3.1 Introduction

- Solve non-linear real equation $f(x) = 0$ for real *root* or *zero* x^*.

$$\text{E.g.} \qquad x^3 + 1.5x - 1.5 = 0\,, \qquad \tan x - x = 0\,.$$

- *Practical existence test for roots*: by intermediate value theorem,

$$f \in C[a,b] \quad \& \quad f(a)f(b) < 0 \quad \Rightarrow \quad \text{at least one root } x^*,\ a < x^* < b.$$
$$f \in C[a,b] \quad \& \quad f(a)f(b) > 0 \quad \Rightarrow \quad \text{even number (possibly 0) of roots.}$$

  Roots of f/f' are simple and coincide with roots of f.

- No general algebraic method for finding x^*.

  Must approximate x^* by $\bar{x}$ within specified tolerances ϵ_1, ϵ_2 or ϵ_3:

$$|\bar{x} - x^*| \leq \epsilon_1\,, \qquad |\bar{x} - x^*| \leq \epsilon_2 |x^*|\,, \qquad |f(\bar{x})| \leq \epsilon_3\,,$$

  Definition: $\bar{x}$ approximates x^* to t *significant decimal digits* (or *figures*) if

$$|\bar{x} - x^*| \leq 0.5 \times 10^{-t} |x^*|\,.$$

- Special methods: matrix eigenvalues, function minimisation, polynomials, non-linear least squares.

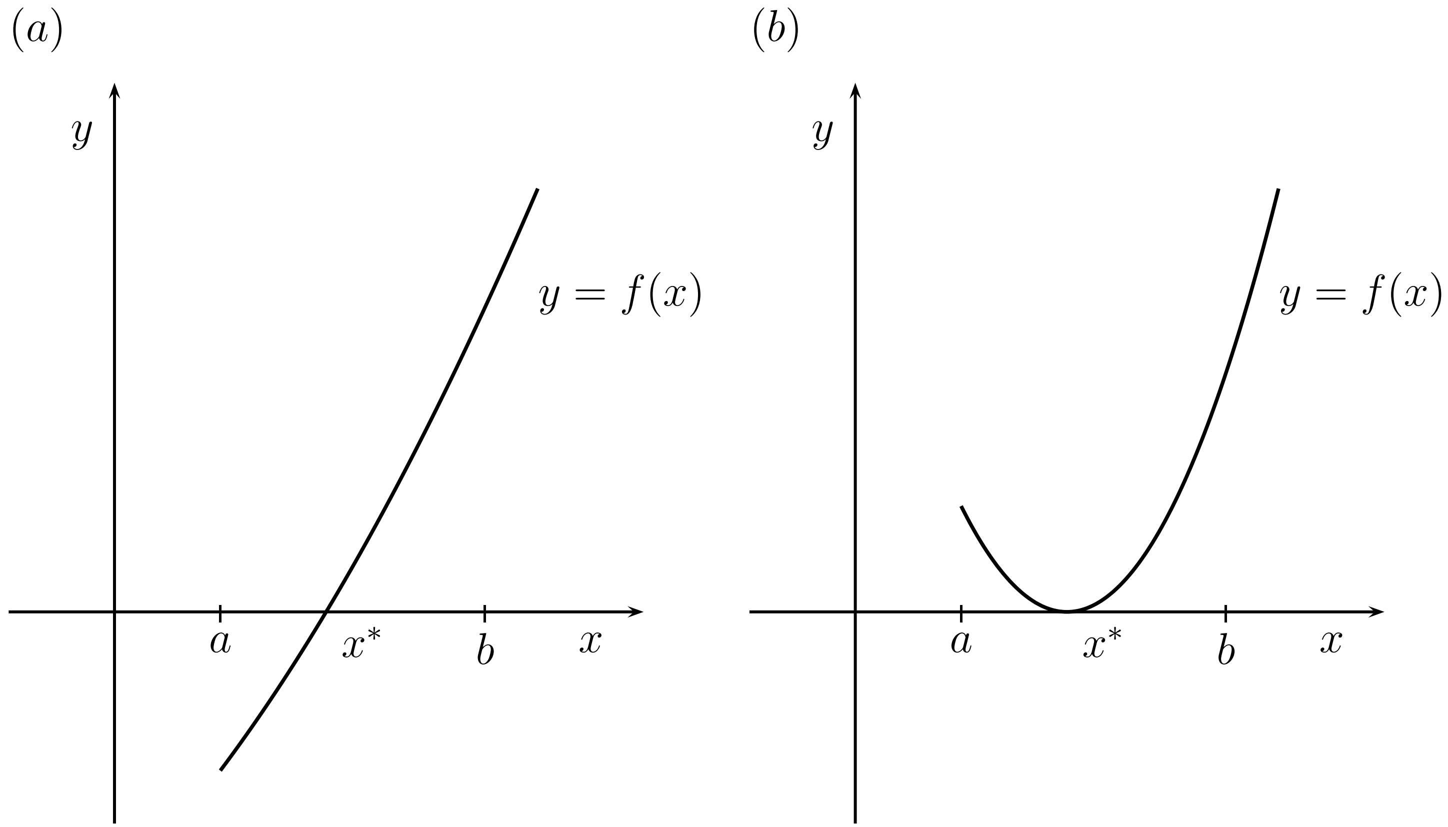

Figure 3.1: Examples of roots where (a) f changes sign and (b) f does not change sign.

3.2 Graphing $y = f(x)$

- 1 – 2 (hand) / 5 – 10 (calculator) significant decimal digits. Re-graph on a smaller scale for improved accuracy.

- Good enough for simple functions & few roots.

  Hopelessly inefficient for complicated functions or many roots.

- Efficiency: *assume the dominant arithmetic cost of numerical algorithms (except for linear algebraic systems) is function evaluation*.

  Valid for functions with complicated definitions: e.g. defined by infinite series, infinite products, continued fractions, recurrence relations, integrals or differential equations.

Example 3.1 Graph $f(x) = 2x^3 + 3x - 3$ in $[0, 1]$ using 11 points to find roots.

Solution Space points evenly & assume f varies smoothly (reasonable for cubic polynomial but could smooth out zeros of wildly oscillating function).

$$f \in C[0,1] \quad \& \quad f(0)f(1) < 0 \quad \Rightarrow \quad \text{one root } x^* \in [0,1] \text{ near } 0.73.$$

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$f(x)$	-3	-2.7	-2.4	-2.0	-1.7	-1.3	-0.8	-0.2	0.4	1.2	2

Table 3.1: Values of $f(x) = 2x^3 + 3x - 3$.

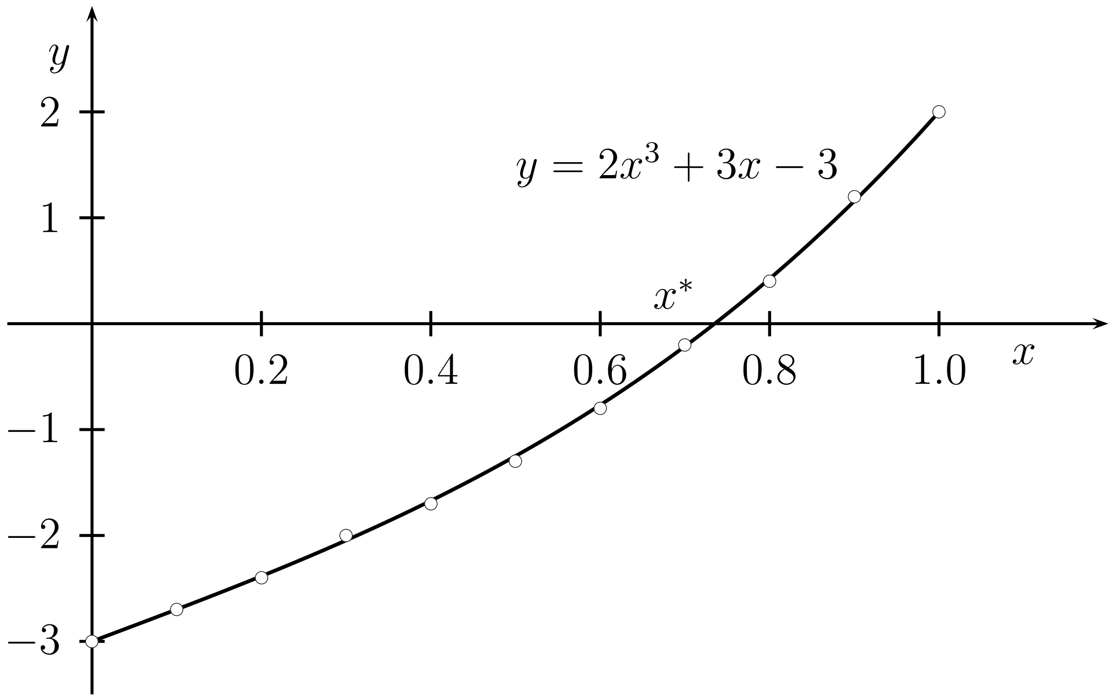

Figure 3.2: Graph of $y = 2x^3 + 3x - 3$ on $[0, 1]$ using 11 points.

3.3 Bisection or Interval-Halving

- $f \in C[a,b]$ & $f(a)f(b) < 0$ $\Rightarrow$ at least one root x^*, $a < x^* < b$.

  Calculate midpoint $m = \frac{1}{2}(a+b)$:

  $$f(a)f(m) < 0 \quad \Rightarrow \quad a < x^* < m\,; \quad \text{otherwise} \quad f(a)f(m) \geq 0 \quad \Rightarrow \quad m \leq x^* < b.$$

  i.e. interval bracketing a root halved: $m - a = b - m = (b-a)/2$.

  Repeat to obtain interval of length $< \epsilon_1$ which contains x^*. Any point $\bar{x}$ in this interval approximates x^*.

Example 3.2 Apply bisection to $f(x) = 2x^3 + 3x - 3$ with $a_0 = 0.7$, $b_0 = 0.8$.

Solution $m_1 = (a_0 + b_0)/2 = 0.75$, $f(0.7) = -0.214$, $f(0.75) = 0.09375$:

$$f(0.7)f(0.75) < 0 \quad \Rightarrow \quad 0.7 = a_0 < x^* < m_1 = 0.75\,.$$

Repeat with $a_1 = 0.7$, $b_1 = 0.75$; etc.

- Bisection generates a sequence of intervals $[a_i, b_i]$, $i = 0, 1, 2, \ldots$, containing x^*. At iteration i the uncertainty $e_i = b_i - a_i$ in x^* is halved: $e_{i+1} = e_i/2$.
  At least one extra bit of x^* is obtained each iteration.
  For 32-bit floating-point numbers with 24-bit mantissa, full 24-bit accuracy achieved within 24 iterations with 1 function evaluation per iteration.
  Convergence to x^* certain but slow if sign of f can be accurately determined.

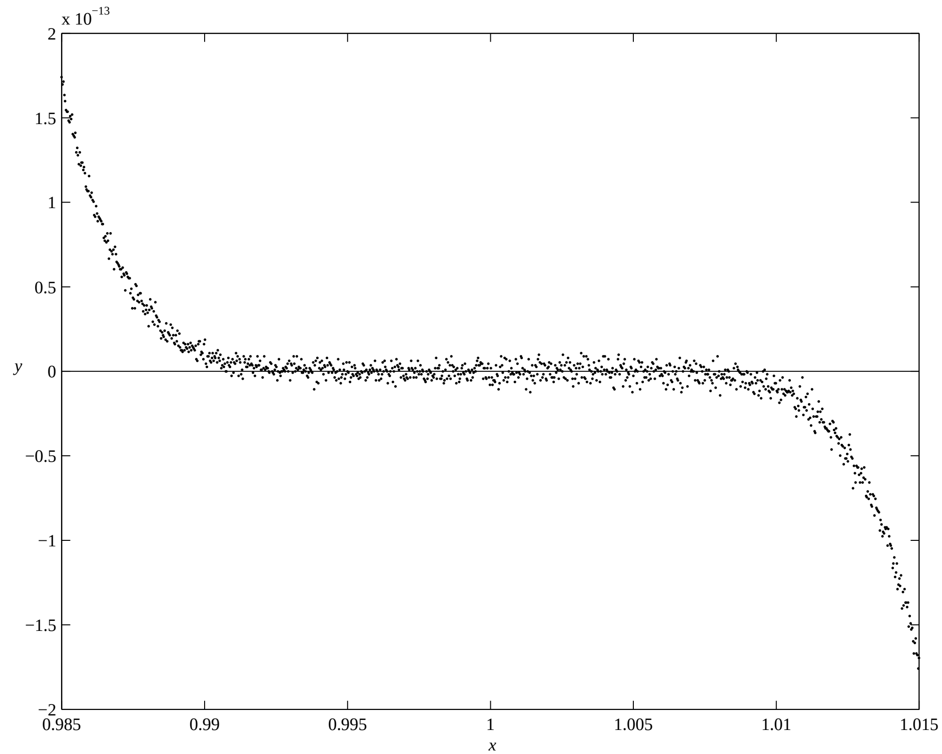

Figure 3.3: Evaluation of $y = 1 - 7x + 21x^2 - 35x^3 + 35x^4 - 21x^5 + 7x^6 - x^7$ near $x = 1$.

- A sequence $\{x_i\}$ converges to x^* with *rate* r and *asymptotic error constant* C if the error e_i in the ith iterate x_i, $e_i = x^* - x_i$, satisfies

$$\lim_{i\to\infty} \frac{|e_{i+1}|}{|e_i|^r} = C\,, \qquad 0 < C < \infty\,.$$

- A numerical method converges to x^* with *rate* r and *asymptotic error constant* C if it generates sequences with these properties.

- Eventually, $|e_{i+1}| \approx C|e_i|^r$. Let $e_i = 10^{-k_i}$, $C = 10^{-\ell}$ & take base 10 logarithms:

$$k_{i+1} \approx r k_i + \ell\,.$$

$$k_i \approx \text{correct number of decimal places in } x_i.$$

Convergence:

linear if $r = 1$ ($C < 1$): $\approx \ell$ extra decimal places per iteration;

superlinear if $r > 1$;

quadratic if $r = 2$: $\approx$ double number of correct decimal places per iteration;
($\approx \ell$ extra decimal places per iteration swamped by doubling).

- Condition for bisection holds with $e_i = b_i - a_i$, $r = 1$ and $C = \frac{1}{2}$.

3.4 The Newton-Raphson Method

- *Geometric Derivation*: approximate curve $y = f(x)$ by tangent at $x = x_i$ & take x-intercept as x_{i+1}. From diagram,

$$\text{slope} = \tan\alpha = \frac{f(x_i)}{x_i - x_{i+1}} = f'(x_i)\,.$$

Solving for x_{i+1} gives *Newton-Raphson formula*:

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

- *Taylor Series Derivation*: expand $f(x)$ in Taylor series about x_i,

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(\eta_i)(x - x_i)^2\,.$$

Set $x = x^*$ & use $x^* = x_i + e_i$,

$$0 = f(x^*) = f(x_i + e_i) = f(x_i) + f'(x_i)e_i + \frac{1}{2}f''(\eta_i)e_i^2\,. \tag{3.1}$$

Thus $f(x_i) + f'(x_i)e_i \approx 0$ & $e_i \approx -f(x_i)/f'(x_i)$. Hence

$$x^* = x_i + e_i \approx x_i - \frac{f(x_i)}{f'(x_i)} = x_{i+1}\,.$$

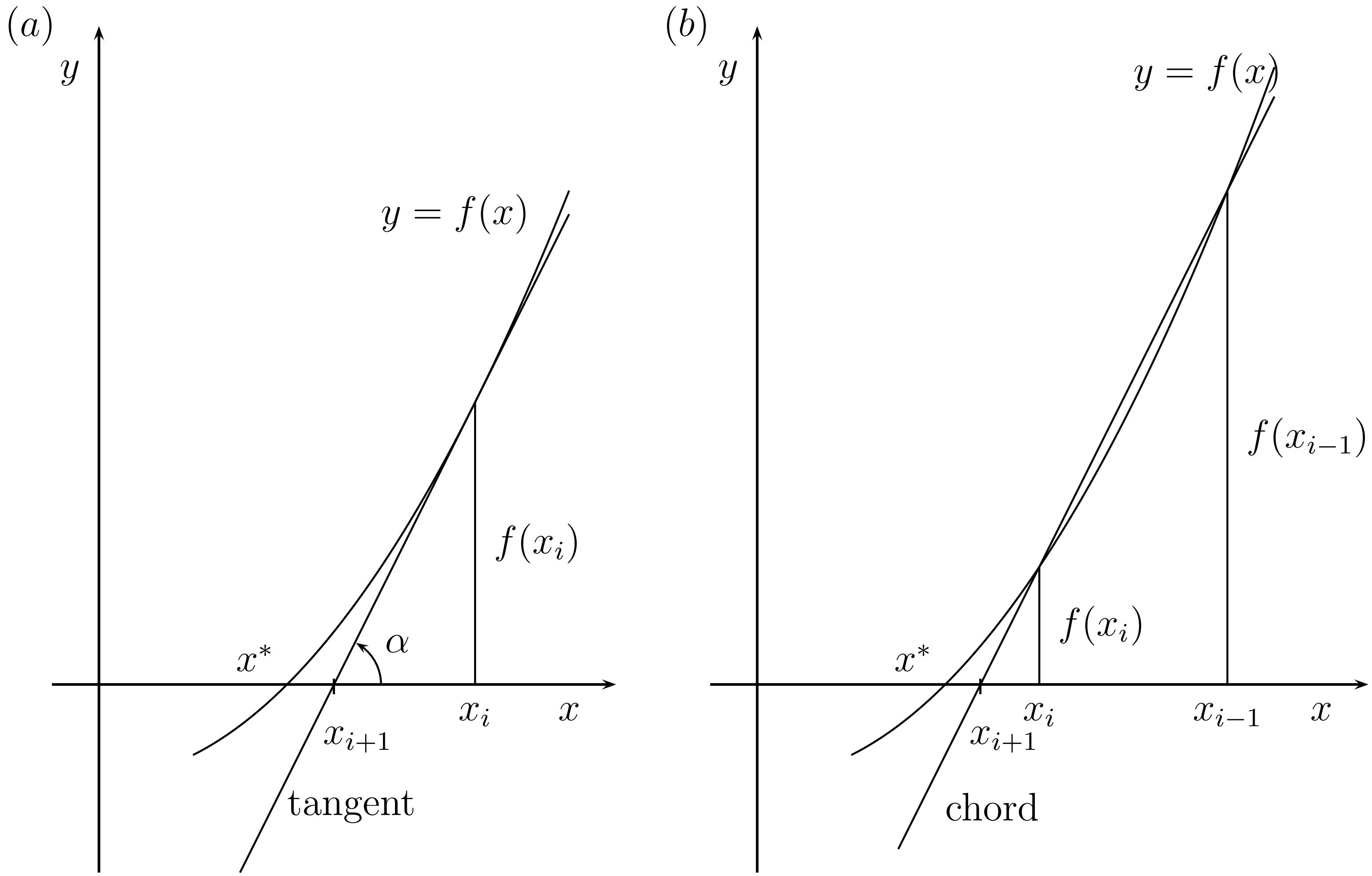

Figure 3.4: (a) The Newton-Raphson method. (b) The secant method.

- Newton-Raphson method converges quadratically for simple roots:

  Divide (3.1) by $f'(x_i)$ & re-arrange,

$$e_i = -\frac{f(x_i)}{f'(x_i)} - \frac{f''(\eta_i)}{2f'(x_i)} e_i^2 \,.$$

  Then

$$e_{i+1} = x^* - x_{i+1} = x^* - \left(x_i - \frac{f(x_i)}{f'(x_i)} \right) = e_i + \frac{f(x_i)}{f'(x_i)} = -\frac{f''(\eta_i)}{2f'(x_i)} e_i^2 = -C_i e_i^2 \,,$$

  where $C_i = f''(\eta_i)/2f'(x_i)$. If $x_i \to x^*$ as $i \to \infty$, then $\eta_i \to x^*$ &

$$\lim_{i\to\infty} \frac{|e_{i+1}|}{|e_i|^2} = \lim_{i\to\infty} |C_i| = \frac{|f''(x^*)|}{|2f'(x^*)|} = C \,.$$

- For pth-order roots, $p > 1$, Newton-Raphson method converges linearly with $C = 1 - 1/p$. If p known modified Newton-Raphson method quadratically convergent:

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

  else, if p unknown, replace f by f/f' — needs f''.

Example 3.4 Apply Newton-Raphson to $f(x) = 2x^3 + 3x - 3$ with $x_0 = 0.7$.

Solution $f(x_0) = f(0.7) = -0.214$. $f'(x) = 6x^2 + 3$, $f'(x_0) = f'(0.7) = 5.94$.

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0.7 - \frac{f(0.7)}{f'(0.7)} = 0.7 - \frac{-0.214}{5.94} = 0.7 + 0.0360269360 = 0.7360269360\,.$$

$$x^* = 0.7351392590\ldots\,, \quad f''(x) = 12x \Rightarrow C = f''(x^*)/2f'(x^*) \approx 0.7066.$$

$e_1 = -0.0008877$ & $e_2 = -0.000000556$. Thus $-Ce_1^2 = -0.000000557 \approx e_2$.

- Newton-Raphson method is of form

$$x_{i+1} = g(x_i)\,, \tag{3.2}$$

with $g(x) = x - f(x)/f'(x)$.

Infinitely many choices of g give root-finding schemes but g is not arbitrary: a root x^* of $f(x)$ must be a *fixed point* of g, i.e. $g(x^*) = x^*$, and conversely.

Simple iteration / functional iteration if linearly convergent.

- *Convergence Test* (not all g give convergent schemes):

$$|g'(x)| \le L < 1 \text{in interval about } x^* \text{ \& } x_0 \quad \Rightarrow \quad \text{convergence to } x^*.$$

Mean value theorem & $g(x^*) = x^*$ $\Rightarrow$

$$x^* - x_i = g(x^*) - g(x_i) = g'(\eta_i)(x^* - x_{i-1})\,.$$

Correction: i-1

Therefore

– x_{i-1}, x_i lie on same [opposite] side of x^* if $g'(\eta_i) > 0$ [$g'(\eta_i) < 0$];

– x_i is closer to x^* than x_{i-1} if $|g'(\eta_i)| < 1$;

– the smaller $g'(\eta_i)$, the closer x_i is to x^*.

If $L < 1$,

$$|x^* - x_i| = |g'(\eta_i)(x^* - x_{i-1})| \leq L|x^* - x_{i-1}| \leq L^i|x^* - x_0| \to 0 \quad \text{as } i \to \infty\,.$$

The smaller L & $|g'|$, the faster $x_i \to x^*$.

- For Newton-Raphson method

$$g' = 1 - \frac{f'^2 - f f''}{f'^2} = \frac{f f''}{f'^2}\,.$$

Thus Newton-Raphson converges if $|f f''/f'^2| \leq L < 1$,

i.e. if x_0 is close to x^*, f does not curve too sharply & f is steep enough.

Example 3.5 Apply simple iteration to $f(x) = 2x^3 + 3x - 3$ with $x_0 = 0.7$.

Solution Three fixed point forms of $f(x) = 0$ are

$$x = 1 - \frac{2x^3}{3}, \qquad x = \frac{3}{2x^2 + 3}, \qquad x = x - \frac{2x^3 + 3x - 3}{5.94}\,.$$

Third is $x = x - f(x)/f'(x_0)$ with $f'(x_0) = f'(0.7) = 5.94$.

Iteration schemes have

$$\text{(i)} \quad g(x) = 1 - \frac{2x^3}{3}, \qquad \text{(ii)} \quad g(x) = \frac{3}{2x^2+3}, \qquad \text{(iii)} \quad g(x) = x - \frac{2x^3+3x-3}{5.94}\,.$$

Difficult to show $|g'| \leq L < 1$ in interval: practical but not foolproof test: $|g'(x_0)|$. Since

$$|g'(x_0)| = |g'(0.7)| = \qquad \text{(i)} \quad 0.98, \quad \text{(ii)} \quad 0.53, \quad \text{(iii)} \quad 0.00\,,$$

expect:

- (i) may not converge or converge very slowly;
- (ii) will probably converge;
- (iii) will converge the fastest.

3.5 The Secant (or Chord) Method

- Newton-Raphson:
  - 2 disadvantages: f' & possible non-convergence;
  - 1 advantage: quadratic convergence.

  Disadvantages of f':

  - f' may not exist, or may be difficult or practically impossible to determine;
  - f' (& f) must be evaluated each iteration.

- *Finite-Difference Approximation*: Approximate f' in Newton-Raphson formula,

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_1 - x_{i-1}},$$

Correction: i

$\Rightarrow$ secant iteration formula:

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

- *Geometric Derivation*: By similar triangles,

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

Solve for x_{i+1} to get secant iteration formula.

Example 3.6 Apply secant method to $f(x) = 2x^3 + 3x - 3 = 0$ with $x_0 = 0.8$ & $x_1 = 0.7$.

Solution

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1) = 0.7 - \frac{0.7 - 0.8}{f(0.7) - f(0.8)} f(0.7) = 0.7335423\,.$$

- Secant method converges superlinearly

$$\boxed{r = (1 + \sqrt{5})/2 \approx 1.618\,, \qquad C = \{|f''(x^*)|/2|f'(x^*)|\}^{1/r}}$$

(60% increase in correct decimal places per iteration).

- Secant method requires only one function evaluation $f(x_i)$ per iteration. [Cf. $f(x_i)$, $f'(x_i)$.]

Thus it can compare favourably with Newton-Raphson, even though it converges more slowly.

- Secant method may not converge like Newton-Raphson method.

- To be robust any algorithm using the secant method must incorporate a strategy to guarantee convergence, e.g. sign f bracketing. Let $[a_i, b_i]$ bracket x^* & set

$$x_i = b_i - \frac{b_i - a_i}{f(b_i) - f(a_i)} f(b_i)\,.$$

If $f(x_i)f(a_i) < 0$, set $a_{i+1} = a_i$, $b_{i+1} = x_i$; otherwise set $a_{i+1} = x_i$, $b_{i+1} = b_i$ — *linear interpolation* (*false position*).

Example 3.7 Apply linear interpolation to $f(x) = 2x^3 + 3x - 3$ with $x_0 = 0.8$ & $x_1 = 0.7$.

Linear interpolation may converge more slowly than bisection, depending on f. To accelerate convergence either:

- take secant through $(a_i, \frac{1}{2}f(a_i))$ & $(b_i, f(b_i))$ if $a_i = a_{i-1}$ or through $(a_i, f(a_i))$ & $(b_i, \frac{1}{2}f(b_i))$ if $b_i = b_{i-1}$ — *modified linear interpolation* (may be slower than bisection);
- do bisection every few iterations, if convergence is slower than bisection — FZERO.

If f has root in $[a_i, b_i]$ but $f(a_i)f(b_i) > 0$, sign f bracketing fails. Instead a $|f|$ bracketing strategy is possible. Such a strategy may give convergence to a local maximum or minimum of f. Instead of $|f|$ bracketing, which requires extra function evaluations, FZERO chooses $[a_{i+1}, b_{i+1}]$ to give best decrease in $|f|$.