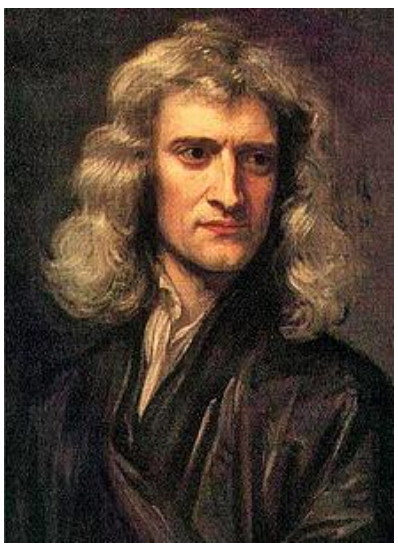
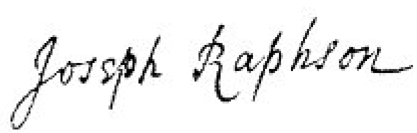
Newton-Raphson Method

1. General development
2. Examples
3. In-class exercise
4. Secant method



Isaac Newton
1669



Joseph Raphson
1690

Thomas Simpson
1740

Newton-Raphson Method

General Development

Fixed-point Method

- Single nonlinear algebraic equation:

$$f(x) = 0 \quad \Rightarrow \quad x = f(x) + x = g(x)$$

- Generate an iterative equation:

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

- A value $x = s$ that satisfies $s = g(s)$ is called a fixed point
- The fixed point also is a solution of $f(s) = 0$

Limitations of the Fixed-Point Method

- The function $g(x)$ is not unique
  - » Not obvious how to construct
- The method often exhibits divergence
  - » Not clear how to select $g(x)$ and x_0 to achieve convergence
- The method can exhibit very slow convergence
  - » Problematic for large systems of nonlinear algebraic equations
- More stable and faster converging methods are needed

Newton-Raphson Method

- First-order Taylor series expansion

$$0 = f(x) \cong f(\bar{x}) + \left(\frac{df}{dx}\right)_{(\bar{x})} (x - \bar{x})$$

- Iterative equation

$$f(x_{n+1}) = f(x_n) + \frac{df(x_n)}{dx}(x_{n+1} - x_n) = 0$$

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

Newton-Raphson Method

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

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

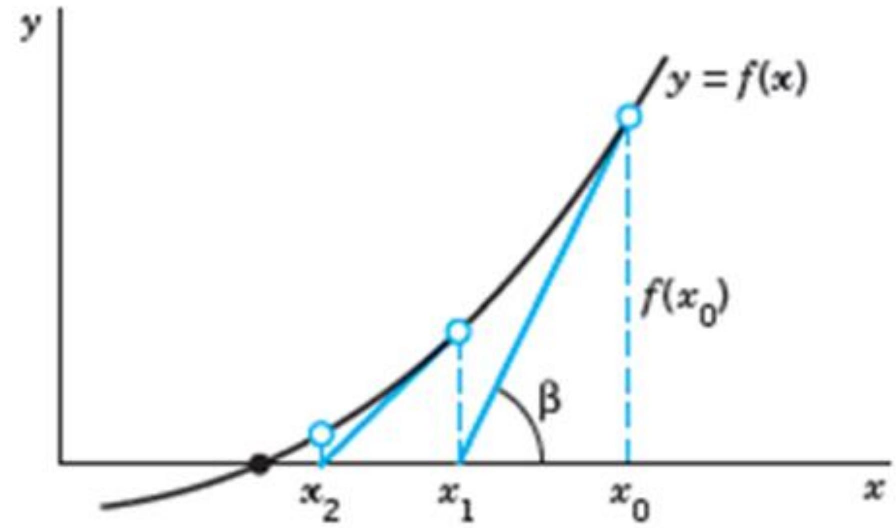


Fig. 428. Newton's method

- Newton-Raphson method offers better convergence properties than the fixed-point method
- This improved performance comes at the expense of needing to compute the derivative of $f(x)$

Convergence of the Newton-Raphson Method

- Let $f(x) = 0$ have a solution $x = s$ and assume $f(x)$ is three times differentiable, its first- and second-order derivatives are non-zero at $x = s$ and x_0 is sufficiently close to s.
- Then the Newton method is second-order and exhibit quadratic converge to s:

$$\varepsilon_n \equiv s - x_n \quad \Rightarrow \quad \left|\varepsilon_{n+1}\right| = c\left|\varepsilon_n\right|^2$$

- Caveats
  - » The method can converge slowly or even diverge for poorly chosen x_0
  - » The solution obtained can depend on x_0
  - » The method fails if the first-order derivative becomes zero (singularity)

UMASS.

Practical Newton-Raphson Algorithm

- Inputs
  - » Functions: $f(x)$ and $df(x)/dx$
  - » Initial guess: x_0
  - » Solution error tolerance: δ
  - » Maximum number of iterations: N
- Given x_n, compute x_{n+1} as:

$$x_{n+1} = x_n - \frac{f(x_n)}{df(x_n)/dx}$$

- Continue until $|x_{n+1} - x_n| < \delta |x_n|$ or $n = N$

Newton-Raphson Method

Examples

Newton-Raphson Example 1

- Nonlinear equation:

$$f(x) = x^2 - 3x + 1 = 0 \quad \Rightarrow \quad x = 0.3820, 2.618$$

- Fixed point iteration:

$$\frac{df}{dx} = 2x - 3 \quad \Rightarrow \quad x_{n+1} = x_n - \frac{x_n^3 - 3x_n + 1}{2x_n - 3}$$

Iteration	$x_0 = 1$	$x_o = 3$
1	0	2.667
2	0.3333	2.619
3	0.3810	2.618
4	0.3820	2.618

Newton-Raphson Example 2

- Redlich-Kwong equation:

$$P = \frac{RT}{V-b} - \frac{a}{\sqrt{T}V(V+b)}$$

- Iterative equation:

$$f(V) = -P + \frac{RT}{V-b} - \frac{a}{\sqrt{T}V(V+b)} = 0$$

$$f'(V) = -\frac{RT(1)}{(V-b)^2} + \frac{a\sqrt{T}(2V+b)}{TV^2(V+b)^2}$$

$$V_{n+1} = V_n - \frac{f(V_n)}{f'(V_n)}$$

Newton-Raphson Example 2

- Data for argon
  - » a = 16.82 $L^2 \cdot bar \cdot mol^{-2} \cdot K^{1/2}$
  - » b = 0.02219 L/mol
- Conditions
  - » P = 174 bar
  - » T = 390 K
- Gas constant: R = 0.08314 $L \cdot bar \cdot mol^{-1} \cdot K^{-1}$

Iteration	$V_0 = 0.1$	$V_0 = 1$
1	0.1423	-3.392
2	0.1759	-68.34
3	0.1871	-2.3×10^{4}
4	0.1879	-3.4×10^{9}
5	0.1879	-6.2×10^{19}

Nonlinear Algebraic Equation Systems

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, \ldots, x_n) &= 0 \\
&\vdots \\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
\quad \Rightarrow \quad \mathbf{f}(\mathbf{x}) = \mathbf{0} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}
$$

- Newton-Raphson method:

$$
\mathbf{x}_{n+1} = \mathbf{x}_n - \left[\frac{d\mathbf{f}(\mathbf{x}_n)}{d\mathbf{x}} \right]^{-1} \mathbf{f}(\mathbf{x}_n)
$$

- Guess $\mathbf{x}_0$

The Jacobian Matrix

- The Jacobian matrix is a *nxn* matrix of partial derivatives:

$$\mathbf{J}(\mathbf{x}) = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

- For a large system of equations, calculating **J** requires considerable effort
- Many of the elements of **J** may be identically zero

Newton-Raphson Example 3

- 2 coupled nonlinear algebraic equations

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_1^2 + x_2^2 - 2 \\ x_1 x_2 - 1 \end{bmatrix} = \mathbf{0}$$

- Jacobian matrix

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 & 2x_2 \\ x_2 & x_1 \end{bmatrix}$$

Newton-Raphson Example 3

- Iterative equation

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \begin{bmatrix} 2x_1 & 2x_2 \\ x_2 & x_1 \end{bmatrix}^{-1} \begin{bmatrix} x_1^2 + x_2^2 - 2 \\ x_1 x_2 - 1 \end{bmatrix}$$

Iteration	$\mathbf{x}_0$ = [3 2]	$\mathbf{x}_o$ = [300 200]
1	[1.700 1.200]	[150.0 100.0]
2	[1.195 0.9448]	[75.01 50.01]
3	[1.065 0.9398]	[37.51 25.01]
4	[1.031 0.9688]	[18.77 12.52]
5	[1.016 0.9844]	[9.418 6.292]
15	[1.000 1.000]	[1.002 0.9985]

Newton-Raphson Method

In-class Exercise

Newton-Raphson Method

Secant Method

Secant Method

- Motivation
  - » Evaluation of $d\mathbf{f}/d\mathbf{x}$ is computationally expensive for large systems of equations
  - » Want an efficient, derivative-free method
- Derivative approximation

$$\frac{df(x_n)}{dx} \cong \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

- Secant algorithm

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

- Convergence
  - » Superlinear: $|\varepsilon_{n+1}| = c|\varepsilon_n|^m \quad 1 < m < 2$
  - » Slower but more efficient than Newton-Raphson ($m = 2$)

Secant Example 1

- Redlich-Kwong equation:

$$P = \frac{RT}{V - b} - \frac{a}{\sqrt{T}V(V + b)}$$

- Iterative equation:

$$f(V_n) = -P + \frac{RT}{V_n - b} - \frac{a}{\sqrt{T}V_n(V_n + b)} = 0$$

$$V_{n+1} = V_n - f(V_n)\frac{V_n - V_{n-1}}{f(V_n) - f(V_{n-1})}$$

Secant Example 1

Iteration	$V_0 = 0.1, V_{-1} = 0.2$
1	0.1941
2	0.1911
3	0.1878
4	0.1879
5	0.1879

Secant Method for Equation Systems

- Newton-Raphson method

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \qquad\qquad \mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}^{-1}(\mathbf{x}_n)\mathbf{f}(\mathbf{x}_n)$$

- Jacobian approximation for 2x2 system

$$\mathbf{J}(\mathbf{x}_n) \cong \tilde{\mathbf{J}}(\mathbf{x}_n) = \begin{bmatrix} \dfrac{f_1(x_{1,n}, x_{2,n}) - f_1(x_{1,n-1}, x_{2,n})}{x_{1,n} - x_{1,n-1}} & \dfrac{f_1(x_{1,n}, x_{2,n}) - f_1(x_{1,n}, x_{2,n-1})}{x_{2,n} - x_{2,n-1}} \\ \dfrac{f_2(x_{1,n}, x_{2,n}) - f_2(x_{1,n-1}, x_{2,n})}{x_{1,n} - x_{1,n-1}} & \dfrac{f_2(x_{1,n}, x_{2,n}) - f_2(x_{1,n}, x_{2,n-1})}{x_{2,n} - x_{2,n-1}} \end{bmatrix}$$

- Secant method:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \tilde{\mathbf{J}}^{-1}(\mathbf{x}_n)\mathbf{f}(\mathbf{x}_n)$$