Numerical Methods II

SSCM 3423

Part 1

Prepared by:

Dr. Yeak Su Hoe
C22-432, U.T.M. Skudai
s.h.yeak@utm.my
012-7116604

August 2013

Single & multistep method, Single step method for system of ODE

Finite difference method for boundary value problem

Well-posed

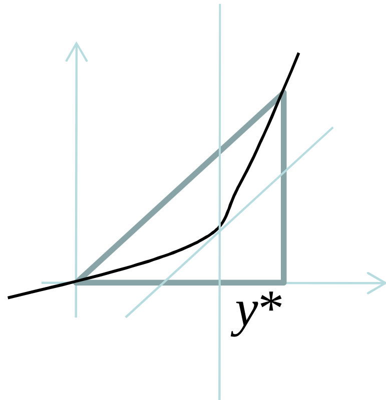
Given initial value problem (IVP)
$y' = f(x,y), \; y(a) = \eta \qquad (1)$

Lipschitz condition:
$|f(x,y) - f(x,y^*)| \le L|y - y^*|$
where L is Lipschitz costant

If η is any given number, there exist a unique solution $y(x)$ where $y(x)$ is continuous and differentiable for all (x,y) in domain D and fulfill Lipschitz condition.

Using **mean value theorem**

$$f(x,y) - f(x,y^*) = \frac{\partial f(x,\bar{y})}{\partial y}(y - y^*)$$



So we choose

$$L = \sup_{(x,y)\in D}\left|\frac{\partial f(x,y)}{\partial y}\right|$$

An initial value problem that has a **unique solution** and is **stable** is called **well-posed**. If $f(t,y)$ is continuous and satisfied a **Lipschitz condition**, then the IVP is well posed.

First-order initial value problems : **single step methods**

Euler method

Generally, *n*-order ODE has the form

$$\frac{d^n y}{dt^n} = y^{(n)} = f\left(t, y, y', y'', y^{(3)}, \cdots, y^{(n-1)}\right)$$

$y'=f(t,y),\ a\le t\le b,\ y(a)=\alpha,$

where y is function of single variable t and n is +ve integer.

If initial condition is given, $y(t=a)=y_0$, where a and y_0 are given constants, then it become **first-order initial value problem**.

Euler method

Taylor's series at $t=a$

$$f(x+h) = f(x) + h\frac{f'(x)}{1!} + h^2\frac{f''(x)}{2!} + \cdots + h^n\frac{f^{(n)}(x)}{n!} + R_n^*.$$

$$y(t) = y(a) + \frac{y'(a)}{1!}(t-a) + \frac{y''(a)}{2!}(t-a)^2 + \cdots + \frac{y^{(n)}(a)}{n!}(t-a)^n + R_n.$$

$$R_n = \frac{y^{(n+1)}(\theta\, t)}{(n+1)!}(t-a)^{n+1},\quad a<\theta\, t<x.$$

Let $y'=f$, $t_i=a$, $t_{i+1}=t$, $h=t-a$, and truncated the series after the second term, for $h\to 0$, we get

$$\boxed{y_{i+1} = y_i + hf(t_i, y_i) + O(h^2), \quad y_{i+1} \approx y_i + hf(t_i, y_i).}$$

Basic Euler formula, First order

Truncation error

"Big O" notation$\to O(g(t))$.

Some finite value $\times\, g(t)$

$O(h^2)$=Some finite value $\times\, h^2$

$|O(h^2)|\le M|h^2|$

Definition: $\boldsymbol{f(x)}$ **has order** $\boldsymbol{O(g(x))}$ as $x\to a$, if and only if $\boldsymbol{|f(x)|\le M|g(x)|}$ for $|x-a|<\delta$, where $0<M<\infty$, $\delta>0$.

First-order initial value problems : **single step methods**

Euler method

Example:

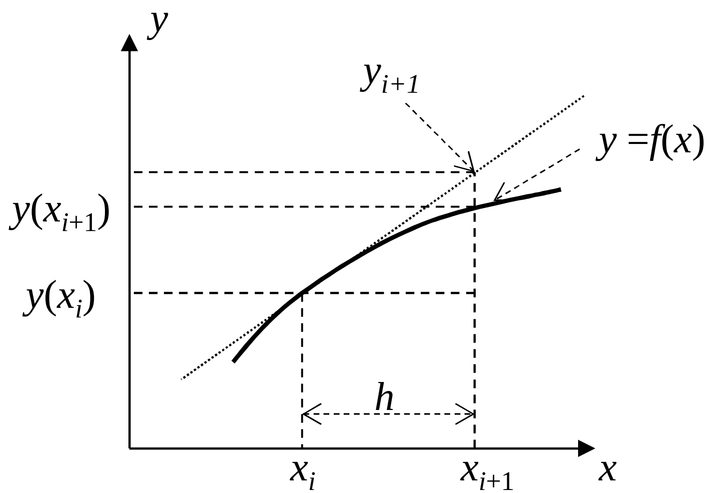
• if $f(x)=6x^4-2x^3$,
$f(x)$ has order $O(x^4)$ as $x\to\infty$, $f(x)$ has order $O(x^3)$ as $x\to 0$. $6x^4-2x^3+5$ has order O(1) as $x\to 0$.

• $(n+1)^2 = n^2 + O(n)+O(1)$.

• $(n+1)/n^2$ has order $O(1/n)$, $5/n+e^{-n}$ has order $O(1/n)$ as $n\to\infty$.

E.g. initial value problem:

$y'=y-t^2+1,\ 0 \le t \le 1,\ y(0)=0.5$ $\qquad$ $y'=f(t,y),\ a\le t\le b,\ y(a)=\alpha,$



Let h =0.2, y_0=0.5, we get (here use 7 decimal places, D.P.)

$$\boxed{y_{i+1} = y_i + hf\left(t_i, y_i\right) = y_i + h\left(y_i - t_i^2 + 1\right), \quad i = 0,\cdots,4.}$$

The exact solution is $y(t)=(t+1)^2-0.5e^t$.

$dy/dt=f(t,y)$

Time, t_i	**Approx, y_i**	**Exact, $y(t_i)$**	**Absolute error, $\lvert y(t_i)-y_i\rvert$**
0.0	0.5	0.5	0
0.2	0.8	0.8292986	0.0292986
0.4	1.152	1.2140877	0.0620877
0.6	1.5504	1.6489406	0.0985406
0.8	1.98848	2.1272295	0.1387495
1.0	2.458176	2.6408591	0.1826831

•48.4 & 48.0 have one **decimal place**
•0.00001845, 0.0001845 and 0.001845 all have
4 **significant figures or digits**.
•4.53×10^4, 4.530×10^4, 4.5300×10^4 have
3, 4 and 5 significant figures.

First-order initial value problems : **single step methods**

Taylor method – higher order one step method

Taylor's series at *t=a* $y' = f(t,y),\ a \le t \le b,\ y(a) = \alpha,$

$$y(t) = y(a) + \frac{y'(a)}{1!}(t-a) + \frac{y''(a)}{2!}(t-a)^2 + \cdots + \frac{y^{(n)}(a)}{n!}(t-a)^n + R_n. \qquad R_n = \frac{y^{(n+1)}(\theta\, t)}{(n+1)!}(t-a)^{n+1},\ \ a < \theta\, t < x.$$

Let $y' = f$, $t_i = a$, $t_{i+1} = t$, $h = t - a$, and truncated the series after the second term, for $h \to 0$, we get

$$\boxed{y_{i+1} = y_i + hf(t_i, y_i) + \tfrac{1}{2} h^2 \tfrac{d}{dt} f(t,y)\Big|_{t_i} + O(h^3), \quad y_{i+1} \approx y_i + hf(t_i, y_i) + \tfrac{1}{2} h^2 \left(\tfrac{\partial f}{\partial t} + \tfrac{\partial f}{\partial y} f \right)\Big|_{t_i}}$$

Second - order Taylor method

E.g. Consider IVP

$$\frac{dx}{dt} = 1 + \frac{x}{t}, \quad (1 \le t \le 6), \ \ x(1) = 1.$$

Exact solution: $x(t) = t(1 + \ln t)$

First-order initial value problems : **single step methods**

Second-order Runge-Kutta Method $y'=f(t,y),\ a\le t\le b,\ y(a)=\alpha,$

Assume

$$\boxed{\frac{y(t+h)-y(t)}{h}=a_1 f(t,y)+a_2 f\left(t+\alpha_2, y+\delta_2 f(t,y)\right)}$$

Higher dimension Taylor expansion

$$f(x+a,y+b)=f(x,y)+\frac{1}{1!}D_1[f(x,y)]+\frac{1}{2!}D_2[f(x,y)]+\cdots+\frac{1}{n!}D_n[f(x,y)]+R_n.$$

$$D_n=\left(a\frac{\partial}{\partial x}+b\frac{\partial}{\partial y}\right)^n \quad R_n=\frac{1}{(n+1)!}D_{n+1}[f(x+\theta_1 a, y+\theta_2 b)]$$

$$0<\theta_1<1,\ 0<\theta_2<1.$$

We get

$$f(t+\Delta t, y+\Delta y)=f(t,y)+\left[\Delta t\frac{\partial f(t,y)}{\partial t}+\Delta y\frac{\partial f(t,y)}{\partial y}\right]$$

$$\left.+\left[\frac{(\Delta t)^2}{2}\frac{\partial^2 f(\xi,\eta)}{\partial t^2}+\Delta t\Delta y\frac{\partial^2 f(\xi,\eta)}{\partial t\partial y}++\frac{(\Delta y)^2}{2}\frac{\partial^2 f(\xi,\eta)}{\partial y^2}\right]\right\} O(\Delta t^{\,2})$$

Second order Taylor method

$$\boxed{y(t+h)=y+hf(t,y)+\tfrac{1}{2}h^2\, \tfrac{d}{dt}\, f(t,y)\Big|_{t_i}+O(h^3), \quad y(t+h)\approx y+hf(t,y)+\tfrac{1}{2}h^2\left(\tfrac{\partial f}{\partial t}+\tfrac{\partial f}{\partial y}\, f\right)}$$

$$\frac{y(t+h)-y(t)}{h}=a_1 f(t,y)+a_2 f\left(t+\alpha_2, y+\delta_2 f(t,y)\right)$$

$$=a_1 f(t,y)+a_2\left[f(t,y)+\alpha_2\frac{\partial f}{\partial t}+\delta_2 f(t,y)\frac{\partial f}{\partial y}+R_1\right]$$

$$=(a_1+a_2)f(t,y)+a_2\alpha_2\frac{\partial f}{\partial t}+a_2\delta_2 f(t,y)\frac{\partial f}{\partial y}+a_2R_1$$

$$R_1=\left[\alpha_2^2\frac{\partial^2 f(\xi,\eta)}{\partial t^2}+\alpha_2\delta_2 f(t,y)\frac{\partial^2 f(\xi,\eta)}{\partial t\partial y}+\delta_2^2 f^2(t,y)\frac{\partial^2 f(\xi,\eta)}{\partial y^2}\right]=O(h^2)$$

$$\left.\begin{array}{l} a_1+a_2=1 \\ a_2\alpha_2=h/2 \\ a_2\delta_2=h/2 \quad \rightarrow \alpha_2=\delta_2=h/(2a_2)\end{array}\right.$$

3 Eqs with **4 unknown** leads to non-uniquely solution. **undetermined systems**

First-order initial value problems : **single step methods**

Second-order Runge-Kutta Method

$y' = f(t,y),\ a \le t \le b,\ y(a) = \alpha,$

Modified Euler method
(Midpoint formulae)

$a_1=0,\ a_2=1;\ \alpha_2=\delta_2=h/2$

$$y^* = y_i + \frac{h}{2} f(t_i, y_i),$$

$$y_{i+1} \approx y_i + hf(t_i + h/2, y^*).$$

Improved Euler's formula
(Heun method)

$a_1=a_2=1/2\ ;\ \alpha_2=\delta_2=h$

$$y_{i+1} \approx y_i + \frac{1}{2}(K_1 + K_2)$$

$$K_1 = hf(t_i, y_i)$$

$$K_2 = hf(t_i + h, y_i + K_1)$$

Optimal RK2 method

$a_1=1/4,\ a_2=3/4;\ \alpha_2=\delta_2=2h/3$

$$y_{i+1} \approx y_i + \frac{1}{4}(K_1 + 3K_2)$$

$$K_1 = hf(t_i, y_i)$$

$$K_2 = hf\left(t_i + \frac{2}{3}h, y_i + \frac{2}{3}K_1\right)$$

E.g. Consider IVP

$$\frac{dx}{dt} = 1 + \frac{x}{t}, \quad (1 \le t \le 6), \ \ x(1) = 1.$$

Exact solution: $x(t)=t(1+\ln t)$

First-order initial value problems : **single step methods**

Classical fourth-order Runge-Kutta Method (RK4) $y' = f(x,y)$, $a \le x \le b$, $y(a) = \alpha$,

RK4 is given

$$y(x+h) \approx y(x) + 1/6[\, k_1 + 2k_2 + 2k_3 + k_4] + O(h^5)$$

$$k_1 = hf(x,y),\ k_2 = hf(x + \tfrac{1}{2}h, y + \tfrac{1}{2}k_1),\ k_3 = hf(x + \tfrac{1}{2}h, y + \tfrac{1}{2}k_2),\ k_4 = hf(x+h, y+k_3)$$

Note: calculation of higher derivatives of $y(x)$ is not required.

Second-order Runge-Kutta (RK2) method (Heun's method).

$$y(x+h) \approx y(x) + \tfrac{1}{2}[\, k_1 + k_2] + O(h^3)$$

$$k_1 = hf(x,y),\ k_2 = hf(x+h, y+k_1)$$

e.g. $y' = f(x,y) = -2x^3 + 12x^2 - 20x + 8.5$, with step size $h=0.5$ and initial condition $y(0)=1$,

Find $y(-0.5)$ using RK2 and RK4. exact answer: $y(x) = -\tfrac{1}{2}x^4 + 4x^3 - 10x^2 + 8.5x + 1$. $y(-0.5) = -6.28125$.

RK2: $k_1 = hf(0,1) = (-0.5)[-2(0)^3 + 12(0)^2 - 20(0) + 8.5] = -4.25$, (here, $\boldsymbol{h=-0.5}$)

$k_2 = hf(0-0.5, 1-4.25) = (-0.5)[-2(-0.5)^3 + 12(-0.5)^2 - 20(-.5) + 8.5] = -10.875$.

Answer: $y(-0.5) = y(0) + \tfrac{1}{2}[\, k_1 + k_2] = 1 + \tfrac{1}{2}[-4.25 - 10.875] = -6.5625$. $\varepsilon_t = -4.5\%$.

First-order initial value problems : **fourth-order Runge-Kutta method (RK4)**

RK4: $k_1=hf(0,1)=(-0.5)[-2(0)3+12(0)2-20(0)+8.5]=-4.25$, (here, $\boldsymbol{h}$**=-0.5**)

$k_2=hf(0\text{-}0.5/2,1\text{-}4.25/2)= (-0.5)[-2(-0.25)3+12(-0.25)2-20(-.25)+8.5]=-7.140625$,

$k_3=hf(0\text{-}0.5/2,1\text{-}7.140625/2)= (-0.5)[-2(-0.25)3+12(-0.25)2-20(-.25)+8.5]=-7.140625$,

$k_4=hf(0\text{-}0.5,1\text{-}7.140625)= (-0.5)[-2(-0.5)3+12(-0.5)2-20(-.5)+8.5]= -10.875$.

Answer: $y(-0.5)=y(0)+ 1/6[k_1+2k_2+2k_3+k_4]=1+1/6 [-43.6875]=-6.28125.\rightarrow$ $\boldsymbol{\varepsilon_t=0\%}$.

Note: RK4 produce exact solution since true solution is a quartic. The fourth-order method gives an exact result.

E.g. $\boldsymbol{y'=4e^{0.8x}-0.5y}$, $0\leq x\leq 0.5$, step size (h=0.5), initial condition, $y(0)=2$.

Analytical solution: $y=(4/1.3)[e^{0.8x}-e^{-0.5x}]+2e^{-0.5x}$. $y(0.5)=3.751521$

RK4: $k_1=hf(0,2)=(0.5)[4e^{0.8(0)}-0.5(2)]=1.5$, (here, $\boldsymbol{h}$**=0.5**)

$k_2=hf(0+0.5/2,2+1.5/2)=(0.5)[4e^{0.8(0.25)}-0.5(2.75)]=1.755306$,

$k_3=hf(0+0.5/2,2+1.755306/2)= (0.5)[4e^{0.8(0.25)}-0.5(2.877653)]=1.723392$,

$k_4=hf(0+0.5,2+1.723392)= (0.5)[4e^{0.8(0.5)}-0.5(3.723392)]=2.052801$,

Answer: $y(0.5)=y(0)+ 1/6[k1+2k2+2k3+k4]=2+1/6 [-43.6875]=3.7516995 \rightarrow$ $\boldsymbol{\varepsilon_t=-0.0048\%}$.

Ordinary differential equations (ODEs)

First-order initial value problems : **multistep methods**

m-step multistep method for solving initial value problem:

$$y' = f(x,y), \quad a \le x \le b, \; y(a) = \alpha,$$

is given

$$y_{i+1} = a_{m-1} y_i + a_{m-2} y_{i-1} + \ldots + a_0 y_{i+1-m}$$
$$+ h[b_m f(x_{i+1}, y_{i+1}) + b_{m-1} f(x_i, y_i) + \ldots + b_0 f(x_{i+1-m}, y_{i+1-m})].$$

for $i = m-1, m, \ldots, N-1$, where $h = (b-a)/N$, a_i and b_i are constants, and the starting values $y_0 = \alpha$, $y_1 = \alpha_1$, $y_2 = \alpha_2$, …, $y_{m-1} = \alpha_{m-1}$.

When $b_m = 0$ the method is called **explicit**, or **open**. When $b_m \neq 0$ the method is called **implicit**, or **closed**, since y_{i+1} occurs on both sides.

Adams-Bashforth – open formulas

$\boxed{f_i = f(x_i, y_i)}$

Taylor series expansion around x_i

$$y_{i+1} = y_i + f_i h + \frac{f_i'}{2} h^2 + \frac{f_i''}{3!} h^3 + \cdots = y_i + h\left(f_i + \frac{f_i'}{2} h + \frac{f_i''}{3!} h^2 + \cdots \right)$$

Backward difference to approximate derivative: $f_i' = \dfrac{f_i - f_{i-1}}{h} + \dfrac{f_i''}{2} h + O(h^2)$

$$y_{i+1} = y_i + h\left\{ f_i + \frac{h}{2}\left[\frac{f_i - f_{i-1}}{h} + \frac{f_i''}{2} h + O(h^2) \right] + \frac{h^2}{6} f_i'' + \cdots \right\} \xrightarrow{\text{simplify}} y_{i+1} = y_i + h\left(\frac{3}{2} f_i - \frac{1}{2} f_{i-1} \right) + \frac{5}{12} h^3 f_i'' + O(h^4)$$

$$\boxed{y_{i+1} = y_i + h\left(\frac{3}{2} f_i - \frac{1}{2} f_{i-1} \right)} + O(h^3), \quad \text{2-step method.}$$

Ordinary differential equations (ODEs)

First-order initial value problems : **multistep methods**

,

Coefficients and truncation error for *n*-order (open) Adams-Bashforth predictors

$$y_{i+1} = y_i + h\sum_{k=0}^{n-1} b_k f_{i-k} + O\left(h^{n+1}\right)$$

$f_i = f(x_i, y_i)$

order	$\boldsymbol{b_0}$	$\boldsymbol{b_1}$	$\boldsymbol{b_2}$	$\boldsymbol{b_3}$	$\boldsymbol{b_4}$	**Local truncation error**
1	1					$\frac{1}{2}h^2 f'(\xi)$
2	3/2	-1/2				$\frac{5}{12}h^3 f''(\xi)$
3	23/12	-16/12	5/12			$\frac{9}{24}h^4 f^{(3)}(\xi)$
4	55/24	-59/24	37/24	-9/24		$\frac{251}{720}h^5 f^{(4)}(\xi)$
5	1901/720	-2774/720	2616/720	-1274/720	251/720	$\frac{475}{1440}h^6 f^{(5)}(\xi)$

Adams-Moulton – closed formulas

Backward Taylor series expansion around x_{i+1}

$$y_i = y_{i+1} - f_{i+1}h + \frac{f'_{i+1}}{2}h^2 - \frac{f''_{i+1}}{3!}h^3 + \cdots \quad \rightarrow \quad y_{i+1} = y_i + h\left(f_{i+1} - \frac{h}{2}f'_{i+1} + \frac{h^2}{6}f''_{i+1} + \cdots\right)$$

Approximate 1st derivative $f'_{i+1} = \frac{f_{i+1} - f_i}{h} + \frac{f''_{i+1}}{2}h + O\left(h^2\right)$

$$\boxed{y_{i+1} = y_i + h\left(\frac{1}{2}f_{i+1} + \frac{1}{2}f_i\right) - \frac{1}{12}h^3 f''_{i+1} - O\left(h^4\right)}$$

$$y_{i+1} = y\left(x_i + h\right) = y_i + hy'_i + \frac{h^2}{2!}y''_i + \cdots$$

$$y_i = y\left(x_i + h - h\right) = y\left(x_i + h\right) - hy'_i + \frac{h^2}{2!}y''_i - \cdots$$

$$y^{(0)}{}_{i+1} \approx y_i + \frac{h}{2}\left(3f_i - f_{i-1}\right)$$

$$y^{(k+1)}{}_{i+1} \approx y_i + \frac{h}{2}\left(f^{(k)}{}_{i+1} + f_i\right)$$

Second order Adams-Moulton formula

Ordinary differential equations (ODEs)

First-order initial value problems : **multistep methods**

Coefficients and truncation error for *n*-order (closed) Adams-Moulton correctors

$$y_{i+1} = y_i + h\sum_{k=0}^{n-1} b_k f_{i+1-k} + O\left(h^{n+1}\right)$$

$f_i=f(x_i,y_i)$

order	b_0	b_1	b_2	b_3	b_4	Local truncation error
2	½	½				$-\frac{1}{12}h^3 f''(\xi)$
3	5/12	8/12	-1/12			$-\frac{1}{24}h^4 f^{(3)}(\xi)$
4	9/24	19/24	-5/24	1/24		$-\frac{19}{720}h^5 f^{(4)}(\xi)$
5	251/720	646/720	-264/720	106/720	-19/720	$-\frac{27}{1440}h^6 f^{(5)}(\xi)$

y_{i+1}^{j} ← iteration

← node

Fourth-order Adams method (requires **4** previous values)

Predictor: $y_{i+1}^0 = y_i^m + h\left(\frac{55}{24}f_i^m - \frac{59}{24}f_{i-1}^m + \frac{37}{24}f_{i-2}^m - \frac{9}{24}f_{i-3}^m\right)$ $\Longrightarrow$ Predictor can be used alone!

Corrector: $y_{i+1}^j = y_i^m + h\left(\frac{9}{24}f_{i+1}^{j-1} + \frac{19}{24}f_i^m - \frac{5}{24}f_{i-1}^m + \frac{1}{24}f_{i-2}^m\right)$

where j=iteration=1,2,…,m.

Iterations terminate on **approximate percent relative error**, ε_a.

$$\boxed{\left|\varepsilon_a\right| = \left|\frac{y_{i+1}^j - y_{i+1}^{j-1}}{y_{i+1}^j}\right| \cdot 100\%}$$

Predictor calculate for y_1^0 (use y_0 and others); get y_2^0 (use y_1^m and others); …
Corrector calculate for $y_1^1, y_1^2, \ldots, y_1^m$. ($m$ iteration); get $y_2^1, y_2^2, \ldots, y_2^m$. ($m$ iteration); …

Ordinary differential equations (ODEs)

First-order initial value problems : **multistep methods**

E.g. $\boldsymbol{y'=4e^{0.8x}-0.5y}$, $0\le x\le 4$, step size (h=1), initial condition, $y(0)=2$.

Analytical solution: $y=(4/1.3)[e^{0.8x}-e^{-0.5x}]+2e^{-0.5x}$.

Previous values can be approximated by Runge-Kutta method or Taylor series.

here we use analytical solution to compute exact values at $x_{-3}=-3$, $x_{-2}=-2$, $x_{-1}=-1$, with $y_{-3}=-4.547302$, $y_{-2}=-2.306160$ & $y_{-1}=-0.3929953$.

$$y'_0 = f(x_0,y_0) = f_0 = f_0^m = f(0,2) = 3,\; y'_{-1} = f(x_{-1},y_{-1}) = f_{-1}^m = f(-1,-0.3929953) = 1.993814.$$

$$y'_{-2} = f(x_{-2},y_{-2}) = f_{-2}^m = f(-2,-2.30616) = 1.960667,\; y'_{-3} = f(x_{-3},y_{-3}) = f_{-3}^m = f(-3,-4.547302) = 2.6365228.$$

By setting <u>number of iterations, m=1</u>, we get

Predictor:

$$y_1^0 = y_0^m + h\left(\frac{55}{24}f_0^m - \frac{59}{24}f_{-1}^m + \frac{37}{24}f_{-2}^m - \frac{9}{24}f_{-3}^m\right)$$

$$= 2 + 1\left(\frac{55}{24}3 - \frac{59}{24}1.993814 + \frac{37}{24}1.960667 - \frac{9}{24}2.6365228\right) = 6.007539.$$

Second order Adams-Moulton formula

$$y^{(0)}{}_{i+1} \approx y_i + \frac{h}{2}\left(3f_i - f_{i-1}\right)$$

$$y^{(k+1)}{}_{i+1} \approx y_i + \frac{h}{2}\left(f^{(k)}{}_{i+1} + f_i\right)$$

True percent relative error, $\varepsilon_t = {}^{\text{(true value-approx)}}/_{\text{(true value)}} \times 100\% = 3.1\%$

Corrector:

$$y_{i+1}^j = y_i^m + h\left(\frac{9}{24}f_{i+1}^{j-1} + \frac{19}{24}f_i^m - \frac{5}{24}f_{i-1}^m + \frac{1}{24}f_{i-2}^m\right)$$

$$f_i^m = f\left(x_i, y_i^m\right)$$

$$f_1^0 = f(x_1, y_1^0) = f(1, 6.007539) = 5.898394.$$

$$y_1^1 = y_0^m + h\left(\frac{9}{24}f_1^0 + \frac{19}{24}f_0^m - \frac{5}{24}f_{-1}^m + \frac{1}{24}f_{-2}^m\right)$$

$$= 2 + 1\left(\frac{9}{24}5.898394 + \frac{19}{24}3 - \frac{5}{24}1.993814 + \frac{1}{24}1.960667\right) = 6.253214.$$

$\varepsilon_t = -0.96\%$ ⇨ improvement

$$y(x+h) = y(x) + h\frac{y'(x)}{1!} + h^2\frac{y''(x)}{2!} + \cdots$$

Ordinary differential equations (ODEs)

First-order initial value problems : **single step methods**

Improved Euler method – Heun's method (Predictor-corrector method)

Slope ,$f(t,y)$, is the average of two derivatives.

$$y_{i+1} \approx y_i + \frac{1}{2}(K_1 + K_2)$$

$$y'_i = f(t_i, y_i) \rightarrow \quad y^0_{i+1} = y_i + hf(t_i, y_i) \qquad\qquad y'_{i+1} = f\left(t_{i+1}, y^0_{i+1}\right)$$

$$K_1 = hf(t_i, y_i)$$

Average slopes: $$\bar{y}' = \frac{y'_i + y'_{i+1}}{2} = \frac{f(t_i, y_i) + f\left(t_{i+1}, y^0_{i+1}\right)}{2}$$

$$K_2 = hf(t_i + h, y_i + K_1)$$

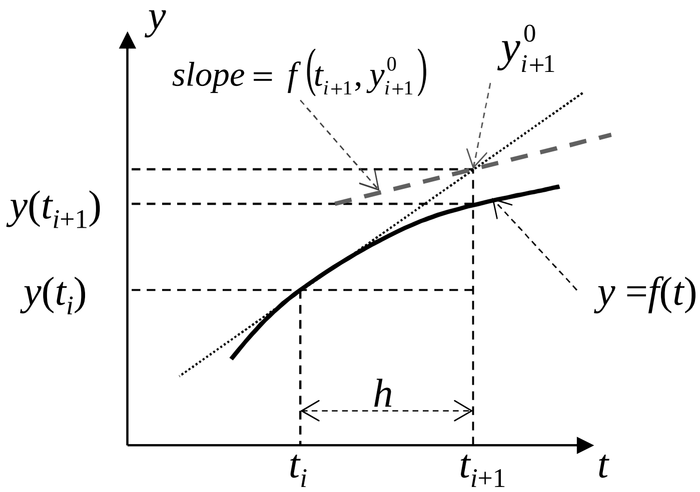
Explicit form: $$\boxed{y_{i+1} = y_i + \bar{y}'h = y_i + \frac{f(t_i, y_i) + f\left(t_{i+1}, y^0_{i+1}\right)}{2}h = y_i + \frac{h}{2}\left(f(t_i, y_i) + f\left(t_{i+1}, y_i + hf(t_i, y_i)\right)\right)}$$

Predictor-corrector approach form:

Predictor: $$y^0_{i+1} = y_i + hf(t_i, y_i)$$

Corrector: $$y_{i+1} = y_i + \frac{f(t_i, y_i) + f\left(t_{i+1}, y^0_{i+1}\right)}{2}h$$

Predictor is basic Euler method



Ordinary differential equations (ODEs)

First-order initial value problems : **single step methods**

Improved Euler method – Heun's method (Predictor-corrector method)

$y(x=1)=?$

E.g. $\boldsymbol{y'=4e^{0.8x}-0.5y}$, $0\le x\le 4$, step size (h=1), initial condition, $y(0)=2$.

Analytical solution: $y=(4/1.3)[e^{0.8x}-e^{-0.5x}]+2e^{-0.5x}$

Standard Euler method

Slope at (x_0,y_0): $y_0' = 4e^0 - 0.5(2) = 3$ predictor→ $y_1^0 = y_0 + 1f(x_0,y_0) = 2+3(1) = 5$

(1 iteration) corrector→

$$y_1 = y_0 + \frac{f(x_0,y_0)+f(x_1,y_1^0)}{2}h = 2+1\frac{f(0,2)+f(1,5)}{2} = 2+\frac{3+4e^{(0.8)1}-0.5(5)}{2}1 = 6.701082. \quad (\varepsilon_t = -8.18\%)$$

True percent relative error, $\varepsilon_t = \dfrac{true\,value - approx}{true\,value}\cdot 100\%$

(2 iteration) corrector $(y_1^0 \leftarrow y_1) \rightarrow$

$$y_1 = 2+1\frac{f(0,2)+f(1,6.701082)}{2} = 2+\frac{3+4e^{(0.8)1}-0.5(6.701082)}{2}1 = 6.275811. \ (\varepsilon_t = -1.31\%)$$

(3 iteration) corrector $(y_1^0 \leftarrow y_1) \rightarrow$

$$y_1 = 2+1\frac{f(0,2)+f(1,6.275811)}{2} = 2+\frac{3+4e^{(0.8)1}-0.5(6.275811)}{2}1 = 6.382129. \quad (\varepsilon_t = -3.03\%)$$

$$y_1^1 = y_0 + \frac{f(x_0,y_0)+f(x_1,y_1^0)}{2}h$$

Repeat the iteration

Normally, we use 3 or 4 D.P. in calculation!

Ordinary differential equations (ODEs)

First-order initial value problems : **single step methods**

Improved Euler method – Heun's method (Predictor-corrector method)

$$y_{i+1} \approx y_i + \frac{1}{2}(K_1 + K_2)$$

$$K_1 = hf\left(t_i, y_i\right)$$

$$K_2 = hf\left(t_i + h, y_i + K_1\right)$$

		Iterations of Heun's method			
		Iteration =1		Iteration = 15	
x	y_{true}	y_{approx}	$\lvert\varepsilon_t\rvert$ (%)	y_{approx}	$\lvert\varepsilon_t\rvert$ (%)
0	2	2	0	2	0
1	6.1946314	6.7010819	8.18	6.3608655	2.68
2	14.8439219	16.3197819	9.94	15.3022367	3.09
3	33.6771718	37.1992489	10.46	34.7432761	3.17
4	75.3389626	83.3377674	10.62	77.7350962	3.18

Normally, we use
3 or 4 D.P. in calculation!

$$\frac{dy}{dx} = f(x) \rightarrow \int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x)dx$$

$$\rightarrow y_{i+1} - y_i = \int_{x_i}^{x_{i+1}} f(x)dx \rightarrow \quad y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x)dx$$

Trapezoidal rule for integration

$$\int_{x_i}^{x_{i+1}} f(x)dx = \frac{f(x_i)+f(x_{i+1})}{2}h - \frac{h^3}{12}f''(\xi), \quad x_i < \xi < x_{i+1}.$$

$$\rightarrow y_{i+1} = y_i + \frac{f(x_i)+f(x_{i+1})}{2}h + O(h^3)$$

Local error

Heun's method is second order since second derivative of ODE is zero. Local error is $O(h^3)$.
Propagated truncated error results from the approximations produced in previous steps. The sum of two is Global truncation error.

Ordinary differential equations (ODEs)

Systems of first-order initial value problems

Standard form for a system of first-order ODEs is:

$$x_1' = f_1(t, x_1, x_2, \cdots, x_n) \qquad x_1' = \frac{d}{dt}x_1$$

$$x_2' = f_2(t, x_1, x_2, \cdots, x_n)$$

$$\vdots$$

$$x_n' = f_n(t, x_1, x_2, \cdots, x_n)$$

Example of system of first-order ODEs is given:

$$\boldsymbol{x' = x + 4y - e^t \quad , \quad y' = x + y + 2e^t.}$$

The general solution is

$$\boldsymbol{x = 2ae^{3t} - 2be^{-t} - 2e^t \quad , \quad y = ae^{3t} + be^{-t} + 1/4\ e^t,}$$

where a and b are arbitrary constants.

If the system were given initial conditions

$x(0)=4, \quad y(0)=5/4,$

Then the particular solution would be

$$\boldsymbol{x = 4e^{3t} + 2e^{-t} - 2e^t \quad , \quad y = 2e^{3t} - e^{-t} + 1/4\ e^t.}$$

Ordinary differential equations (ODEs)

Systems of first-order initial value problems

Let X denote column vector whose components are x_1, x_2,…, x_n. These components are functions of t. And let F denote column vector with components f_1, f_2, …, f_n.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad \text{system of ODE} \rightarrow X' = F(t, X) \Leftrightarrow \begin{bmatrix} \frac{d}{dt} x_1 \\ \frac{d}{dt} x_2 \\ \vdots \\ \frac{d}{dt} x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}.$$

e.g. convert the initial-value problem

$(\sin t)y''' + \cos(ty) + \sin(t^2 + y'') + (y')^3 = \ln t$

$y(2)=7, \quad y'(2)=3, \quad y''(2)=-4,$

into a system of ODEs.

Solution: introduce new variables x_1,x_2 & x_3 as: $x_1=y$, $x_2=y'$, and $x_3=y''$. The system of ODEs for $X=[x_1, x_2, x_3]^T$ is

$x_1'=x_2, \quad x_2'=x_3, \quad x_3'=[\ln t - x_2^3 - \sin(t^2+x_3) - \cos(tx_1)]/\sin t,$

with initial conditions at t=2 are $X=(7,3,-4)^T$.

$$X' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}' = F = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \left[\ln t - x_2^3 - \sin\left(t^2 + x_3\right) - \cos(tx_1)\right] \Big/ \sin t \end{bmatrix}$$

Ordinary differential equations (ODEs)

Systems of first-order initial value problems

e.g. convert the problem

$(x'')^2 + te^y + y' = x' - x, \qquad y'y'' - \cos(xy) + \sin(tx'y) = x.$

into a system of first order ODEs.

Solution: introduce new variables as: $x_1=x$, $x_2=x'$, $x_3=y$ and $x_4=y'$. The system of ODEs for $X=[x_1, x_2, x_3, x_4]^T$ is

$$
\begin{aligned}
x_1' &= x_2 \\
x_2' &= \left(x_2 - x_1 - x_4 - te^{x_3}\right)^{1/2} \\
x_3' &= x_4 \\
x_4' &= \left[x_1 - \sin(tx_2x_3) + \cos(x_1x_3)\right] / x_4
\end{aligned}
$$

Taylor series for column vector of X can be written as:

$$\boxed{\begin{array}{l} X''(t) = \frac{d}{dt} F(t, X) \\ x_1(t),\ x_2(t), \cdots \end{array}}$$

$$X(t+h) = X(t) + hX'(t) + \frac{h^2}{2!} X''(t) + \frac{h^3}{3!} X^{(3)}(t) + \cdots$$

$$X(t+h) \approx X(t) + hX'(t) + \frac{h^2}{2!} X''(t) + O(h^3) \longleftarrow \text{Second-order Taylor series method}$$

$$X(t+h) \approx X(t) + hX'(t) + O(h^2) \longleftarrow \text{First-order Taylor series method}$$

First-order Taylor series or Euler method for system of ODEs, $X'=F(t,X)$ is

$$\boldsymbol{X(t+h) = X(t) + hF(t,X)}$$

Ordinary differential equations (ODEs)

Systems of first-order initial value problems

Example of system of first-order ODEs is given:

$$\boldsymbol{x' = x + 4y - e^{t}} \qquad , \qquad \boldsymbol{y' = x + y + 2e^{t}}.$$

with initial conditions, $x(0)=4$, $y(0)=5/4$. Calculate $x(0.2)$ and $y(0.2)$ with Euler method.

The particular solution is given $\boldsymbol{x=4e^{3t}+2e^{-t}-2e^{t}}$, $\boldsymbol{y=2e^{3t}-e^{-t}+1/4\, e^{t}}$. $x(0.2)=6.483131$, $y(0.2)=3.130858$.

Here, $h=0.2$.

$$X(0+h) = X(0.2) = X(0) + hF(t=0)$$

$$\begin{bmatrix} x_{0.2} \\ y_{0.2} \end{bmatrix} \approx \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + 0.2 \begin{bmatrix} 4+4(5/4)-e^{0} \\ 4+5/4+2e^{0} \end{bmatrix} = \begin{bmatrix} 4 \\ 5/4 \end{bmatrix} + 0.2 \begin{bmatrix} 8 \\ 7.25 \end{bmatrix} = \begin{bmatrix} 5.6 \\ 2.7 \end{bmatrix}.$$

$$F' = d/dt\, F = \frac{d}{dt}\begin{bmatrix} x+4y-e^{t} \\ x+y+2e^{t} \end{bmatrix} = \begin{bmatrix} x'+4y'-e^{t} \\ x'+y'+2e^{t} \end{bmatrix} = \begin{bmatrix} 5x+8y+6e^{t} \\ 2x+5y+3e^{t} \end{bmatrix}$$

The error vector, E is given as:

E=trues value-approximate values=[6.483131, 3.130858]-[5.6, 2.7]=[0.883131, 0.430858]

The size of error vector can be measured using different norms as below:

Euclidean norm: -----------→ $\|E\|_e = \sqrt{\sum_{i=1}^{n} e_i^2} = \sqrt{0.883131^2 + 0.430858^2} = 0.9826$

***p*-norm:** --------------------→ $\|E\|_p = \left(\sum_{i=1}^{n} |e_i|^p\right)^{1/p} = \left(|0.883131|^p + |0.430858|^p\right)^{1/p}$

1-norm: --------------------→ $\|E\|_1 = \sum_{i=1}^{n} |e_i| = |0.883131| + |0.430858| = 1.313989$

Maximum-magnitude -----→ $\|E\|_\infty = \max_{1\le i\le n} |e_i| = \max\left(|0.883131|, |0.430858|\right) = 0.883131$

or ***uniform-vector* norm:**

	$X_0 \rightarrow$	$X_{0.1} \rightarrow$	$X_{0.2}$
Error:	$O(0.1^2)$	$O(0.1^2)$	Σerror=const$\times$0.02

	$X_0 \rightarrow$	$X_{0.2}$
Error:	$O(0.2^2)$	Σerror=const$\times$0.04

Ordinary differential equations (ODEs)

Systems of first-order initial value problems

Let X denote column vector whose components are $x_1, x_2, \dots, x_n$. These components are functions of t.
And let F denote column vector with components $f_1, f_2, \dots, f_n$.
The classical fourth-order Runge-Kutta (RK4), in vector form, for system of ODE are:

$$X(t+h)=X(t)+1/6(F_1+2F_2+2F_3+F_4)+O(h^5)$$

where

$$F_1=hF(t,X),\ F_2=hF(t+½h,X+½F_1),\ F_3=hF(t+½h,X+½F_2),\ F_4=hF(t+h,X+F_3).$$

$$X' = F(t,X) = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}$$

Previous e.g. : **$x'=x+4y-e^t$,** **$y'=x+y+2e^t$.**

with initial conditions, $x(0)=4$, $y(0)=5/4$. Calculate $x(0.2)$ and $y(0.2)$ with RK4.

The particular solution is given **$x=4e^{3t}+2e^{-t}-2e^t$, $y=2e^{3t}-e^{-t}+1/4\ e^t$.** $x(0.2)=6.483131$, $y(0.2)=3.130858$.

Here, $h=0.2$, $t=0$.

$$X = X(t=0) = \begin{bmatrix} 4 \\ 5/4 \end{bmatrix},\quad F(t,X) = \begin{bmatrix} 4+4(5/4)-e^0 \\ 4+5/4+2e^0 \end{bmatrix} = \begin{bmatrix} 8 \\ 7.25 \end{bmatrix},\quad F_1 = hF(t,X) = 0.2\begin{bmatrix} 8 \\ 7.25 \end{bmatrix} = \begin{bmatrix} 1.6 \\ 1.45 \end{bmatrix}.$$

$$F_2 = hF(t+½h, X+½F_1) = 0.2F\left(0.1, \begin{bmatrix} 4 \\ 5/4 \end{bmatrix} + ½\begin{bmatrix} 1.6 \\ 1.45 \end{bmatrix}\right) = 0.2F\left(0.1, \begin{bmatrix} 4.8 \\ 1.975 \end{bmatrix}\right) = 0.2\begin{bmatrix} 4.8+4(1.975)-e^{0.1} \\ 4.8+1.975+2e^{0.1} \end{bmatrix} = \begin{bmatrix} 2.318966 \\ 1.797068 \end{bmatrix}$$

$$F_3 = hF(t+½h, X+½F_2) = 0.2F\left(0.1, \begin{bmatrix} 4 \\ 5/4 \end{bmatrix} + ½\begin{bmatrix} 2.318966 \\ 1.797068 \end{bmatrix}\right) = 0.2F\left(0.1, \begin{bmatrix} 5.159483 \\ 2.148534 \end{bmatrix}\right) = 0.2\begin{bmatrix} 5.159483+4(2.148534)-e^{0.1} \\ 5.159483+2.148534+2e^{0.1} \end{bmatrix} = \begin{bmatrix} 2.52969 \\ 1.903672 \end{bmatrix}$$

$$F_4 = hF(t+h, X+F_3) = 0.2F\left(0.2, \begin{bmatrix} 4 \\ 5/4 \end{bmatrix} + \begin{bmatrix} 2.52969 \\ 1.903672 \end{bmatrix}\right) = 0.2F\left(0.2, \begin{bmatrix} 6.52969 \\ 3.153672 \end{bmatrix}\right) = 0.2\begin{bmatrix} 6.52969+4(3.153672)-e^{0.2} \\ 6.52969+3.153672+2e^{0.2} \end{bmatrix} = \begin{bmatrix} 3.584595 \\ 2.425234 \end{bmatrix}$$

$$X(0.2) = X(0) + ⅙(F_1+2F_2+2F_3+F_4) = \begin{bmatrix} 4 \\ 5/4 \end{bmatrix} + ⅙\left(\begin{bmatrix} 1.6 \\ 1.45 \end{bmatrix} + 2\begin{bmatrix} 2.318966 \\ 1.797068 \end{bmatrix} + 2\begin{bmatrix} 2.52969 \\ 1.903672 \end{bmatrix} + \begin{bmatrix} 3.584595 \\ 2.425234 \end{bmatrix}\right) = \begin{bmatrix} 6.480318 \\ 3.129452 \end{bmatrix}.$$

Error vector, $E=[6.483131,3.130858]^T-[6.480318,3.129452]^T=[0.002813,0.001406]^T$.

Maximum-magnitude norm, $\|E\|_\infty=\max(|0.002813|,|0.001406|)=\mathbf{0.002813}$.

RK2 Heun

$$X_{i+1} \approx X_i + \frac{1}{2}(K_1+K_2)$$

$$K_1 = hF(t_i, X_i)$$

$$K_2 = hF(t_i+h, X_i+K_1)$$

Ordinary differential equations (ODEs)

Systems of first-order initial value problems

Exercise: $$\frac{d^2y}{dx^2} - 2\frac{dy}{dx} + y = 2x^2 - 1$$

$$X' = F(t, X) = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}$$

Exact solution: $ae^x + bxe^x + 2x^2 + 8x + 11$.

Initial value problem: $y(0)=1$, $y'(0)=2$,
$y = -10e^x + 4xe^x + 2x^2 + 8x + 11$.

RK2 Heun

$$X_{i+1} \approx X_i + \frac{1}{2}(K_1 + K_2)$$
$$K_1 = hF(t_i, X_i)$$
$$K_2 = hF(t_i + h, X_i + K_1)$$

Newton's Method for Approximating Roots

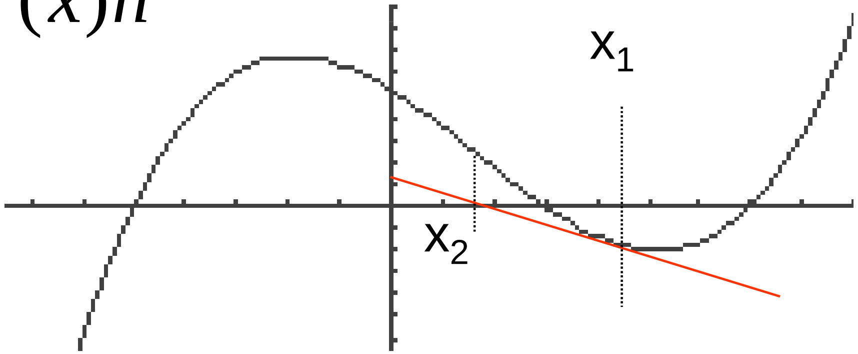
Given: x_i an initial guess of the root of $f(x) = 0$

Question: How do we obtain a better estimate x_{i+1}?

Taylor Therorem: $f(x+h) \approx f(x) + f'(x)h$

Find h such that $f(x+h) = 0$.

$$\Rightarrow h \approx -\frac{f(x)}{f'(x)}$$



A new guess of the root: $$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

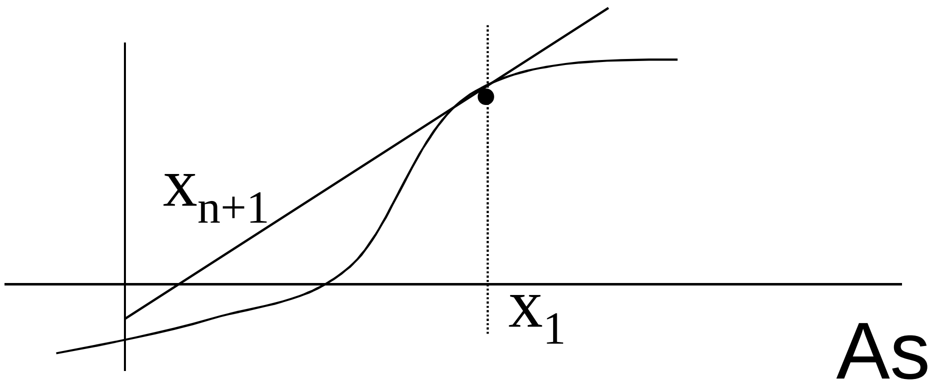
Newton's Method for Approximating Roots

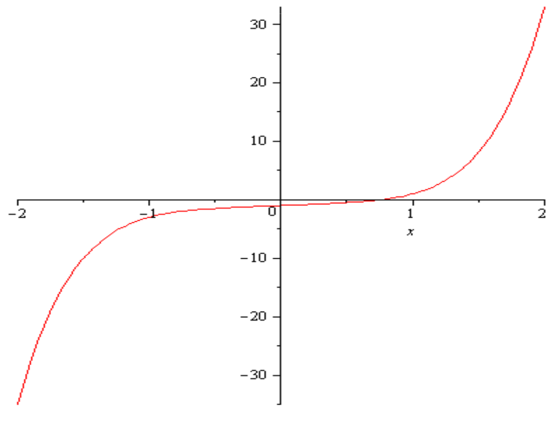
- Given f(x) we seek a root
- If x_n is an approximation for the root

Then we claim

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

is a better approximation





example

$$f(x) = x^5 + x - 1 = 0$$

Answer:
0.7548776667

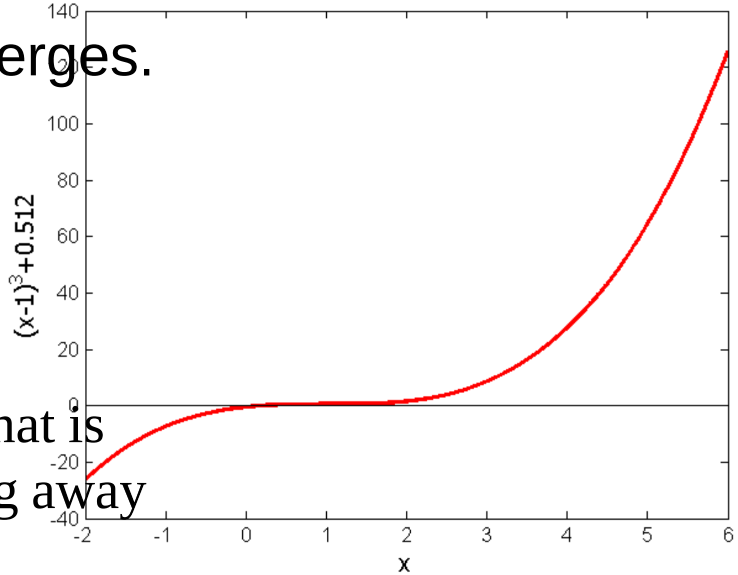
Assumptions:

f(x) is continuous and the first derivative is known
An initial guess x_0 such that $f'(x_0) \neq 0$ is given

Advantages

- Converges fast (quadratic convergence), if it converges.
- Requires only one guess

Drawbacks



1. <u>Divergence at inflection points</u>
   Selection of the initial guess or an iteration value of the root that is close to the inflection point of the function may start diverging away from the root in ther Newton-Raphson method.

For example, to find the root of the equation $f(x) = (x-1)^3 + 0.512 = 0$

The Newton-Raphson method reduces to $x_{i+1} = x_i - \dfrac{(x_i^3 - 1)^3 + 0.512}{3(x_i - 1)^2}$

Table 1 shows the iterated values of the root of the equation.

The root starts to diverge at Iteration 6 because the previous estimate of 0.92589 is close to the inflection point of x=1.

Eventually after 12 more iterations the root converges to the exact value of x=0.2

Drawbacks – Oscillations near local maximum and minimum

Table 3 Oscillations near local maxima and mimima in Newton-Raphson method.

Iteration Number	x_i	$f(x_i)$	$\left\|\in_a\right\|\%$
0	–1.0000	3.00	
1	0.5	2.25	300.00
2	–1.75	5.063	128.571
3	–0.30357	2.092	476.47
4	3.1423	11.874	109.66
5	1.2529	3.570	150.80
6	–0.17166	2.029	829.88
7	5.7395	34.942	102.99
8	2.6955	9.266	112.93
9	0.97678	2.954	175.96

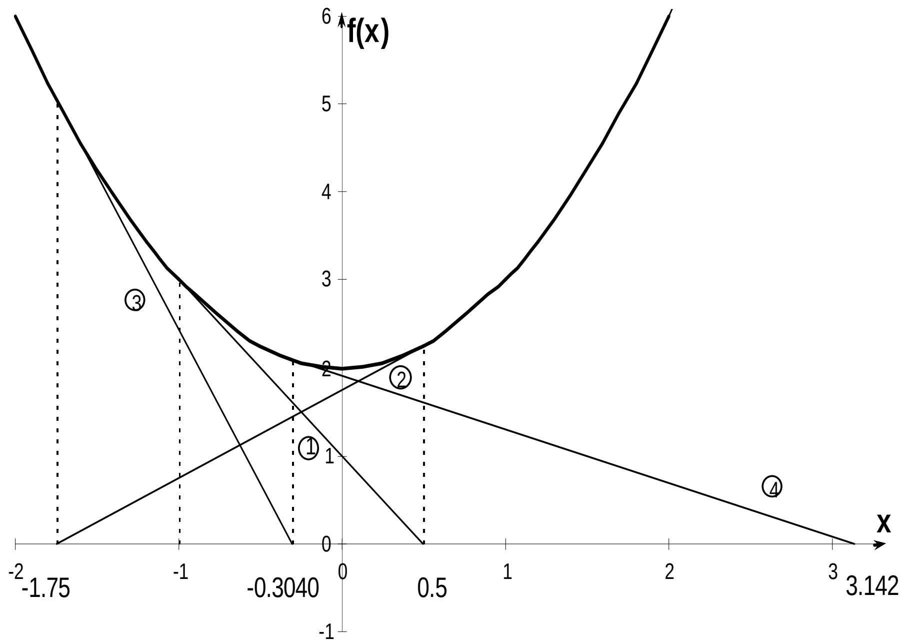


Figure 10 Oscillations around local minima for $f(x) = x^2 + 2$.

Application of Newton method
Finding a square-root

- Example: $\sqrt{2} = 1.41421356237309504880168872420 97$
- Let x_0 be one and apply Newton's method.

$f(x) = x^2 - a = 0$

$$f'(x) = 2x$$

$$x_{i+1} = x_i - \frac{x_i^{\,2} - 2}{2x_i} = \frac{1}{2}\left(x_i + \frac{2}{x_i}\right)$$

$$x_0 = 1$$

$$x_1 = \frac{1}{2}\left(1 + \frac{2}{1}\right) = \frac{3}{2} = 1.5000000000$$

$$x_2 = \frac{1}{2}\left(\frac{3}{2} + \frac{4}{3}\right) = \frac{17}{12} \approx 1.4166666667$$

Note the rapid convergence

$$x_3 = \frac{1}{2}\left(\frac{17}{12} + \frac{24}{17}\right) = \frac{577}{408} \approx 1.414215686$$

$$x_4 = 1.4142135623746$$

$$x_5 = 1.414213562373095048801696$$

$$x_6 = 1.4142135623730950488016887242097$$

Convergence Notation

Let $x_1, x_2, \ldots.,$ converge to x.

Linear Convergence: $\dfrac{|x_{n+1} - x|}{|x_n - x|} \leq C$

Quadratic Convergence: $\dfrac{|x_{n+1} - x|}{|x_n - x|^2} \leq C$

Convergence of order P: $\dfrac{|x_{n+1} - x|}{|x_n - x|^p} \leq C$

- Quadratic convergence is faster than linear convergence.
- A method with convergence order ***q*** converges faster than a method with convergence order ***p*** if ***q>p***.
- Methods of convergence order ***p>1*** are said to have super linear convergence.

Convergence Rate of Newton's

$$e_n = \bar{x} - x_n \quad or \quad \bar{x} = x_n + e_n$$
$$0 \equiv f(\bar{x}) = f(x_n + e_n)$$
$$f(x_n + e_n) = f(x_n) + e_n f'(x_n) + \frac{1}{2} e_n^2 f''(\xi_n), \; for\; some\; \xi_n \in \left(\bar{x}, x_n\right)$$
$$\therefore\; f(x_n) + e_n f'(x_n) = -\frac{1}{2} e_n^2 f''(\xi_n)$$

$$e_{n+1} = \bar{x} - x_{n+1} = \bar{x} - x_n + \frac{f(x_n)}{f'(x_n)} = e_n + \frac{f(x_n)}{f'(x_n)}$$
$$= \frac{e_n f'(x_n) + f(x_n)}{f'(x_n)}$$
$$\therefore\; e_{n+1} = -\frac{1}{2}\left(\frac{f''(\xi_n)}{f'(x_n)}\right) e_n^2$$

Converges **quadratically**.

$$if\; \left|e_n\right| \le 10^{-k} \quad then,$$
$$\left|e_{n+1}\right| \le c10^{-2k}$$

Order of Convergence for Fixed Point Iteration Scheme

Fixed Point Iteration Scheme

Theorem. Let g be a continuous function on closed interval $[a,b]$ with $\alpha>1$ continuous derivatives on open interval (a,b). Let $p\in(a,b)$ be a fixed point of g.

If $\boldsymbol{g'(p)=g''(p)=\ldots=g^{(\alpha-1)}(p)=0}$,

But $\boldsymbol{g^{(\alpha)}(p)\neq 0}$, then there exists a $\delta>0$ such that for any $p_0\in[p-\delta, p+\delta]$, the sequence $\boldsymbol{p_n=g(p_{n-1})}$ converges to fixed point p of order α with asymptotic error constant

$$\lim_{n\to\infty}\frac{|e_{n+1}|}{|e_n|^{\alpha}}=\frac{\left|g^{(\alpha)}(p)\right|}{\alpha!}$$

Let, $x_n=r-\varepsilon_n$, $x_{n+1}=r-\varepsilon_{n+1}$, Where $r=G(r)$

$x_{n+1}=G(x_n) \rightarrow r-\varepsilon_{n+1}=G(r-\varepsilon_n)$ Using Taylor series, we get

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

The leading term in Taylor series gives $\varepsilon_{n+1}\approx G'(r)\,\varepsilon_n$ $\varepsilon_1\approx G'(r)\,\varepsilon_0$ $\varepsilon_n\approx [G'(r)]^n\varepsilon_0$

So, the fixed-point iteration is a first order scheme, provided $G'(r)\neq 0$

The scheme converges if $|G'(r)|<1$, diverges if $|G'(r)|>1$

e.g. $x-e^{-x}=0$

The error decreases monotonically if $0\le G'(r)<1$,

The error decreases oscillatory if $-1\le G'(r)<0$.

$x_{n+1}=e^{-xn}$ Or $x_{n+1}=-\ln(x_n)$

$x^2-2x-3=0$, roots are -1, 3. $x_{n+1}=\sqrt{2x_n+3}$ $x_{n+1}=\dfrac{3}{x_n-2}$

Newton's method for nonlinear systems

$$\begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

The system of equations $\quad g_i(y_1,y_2,\ldots,y_n)=0 \quad (1\le i\le n)$
can be expressed simply as $\quad \mathbf{G}(\mathbf{Y})=\mathbf{0}$
by letting $\mathbf{Y}=(y_1,y_2,\ldots,y_n)^T$ and $\mathbf{G}=(g_1,g_2,\ldots,g_n)^T$. Using the Taylor's series expansion, we get

$$\mathbf{0}=\mathbf{G}(\mathbf{Y}+\mathbf{H}) \approx \mathbf{G}(\mathbf{Y})+\mathbf{G}'(\mathbf{Y})\mathbf{H}\ , \quad \textit{(where } \mathbf{Y}+\mathbf{H} \textit{ is more accurate solution)}$$

where $\mathbf{H}=(h_1,h_2,\ldots,h_n)^T$ and $\mathbf{G}'(\mathbf{Y})$ is the $n\times n$ Jacobian matrix $\mathbf{J}(\mathbf{Y})$:

$$\mathbf{J}(\mathbf{Y}) = \mathbf{G}'(\mathbf{Y}) = \begin{bmatrix} \partial g_1/\partial y_1 & \partial g_1/\partial y_2 & \cdots & \partial g_1/\partial y_n \\ \partial g_2/\partial y_1 & \partial g_2/\partial y_2 & \cdots & \partial g_2/\partial y_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial g_n/\partial y_1 & \partial g_n/\partial y_2 & \cdots & \partial g_n/\partial y_n \end{bmatrix}$$

$$d\mathbf{F} = \mathbf{F}'\left(\mathbf{x}^{(n)}\right)\Delta\mathbf{x}$$

$$df = \frac{\partial f}{\partial x_1}\Delta x_1 + \frac{\partial f}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial f}{\partial x_n}\Delta x_n$$

The correction vector $\mathbf{H}$ is obtained by solving linear system

$$\mathbf{J}(\mathbf{Y})\mathbf{H} = -\mathbf{G}(\mathbf{Y})$$

If Jacobian matrix is tridiagonal matrix, then $\mathbf{H}$ can be solved using Thomas algorithm. If the matrix size is 2×2, then just use the inverse of matrix $\mathbf{J}$, $\mathbf{H}=\mathbf{J}^{-1}(-\mathbf{G})$.
Finally, Newton's iteration for n nonlinear equations in n variables is given by

$$\mathbf{Y}^{(k+1)}=\mathbf{Y}^{(k)}+\mathbf{H}^{(k)} \ \rightarrow \ \mathbf{Y}^{(k+1)}=\mathbf{Y}^{(k)}-\mathbf{J}^{-1}\mathbf{G}$$

where the Jacobian system is

$$\mathbf{J}(\mathbf{Y}^{(k)})\mathbf{H}^{(k)} = -\mathbf{G}(\mathbf{Y}^{(k)}).$$

Preliminaries

Taylor's series

$$\boxed{f(x+h) = f(x) + h\frac{f'(x)}{1!} + h^2\frac{f''(x)}{2!} + \cdots + h^n\frac{f^{(n)}(x)}{n!} + R_n^*.}$$

$$R_n = \frac{f^{(n+1)}(\theta x)}{(n+1)!}(x-a)^{n+1}, \quad a < \theta x < x.$$

$$X(t+h) = X(t) + hX'(t) + \frac{h^2}{2!}X''(t) + \frac{h^3}{3!}X^{(3)}(t) + \cdots$$

$$\boxed{\begin{array}{l} X''(t) = {}^{d}\!/_{dt}\, F(t, X) \\ x_1(t),\ x_2(t), \cdots \end{array}}$$

y (scalar or a vector)	$\partial y/\partial \mathbf{x}$
$A\mathbf{x}$	A^T
$\mathbf{x}^T A$	A
$\mathbf{x}^T\mathbf{x}$	$2\mathbf{x}$
$\mathbf{x}^T A\mathbf{x}$	$A\mathbf{x}+A^T\mathbf{x}$

$$f(x+a, y+b) = f(x,y) + \frac{1}{1!}D_1[f(x,y)] + \frac{1}{2!}D_2[f(x,y)] + \cdots + \frac{1}{n!}D_n[f(x,y)] + R_n.$$

$$D_n = \left(a\frac{\partial}{\partial x} + b\frac{\partial}{\partial y}\right)^n \quad R_n = \frac{1}{(n+1)!}D_{n+1}[f(x+\theta_1 a, y+\theta_2 b)]$$

$$0 < \theta_1 < 1,\ 0 < \theta_2 < 1.$$

Newton's method for nonlinear systems

Example
$$y + x^2 - 0.5 - x = 0$$
$$x^2 - 5xy - y = 0$$
$$\text{Initial guess } \; x = 1, \;\; y = 0$$

$$\mathbf{J}(\mathbf{Y}) = \mathbf{G}'(\mathbf{Y}) = \begin{bmatrix} \partial g_1/\partial y_1 & \partial g_1/\partial y_2 & \cdots & \partial g_1/\partial y_n \\ \partial g_2/\partial y_1 & \partial g_2/\partial y_2 & \cdots & \partial g_2/\partial y_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial g_n/\partial y_1 & \partial g_n/\partial y_2 & \cdots & \partial g_n/\partial y_n \end{bmatrix}$$

$$F\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} y + x^2 - 0.5 - x \\ x^2 - 5xy - y \end{bmatrix}, \; F' = \begin{bmatrix} 2x-1 & 1 \\ 2x - 5y & -5x - 1 \end{bmatrix}, X_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Exact solution:
[x = 1.233317793, y = .2122450145],
[x = -.1781281996, y = .2901421450],
[x = -.4551895934, y = -.1623871594].

Iteration 1:

$$F\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y + x^2 - 0.5 - x \\ x^2 - 5xy - y \end{bmatrix} = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} =, \; F' = \begin{bmatrix} 2x-1 & 1 \\ 2x - 5y & -5x - 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -6 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 2 & -6 \end{bmatrix}^{-1} \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.25 \\ 0.25 \end{bmatrix}$$

Iteration 2:

$$F\begin{bmatrix} 1.25 \\ 0.25 \end{bmatrix} = \begin{bmatrix} 0.0625 \\ -0.25 \end{bmatrix} =, \; F' = \begin{bmatrix} 1.5 & 1 \\ 1.25 & -7.25 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 1.25 \\ 0.25 \end{bmatrix} - \begin{bmatrix} 1.5 & 1 \\ 1.25 & -7.25 \end{bmatrix}^{-1} \begin{bmatrix} 0.0625 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 1.2332 \\ 0.2126 \end{bmatrix}$$

Try initial guess:
$x = -0.5, 0.5$
$x = -0.5, -0.5$

$$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \mathbf{H}^{(k)} \;\rightarrow\; \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} - \mathbf{J}^{-1}\mathbf{G}$$

Newton's method for nonlinear systems

Example

$$x_1^3 - 2x_2 - 2 = 0$$

$$x_1^3 - 5x_3^2 + 7 = 0$$

$$x_2 x_3^2 - 1 = 0$$

$$\text{Initial guess } \mathbf{x} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$$

Exact solution:

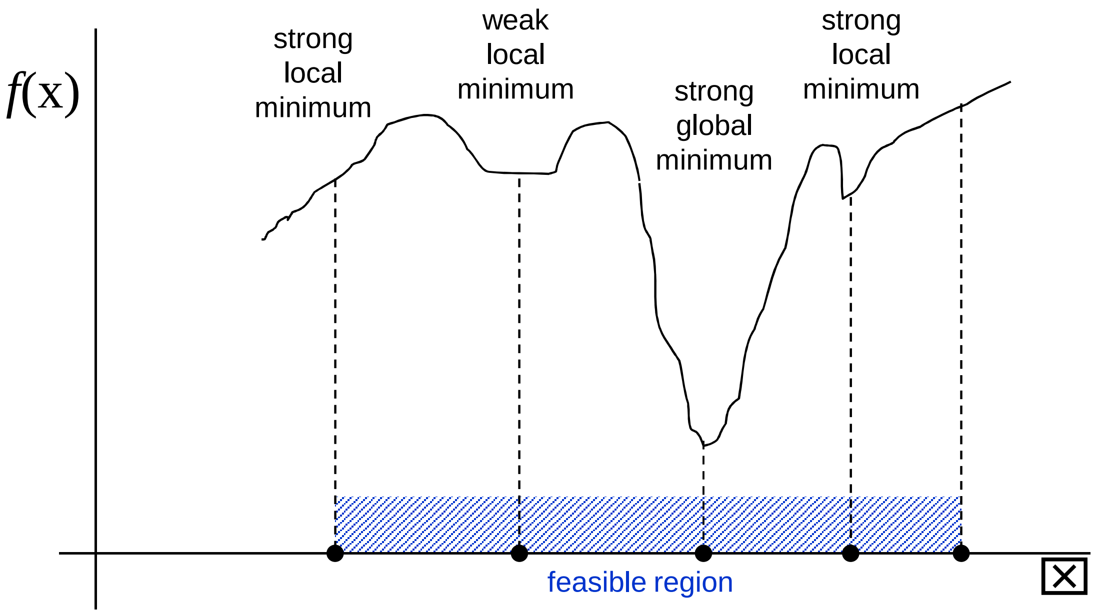
$$\mathbf{x} = \begin{bmatrix} 3^{1/3} & 0.5 & \sqrt{2} \end{bmatrix}^T$$

$$F = \begin{bmatrix} x_1^3 - 2x_2 - 2 \\ x_1^3 - 5x_3^2 + 7 \\ x_2 x_3^2 - 1 \end{bmatrix}, \quad F' = \begin{bmatrix} 3x_1^2 & -2 & 0 \\ & & \\ & & \end{bmatrix}, \quad X_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \mathbf{H}^{(k)} \rightarrow \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} - \mathbf{J}^{-1}\mathbf{F}^{(k)}$$

$$\mathbf{J}(\mathbf{Y}) = \mathbf{G}'(\mathbf{Y}) = \begin{bmatrix} \partial g_1/\partial y_1 & \partial g_1/\partial y_2 & \cdots & \partial g_1/\partial y_n \\ \partial g_2/\partial y_1 & \partial g_2/\partial y_2 & \cdots & \partial g_2/\partial y_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial g_n/\partial y_1 & \partial g_n/\partial y_2 & \cdots & \partial g_n/\partial y_n \end{bmatrix}$$

Types of minima



- which of the minima is found depends on the starting point
- such minima often occur in real applications

First-order optimality condition

Minimize $f(x)$

- For function of one variable, one can find extremum by differentiating function and setting derivative to zero
- Generalization to function of n variables is to find *critical point*, i.e., solution of nonlinear system

$$\boxed{\nabla f(\boldsymbol{x}) = \mathbf{0}}$$

  where $\nabla f(\boldsymbol{x})$ is *gradient* vector of f, whose ith component is $\partial f(\boldsymbol{x})/\partial x_i$
- For continuously differentiable $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$, any interior point x^* of S at which f has local minimum must be critical point of f
- But not all critical points are minima: they can also be maxima or saddle points

Second-order optimality condition

A Hermitian matrix which is neither positive definite, negative definite, positive-semidefinite, nor negative-semidefinite is called *indefinite*. (having both positive and negative eigenvalues).

- For twice continuously differentiable $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$, we can distinguish among critical points by considering *Hessian matrix* $\boldsymbol{H}_f(x)$ defined by

$$\{\boldsymbol{H}_f(x)\}_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

which is symmetric

Minimize $f(x)$

$$\nabla f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

$$\mathbf{H} = \nabla^2 f(\mathbf{x}) = \frac{\partial(\nabla f(\mathbf{x}))}{\partial \mathbf{x}} = \nabla(\nabla f(\mathbf{x})) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix}$$

- At critical point x^*, if $\boldsymbol{H}_f(x^*)$ is
  - positive definite, then x^* is minimum of f
  - negative definite, then x^* is maximum of f
  - indefinite, then x^* is saddle point of f
  - singular, then various pathological situations are possible

A is positive definite if
$\mathbf{x}^T A \mathbf{x} > 0$, for all $\mathbf{x}$ (all eigenvalue>0)
A is negative definite if
$\mathbf{x}^T A \mathbf{x} < 0$, for all $\mathbf{x}$ (all eigenvalue<–0)

Newton's method

Extra notes

- Another local quadratic approximation is truncated Taylor series

$$f(x+h) \approx f(x) + f'(x)h + \frac{f''(x)}{2}h^2$$

Minimize $f(x)$

- By differentiation, minimum of this quadratic function of h is given by $h = -f'(x)/f''(x)$

$$\frac{d}{dh} f(x+h) = f'(x) + hf''(x) = 0$$

- Suggests iteration scheme

$$x_{k+1} = x_k - f'(x_k)/f''(x_k)$$

which is *Newton's method* for solving nonlinear equation $f'(x) = 0$

Newton's method for finding minimum normally has quadratic convergence rate, but must be started close enough to solution to converge

Example

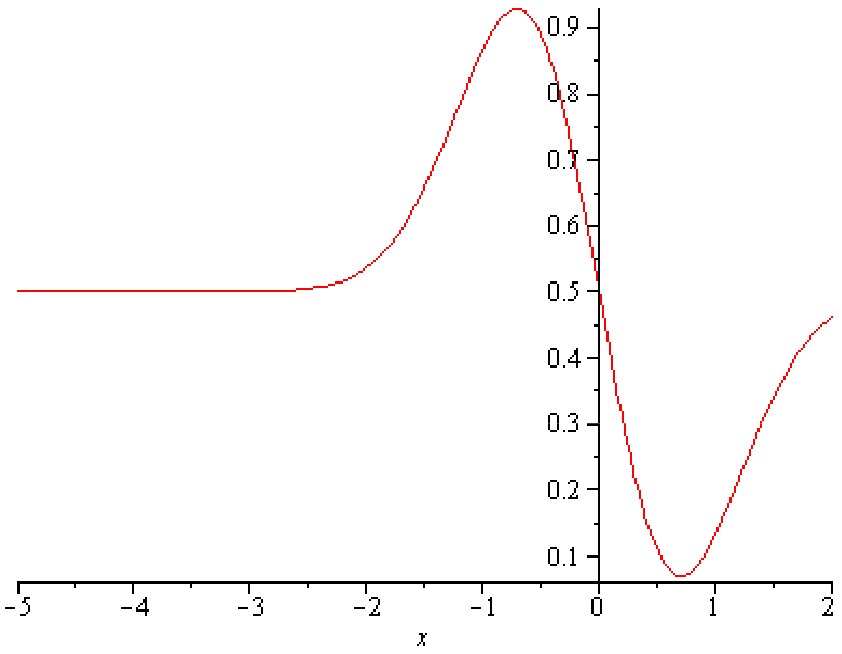
Extra notes

- Use Newton's method to minimize $f(x) = 0.5 - x\exp(-x^2)$
- First and second derivatives of f are given by

$$f'(x) = (2x^2 - 1)\exp(-x^2)$$

and

$$f''(x) = 2x(3 - 2x^2)\exp(-x^2)$$



- Newton iteration for zero of f' is given by

$$x_{k+1} = x_k - (2x_k^2 - 1)/(2x_k(3 - 2x$$

- Using starting guess $x_0 = 1$, we obtain

x_k	$f(x_k)$
1.000	0.132
0.500	0.111
0.700	0.071
0.707	0.071

Newton method

Extra notes

Expand $f(\mathbf{x})$ by its Taylor series about the point $\mathbf{x}_k$

$$f(\mathbf{x}_k + \delta\mathbf{x}) \approx f(\mathbf{x}_k) + \mathbf{g}_k^T \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T \mathbf{H}_k \delta\mathbf{x}$$

where the gradient is the vector

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k) = \left[\frac{\partial f}{x_1} \ldots \frac{\partial f}{x_N}\right]^T$$

and the Hessian is the symmetric matrix

$$\nabla f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

$$\mathbf{H}_k = \mathbf{H}(\mathbf{x}_k) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix}$$

$$\mathbf{H} = \nabla^2 f(\mathbf{x}) = \frac{\partial(\nabla f(\mathbf{x}))}{\partial \mathbf{x}} = \nabla(\nabla f(\mathbf{x})) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix}$$

Newton method

Extra notes

For a minimum we require that $\nabla f(\mathbf{x}) = \mathbf{0}$, and so

$$f(\mathbf{x}_k + \delta\mathbf{x}) \approx f(\mathbf{x}_k) + \mathbf{g}_k^T \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T \mathbf{H}_k \delta\mathbf{x}$$

$$\nabla f(\mathbf{x}) = \mathbf{g}_k + \mathbf{H}_k \delta\mathbf{x} = \mathbf{0}$$

$$\frac{\partial f(\mathbf{x}_k + \delta\mathbf{x})}{\partial \delta\mathbf{x}}$$

y (scalar or a vector)	$\partial y / \partial \mathbf{x}$
$A\mathbf{x}$	A^T
$\mathbf{x}^T A$	A
$\mathbf{x}^T \mathbf{x}$	$2\mathbf{x}$
$\mathbf{x}^T A \mathbf{x}$	$A\mathbf{x} + A^T\mathbf{x}$

with solution $\delta\mathbf{x} = -\mathbf{H}_k^{-1}\mathbf{g}_k$. This gives the iterative update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{g}_k$$

- If $f(\mathbf{x})$ is quadratic, then the solution is found in one step.
- The method has quadratic convergence (as in the 1D case).
- The solution $\delta\mathbf{x} = -\mathbf{H}_k^{-1}\mathbf{g}_k$ is guaranteed to be a downhill direction.
- Rather than jump straight to the minimum, it is better to perform a line minimization which ensures global convergence

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1}\mathbf{g}_k$$

- If $\mathbf{H}=\mathbf{I}$ then this reduces to steepest descent.

Steepest descent

- Basic principle is to minimize the N-dimensional function by a series of 1D line-minimizations:

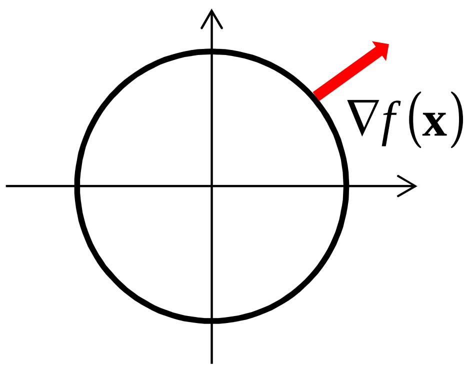
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

- The steepest descent method chooses $\mathbf{p}_k$ to be parallel to the gradient

$$\mathbf{p}_k = -\nabla f(\mathbf{x}_k)$$

Try with $f=x^2+y^2$

- Step-size α_k is chosen to minimize $f(\mathbf{x}_k + \alpha_k\mathbf{p}_k)$.
  For quadratic forms there is a closed form solution:

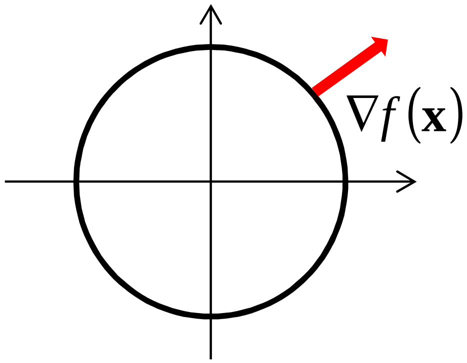


$$\alpha_k = \frac{\mathbf{p}_k^T\mathbf{p}_k}{\mathbf{p}_k^T\mathbf{H}\mathbf{p}_k}$$

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T H\mathbf{x} - \mathbf{b}^T\mathbf{x}$$

Steepest descent

Try with
$f=x^2+y^2$



Let $\mathbf{x}_k=[1\ 0]^T$, $\nabla f=2x\mathbf{i}+2y\mathbf{j}$, $\mathbf{z}_k= -\nabla f/|\nabla f|$,
Minimize $f(\mathbf{x}_k+\alpha\mathbf{z}_k)$

Let $\alpha_0=0$, $\alpha_1=0.2$, $\alpha_2=0.4$, we get

$$f_1 = f(\mathbf{x}_k - \alpha_1 \mathbf{z}_k) = f\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = 1,$$

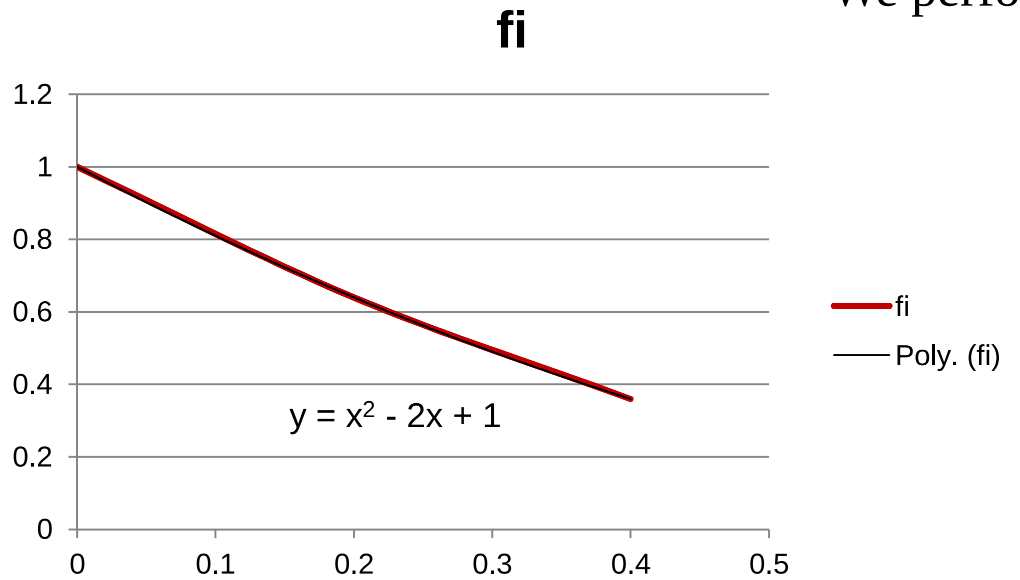
$$f_2 = f(\mathbf{x}_k - \alpha_2 \mathbf{z}_k) = f\left(\begin{bmatrix}0.8\\0\end{bmatrix}\right) = 0.64,$$

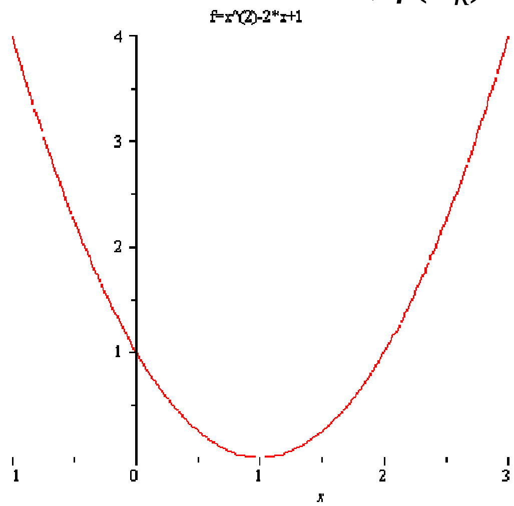
$$f_3 = f(\mathbf{x}_k - \alpha_3 \mathbf{z}_k) = f\left(\begin{bmatrix}0.6\\0\end{bmatrix}\right) = 0.36,$$

i	**1**	**2**	**3**
α_i	0	0.2	0.4
f_i	1	0.64	0.36

We perform interpolation using Ms Excel

So, $f(\mathbf{x}_k)$ is minimum when α is 1





Steepest descent method

- Let $f: \mathbb{R}^n \to \mathbb{R}$ be real-valued function of n real variables
- At any point x where gradient vector is nonzero, negative gradient, $-\nabla f(x)$, points downhill toward lower values of f
- In fact, $-\nabla f(x)$ is locally direction of steepest descent: f decreases more rapidly along direction of negative gradient than along any other
- *Steepest descent* method: starting from initial guess x_0, successive approximate solutions given by

$$\boxed{x_{k+1} = x_k - \alpha_k \nabla f(x_k)}$$

where α_k is *line search* parameter that determines how far to go in given direction

Try with $f = x^2 + y^2$

Steepest Descent

Problem

$$\min_x f(x)$$

- Steepest descent algorithm:

Data: $x_0 \in R^n$

Step 0: set i=0

Step 1: if $\nabla f(x_i) = 0$ stop,

else, compute ***search direction*** $h_i = -\nabla f(x_i)$

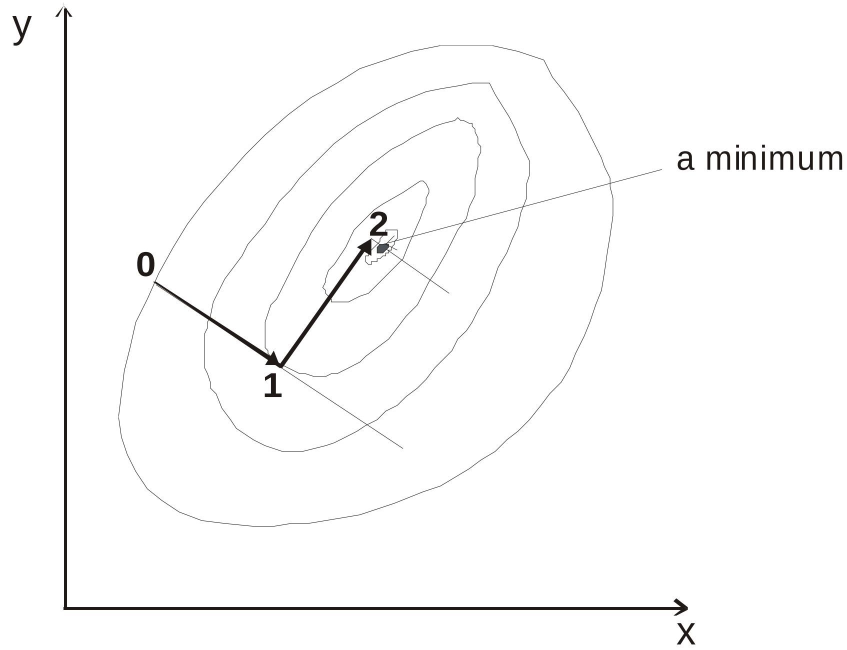
Step 2: compute the ***step-size***

$$\alpha_i \in \arg\min_{\alpha \geq 0} f(x_i + \alpha \cdot h_i)$$

Step 3: set $x_{i+1} = x_i + \alpha_i \cdot h_i$ go to step 1

Steepest Descent

steepest descent method to find the minimum can be applied to solve a system of nonlinear equations $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$



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

$$g(x_1, \cdots, x_n) = \sum_{i=1}^{n} [f_i(x_1, \cdots, x_n)]^2$$

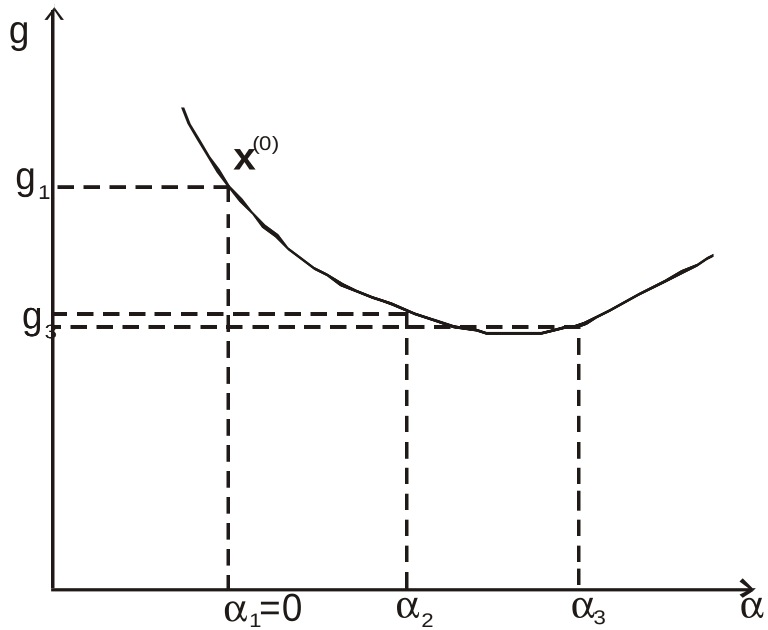
$$g = \boldsymbol{f} \times \boldsymbol{f}^{\mathrm{T}}$$

where $\boldsymbol{f} = [f_1 \ f_2 \ \ldots \ f_n]$ and $\boldsymbol{f}^{\mathrm{T}}$ is the transpose of $\boldsymbol{f}$

Where, we get $\min_{\mathbf{x}} g(\mathbf{x}) = 0$

Steepest Descent

1. Evaluate g at an initial approximation $\boldsymbol{x}^{(0)}$.
2. Determine a direction from $\boldsymbol{x}^{(0)}$ using the gradient of g.
3. Move to a new appropriate position $\boldsymbol{x}^{(1)}$ so that $g(\boldsymbol{x}^{(1)}) < g(\boldsymbol{x}^{(0)})$.
4. Repeat steps 2-4 with $\boldsymbol{x}^{(0)}$ replaced by $\boldsymbol{x}^{(1)}$.

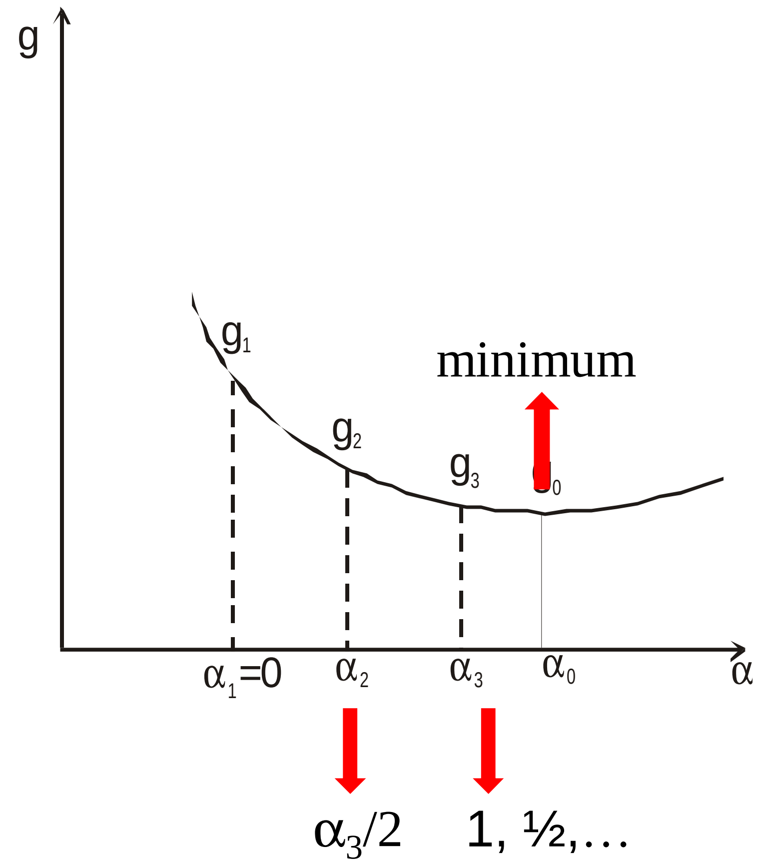


The direction of greatest decrease in the value of g at $\boldsymbol{x}$ is the direction given by $-\nabla g(\boldsymbol{x})$, therefore $\boldsymbol{x}^{(1)}$ is given by

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \alpha \nabla g(\boldsymbol{x})$$

where α is to be determined

Steepest Descent



The procedure is to fit a quadratic to three points (α_1, g_1), (α_2, g_2), and (α_3, g_3) then choose α so that g is a minimum in the direction of steepest descent. The steps are

a) Let $\alpha_1 = 0$ at $\boldsymbol{x}^{(0)}$, therefore $g_1 = g(\boldsymbol{x}^{(0)})$.
b) Let $\alpha_3 = 1$ and evaluate $g_3 = g(\boldsymbol{x}^{(0)} - \alpha_3 \boldsymbol{z})$
c) If $g_3 > g_1$, let $\alpha_3 = \alpha_3/2$ and repeat step (b)
d) Let $\alpha_2 = \alpha_3/2$ and evaluate $g_2 = g(\boldsymbol{x}^{(0)} - \alpha_2 \boldsymbol{z})$

The quadratic through three points (α_1, g_1), (α_2, g_2), and (α_3, g_3) has the form

$$P(\alpha) = g_1 + h_1\alpha + h_3\alpha(\alpha - \alpha_2) = g_1 + g^1{}_1\alpha + g^2{}_1\alpha(\alpha - \alpha_2)$$

where

$$h_1 = \frac{g_2 - g_1}{\alpha_2 - \alpha_1}, \quad h_2 = \frac{g_3 - g_2}{\alpha_3 - \alpha_2}, \quad h_3 = \frac{h_2 - h_1}{\alpha_3 - \alpha_1}$$

$$g_1^1 = \frac{g_2 - g_1}{\alpha_2 - \alpha_1}, \; g_2^1 = \frac{g_3 - g_2}{\alpha_3 - \alpha_2}, \quad g_1^2 = \frac{g_2^1 - g_1^1}{\alpha_3 - \alpha_1}$$

$$z_0 = \left\| \nabla g\left(\mathbf{x}^{(0)}\right) \right\|_2$$

$$\mathbf{z} = \frac{1}{z_0} \nabla g\left(\mathbf{x}^{(0)}\right) = \nabla g\left(\mathbf{x}^{(0)}\right) \Big/ \left\| \nabla g\left(\mathbf{x}^{(0)}\right) \right\|_2$$

At the location where

$$\frac{dP}{d\alpha} = h_1 + 2h_3\alpha - h_3\alpha_2 = 0 \;\rightarrow \alpha_0 = 0.5\left(\alpha_2 - \frac{h_1}{h_3}\right) = 0.5\left(\alpha_2 - \frac{g_1^1}{g_1^2}\right)$$

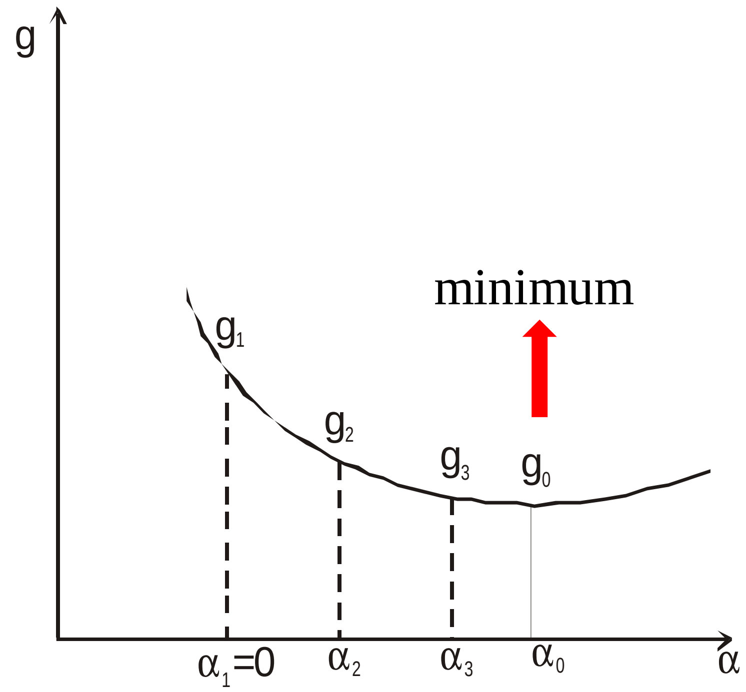
Steepest Descent

e) Evaluate $g_0 = g(\boldsymbol{x}^{(0)} - \alpha_0 \boldsymbol{z})$

Since a quadratic through three points can have a minimum or a maximum as shown in Fig below, α is chosen so that g is the lowest value between g_0 and g_3 as follows

f) If $g_0 < g_3$ then $\alpha = \alpha_0$ else $\alpha = \alpha_3$

4. Repeat steps 2-4 with $\boldsymbol{x}^{(0)}$ replaced by $\boldsymbol{x}^{(1)}$.



a) Let $\alpha_1 = 0$ at $\boldsymbol{x}^{(0)}$, therefore $g_1 = g(\boldsymbol{x}^{(0)})$.

b) Let $\alpha_3 = 1$ and evaluate $g_3 = g(\boldsymbol{x}^{(0)} - \alpha_3 \boldsymbol{z})$

c) If $g_3 > g_1$, let $\alpha_3 = \alpha_3/2$ and repeat step (b)

d) Let $\alpha_2 = \alpha_3/2$ and evaluate $g_2 = g(\boldsymbol{x}^{(0)} - \alpha_2 \boldsymbol{z})$

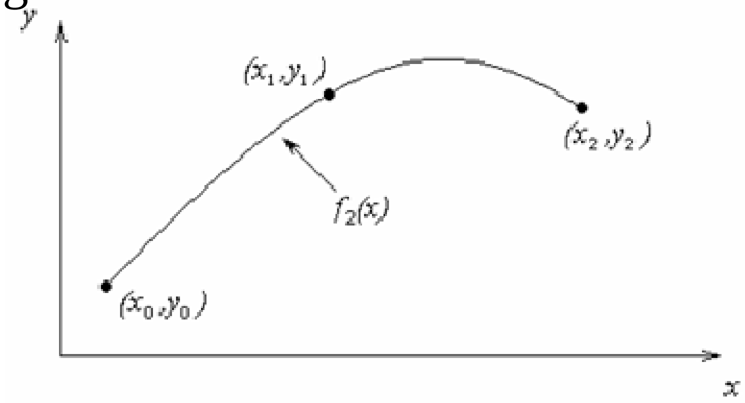
1. Evaluate g at an initial approximation $\boldsymbol{x}^{(0)}$.
2. Determine a direction from $\boldsymbol{x}^{(0)}$ using the gradient of g.
3. Move to a new appropriate position $\boldsymbol{x}^{(1)}$ so that $g(\boldsymbol{x}^{(1)}) < g(\boldsymbol{x}^{(0)})$.
4. Repeat steps 2-4 with $\boldsymbol{x}^{(0)}$ replaced by $\boldsymbol{x}^{(1)}$.

$$\mathbf{z} = \frac{1}{z_0}\nabla g\left(\mathbf{x}^{(0)}\right) = \nabla g\left(\mathbf{x}^{(0)}\right) \Big/ \left\|\nabla g\left(\mathbf{x}^{(0)}\right)\right\|_2$$

Steepest Descent

Newton's forward divided-difference interpolation

Given (x_0,y_0), (x_1,y_1) and (x_2,y_2), fit a quadratic interpolant through the data



$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

$$b_0 = f(x_0)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

$$b_0 = f[x_0] = f(x_0)$$

$$b_1 = f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$b_2 = f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

General form

$$f(x) = f[x_0] + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1)$$

alternative form

$$f(x) = f_0 + f_0^{[1]}(x - x_0) + f_0^{[2]}(x - x_0)(x - x_1)$$

where

$$f(x_i) = f_i, \quad f_i^{[0]} = f_i, \quad f_i^{[j]} = \frac{f_{i+1}^{[j-1]} - f_i^{[j-1]}}{x_{i+j} - x_i}$$

Steepest Descent

Use the method of steepest descent with the initial guess $\boldsymbol{x}$ = [0 0 0] to obtain the solutions to the following equations

$$f_1(x_1, x_2, x_3) = 3x_1 - \cos(x_2\, x_3) - ½ = 0$$
$$f_2(x_1, x_2, x_3) = x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0$$
$$f_3(x_1, x_2, x_3) = e^{-x_1 x_2} + 20x_3 + (10\pi - 3)/3 = 0$$

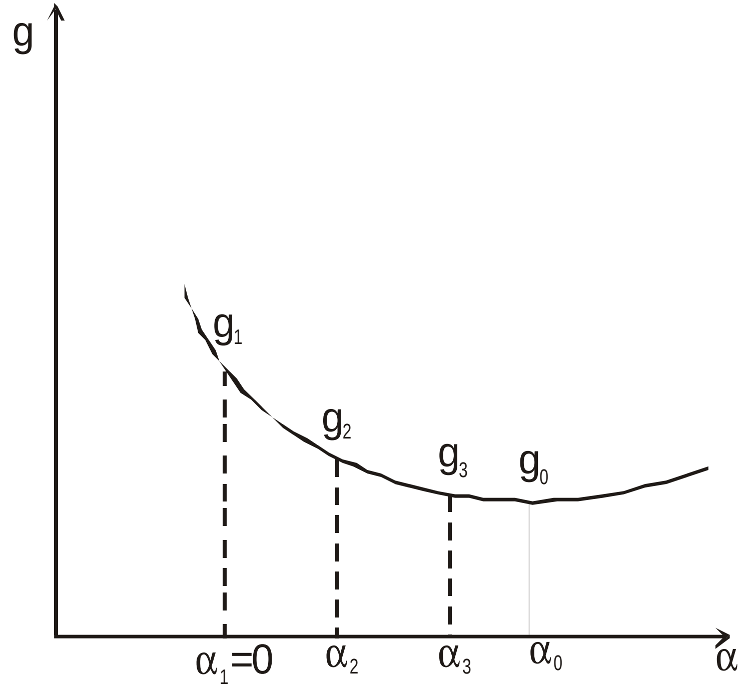
Exact, $\mathbf{x}=(0.5,0,-0.5235988)^T$

$$\boldsymbol{x}^{(0)} = [0\ 0\ 0];\ \boldsymbol{f} = [f_1\ \ f_2\ \ f_3]$$
$$g = \boldsymbol{f}\times\boldsymbol{f}^{T} = f_1^2 + f_2^2 + f_3^2 = 111.975$$

$$\nabla g(\mathbf{x}) = \frac{\partial g}{\partial \mathbf{x}} = \begin{pmatrix} \dfrac{\partial g(\mathbf{x})}{\partial x_1} \\ \vdots \\ \dfrac{\partial g(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

$$\mathbf{H} = \nabla^2 g(\mathbf{x}) = \frac{\partial(\nabla g(\mathbf{x}))}{\partial \mathbf{x}} = \nabla(\nabla g(\mathbf{x})) = \begin{bmatrix} \dfrac{\partial^2 g(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \dfrac{\partial^2 g(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial^2 g(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \dfrac{\partial^2 g(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix}$$

Steepest Descent



$f_1(x_1, x_2, x_3) = 3x_1 - \cos(x_2\, x_3) - ½ = 0$
$f_2(x_1, x_2, x_3) = x_1^{\,2} - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0$
$f_3(x_1, x_2, x_3) = e^{-x_1 x_2} + 20x_3 + (10\pi - 3)/3 = 0$

Exact, $\mathbf{x}$=(0.5,0,–0.5235988)^T

$$\nabla g(\mathbf{x}) = \frac{\partial g}{\partial \mathbf{x}} = \begin{pmatrix} \dfrac{\partial g(\mathbf{x})}{\partial x_1} \\ \vdots \\ \dfrac{\partial g(\mathbf{x})}{\partial x_n} \end{pmatrix} = \begin{bmatrix} 2f_1 \cdot 3 + 2f_2 \cdot 2x_1 + 2f_3 \cdot e^{-x_1 x_2}(-x_2) \\ 2f_1 \cdot (\sin(x_2 x_3)x_3) + 2f_2 \cdot (-162(x_2 + 0.1)) + 2f_3 \cdot e^{-x_1 x_2}(-x_1) \\ 2f_1 \cdot (\sin(x_2 x_3)x_2) + 2f_2 \cdot (\cos x_3) + 2f_3 \cdot 20 \end{bmatrix}$$

$\boldsymbol{x}^{(0)} = [0\ 0\ 0];$
$g(\boldsymbol{x}^{(0)}) = f_1^{\,2} + f_2^{\,2} + f_3^{\,2} = 111.975 \qquad z_0 = \left\|\nabla g\left(\mathbf{x}^{(0)}\right)\right\|_2 = 419.554$

$$f_2(x) = f[x_0] + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1)$$

$$b_0 = f[x_0] = f(x_0),$$
$$b_1 = f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0},$$
$$b_2 = f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0}.$$

Let $\mathbf{z} = \dfrac{1}{z_0}\nabla g\left(\mathbf{x}^{(0)}\right) = (-0.0214514, -0.0193062, 0.999583)^T$

With α_1=0, $g_1 = g\left(\mathbf{x}^{(0} - \alpha_1 \mathbf{z}\right) = g\left(\mathbf{x}^{(0}\right) = 111.975$ Now, let α_3=1,

$g_3 = g\left(\mathbf{x}^{(0} - \alpha_3 \mathbf{z}\right) = 93.5649$

Since, $g_3 < g_1$, we accept α_3, set $\alpha_2 = \alpha_3/2 = 0.5$

$g_2 = g\left(\mathbf{x}^{(0} - \alpha_2 \mathbf{z}\right) = 2.53557$

$$\frac{dP}{d\alpha} = h_1 + 2h_3\alpha - h_3\alpha_2 = 0 \ \rightarrow \alpha_0 = 0.5\left(\alpha_2 - \frac{h_1}{h_3}\right) = 0.5\left(\alpha_2 - \frac{g_1^1}{g_1^2}\right)$$

Find quadratic polynomial that interpolate data (0,111.975), (0.5, 2.53557) &(1,93.5649). We use Newton's forward divided-difference interpolation

$$P(\alpha) = g_1 + h_1\alpha + h_3\alpha(\alpha - \alpha_2) = g_1 + g^1{}_1\alpha + g^2{}_1\,\alpha(\alpha - \alpha_2)$$

Steepest Descent

$P(\alpha) = g_1 + h_1\alpha + h_3\alpha(\alpha - \alpha_2) = g_1 + g^1{}_1\alpha + g^2{}_1\,\alpha(\alpha - \alpha_2)$

interpolate data (0,111.975), (0.5, 2.53557) & (1,93.5649).

$$\alpha_1 = 0, \quad g_1 = 111.975,$$

$$\alpha_2 = 0.5, \quad g_2 = 2.53557, h_1 = \frac{g_2 - g_1}{\alpha_2 - \alpha_1} = -218.878,$$

$$\alpha_3 = 1, \quad g_3 = 93.5649, h_2 = \frac{g_3 - g_2}{\alpha_3 - \alpha_2} = 182.059, \quad h_3 = \frac{h_2 - h_1}{\alpha_3 - \alpha_1} = 400.937$$

x =[0.011218 0.010096 -0.522741], g = 2.327617
x =[0.137860 -0.205453 -0.522059], g = 1.274058
x =[0.266959 0.005511 -0.558494], g = 1.068131
x =[0.272734 -0.008118 -0.522006], g = 0.468309
x =[0.308689 -0.020403 -0.533112], g = 0.381087
x =[0.314308 -0.014705 -0.520923], g = 0.318837
x =[0.324267 -0.008525 -0.528431], g = 0.287024
x =[0.330809 -0.009678 -0.520662], g = 0.261579
x =[0.339809 -0.008592 -0.528080], g = 0.238486
x =[0.345746 -0.009034 -0.520941], g = 0.217440

$$\Rightarrow \quad P(\alpha) = g_1 + h_1\alpha + h_3\alpha(\alpha - \alpha_2) = 111.975 - 218.878\alpha + 400.937\alpha(\alpha - 0.5)$$

We have $P'(\alpha)=0$ when $\alpha=\alpha_0=0.522959$. Since $g_0=g(\mathbf{x}^{(0)}-\alpha_0\mathbf{z})=2.32762$ is smaller than g_1 and g_3, we set

$\mathbf{x}^{(1)}=\mathbf{x}^{(0)}-\alpha_0\mathbf{z}=\mathbf{x}^{(0)}-0.522959\mathbf{z}=(0.0112182,0.0100964,-0.522741)^T$ We get, $g(\mathbf{x}^{(1)})=2.32762$

$$\frac{dP}{d\alpha} = h_1 + 2h_3\alpha - h_3\alpha_2 = 0 \rightarrow \alpha_0 = 0.5\left(\alpha_2 - \frac{h_1}{h_3}\right) = 0.5\left(\alpha_2 - \frac{g_1^1}{g_1^2}\right)$$

Exact, $\mathbf{x}=(0.5,0,-0.5235988)^T$

$$f_2(x) = f[x_0] + f[x_1,x_0](x-x_0) + f[x_2,x_1,x_0](x-x_0)(x-x_1)$$

$$f[x_0] = f(x_0),$$
$$f[x_1,x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0},$$
$$f[x_2,x_1,x_0] = \frac{f[x_2,x_1] - f[x_1,x_0]}{x_2 - x_0}.$$

$$f(x) = f_0 + f_0^{[1]}(x-x_0) + f_0^{[2]}(x-x_0)(x-x_1)$$

$$f(x_i) = f_i, \quad f_i^{[0]} = f_i, \quad f_i^{[j]} = \frac{f_{i+1}^{[j-1]} - f_i^{[j-1]}}{x_{i+j} - x_i}$$

Ordinary differential equations (ODEs)

Finite difference method for linear second-order boundary value problem

General form: $y'' + p(x)y' + q(x)y = r(x),\ \ a \le x \le b,\ y(a)=\alpha,\ y(b)=\beta.$ $\quad(a)$

Centered-difference formula for second derivative:

$$y''(x_i) = \frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] - \frac{h^2}{12}y^{(4)}(\xi_i), \quad x_{i-1} < \xi_i < x_{i+1}.$$

Centered-difference formula for first derivative:

$$y'(x_i) = \frac{1}{2h}\left[y(x_{i+1}) - y(x_{i-1})\right] - \frac{h^2}{6}y^{(3)}(\xi_i), \quad x_{i-1} < \xi_i < x_{i+1}.$$

Let divide the interval $[a,b]$ into N equal subintervals where $x_0=a$, $x_i=x_0+ih$, $\{i=1,2,\ldots,N\}$, $x_N=b$ and $h=(b-a)/N$.
At point $x=x_i$, equation (a) becomes

$$y_i'' + p_i y_i' + q_i y_i = r_i,$$

Using centered-difference formula, we get

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i\frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = r_i\,,\ (\times h^2) \rightarrow (y_{i+1} - 2y_i + y_{i-1}) + p_i\frac{h}{2}(y_{i+1} - y_{i-1}) + h^2 q_i y_i = h^2 r_i$$

$$\left(1 - \frac{h}{2}p_i\right)y_{i-1} - \left(2 - h^2 q_i\right)y_i + \left(1 + \frac{h}{2}p_i\right)y_{i+1} = h^2 r_i.$$

For $i=1,2,\ldots,N-1$, the above equation will produce system $(N-1)$ equations with unknowns $y_0, y_1, \ldots, y_N$.
With the given boundary condition, $y_0=\alpha$ and $y_N=\beta$, the system can be solved for $y_1, y_2, \ldots, y_{N-1}$ in matrix form, $\mathbf{A}\mathbf{y}=\mathbf{b}$ (where matrix $\mathbf{A}$ is tridiagonal matrix with diagonally dominant, $|a_{ii}| > \sum |a_{ij}|$, $j=1\ldots n$, $j \ne n$, row direction).
Tridiagonal system, $\mathbf{A}\mathbf{y}=\mathbf{b}$, can be solved using Thomas algorithm.

Ordinary differential equations (ODEs)

Finite difference method for linear second-order boundary value problem

Thomas algorithm

For tridiagonal system with size $n \times n$, $\mathbf{Ax}=\mathbf{b}$, matrix $\mathbf{A}$ can be factored into $\mathbf{A}=\mathbf{LU}$, where $\mathbf{L}$ (lower triangular Matrix) and $\mathbf{U}$(upper triangular matrix) as below:

$$A = LU \rightarrow \begin{pmatrix} d_1 & e_1 & 0 & \cdots & 0 \\ c_2 & d_2 & e_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & c_{n-1} & d_{n-1} & e_{n-1} \\ 0 & \cdots & 0 & c_n & d_n \end{pmatrix} = \begin{pmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ c_2 & \alpha_2 & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & c_{n-1} & \alpha_{n-1} & 0 \\ 0 & \cdots & 0 & c_n & \alpha_n \end{pmatrix} \cdot \begin{pmatrix} 1 & \beta_1 & 0 & \cdots & 0 \\ 0 & 1 & \beta_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & 1 & \beta_{n-1} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

According to the above system, α_i and β_i can be calculated as

$\alpha_1=d_1$; $\alpha_i=d_i-c_i\beta_{i-1}$, $i=2,3,\ldots,n$; $\beta_i=e_i/\alpha_i$, $i=1,2,\ldots,n-1$.

The system $\mathbf{Ax}=\mathbf{b}$ can be factorized as $\mathbf{LUx}=\mathbf{b}$, by letting $\mathbf{w}=\mathbf{Ux}$, we get $\mathbf{Lw}=\mathbf{b}$, then

(1) Solve $\mathbf{Lw}=\mathbf{b}$ by forward substitution, we get

$$w_1=b_1/\alpha_1, \qquad w_i=(b_i-c_iw_{i-1})/\alpha_i,\ i=2,3,\ldots,n.$$

(2)Solve $\mathbf{Ux}=\mathbf{w}$ by backward substitution, we get

$$x_n=w_n, \qquad x_i=w_i-\beta_i x_{i+1},\ i=n-1,\ n-2,\ldots,1.$$

The whole Thomas algorithm can be summarized as:

1. $\alpha_1=d_1$
2. $\alpha_i=d_i-c_i\beta_{i-1}$, $i=2,3,\ldots,n$
3. $\beta_i=e_i/\alpha_i$, $i=1,2,\ldots,n-1$.
4. $w_1=b_1/\alpha_1$
5. $w_i=(b_i-c_iw_{i-1})/\alpha_i$, $i=2,3,\ldots,n$.
6. $x_n=w_n$
7. $x_i=w_i-\beta_i x_{i+1}$, $i=n-1,\ n-2,\ldots,1$.

Ordinary differential equations (ODEs)

Finite difference method for linear second-order boundary value problem

Solve the linear boundary value problem

$$\boldsymbol{y'' + (1/x)y' - (1/x^2)y = 3, \quad y(1)=2, \; y(2)=3}$$

for x=1(0.2)2 using finite difference method. Analytical solution: $y(x)=x(x-1)+2/x$.

Let h=0.2, $x_0=a=1$, $x_1=1.2$, $x_2=1.4$, $x_3=1.6$, $x_4=1.8$ and $x_5=b=2$. Find $y_i \approx y(x_i)$, i=1,2,3,4.

At x_i, we get

$$y_i'' + \left(\frac{1}{x_i}\right)y_i' - \left(\frac{1}{x_i^2}\right)y_i = 3 \;\rightarrow\; \left(\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}\right) + \left(\frac{1}{x_i}\right)\left(\frac{y_{i+1} - y_{i-1}}{2h}\right) - \left(\frac{1}{x_i^2}\right)y_i = 3$$

Multiply with h^2, here we use 4 decimal place.

$$(y_{i+1} - 2y_i + y_{i-1}) + \frac{h}{2}\left(\frac{1}{x_i}\right)(y_{i+1} - y_{i-1}) - h^2\left(\frac{1}{x_i^2}\right)y_i = 3h^2 \;\rightarrow\; \left(1 - \frac{0.1}{x_i}\right)y_{i-1} - \left[2 + \left(\frac{0.2}{x_i}\right)^2\right]y_i + \left(1 + \frac{0.1}{x_i}\right)y_{i+1} = 0.12$$

For i=1,

$$\left(1 - \frac{0.1}{x_1}\right)y_0 - \left[2 + \left(\frac{0.2}{x_1}\right)^2\right]y_1 + \left(1 + \frac{0.1}{x_1}\right)y_2 = 0.12 \;\rightarrow\; \left(1 - \frac{0.1}{1.2}\right)2 - \left[2 + \left(\frac{0.2}{1.2}\right)^2\right]y_1 + \left(1 + \frac{0.1}{1.2}\right)y_2 = 0.12$$

$$\rightarrow \quad -2.0278y_1 + 1.0833y_2 = -1.7133$$

For i=2,

$$\left(1 - \frac{0.1}{x_2}\right)y_1 - \left[2 + \left(\frac{0.2}{x_2}\right)^2\right]y_2 + \left(1 + \frac{0.1}{x_2}\right)y_3 = 0.12 \;\rightarrow\; \left(1 - \frac{0.1}{1.4}\right)y_1 - \left[2 + \left(\frac{0.2}{1.4}\right)^2\right]y_2 + \left(1 + \frac{0.1}{1.4}\right)y_3 = 0.12$$

$$\rightarrow \quad 0.9286y_1 - 2.0204y_2 + 1.0714y_3 = 0.12$$

For i=3,

$$\left(1 - \frac{0.1}{x_3}\right)y_2 - \left[2 + \left(\frac{0.2}{x_3}\right)^2\right]y_3 + \left(1 + \frac{0.1}{x_3}\right)y_4 = 0.12 \;\rightarrow\; \left(1 - \frac{0.1}{1.6}\right)y_2 - \left[2 + \left(\frac{0.2}{1.6}\right)^2\right]y_3 + \left(1 + \frac{0.1}{1.6}\right)y_4 = 0.12$$

$$\rightarrow \quad 0.9375y_2 - 2.0156y_3 + 1.0625y_4 = 0.12$$

For i=4,

$$\left(1 - \frac{0.1}{x_4}\right)y_3 - \left[2 + \left(\frac{0.2}{x_4}\right)^2\right]y_4 + \left(1 + \frac{0.1}{x_4}\right)y_5 = 0.12 \;\rightarrow\; \left(1 - \frac{0.1}{1.8}\right)y_3 - \left[2 + \left(\frac{0.2}{1.8}\right)^2\right]y_4 + \left(1 + \frac{0.1}{1.8}\right)(3) = 0.12$$

$$\rightarrow \quad 0.9444y_3 - 2.0123y_4 = -3.0468$$

Ordinary differential equations (ODEs)

Finite difference method for linear second-order boundary value problem

Finally, we get the tridiagonal system as below:

$$\mathbf{Ay}=\mathbf{b} \quad \rightarrow \quad \begin{pmatrix} -2.0278 & 1.0833 & 0 & 0 \\ 0.9286 & -2.0204 & 1.0714 & 0 \\ 0 & 0.9375 & -2.0156 & 1.0625 \\ 0 & 0 & 0.9444 & -2.0123 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}=\begin{pmatrix} -1.7133 \\ 0.1200 \\ 0.1200 \\ -3.0468 \end{pmatrix}$$

Using Thomas algorithm, we get

i	**1**	**2**	**3**	**4**
d_i e_i c_i b_i	-2.0278 - 1.0833 -1.7133	-2.0204 0.9286 1.0714 0.1200	-2.0156 0.9375 1.0625 0.1200	-2.0123 0.9444 - -3.0468
$(\alpha_1=d_1)$ $\alpha_i=d_i-c_i\beta_{i-1}$,	-2.0278	-1.5243	-1.3566	-1.2726
$\beta_i=e_i/\alpha_i$,	-0.5342	-0.7029	-0.7832	-
$(w_1=b_1/\alpha_1)$ $w_i=(b_i-c_iw_{i-1})/\alpha_i$,	0.8449	0.4360	0.2128	2.5521
$(y_n=w_n)$ $y_i=w_i-\beta_i y_{i+1}$,	1.9082	1.9905	2.2116	2.5521

Finally, we get $y(1.2)\approx y_1=1.9082$, $y_2=1.9905$, $y_3=2.2116$ and $y(1.8)\approx y_4=2.5521$.
The exact solution is given as $y(1.2)=1.9067$, $y(1.4)=1.9886$, $y(1.6)=2.2100$, $y(1.8)=2.5511$.
So, finite difference method produce results accurate up to 2 decimal places.

Ordinary differential equations (ODEs)

Finite difference method for nonlinear second-order boundary value problem

For the general nonlinear boundary value problem

$$\boldsymbol{y''=f(x,y,y'), \qquad a\le x\le b,\ y(a)=\alpha,\ y(b)=\beta,} \qquad (a)$$

Let divide the interval $[a,b]$ into N equal subintervals where $x_0=a$, $x_i=x_0+ih$, $\{i=1,2,\ldots,N\}$, $x_N=b$ and $h=(b-a)/N$.

At point $x=x_i$, equation (a) becomes

$$-\frac{y_{i+1}-2y_i+y_{i-1}}{h^2}+f\left(x_i,y_i,\frac{y_{i+1}-y_{i-1}}{2h}\right)=0 \quad \to \quad \boxed{-y_{i-1}+2y_i-y_{i+1}+h^2 f\left(x_i,y_i,\frac{y_{i+1}-y_{i-1}}{2h}\right)=0=f_i\left(y_1,\cdots,y_{N-1}\right)}$$

For $i=1,2,\ldots,N$-1, the above equation will produce nonlinear system $(N$-1) equations with unknowns $y_0,y_1,\ldots,y_N$. The above nonlinear system has a unique solution if $h<2/L$, $L=\max|f_{y'}(x,y,y')|$. With the given boundary condition, $y_0=\alpha$ and $y_N=\beta$, the system can be solved by Newton's method for nonlinear systems. A sequence of iteration will converge to solution if the guess initial approximation is sufficiently close to solution.

The Jacobian matrix, $J(y_1,\ldots,y_{N-1})$ is tridiagonal with ij-th entry:

$$J\left(y_1,\cdots,y_{N-1}\right)_{ij}=\begin{cases}-1+\dfrac{h}{2}f_{y'}\left(x_i,y_i,\dfrac{y_{i+1}-y_{i-1}}{2h}\right), & \text{for } i=j-1 \text{ and } j=2,\cdots,N-1,\\ 2+h^2 f_y\left(x_i,y_i,\dfrac{y_{i+1}-y_{i-1}}{2h}\right), & \text{for } i=j \text{ and } j=1,\cdots,N-1,\\ -1-\dfrac{h}{2}f_{y'}\left(x_i,y_i,\dfrac{y_{i+1}-y_{i-1}}{2h}\right), & \text{for } i=j+1 \text{ and } j=1,\cdots,N-2.\end{cases}$$

Correction vector can be calculated using Thomas algorithm:

$$\boxed{J\cdot\begin{bmatrix}h_1^{(k)}\\ \vdots\\ h_{N-1}^{(k)}\end{bmatrix}=-\begin{bmatrix}f_1\left(y_1,\cdots,y_{N-1}\right)\\ \vdots\\ f_{N-1}\left(y_1,\cdots,y_{N-1}\right)\end{bmatrix} \quad\to\quad \begin{bmatrix}y_1^{(k+1)}\\ \vdots\\ y_{N-1}^{(k+1)}\end{bmatrix}=\begin{bmatrix}y_1^{(k)}\\ \vdots\\ y_{N-1}^{(k)}\end{bmatrix}+\begin{bmatrix}h_1^{(k)}\\ \vdots\\ h_{N-1}^{(k)}\end{bmatrix}.}$$

The Newton iteration will stop when the solutions converge to certain decimal places or some norm stopping criteria.

Ordinary differential equations (ODEs)

Finite difference method for nonlinear second-order boundary value problem

Newton's method for nonlinear systems

$$\begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

The system of equations $\quad g_i(y_1, y_2, \ldots, y_n) = 0 \quad (1 \le i \le n)$
can be expressed simply as $\quad \mathbf{G}(\mathbf{Y}) = \mathbf{0}$
by letting $\mathbf{Y} = (y_1, y_2, \ldots, y_n)^T$ and $\mathbf{G} = (g_1, g_2, \ldots, g_n)^T$. Using the Taylor's series expansion, we get

$$\mathbf{0} = \mathbf{G}(\mathbf{Y}+\mathbf{H}) \approx \mathbf{G}(\mathbf{Y}) + \mathbf{G}'(\mathbf{Y})\mathbf{H}\ ,$$ *(where* $\mathbf{Y}+\mathbf{H}$ *is more accurate solution)*

where $\mathbf{H} = (h_1, h_2, \ldots, h_n)^T$ and $\mathbf{G}'(\mathbf{Y})$ is the $n \times n$ Jacobian matrix $\mathbf{J}(\mathbf{Y})$:

$$\mathbf{J}(\mathbf{Y}) = \mathbf{G}'(\mathbf{Y}) = \begin{bmatrix} \partial g_1/\partial y_1 & \partial g_1/\partial y_2 & \cdots & \partial g_1/\partial y_n \\ \partial g_2/\partial y_1 & \partial g_2/\partial y_2 & \cdots & \partial g_2/\partial y_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial g_n/\partial y_1 & \partial g_n/\partial y_2 & \cdots & \partial g_n/\partial y_n \end{bmatrix}$$

$$y'' + (y')^3 y = 0$$
$$ans: y^3/3 - 2c_1 y = 2x + c2$$
$$let\ c_1 = c_2 = 0$$
$$y^3 = 6x$$
$$x = 1(0.25)2$$

The correction vector $\mathbf{H}$ is obtained by solving linear system

$$\mathbf{J}(\mathbf{Y})\mathbf{H} = -\mathbf{G}(\mathbf{Y})$$

If Jacobian matrix is tridiagonal matrix, then $\mathbf{H}$ can be solved using Thomas algorithm. If the matrix size is 2×2, then just use the inverse of matrix $\mathbf{J}$, $\mathbf{H} = \mathbf{J}^{-1}(-\mathbf{G})$.
Finally, Newton's iteration for n nonlinear equations in n variables is given by

$$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \mathbf{H}^{(k)} \quad \rightarrow \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} - \mathbf{J}^{-1}\mathbf{G}$$

where the Jacobian system is

$$\mathbf{J}(\mathbf{Y}^{(k)})\mathbf{H}^{(k)} = -\mathbf{G}(\mathbf{Y}^{(k)}).$$

Ordinary differential equations (ODEs)

Finite difference method for nonlinear second-order boundary value problem

E.g. use nonlinear finite difference method to solve boundary value problem

$$y'' = -(y')^2 - y + \ln x,\quad 1 \le x \le 2,\ y(1)=0,\ y(2)=\ln 2 = 0.6931.$$

for x=1(0.2)2. Analytical solution: $y=\ln x$. (use 4 decimal places). Stopping criterion: Tolerance, ε=0.02 using *maximum-magnitude* norm.

Let h=0.2, $x_0=a=1$, $x_1=1.2$, $x_2=1.4$, $x_3=1.6$, $x_4=1.8$ and $x_5=b=2$. Find $y_i \approx y(x_i)$, i=1,2,3,4. N=5.

At x_i, we get

$$-\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} + f\left(x_i, y_i, \frac{y_{i+1}-y_{i-1}}{2h}\right) = 0 \quad \rightarrow \boxed{-y_{i-1}+2y_i-y_{i+1}+h^2\left(-\left(\frac{y_{i+1}-y_{i-1}}{2h}\right)^2 - y_i + \ln x_i\right) = 0 = g_i}$$

For i=1, $$-y_0+2y_1-y_2+0.2^2\left(-\left(\frac{y_2-y_0}{2(0.2)}\right)^2 - y_1 + \ln x_1\right) = 0 \rightarrow -0+2y_1-y_2+\left(-\frac{1}{4}(y_2-0)^2 - 0.2^2 y_1 + 0.2^2\cdot 0.1823\right) = 0 = g_1$$

For i=2, $$-y_1+2y_2-y_3+0.2^2\left(-\left(\frac{y_3-y_1}{2(0.2)}\right)^2 - y_2 + \ln x_2\right) = 0 \rightarrow -y_1+2y_2-y_3+\left(-\frac{1}{4}(y_3-y_1)^2 - 0.2^2 y_2 + 0.2^2\cdot 0.3365\right) = 0 = g_2$$

For i=3, $$-y_2+2y_3-y_4+0.2^2\left(-\left(\frac{y_4-y_2}{2(0.2)}\right)^2 - y_3 + \ln x_3\right) = 0 \rightarrow -y_2+2y_3-y_4+\left(-\frac{1}{4}(y_4-y_2)^2 - 0.2^2 y_3 + 0.2^2\cdot 0.4700\right) = 0 = g_3$$

For i=4, $$-y_3+2y_4-y_5+0.2^2\left(-\left(\frac{y_5-y_3}{2(0.2)}\right)^2 - y_4 + \ln x_4\right) = 0 \rightarrow -y_3+2y_4-0.6931+\left(-\frac{1}{4}(0.6931-y_3)^2 - 0.2^2 y_4 + 0.2^2\cdot 0.5878\right) = 0 = g_4$$

$$\mathbf{J}(\mathbf{Y}) = \begin{bmatrix} \partial g_1/\partial y_1 & \partial g_1/\partial y_2 & \cdots & \partial g_1/\partial y_4 \\ \partial g_2/\partial y_1 & \partial g_2/\partial y_2 & \cdots & \partial g_2/\partial y_4 \\ \partial g_3/\partial y_1 & \partial g_3/\partial y_2 & \ddots & \vdots \\ \partial g_4/\partial y_1 & \partial g_4/\partial y_2 & \cdots & \partial g_4/\partial y_4 \end{bmatrix} = \begin{bmatrix} 2-0.2^2 & -1-\tfrac{1}{2}(y_2-\mathbf{0}) & 0 & 0 \\ -1-\tfrac{1}{2}(y_3-y_1)(-1) & 2-0.2^2 & -1-\tfrac{1}{2}(y_3-y_1) & 0 \\ 0 & -1-\tfrac{1}{2}(y_4-y_2)(-1) & 2-0.2^2 & -1-\tfrac{1}{2}(y_4-y_2) \\ 0 & 0 & -1-\tfrac{1}{2}(\mathbf{0.6931}-y_3)(-1) & 2-0.2^2 \end{bmatrix}$$

Ordinary differential equations (ODEs)

Finite difference method for nonlinear second-order boundary value problem

$$\mathbf{J}(\mathbf{Y}) = \begin{bmatrix} 1.96 & -1-\tfrac{1}{2}y_2 & 0 & 0 \\ -1+\tfrac{1}{2}(y_3-y_1) & 1.96 & -1-\tfrac{1}{2}(y_3-y_1) & 0 \\ 0 & -1+\tfrac{1}{2}(y_4-y_2) & 1.96 & -1-\tfrac{1}{2}(y_4-y_2) \\ 0 & 0 & -0.6534-\tfrac{1}{2}y_3 & 1.96 \end{bmatrix}$$

To guess the initial values, we use linear interpolation, h=(ln2-0)/5≈0.14; where y_0=0, y_5=0.7. So, we get

$$\mathbf{Y}^{(0)} = \begin{bmatrix} y_1^{(0)} \\ y_2^{(0)} \\ y_3^{(0)} \\ y_4^{(0)} \end{bmatrix} = \begin{bmatrix} 0.14 \\ 0.28 \\ 0.42 \\ 0.56 \end{bmatrix}, \mathbf{J}\left(\mathbf{Y}^{(0)}\right) = \begin{bmatrix} 1.96 & -1-\tfrac{1}{2}0.28 & 0 & 0 \\ -1+\tfrac{1}{2}(0.42-0.14) & 1.96 & -1-\tfrac{1}{2}(0.42-0.14) & 0 \\ 0 & -1+\tfrac{1}{2}(0.56-0.28) & 1.96 & -1-\tfrac{1}{2}(0.56-0.28) \\ 0 & 0 & -0.6534-\tfrac{1}{2}0.42 & 1.96 \end{bmatrix}$$

$$\rightarrow \mathbf{J}\left(\mathbf{Y}^{(0)}\right) = \begin{bmatrix} 1.96 & -1.14 & 0 & 0 \\ -0.86 & 1.96 & -1.14 & 0 \\ 0 & -0.86 & 1.96 & -1.14 \\ 0 & 0 & -0.8634 & 1.96 \end{bmatrix}, \quad -\mathbf{G}\left(\mathbf{Y}^{(0)}\right) = -\begin{bmatrix} 2(0.14)-0.28-\tfrac{1}{4}0.28^2-0.2^2 0.14+0.0073 \\ -0.14+2\cdot 0.28-0.42-\tfrac{1}{4}(0.42-0.14)^2-0.2^2 0.28+0.0135 \\ -0.28+2\cdot 0.42-0.56-\tfrac{1}{4}(0.56-0.28)^2-0.2^2 0.42+0.0188 \\ -0.42+2\cdot 0.56-0.6931-\tfrac{1}{4}(0.6931-0.42)^2-0.2^2 0.56+0.0235 \end{bmatrix} = -\begin{bmatrix} -0.0179 \\ -0.0173 \\ -0.0176 \\ -0.0106 \end{bmatrix}$$

$$\begin{bmatrix} 1.96 & -1.14 & 0 & 0 \\ -0.86 & 1.96 & -1.14 & 0 \\ 0 & -0.86 & 1.96 & -1.14 \\ 0 & 0 & -0.8634 & 1.96 \end{bmatrix} \begin{bmatrix} h_1^{(0)} \\ h_2^{(0)} \\ h_3^{(0)} \\ h_4^{(0)} \end{bmatrix} = -\begin{bmatrix} -0.0179 \\ -0.0173 \\ -0.0176 \\ -0.0106 \end{bmatrix} \rightarrow \text{Thomas algorithm} \rightarrow \begin{bmatrix} h_1^{(0)} \\ h_2^{(0)} \\ h_3^{(0)} \\ h_4^{(0)} \end{bmatrix} = \begin{bmatrix} 0.0414 \\ 0.0556 \\ 0.0491 \\ 0.0270 \end{bmatrix}.$$

$$\begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \\ y_4^{(1)} \end{bmatrix} = \begin{bmatrix} y_1^{(0)} \\ y_2^{(0)} \\ y_3^{(0)} \\ y_4^{(0)} \end{bmatrix} + \begin{bmatrix} h_1^{(0)} \\ h_2^{(0)} \\ h_3^{(0)} \\ h_4^{(0)} \end{bmatrix} = \begin{bmatrix} 0.14 \\ 0.28 \\ 0.42 \\ 0.56 \end{bmatrix} + \begin{bmatrix} 0.0414 \\ 0.0556 \\ 0.0491 \\ 0.0270 \end{bmatrix} = \begin{bmatrix} 0.1814 \\ 0.3356 \\ 0.4691 \\ 0.587 \end{bmatrix}.$$

Ordinary differential equations (ODEs)

Finite difference method for nonlinear second-order boundary value problem

$$\mathbf{Y}^{(1)} = \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \\ y_4^{(1)} \end{bmatrix} = \begin{bmatrix} 0.1814 \\ 0.3356 \\ 0.4691 \\ 0.587 \end{bmatrix}, \mathbf{J}\left(\mathbf{Y}^{(1)}\right) = \begin{bmatrix} 1.96 & -1-\tfrac{1}{2}0.3356 & 0 & 0 \\ -1+\tfrac{1}{2}(0.4691-0.1814) & 1.96 & -1-\tfrac{1}{2}(0.4691-0.1814) & 0 \\ 0 & -1+\tfrac{1}{2}(0.587-0.3356) & 1.96 & -1-\tfrac{1}{2}(0.587-0.3356) \\ 0 & 0 & -0.6534-\tfrac{1}{2}0.4691 & 1.96 \end{bmatrix}$$

$$\rightarrow \mathbf{J}\left(\mathbf{Y}^{(1)}\right) = \begin{bmatrix} 1.96 & -1.1678 & 0 & 0 \\ -0.8562 & 1.96 & -1.1439 & 0 \\ 0 & -0.8743 & 1.96 & -1.1257 \\ 0 & 0 & -0.8880 & 1.96 \end{bmatrix}, \quad -\mathbf{G}\left(\mathbf{Y}^{(1)}\right) = -\begin{bmatrix} -0.0009 \\ 0.00004 \\ -0.00016 \\ -0.0007 \end{bmatrix}$$

$$\begin{bmatrix} 1.96 & -1.1678 & 0 & 0 \\ -0.8562 & 1.96 & -1.1439 & 0 \\ 0 & -0.8743 & 1.96 & -1.1257 \\ 0 & 0 & -0.8880 & 1.96 \end{bmatrix} \begin{bmatrix} h_1^{(1)} \\ h_2^{(1)} \\ h_3^{(1)} \\ h_4^{(1)} \end{bmatrix} = -\begin{bmatrix} -0.0009 \\ 0.00004 \\ -0.00016 \\ -0.0007 \end{bmatrix} \rightarrow \text{Thomas algorithm} \rightarrow \begin{bmatrix} h_1^{(1)} \\ h_2^{(1)} \\ h_3^{(1)} \\ h_4^{(1)} \end{bmatrix} = \begin{bmatrix} 0.0011 \\ 0.0010 \\ 0.0010 \\ 0.0008 \end{bmatrix}.$$

$$\begin{bmatrix} y_1^{(2)} \\ y_2^{(2)} \\ y_3^{(2)} \\ y_4^{(2)} \end{bmatrix} = \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \\ y_4^{(1)} \end{bmatrix} + \begin{bmatrix} h_1^{(1)} \\ h_2^{(1)} \\ h_3^{(1)} \\ h_4^{(1)} \end{bmatrix} = \begin{bmatrix} 0.1814 \\ 0.3356 \\ 0.4691 \\ 0.587 \end{bmatrix} + \begin{bmatrix} 0.0011 \\ 0.0010 \\ 0.0010 \\ 0.0008 \end{bmatrix} = \begin{bmatrix} 0.1825 \\ 0.3366 \\ 0.4701 \\ 0.5878 \end{bmatrix}.$$

Stopping criterion: *maximum-magnitude* norm of increment solution vector < ε=0.02. $\left\|\mathbf{Y}^{(2)} - \mathbf{Y}^{(1)}\right\|_\infty = \left\|\mathbf{H}^{(1)}\right\|_\infty = 0.0011 < \varepsilon.$

The final solution is

$$\begin{bmatrix} y_1^{(2)} \\ y_2^{(2)} \\ y_3^{(2)} \\ y_4^{(2)} \end{bmatrix} = \begin{bmatrix} 0.1825 \\ 0.3366 \\ 0.4701 \\ 0.5878 \end{bmatrix}. \quad \textit{exact solution}: \begin{bmatrix} \ln 1.2 \\ \ln 1.4 \\ \ln 1.6 \\ \ln 1.8 \end{bmatrix} = \begin{bmatrix} 0.1823 \\ 0.3365 \\ 0.4700 \\ 0.5878 \end{bmatrix}.$$

Note: If the problem is simplified by only finding 2 points (y_1 and y_2), Then Thomas algorithm is not required since the matrix is 2×2. Use the below simple formula.

$$\mathbf{J}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}. \rightarrow \quad \mathbf{H} = \mathbf{J}^{-1}(-\mathbf{G}).$$