C477: One-Dimensional Optimisation

Ruth Misener
r.misener@imperial.ac.uk

Panos Parpas
p.parpas@imperial.ac.uk

Computational Optimisation Group
Department of Computing

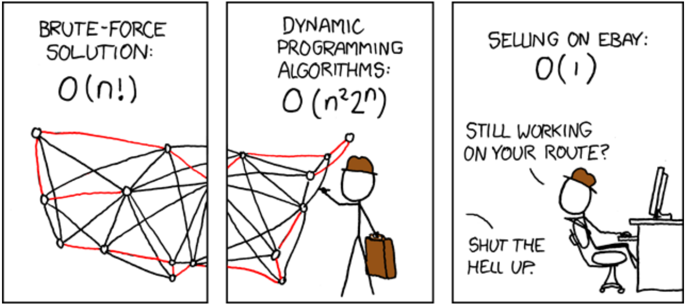
Imperial College London

computational optimisation group

20 January 2020

Trade-offs in Optimisation Algorithms

Traveling Salesman Problem:



- **Effectiveness:** Does the algorithm find what we want?
- **Efficiency:** What is the computation cost for doing it?

effectiveness

Find a point with as good objective value as possible

Find at least one global optimum point

Find all global optimum points

efficiency

Outline

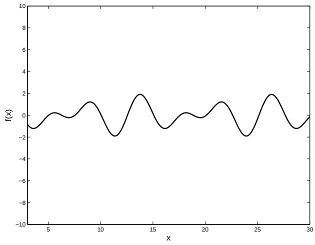
- **Topics**
  - Golden Section Search Method — 0^{th} Order Method
  - One Dimensional Newton's Method — 2^{nd} Order Method
    - ⋆ Newton's Method for computing Roots of Equations
    - ⋆ Convergence issues (cycling, local maxima, sensitivity to initial conditions)
    - ⋆ Fractal behaviour of Newton's Method (optional)
  - Secant Method — Quasi-Newton Method
  - Shubert's algorithm — Deterministic Global Optimisation
- **Definition: Lipschitz Continuity**
- **Reading**
  - Chapters 7 (One-Dimensional Search Methods) & 14 (Global Search Algorithms) in *An Introduction to Optimization*, Chong & Żak, Third Edition.
- **Acknowledgements**
  - Parts of these slides were originally developed by Benoit Chachuat and Panos Parpas. LaTeX design and proof reading by Miten Mistry. Mistakes by Ruth Misener.

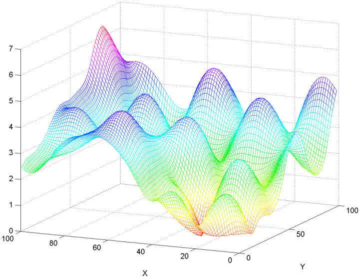
This sounds boring

Why only 1D? I want to solve problems in many dimensions!

Why we study optimisation problems in 1D

- Get **insight** into multivariable solution techniques;
- Single-variable optimisation is a **subproblem** for many nonlinear optimisation methods and software, e.g., linesearch;
- Mastering this lecture will help a lot with understanding the multivariate case.





0^{th} Order, 1^{st} Order, 2^{nd} Order, & Deterministic Global Optimisation Methods

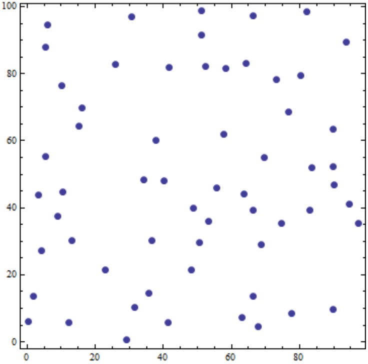
0^{th} Order Methods

- Only evaluate function $f(\boldsymbol{x})$;
- **Pro:** Best possible solution method for difficult-to-evaluate functions, e.g., black-box optimisation;†
- **Con:** Not great for many dimensions.†

†Exceptions exist; problem dependent

Examples of 0^{th} Order Methods

Nedler-Mead, Bayesian Optimisation

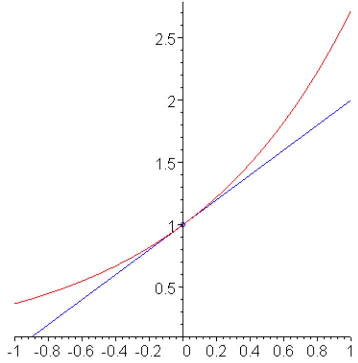


0^{th} Order, 1^{st} Order, 2^{nd} Order, & Deterministic Global Optimisation Methods

1^{st} Order Methods

- Function evaluations $f(\boldsymbol{x})$ & first-order derivatives $\nabla f(\boldsymbol{x})$;
- **Pro:** Only reasonable method for big problems;†
- **Con:** May converge slowly (compared to 2^{nd} order methods).†

†Exceptions exist; problem dependent



Examples of 1^{st} Order Methods

Steepest descent, Stochastic gradient descent, Alternating direction method of multipliers

0^{th} Order, 1^{st} Order, 2^{nd} Order, & Deterministic Global Optimisation Methods

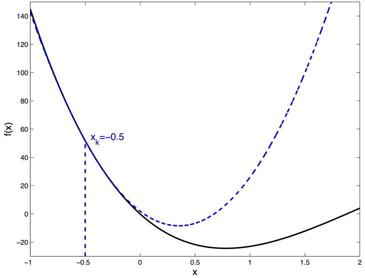
2^{nd} Order Methods

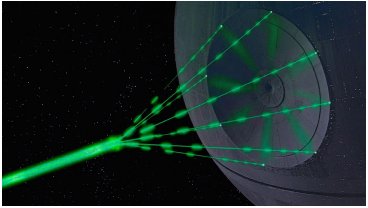
- Function evals $f(\boldsymbol{x})$, 1^{st} & 2^{nd} derivatives $\nabla f(\boldsymbol{x})$, $\nabla^2 f(\boldsymbol{x})$;
- **Pro:** Possibly quadratic (nice) convergence.†
- **Con:** May not be able to calculate second order derivatives for big problems.†

†Exceptions exist; problem dependent

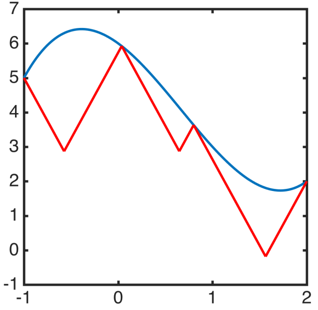
Examples of 2^{nd} Order Methods

Newton Methods, Quasi-Newton Methods*, BFGS* (*Approximate the Hessian)





0^{th} Order, 1^{st} Order, 2^{nd} Order, & Deterministic Global Optimisation Methods



Deterministic Global Optimisation

- Global information, e.g. # local minima, convexity properties;
- **Pro:** Guarantee best possible solution;
- **Con:** May not be appropriate for large problems.†

†Exceptions exist; problem dependent

Deterministic Global Methods

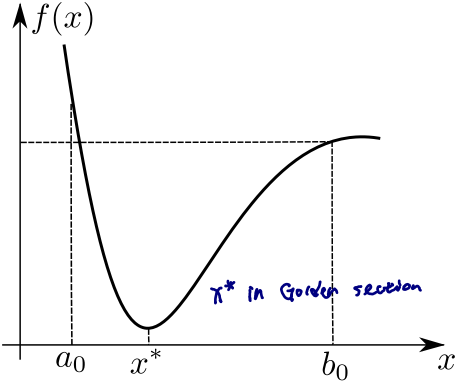
Branch & Cut, Branch & Bound, DIRECT, Lipschitz optimisation

Golden Section Search Method (0^{th} Order)

Assumptions

1. Unimodal (unique global minimum x^* in a given range $[a_0, b_0]$)
2. If $a_1 < a_2 < x^*$ then $f(x^*) < f(a_2) < f(a_1)$;
   If $x^* < a_1 < a_2$ then $f(x^*) < f(a_1) < f(a_2)$.

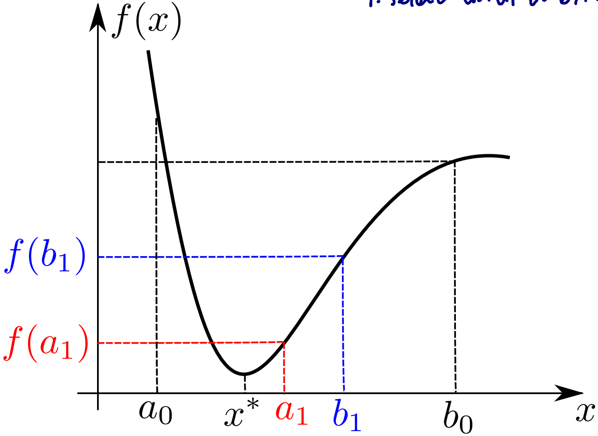
$f(a_i) \sim d(x^*, a_i)$



Golden Section Search Method (0^{th} Order)

1. Select two points such that $a_1 > a_0$, and $b_1 < b_0$.
2. Symmetric reduction: $(a_1 - a_0) = b_0 - b_1 = \varrho(b_0 - a_0)$, with $\varrho < \frac{1}{2}$.

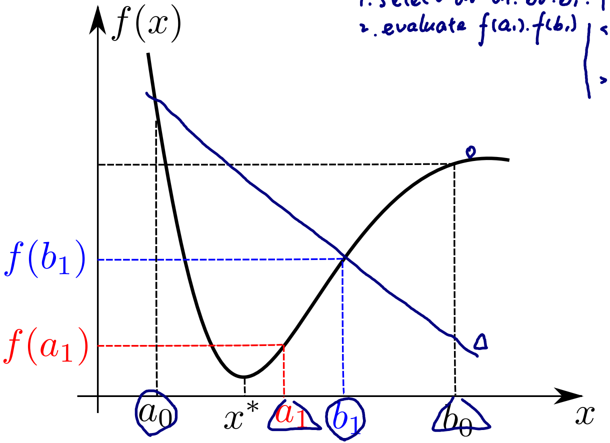
1. select $a_0 . a_1$ b_0 b_1. $\rho < \frac{1}{2}$



Golden Section Search Method (0^{th} Order)

1. Since $f(a_1) < f(b_1)$, reduce the search space to $[a_0, b_1]$.

1. select $a_0 \cdot a_1 \cdot b_0 \cdot b_1 \cdot f(\cdot)$
2. evaluate $f(a_1) \cdot f(b_1)$ $\begin{cases} <: [a_0, b_1] \\ >: [a_1, b_0] \end{cases}$



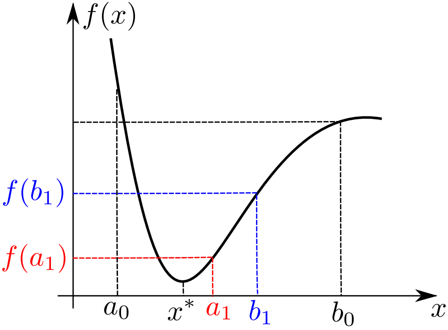
Golden Section Search Method (0^{th} Order)

Summary

1. Unique minimum between a known range;
2. 2 function evals reduce space: $(a_1 - a_0) = (b_0 - b_1) = \varrho(b_0 - a_0)$

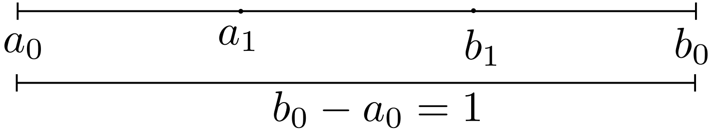
$$a_1 = a_0 + \varrho(b_0 - a_0),\ b_1 = a_0 + (1-\varrho)(b_0 - a_0)$$

3. Can we reduce the space with a single function evaluation?



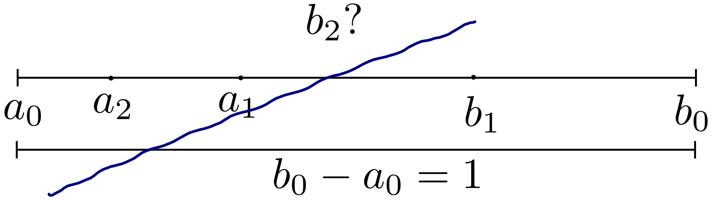
Golden Section Search Method (0^{th} Order)

1. $[a_0, b_0] = [0, 1]$.
2. After one iteration, assume $f(a_1) < f(b_1)$, reduce the search space to $[a_0, b_1]$



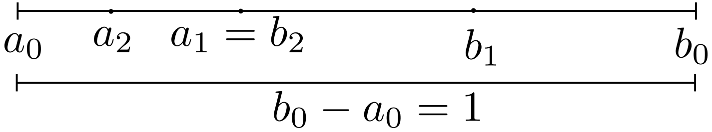
Golden Section Search Method (0^{th} Order)

1. Place a_2 such that $a_0 < a_2 < a_1$,
2. What about b_2?



Golden Section Search Method (0^{th} Order)

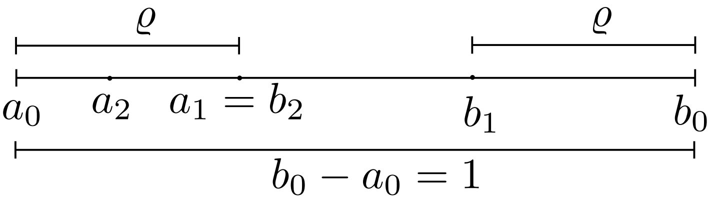
1. Take $b_2 = a_1$ to save one function evaluation. use previous result.
2. Choose ϱ for symmetric search space reduction.



Golden Section Search Method (0^{th} Order)

1. How to chose ϱ?
2. In first iteration

$$a_1 - a_0 = b_0 - b_1 = \varrho(b_0 - a_0)$$



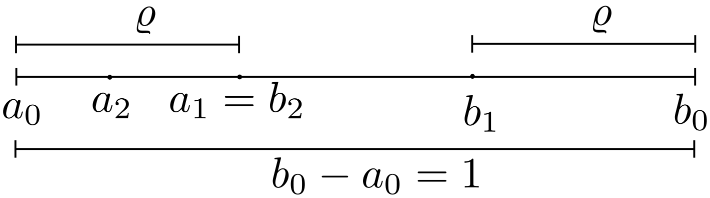
Golden Section Search Method (0^{th} Order)

1. How to chose ϱ?
2. In first iteration

$$a_1 - a_0 = b_0 - b_1 = \varrho(b_0 - a_0)$$

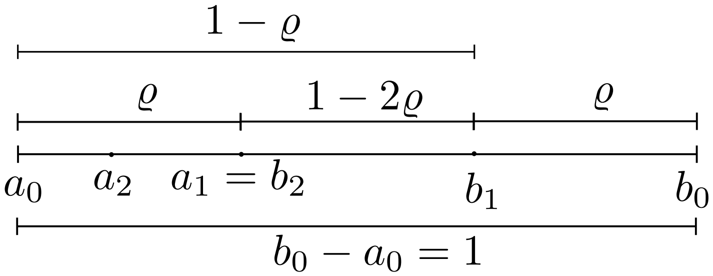
3. Choose ϱ such that

$$b_1 - b_2 = \varrho(b_1 - a_0)$$



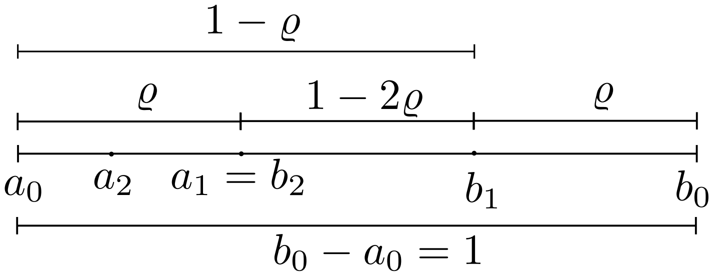
Golden Section Search Method (0^{th} Order)

1. Choose ϱ such that
$$b_1 - b_2 = \varrho(b_1 - a_0)$$
2. But $b_1 - b_2 = 1 - 2\varrho$ & $b_1 - a_0 = 1 - \varrho$
3. $1 - 2\varrho = \varrho(1 - \varrho)$



Golden Section Search Method (0^{th} Order)

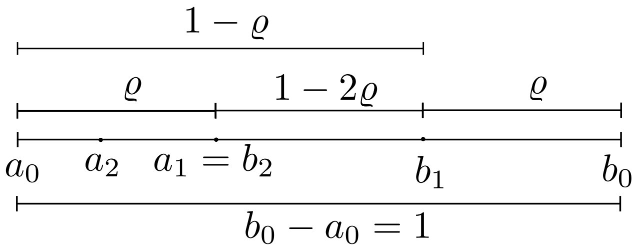
1. Choose ϱ such that $1 - 2\varrho = \varrho(1 - \varrho)$,
$$\varrho^2 - 3\varrho + 1 = 0$$
2. $\varrho = \frac{3 \pm \sqrt{5}}{2}$
3. Choose $\varrho = \frac{3 - \sqrt{5}}{2} \approx 0.382 < \frac{1}{2}$



Golden Section Search Method (0^{th} Order)

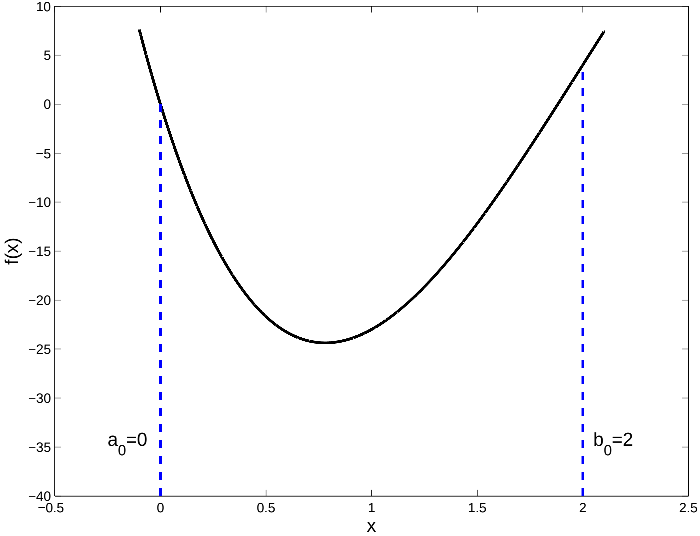
Summary

1. Main Assumption: Unique minimum between a known range
2. Use one function evaluation to reduce search space
3. Search space is reduced by $1-\varrho \approx 0.61803$ at every iteration



Example: Golden Section Search Method

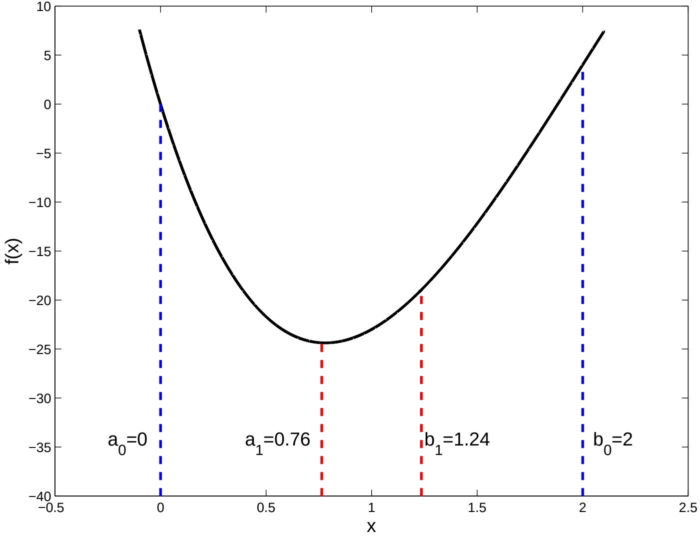
- $f(x) = x^4 - 14x^3 + 60x^2 - 70x$, $x^* \in [0, 2]$
- Locate x^* with an error tolerance of 0.3.



Example: Golden Section Search Method

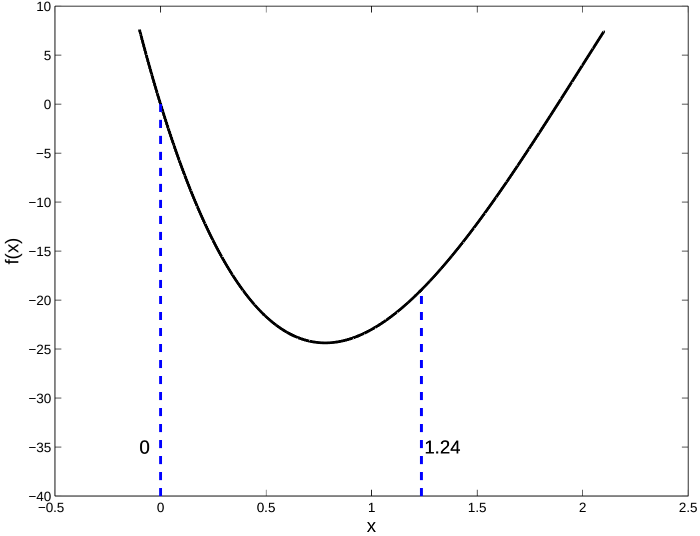
$\rho = \frac{3-\sqrt{5}}{2}$

- Iteration 1: $a_1 = a_0 + \varrho(b_0 - a_0)$, $b_1 = b_0 - \varrho(b_0 - a_0)$
- $f(a_1) = -24.36 < f(b_1) = -18.96 \implies x^* \in [a_0, b_1 = 1.24]$.



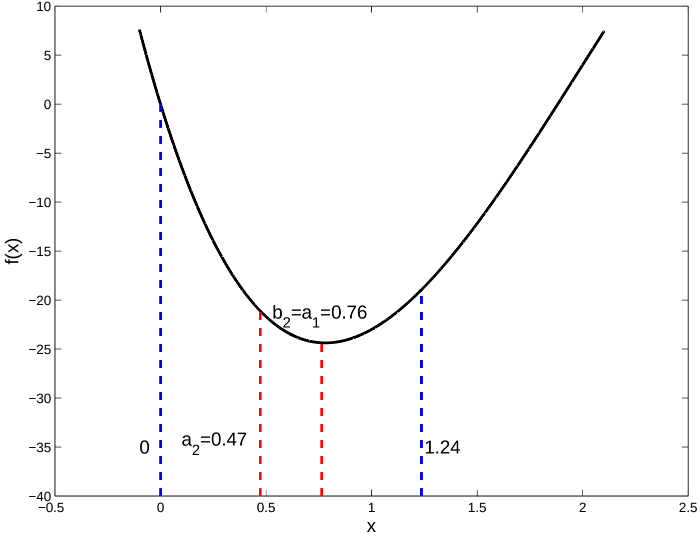
Example: Golden Section Search Method

- Iteration 2: $x^* \in [0, 1.24]$, $a_1 = 0.76, b_1 = 1.24$
- $a_2 = a_0 + \varrho(b_1 - a_0)$, $b_2 = a_1$.



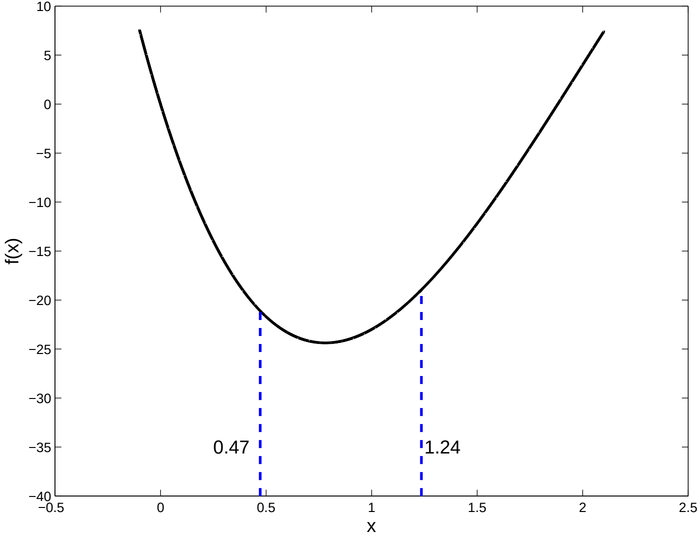
Example: Golden Section Search Method

- $a_2 = a_0 + \varrho(b_1 - a_0) = 0.47$, $b_2 = a_1 = 0.76$.
- $f(b_2) = -24.36 < f(a_2) = -21.10 \implies x^* \in [0.47, 1.24]$



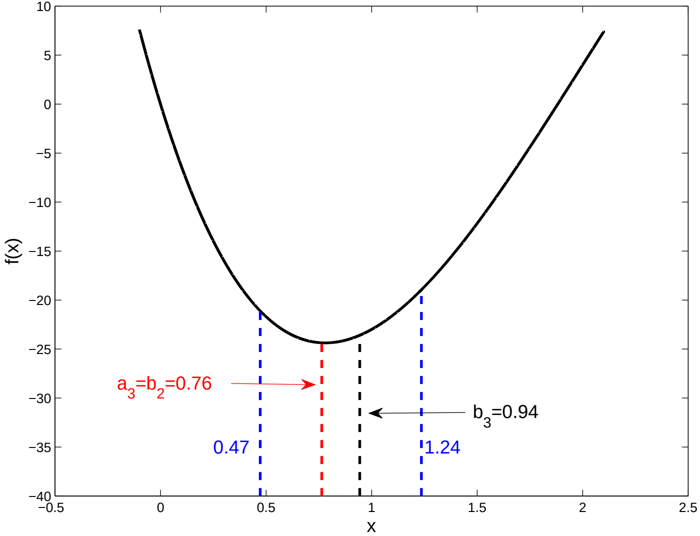
Example: Golden Section Search Method

- Iteration 3: $x^* \in [0.47, 1.24]$, $a_2 = 0.47, b_2 = 0.76$
- $a_3 = b_2$, $b_3 = b_1 - \varrho(b_1 - a_2)$



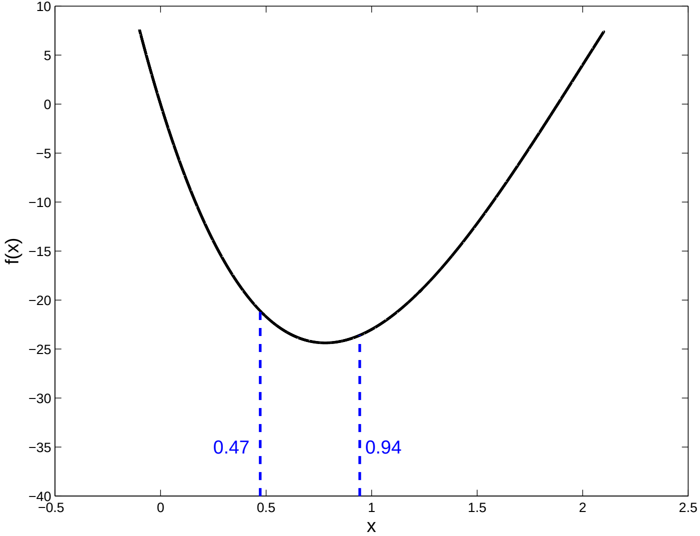
Example: Golden Section Search Method

- $a_3 = b_2 = 0.76,\ b_3 = b_1 - \varrho(b_1 - a_2) = 0.94$
- $f(b_3) = -23.59 > f(a_3) = f(b_2) = -24.36 \implies x^* \in [0.47, 0.94]$



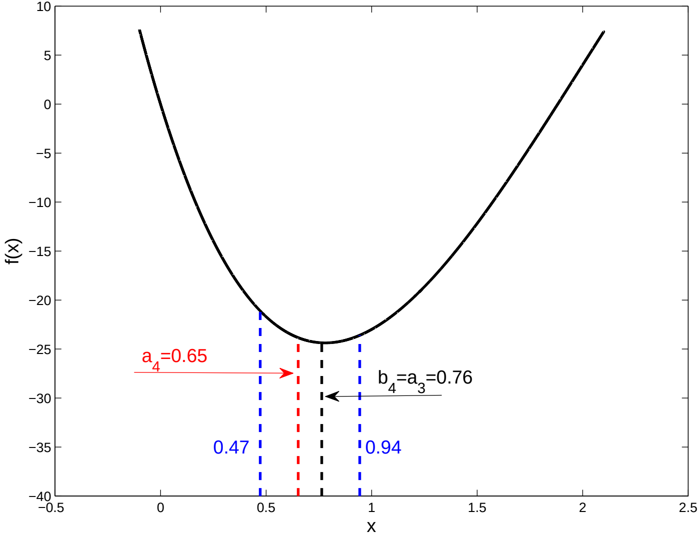
Example: Golden Section Search Method

- Iteration 4: $x^* \in [0.47, 0.94]$, $a_3 = 0.76, b_3 = 0.94$
- $a_4 = a_2 + \varrho(b_3 - a_2)$, $b_4 = a_3$



Example: Golden Section Search Method

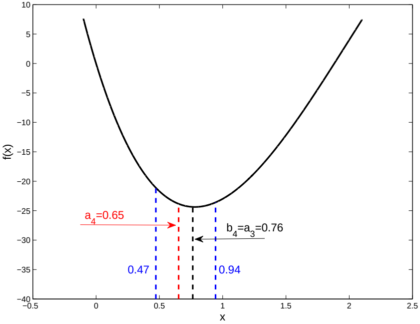
- $a_4 = a_2 + \varrho(b_3 - a_2) = 0.65,\ b_4 = a_3 = 0.76$
- $f(a_4) = -23.84 > f(b_4) = f(a_3) = -24.36 \implies x^* \in [0.65, 0.94]$



Example: Golden Section Search Method

Summary:

k	a_k	b_k	Interval
1	0.76	1.23	$[0\quad, 1.23]$
2	$a_2 = a_0 + \varrho(b_1 - a_0) = 0.47$	$b_2 = a_1 = 0.76$	$[0.47, 1.23]$
3	$b_2 = 0.76$	$b_3 = b_1 - \varrho(b_1 - a_2) = 0.94$	$[0.47, 0.94]$
4	$a_4 = a_2 + \varrho(b_3 - a_2) = 0.65$	$b_4 = a_3 = 0.76$	$[0.65, 0.94]$



One Dimensional Newton's Method (2^{nd} Order)

1-D Newton:
nonlinear
↓
quadratic approximation
↓
zero derivative
↓
iterate

1. Minimising a general non-linear function is difficult

$$\min \; f(x)$$

2. Basic idea: minimise a quadratic approximation

$$\min \; q(x)$$

Truncated Taylor series

where $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$

3. Minimise quadratic approximation

$$0 = q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

4. Use approximate minimiser as new starting point

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

Example: 1D Newton's Method (2^{nd} Order)

Example

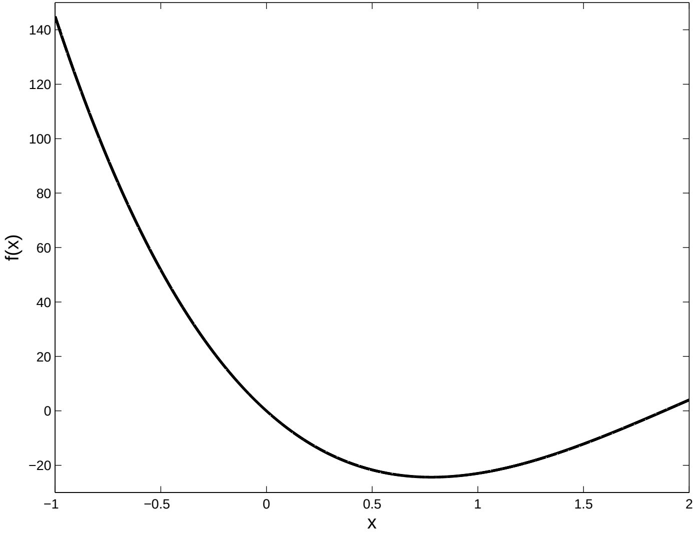
Use Newton's Method to find a minimiser of,

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x.$$

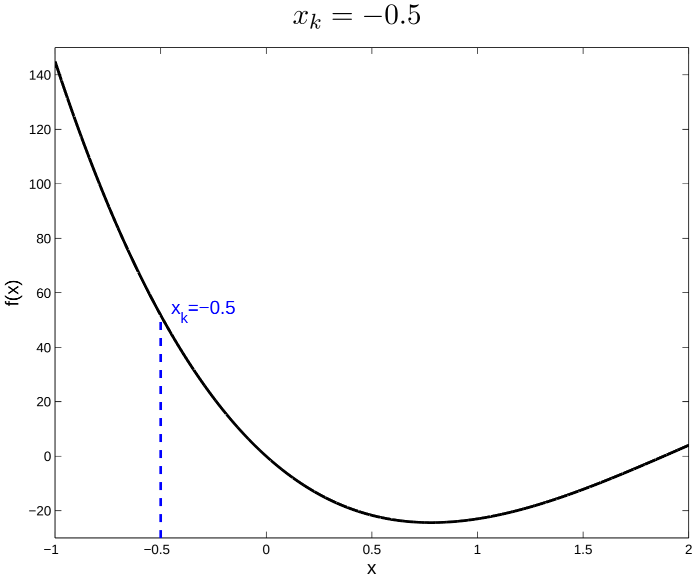
Start at $x(0) = -0.5$

Example: 1D Newton's Method (2^{nd} Order)

$$\min f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

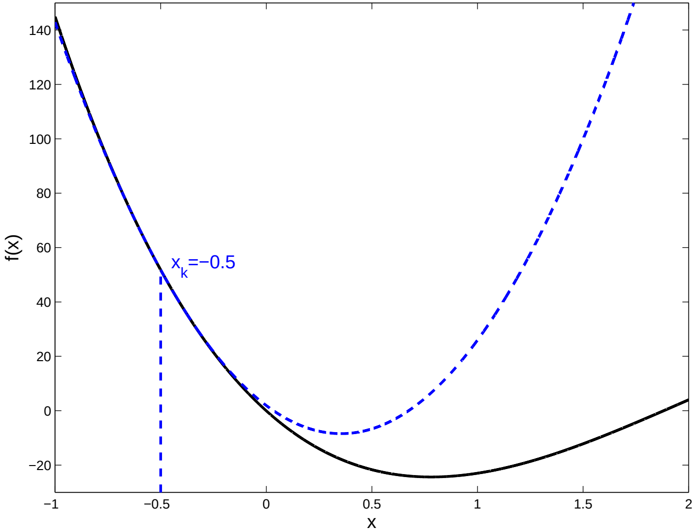


Example: 1D Newton's Method (2^{nd} Order)



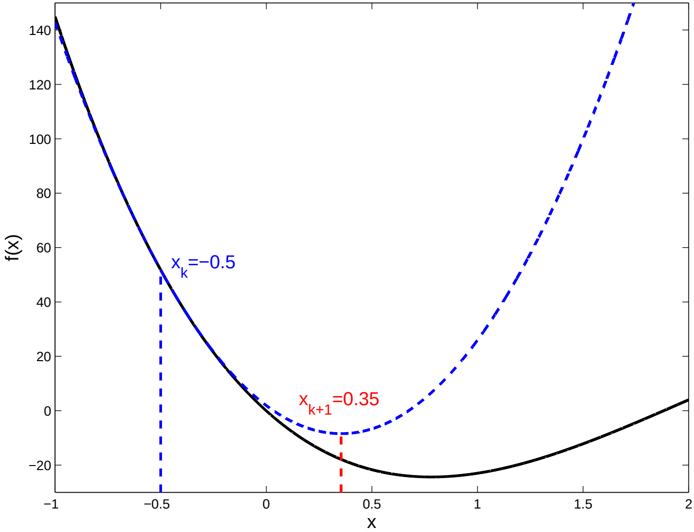
Example: 1D Newton's Method (2^{nd} Order)

$$q(x) = f(-0.5) + f'(-0.5)(x+0.5) + \frac{1}{2}f''(-0.5)(x+0.5)^2$$



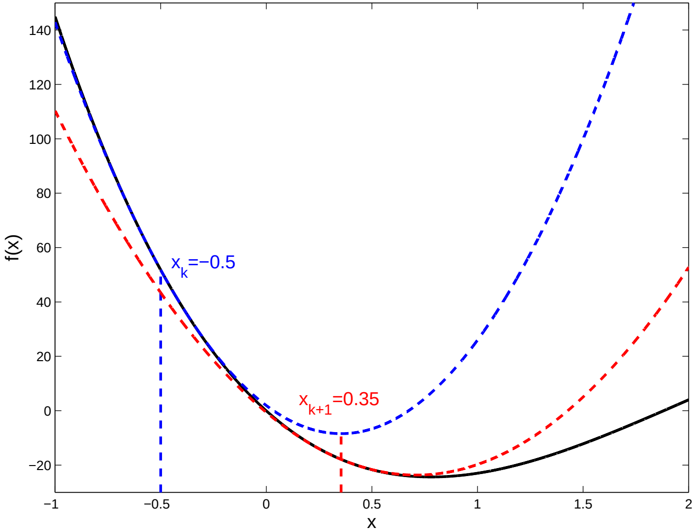
Example: 1D Newton's Method (2^{nd} Order)

$$x_{k+1} = \arg\min \;\; f(-0.5) + f'(-0.5)(x+0.5) + \frac{1}{2} f''(-0.5)(x+0.5)^2$$



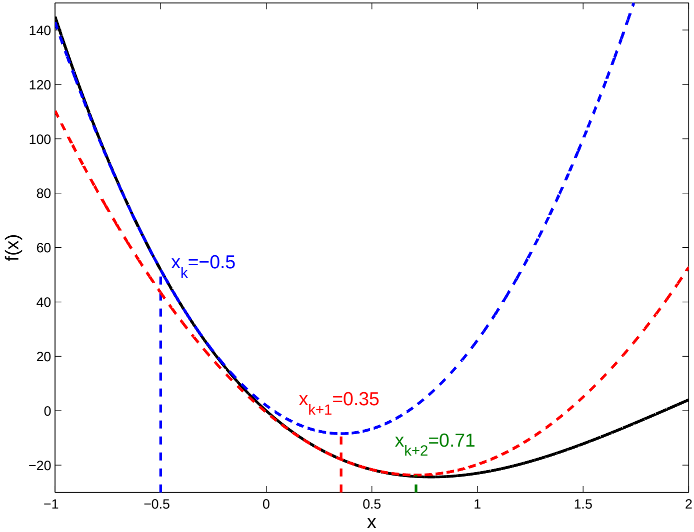
Example: 1D Newton's Method (2nd Order)

$$q(x) = f(0.35) + f'(0.35)(x - 0.35) + \frac{1}{2}f''(0.35)(x - 0.35)^2$$



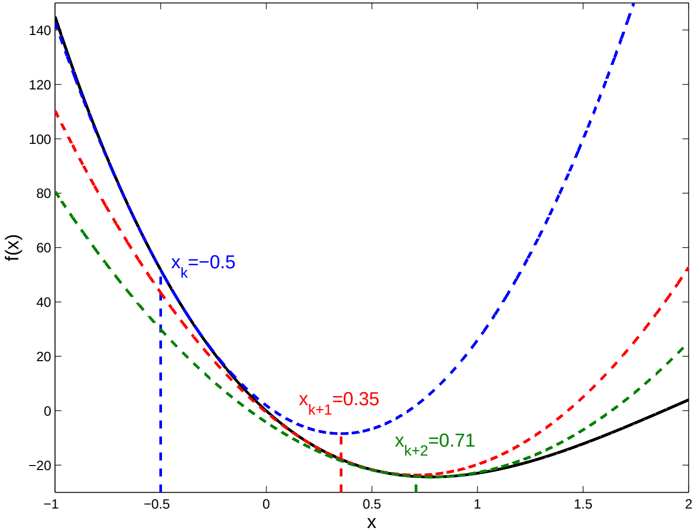
Example: 1D Newton's Method (2$^{\text{nd}}$ Order)

$$x_{k+2} = \arg\min \; f(0.35) + f'(0.35)(x - 0.35) + \frac{1}{2} f''(0.35)(x - 0.35)^2$$



Example: 1D Newton's Method (2^{nd} Order)

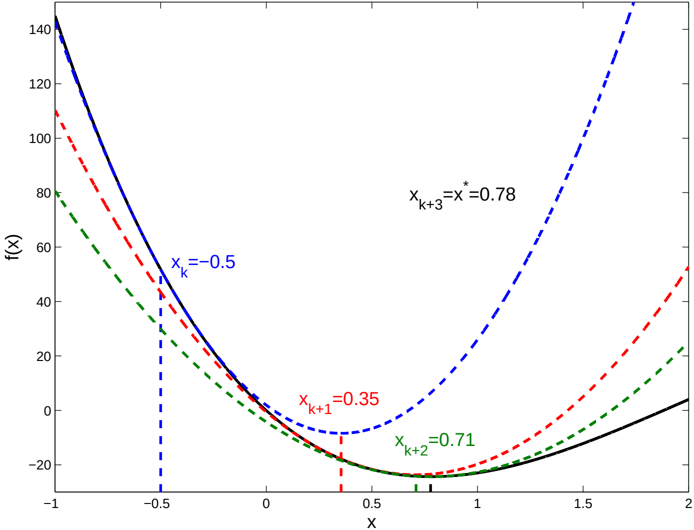
$$q(x) = f(0.71) + f'(0.71)(x - 0.71) + \frac{1}{2} f''(0.71)(x - 0.71)^2$$



Example: 1D Newton's Method (2^{nd} Order)

$$x_{k+3} = \arg\min \; f(0.71) + f'(0.71)(x - 0.71) + \frac{1}{2} f''(0.71)(x - 0.71)^2$$

Convergence after 3 iterations!



Newton's Method for computing Roots of Equations

Drive the first derivative of f to 0

- Newton's method can also be seen as a way to solve for

$$f'(x) = 0$$

using the iterative procedure,

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

- But if we set $g(x) = f'(x)$ then we obtain an algorithm for solving for $g(x) = 0$:

minimize $f(x)$

$\Updownarrow$

solve $g(x) = f'(x) = 0$

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$

Newton: $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = x_k - \frac{g(x_k)}{g'(x_k)}$

Example: Roots of Equations

Example

Use Newton's Method to find a root of,

$$f(x) = x^3 - 12.2x^2 + 7.45x + 42 = 0$$

Start at $x(0) = 12$. Perform two iterations.

Answer

$$\begin{aligned} x_1 &= 12.00 - \tfrac{102.6}{146.65} &= 11.30 \\ x_2 &= 11.30 - \tfrac{14.73}{116.11} &= 11.20 \end{aligned}$$

Equivalent Matlab Code

```
x      = 12
f      = x^3 - 12.2 * x^2 + 7.45 * x + 42
fprime = 3 * x^2 - 12.2 * 2 * x + 7.45
x_new  = x - f/fprime
```

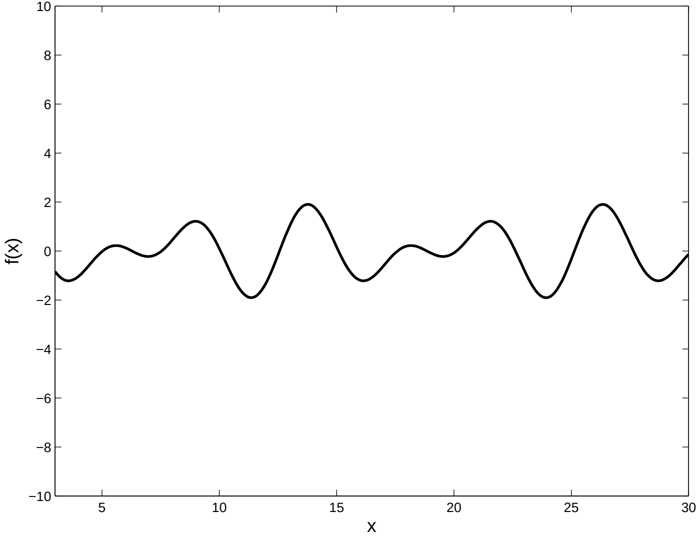
Failure to Converge (1D Newton's Method)

Warning!

- The algorithm can fail to converge if $f''(x) < 0$;
- Algorithm may find a point that satisfies the first order condition, not necessarily a minimiser;
- Algorithm may cycle;
- These issues will be addressed later on in the course.

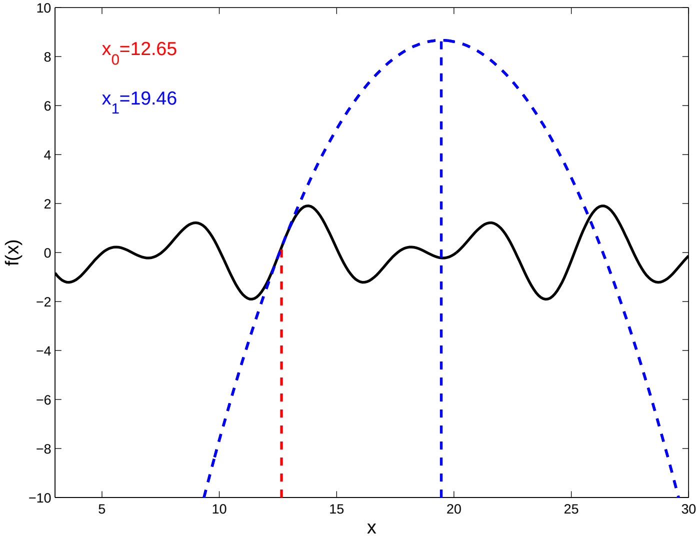
Example: Convergence Problems (1D Newton's Method)

$\min\ \sin(x) + \sin(3x/2)$, Initial Point $x_0 = 12.65$



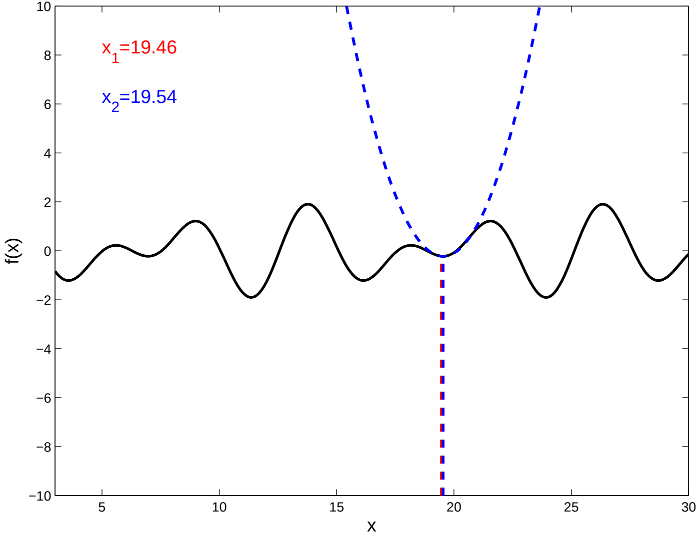
Example: Convergence Problems (1D Newton's Method)

$$x_0 = 12.65,\ f''(x_0) < 0,\ x_1 = 19.46$$



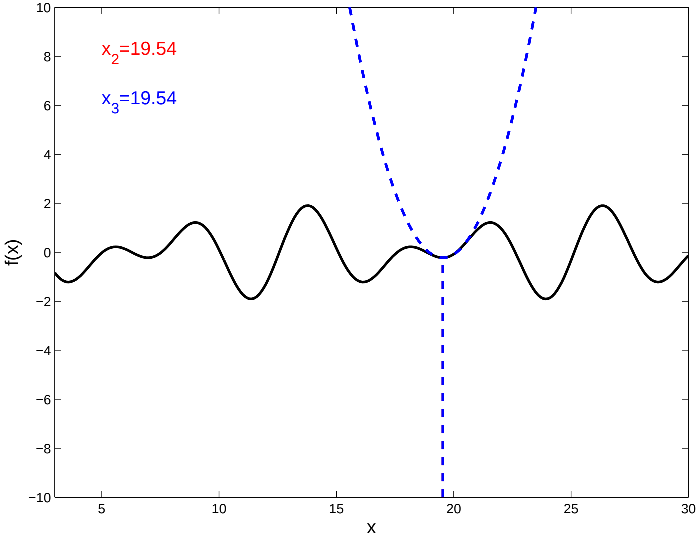
Example: Convergence Problems (1D Newton's Method)

$$x_1 = 19.46, \; f''(x_1) > 0, \; x_2 = 19.54$$



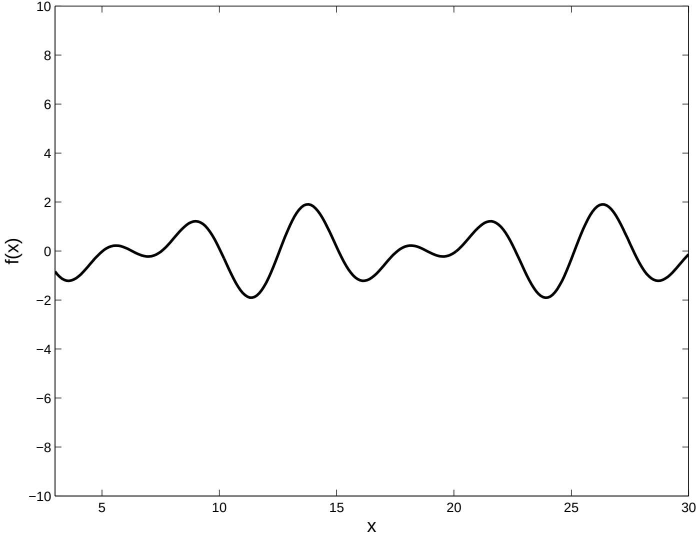
Example: Convergence Problems (1D Newton's Method)

$$x_2 = 19.54, \;\; x_3 = 19.54, \;\; f''(x_3) > 0, f'(x_3) \approx 0$$



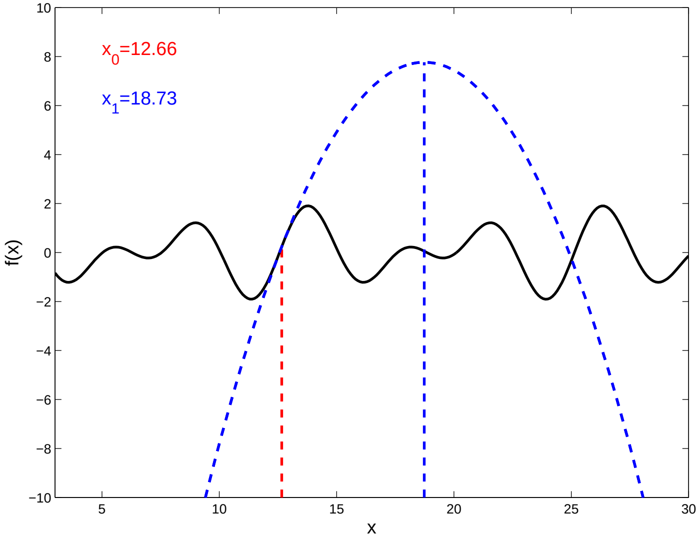
Example: Convergence Problems

$\min \; \sin(x) + \sin(3x/2)$, Initial Point $x_0 =$ ~~12.65~~ 12.66



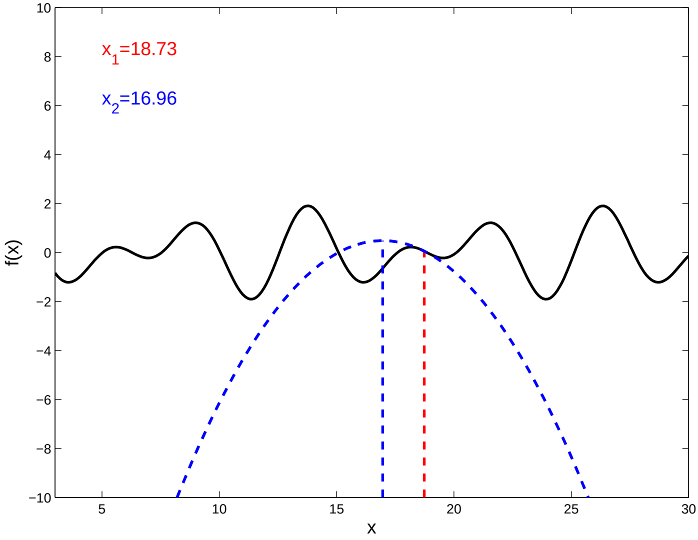
Example: Convergence Problems

$$x_0 = 12.66,\ f''(x_0) < 0,\ x_1 = 18.73$$



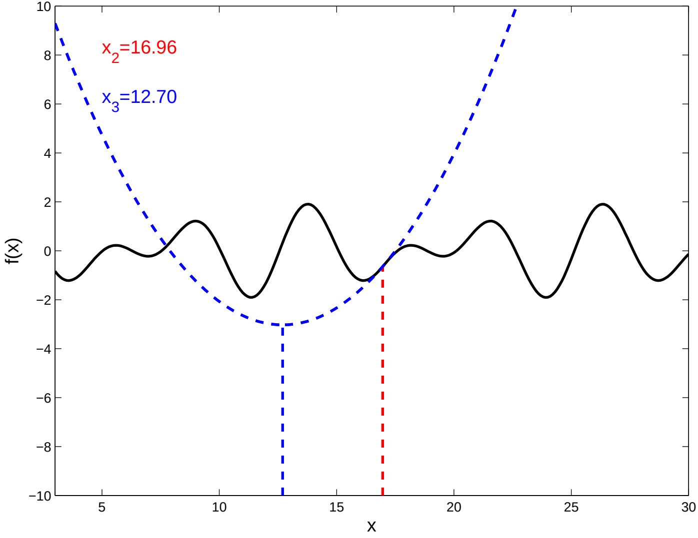
Example: Convergence Problems

$x_1 = 18.73,\ f''(x_1) < 0,\ x_2 = 16.96$



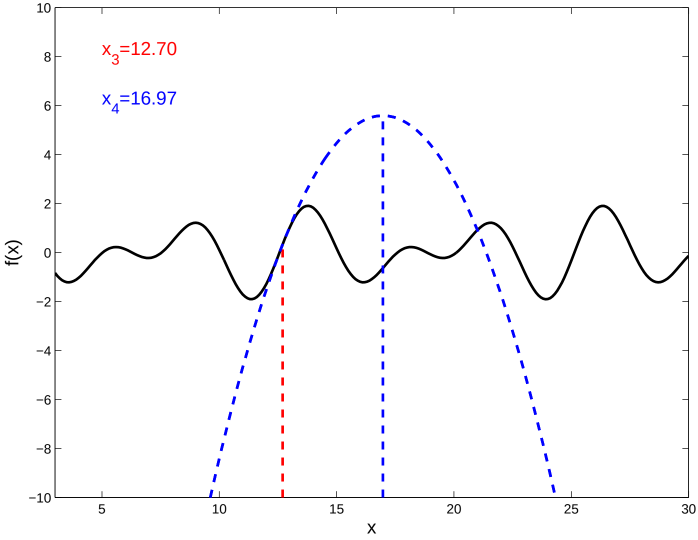
Example: Convergence Problems

$$x_2 = 16.96, \; f''(x_2) > 0, \; x_3 = 12.70$$



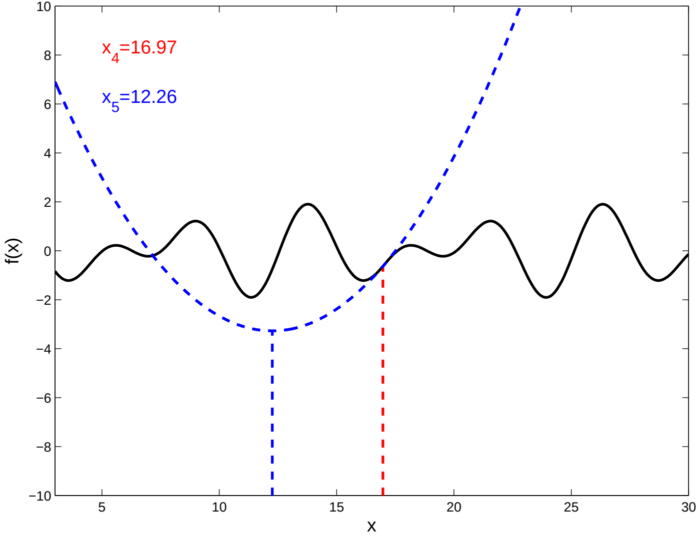
Example: Convergence Problems

$$x_3 = 12.70,\ f''(x_3) < 0,\ x_4 = 16.97$$



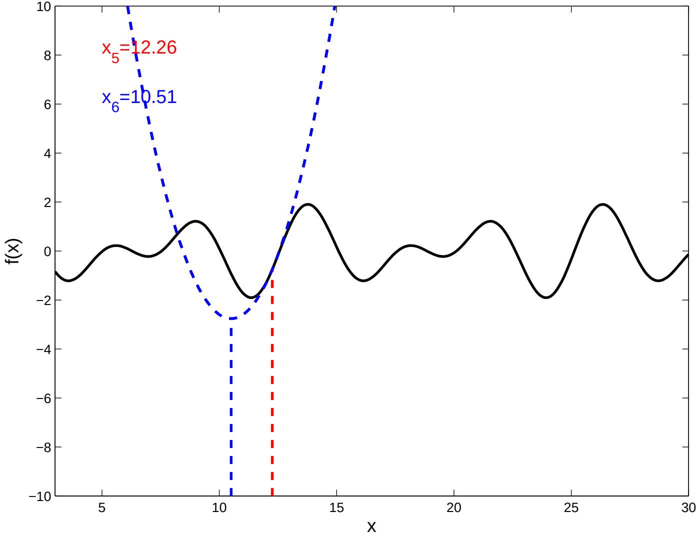
Example: Convergence Problems

$$x_4 = 16.97, \ f''(x_4) > 0, \ x_5 = 12.26$$



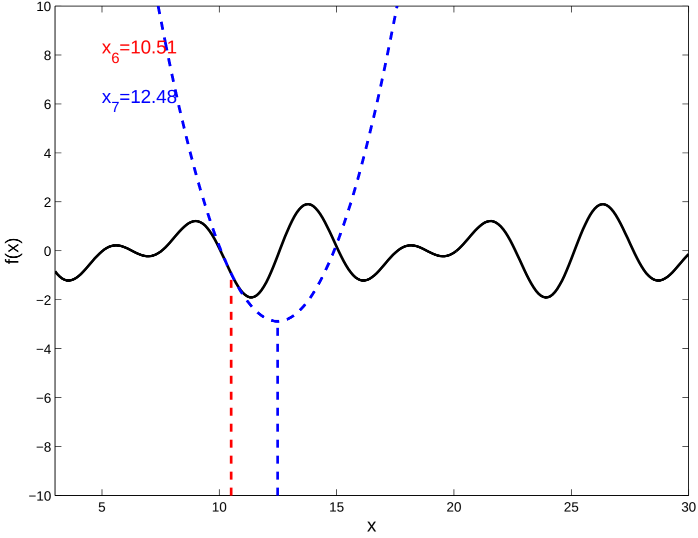
Example: Convergence Problems

$x_5 = 12.26, \ f''(x_5) > 0, \ x_6 = 10.51$



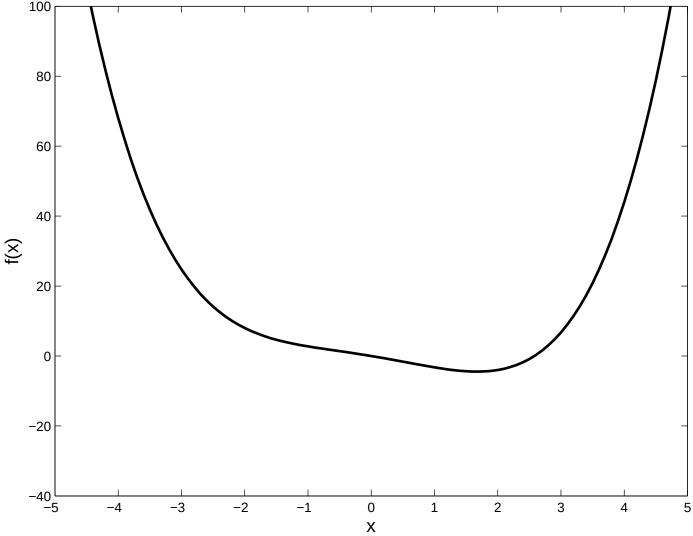
Example: Convergence Problems

$x_6 = 10.51$, $f''(x_6) > 0$, $x_7 = 12.48$

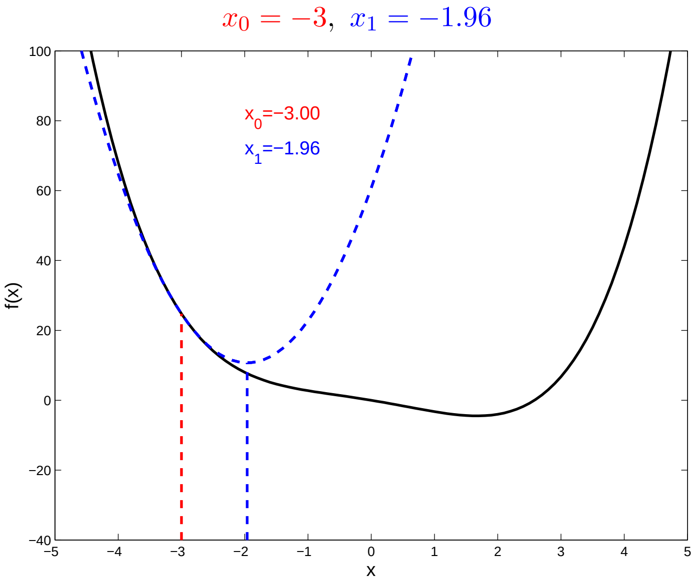


Example: Convergence Problems

$$\min\ \frac{1}{4}x^4 - \frac{1}{2}x^2 - 3x,$$ Initial Point $x_0 = -3.0$

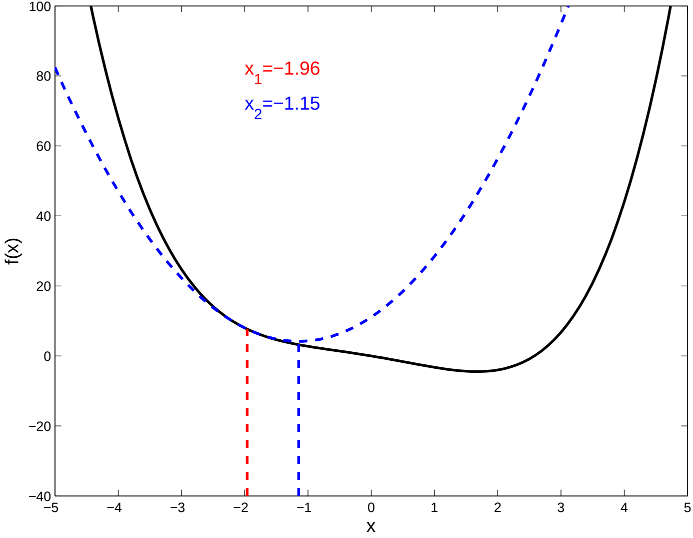


Example: Convergence Problems

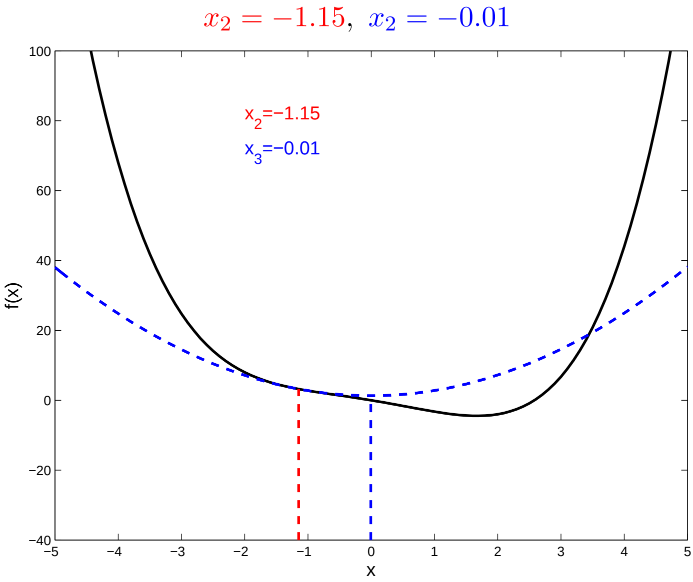


Example: Convergence Problems

$x_1 = -1.96,\ x_2 = -1.15$



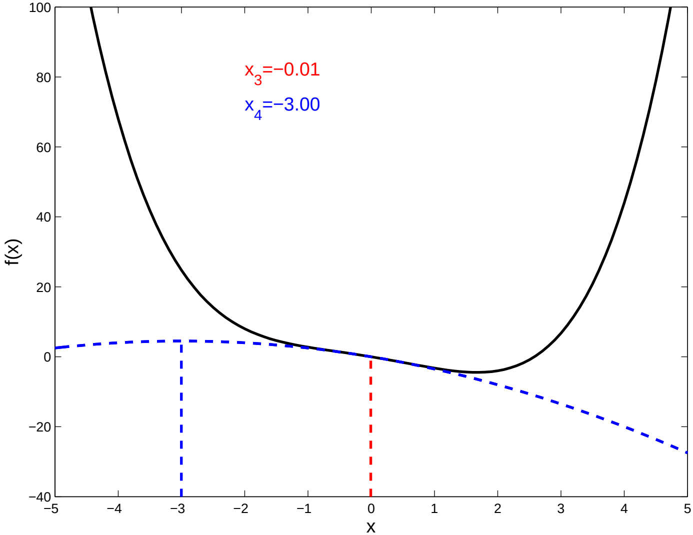
Example: Convergence Problems



Example: Convergence Problems

$$x_3 = -0.01, \; x_4 = -3.00 = x_0$$

The algorithm returns to the initial point!



Newton's Method the complex case (Optional)

- Newton's method can be used to find complex roots
- Same algorithm, but z may be complex i.e. $z = x + iy$

$$z_{k+1} = z_k - \frac{g(z_k)}{g'(z_k)}$$

- Hint: Starting point must be complex, otherwise algorithm will never leave the real plane.

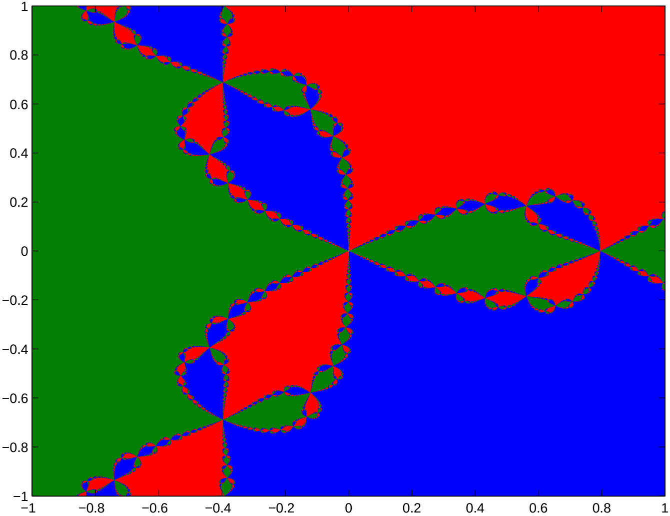
(Optional) Example: `fractal.m`

- Use Newton's Algorithm to find all roots of the equation,

$$g(z) = z^3 + 1$$

- The roots are given by, $\{\exp(i\pi/3), \exp(i\pi), \exp(5i\pi/3)\}$
- Initialise the algorithm from different points in the complex plane
- Set the initial condition to $z_0 = x_0 + iy_0$, and let x_0, y_0 range from -1 to 1 with an interval of 0.01.
- Use different colour to colour each of the different roots
- The result is a fractal.

(Optional) Example: `fractal.m`



Secant Method (Quasi-Newton Method)

Newton's Method

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

- Newton's Method uses first & second derivatives.
- We can approximate the second derivative with,

$$\frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}.$$

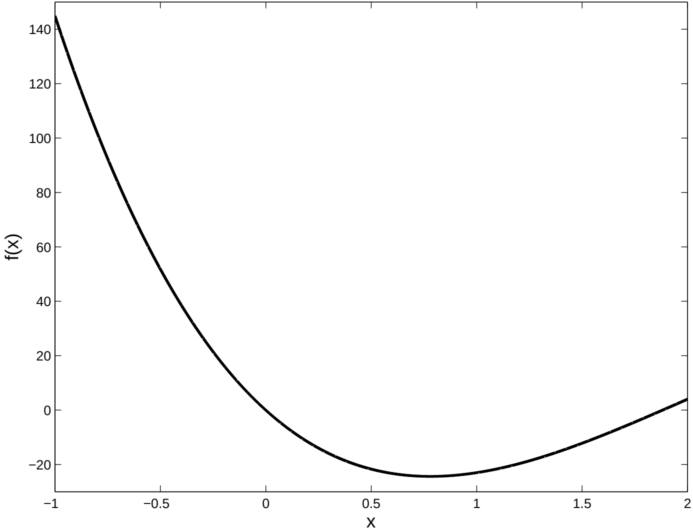
- Secant Method Uses this approximation into Newton's iteration.

Secant Method

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

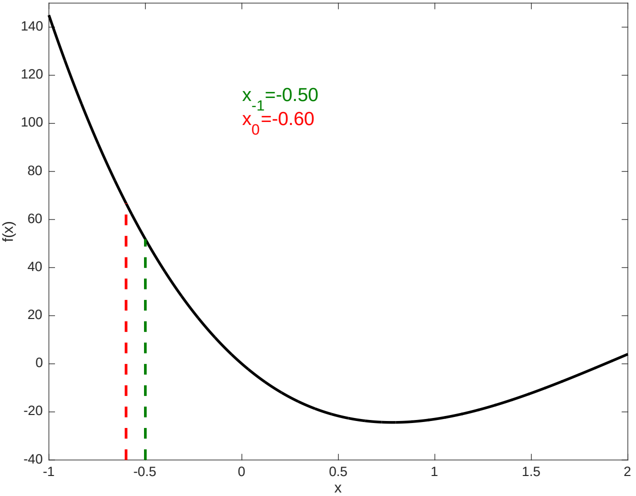
Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



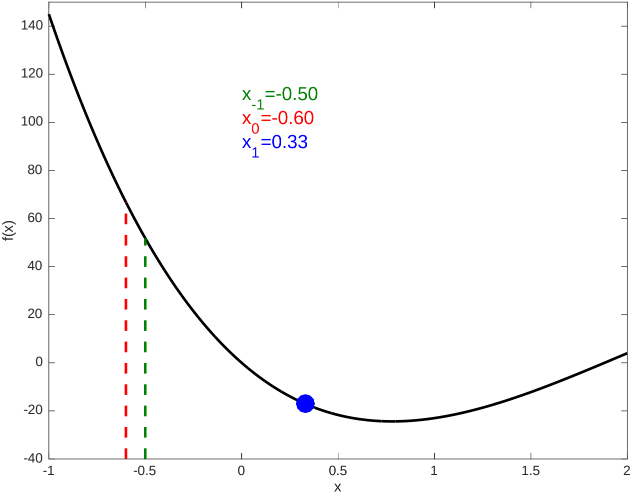
Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



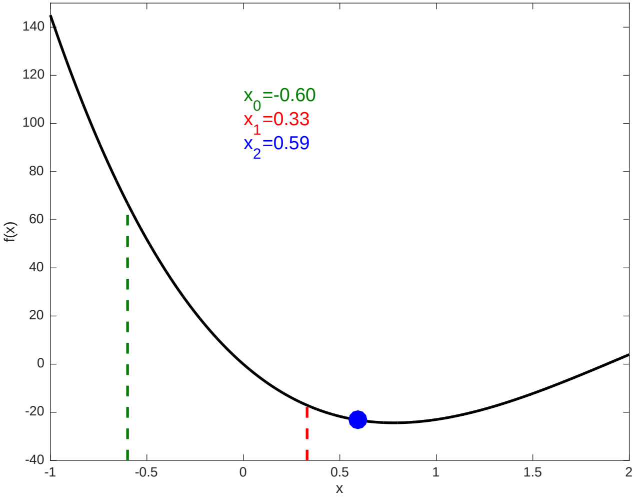
Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



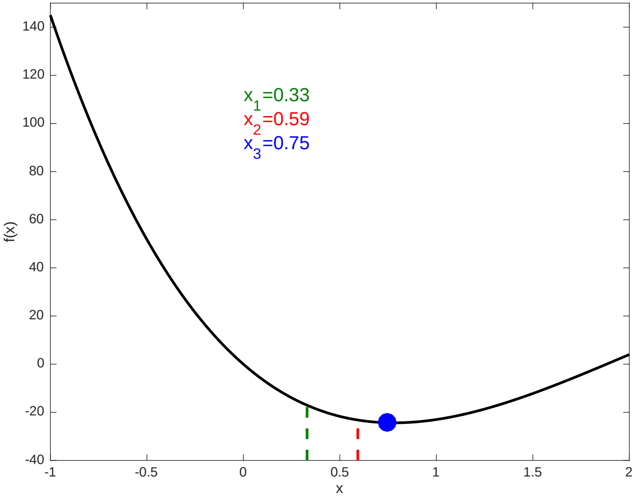
Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



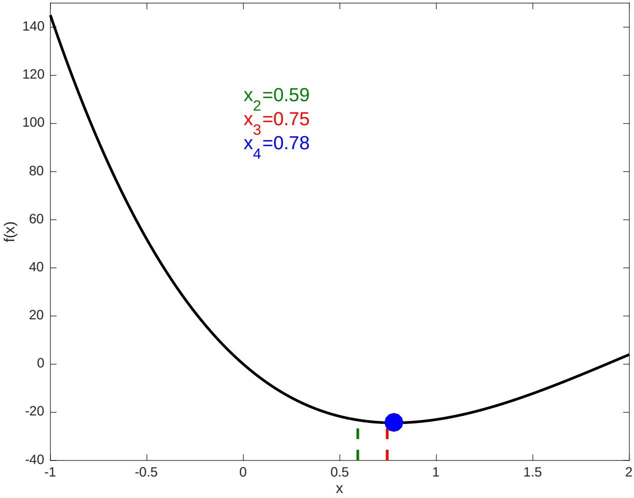
Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



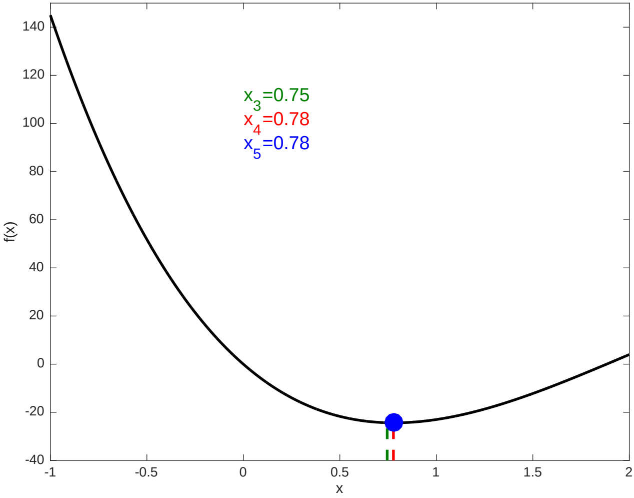
Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



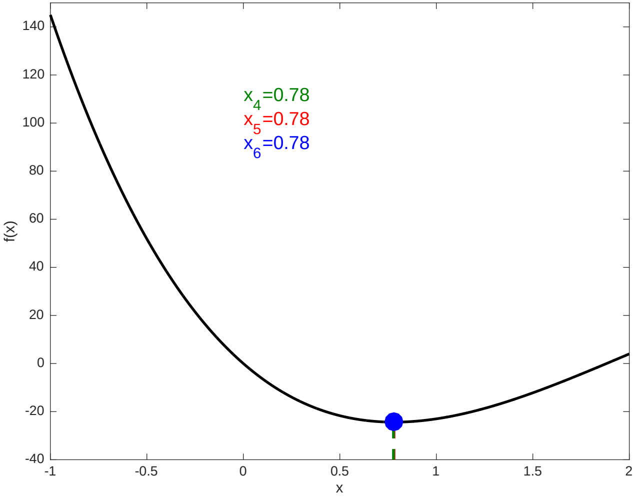
Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



Example: Secant Method (Quasi-Newton Method)

Use the Secant Method to find a minimiser of $x^4 - 14x^3 + 60x^2 - 70x$



Global Information

Level of Information

Without global information, no algorithm can provide a certificate of global optimality, unless it generates a dense sample

Examples of Global Information:

- Number of local optima
- Global optimum value
- Convexity of the objective function and feasible region
- Lipschitz constant L,

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \leq L\,\|\mathbf{x} - \mathbf{y}\|_2\,, \quad \forall \mathbf{x}, \mathbf{y} \in S$$

Algorithmic Strategies:

1. Unstructured Problems: Global information unavailable
   ➥ Global optimum certificate is hopeless!
2. Structured Problems: Global information available
   ➥ Global optimality may be certified, but...

Lipschitz Continuity

Definition: Lipschitz Continuity

A function $f : S \mapsto \mathbb{R}^m$ where $S \subseteq \mathbb{R}^n$ is called Lipschitz continuous if there is a constant $L \in \mathbb{R}$ such that:

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \leq L\,\|\mathbf{x} - \mathbf{y}\|_2\,, \quad \forall \mathbf{x}, \mathbf{y} \in S$$

Intuition?

A function is not allowed to change too quickly.

Sanity Check

Are the following functions Lipschitz continuous?

1. $f(x) = x^{1/3}$ on the domain $x \in [0,\, 1]$;
2. $f(x) = x^2$ on the domain $x \in (-\infty,\, \infty)$;
3. $f(x) = \exp(x)$ on the domain $x \in (-\infty,\, 1]$;
4. $f(x) = |x|$ on the domain $x \in (-\infty,\, \infty)$.

Lipschitz Continuity

1. $f(x) = x^{1/3}$ on the domain $x \in [0, 1]$

2. $f(x) = x^2$ on the domain $x \in (-\infty, \infty)$

Lipschitz Continuity

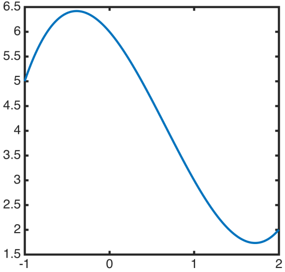
3. $f(x) = \exp(x)$ on the domain $x \in (-\infty,\, 1]$

4. $f(x) = |x|$ on the domain $x \in (-\infty,\, \infty)$

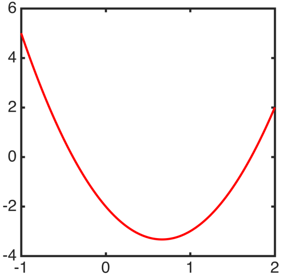
Lipschitz Optimisation

$f(x) = (x - 2) + (x - 3)^2 + (x - 1)^3,\ x \in [-1,\ 2]$

1. What are the local minima? What is the global minimum?
2. Convex function?
3. Satisfies conditions for Golden Section method?
4. What is the Lipschitz constant?



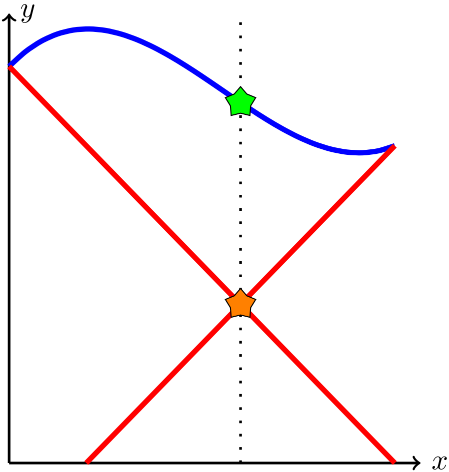
$f(x) = (x-2) + (x-3)^2 + (x-1)^3$



$f'(x) = 1 + 2(x-3) + 3(x-1)^2$

Use Lipschitz Constant for **Global Optimisation**? [1/2]

Deduce lower bounds on the global solution!!



$$f(x) = (x-2) + (x-3)^2 + (x-1)^3$$

$$f(x) \geq -5(x - x_1) + y_1$$
$$= -5(x+1) + 5$$

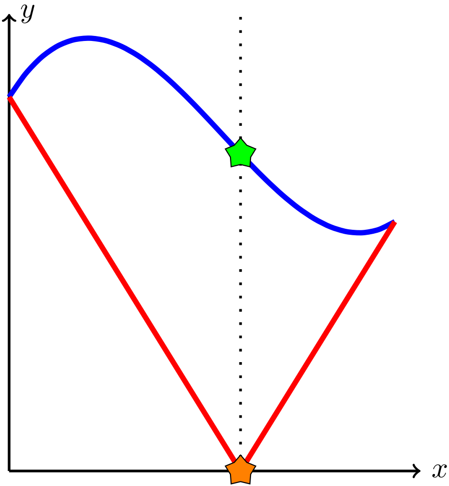
$$f(x) \geq 5(x - x_2) + y_2$$
$$= 5(x-2) + 2$$

Intersection Point? $\hat{x} = [0.8, -4]$

Converged? $f(\bigstar) - f(\bigstar) < \epsilon$?

Use Lipschitz Constant for **Global Optimisation**? [2/2]

Initiate divide and conquer algorithm similar to **Branch & Bound**



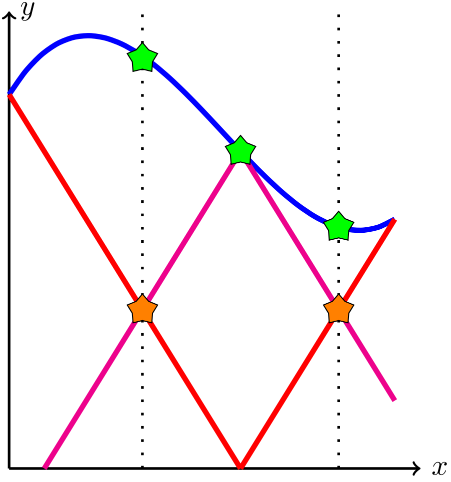
Key Observations

- This is Shubert's algorithm;
- Easy to find intersection of two lines!
- Typical convergence criteria?
  $f^{\mathsf{UB}} - f^{\mathsf{LB}} < \epsilon$
  $\min\ f(\bigstar) - \min\ f(\bigstar) < \epsilon$?
- Is this as bad as complete search?

Function Evaluation Points: $\{0.8,\ 0.0368,\ 1.5632,\ -0.5740,\ 0.6476\}$

Use Lipschitz Constant for **Global Optimisation**? [2/2]

Initiate divide and conquer algorithm similar to **Branch & Bound**



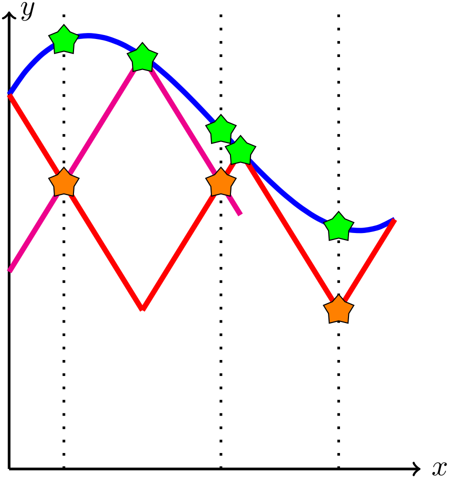
Key Observations

- This is Shubert's algorithm;
- Easy to find intersection of two lines!
- Typical convergence criteria?
  $f^{\mathsf{UB}} - f^{\mathsf{LB}} < \epsilon$
  $\min\ f(\ ) - \min\ f(\ ) < \epsilon$?
- Is this as bad as complete search?

Function Evaluation Points: $\{0.8,\, 0.0368,\, 1.5632,\, -0.5740,\, 0.6476\}$

Use Lipschitz Constant for **Global Optimisation**? [2/2]

Initiate divide and conquer algorithm similar to **Branch & Bound**



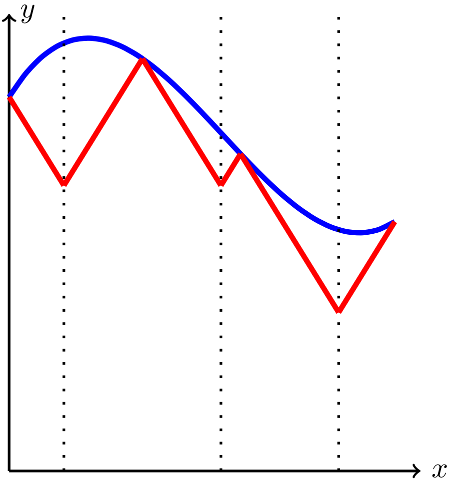
Key Observations

- This is Shubert's algorithm;
- Easy to find intersection of two lines!
- Typical convergence criteria?
  $f^{\text{UB}} - f^{\text{LB}} < \epsilon$
  $\min\ f(\text{green star}) - \min\ f(\text{orange star}) < \epsilon$?
- Is this as bad as complete search?

Function Evaluation Points: $\{0.8,\, 0.0368,\, 1.5632,\, -0.5740,\, 0.6476\}$

Use Lipschitz Constant for **Global Optimisation**? [2/2]

Initiate divide and conquer algorithm similar to **Branch & Bound**

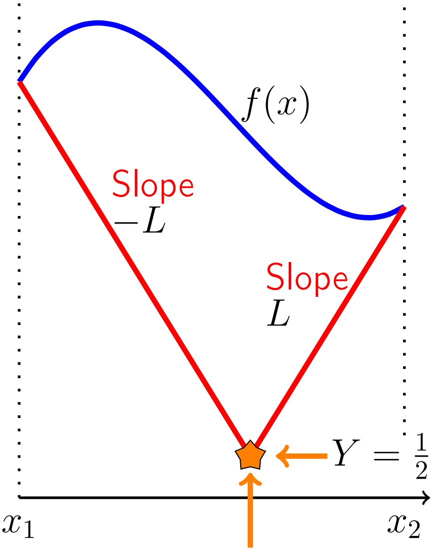


Key Observations

- This is Shubert's algorithm;
- Easy to find intersection of two lines!
- Typical convergence criteria?
  $f^{\mathsf{UB}} - f^{\mathsf{LB}} < \epsilon$
  $\min\ f(\star) - \min\ f(\star) < \epsilon$?
- Is this as bad as complete search?

Function Evaluation Points: $\{0.8,\, 0.0368,\, 1.5632,\, -0.5740,\, 0.6476\}$

Lipschitz Optimisation: Shubert's Algorithm in 1D



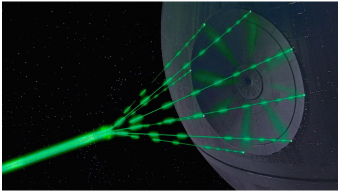
$$f(x) \geq f(x_1) - L(x - x_1)$$
$$f(x) \geq f(x_2) + L(x - x_2)$$

$$Y = \frac{1}{2}\left(f(x_1) + f(x_2) - L\left(x_2 - x_1\right)\right)$$

$$\hat{x} = \frac{1}{2}\left(x_1 + x_2 + \frac{1}{L}\left(f(x_1) - f(x_2)\right)\right)$$

Summary

0^{th} Order Methods



2^{nd} Order Methods

1^{st} Order Methods

Deterministic Global Methods