f(x_{n+1},y_{n+1})\approx 0$ and $g(x_{n+1},y_{n+1})\approx 0$. Then by assuming that $x_{n+1}=x_n+\varepsilon_n$ and $y_{n+1}=y_n+\delta_n$ we can Taylor expand around the solution i.e.

$$0\approx f(x_{n+1},y_{n+1})=f(x_n+\varepsilon_n,y_n+\delta_n)\approx f(x_n,y_n)+\varepsilon_n\left(\frac{\partial f}{\partial x}\right)_n+\delta_n\left(\frac{\partial f}{\partial y}\right)_n$$

$$0\approx g(x_{n+1},y_{n+1})=g(x_n+\varepsilon_n,y_n+\delta_n)\approx g(x_n,y_n)+\varepsilon_n\left(\frac{\partial g}{\partial x}\right)_n+\delta_n\left(\frac{\partial g}{\partial y}\right)_n$$

Newton's method for 2 equations ii

And by solving the system with respect to ε_n and δ_n we get :

$$\varepsilon_n = \frac{-f\frac{\partial g}{\partial y} + g\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial g}{\partial x}\frac{\partial f}{\partial y}} \quad \text{and} \quad \delta_n = \frac{-g\frac{\partial f}{\partial x} + f\frac{\partial g}{\partial x}}{\frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial g}{\partial x}\frac{\partial f}{\partial y}} \tag{32}$$

Since $x_{n+1} = x_n + \varepsilon_n$ and $y_{n+1} = y_n + \delta_n$ we come to the following pair of recurrence relations, which are generalizations of Newton's method:

$$x_{n+1} = x_n - \left(\frac{f \cdot g_y - g \cdot f_y}{f_x \cdot g_y - g_x \cdot f_y}\right)_n \tag{33}$$

$$y_{n+1} = y_n - \left(\frac{g \cdot f_x - f \cdot g_x}{f_x \cdot g_y - g_x \cdot f_y}\right)_n \tag{34}$$

Newton's method n non-linear equations i

If we assume the following system of N equations

$$\begin{aligned} f_1(x^1, x^2, ..., x^N) &= 0 \\ &\vdots \\ f_n(x^1, x^2, ..., x^N) &= 0 \end{aligned}$$

with N unknowns $(x^1, x^2, ..., x^N)$. Then by considering a solution of the system $(x^1_{n+1}, x^2_{n+1}, ..., x^N_{n+1})$ for which we assume the following relations

$$\begin{aligned} x^1_{n+1} &= x^1_n + \Delta x^1_n \\ &\vdots \\ x^N_{n+1} &= x^N_n + \Delta x^N_n \end{aligned}$$

Newton's method n non-linear equations ii

We can create expansion of the form:

$$\begin{aligned} 0 \cong f_1(x^1_{n+1}, ..., x^N_{n+1}) &= f_1(x^1_n + \Delta x^1_n, ..., x^N_n + \Delta x^N_n) \\ &\approx f_1 + \frac{\partial f_1}{\partial x^1}\Delta x^1_n + ... + \frac{\partial f_1}{\partial x^N}\Delta x^N_n \\ &\vdots \\ 0 \cong f_N(x^1_{n+1}, ..., x^N_{n+1}) &= f_N(x^1_n + \Delta x^1_n, ..., x^N_n + \Delta x^N_n) \\ &\approx f_N + \frac{\partial f_N}{\partial x^1}\Delta x^1_n + ... + \frac{\partial f_N}{\partial x^N}\Delta x^N_n \end{aligned} \tag{35}$$

Then the quantities Δx^i_n will be found as solutions of the linear system

$$\begin{pmatrix} \frac{\partial f_1}{\partial x^1} & \frac{\partial f_1}{\partial x^2} & \cdots & \frac{\partial f_1}{\partial x^N} \\ \vdots & & & \vdots \\ \frac{\partial f_N}{\partial x^1} & \frac{\partial f_N}{\partial x^2} & \cdots & \frac{\partial f_N}{\partial x^N} \end{pmatrix} \cdot \begin{pmatrix} \Delta x^1_n \\ \vdots \\ \Delta x^N_n \end{pmatrix} = - \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} \tag{36}$$

Newton's method n non-linear equations iii

Thus if we start with N initial values $(x_0^1, x_0^2, ..., x_0^N)$ then from the solution of the above system we will calculate the quantities Δx_n^i which lead to the "new" values $(x_1^1, x_1^2, ..., x_1^N)$ via the relations:

$$\begin{aligned} x_1^1 &= x_0^1 + \Delta x_0^1 \\ &\vdots \\ x_1^N &= x_0^N + \Delta x_0^N \end{aligned} \tag{37}$$

This procedure will be repeated till the a given accuracy is reached i.e. $\max |\Delta x_n^i| < E$.

Methods of type $x = g(x)$ i

Let's assume the system on N nonlinear equations:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_N) &= 0 \\ &\vdots \quad \vdots \\ f_N(x_1, x_2, \ldots, x_N) &= 0 \end{aligned} \tag{38}$$

The system can be rewritten in the form:

$$\begin{aligned} x_1 &= F_1(x_1, x_2, \ldots, x_N) \\ &\vdots \\ x_N &= F_N(x_1, x_2, \ldots, x_N) \end{aligned} \tag{39}$$

A SIMPLE CHOICE: Give some N initial values $(x_1, \ldots, x_N)^{(0)}$ in the right hand side of the equations and find a set on N new values $(x_1, \ldots, x_N)^{(1)}$. Repeat the procedure till you reach a solution with a given accuracy.

Methods of type $x = g(x)$ ii

AN IMPROVED APPROACH:

Give some N initial values $(x_1, \ldots, x_N)^{(0)}$ in the right hand side of the 1st of the equations (39) and find a new value $(x_1)^{(1)}$.

Then in the right hand side of the 2nd equations apply the following set of "guess" values $(x_1)^{(1)}, (x_2, \ldots, x_N)^{(0)}$ and estimate $(x_2)^{(1)}$. Continue in the third equation by applying the improved set of values : $(x_1, x_2)^{(1)}$, $(x_3, \ldots, x_N)^{(0)}$ and so on.

CONVERGENCE: The system $x_i = F_i(x_1, x_2, \ldots, x_N)$ with $i = 1, \ldots, N$ will converge to a solution, if near this solution the following criterion is satisfied:

$$\begin{aligned} \left|\frac{\partial F_1}{\partial x_1}\right| + \left|\frac{\partial F_1}{\partial x_2}\right| + \cdots + \left|\frac{\partial F_1}{\partial x_N}\right| &< 1 \\ &\vdots \\ \left|\frac{\partial F_N}{\partial x_1}\right| + \left|\frac{\partial F_N}{\partial x_2}\right| + \cdots + \left|\frac{\partial F_N}{\partial x_N}\right| &< 1 \end{aligned} \tag{40}$$

Methods of type $x = g(x)$ iii

Example

The non-linear system of equations

$$x^2 + y^2 = 4 \quad \text{and} \quad e^x - 3y = 1$$

with solutions (1.5595,1.2522) & (-1.9793,-0.2873) can be written in the form:

$$x_{n+1} = -\sqrt{4 - y_n^2} \quad \text{and} \quad y_{n+1} = \frac{1}{3}(e^{x_n} - 1)$$

for an initial choice $(-1, 0)$, we create the following sequence of solutions

n	0	1	2	3	4	5
x	-1	-2	-1.9884	-1.9791	-1.9792	-1.9793
y	0	-0.2107	-0.2882	-0.2877	-0.2873	-0.2873

which after 5 iterations approaches the exact solution.

Example II

If we use the 2nd method then the results are:

n	0	1	2	3
x	-1	-2	-1.9791	-1.9793
y	0	-0.2882	-0.2873	-0.2873

i.e. the same accuracy has been achieved with only 3 iterations.

In order to find the 2nd solution of the system we modify the recurrence relations as follows:

$$x_{n+1} = \sqrt{4 - y_n^2} \quad \text{and} \quad y_{n+1} = \frac{1}{3}(e^{x_n} - 1)$$

If we start with values close to the exact solution e.g. $x_0 = 1.5$ and $y_0 = 1$ the results are :

Methods of type $x = g(x)$ v

n	0	1	2	3	4	5
x	1.5	1.7321	1.2630	1.8126	1.0394	1.9050
y	1	1.5507	0.8453	1.7087	0.6092	1.9064

we notice that we diverge from the solution and the reason is that the criteria for convergence set earlier are not satisfied.

While if we write the recurrence relation in the form

$$\begin{aligned} y_{n+1} &= \sqrt{4 - x_n^2} \\ x_{n+1} &= \ln(1 + 3y_n) \end{aligned}$$

after some iterations we will find the 2nd solution.

Selected problems

1. Solve the following system of equations using both methods and estimate the rate of convergence:

$$e^x - y = 0 \quad \text{and} \quad xy - e^x = 0$$

2. Consider the nonlinear system of equations

$$f_1(x, y) = x^2 + y^2 - 2 = 0, \quad \text{and} \quad f_2(x, y) = xy - 1 = 0\,.$$

   It is obvious that the solutions are $(1, 1)$ and $(-1, -1)$, examine the difficulties that might arise if we try to use Newton's method to find the solution.