## Predictor - Corrector Methods

$$y(x_{n+1}) = y(x_n) + \underbrace{\int_{x_n}^{x_{n+1}} dx' f(x', y)}$$

Key idea: Use the classical extrapolation formulas for finding integration based on function interpolation (see chapter on function interpolation and finite difference schemes).

Let $f_n = f(x_n, y_n)$

Then

$$y_{n+1} = y_n + h\left(\beta_0 f_{n+1} + \beta_1 f_n + \beta_2 f_{n-1} + \beta_3 f_{n-2} + \cdots\right)$$

with $\beta_0 = 0$, the integration is explicit, otherwise it is implicit.

Basic idea: Integrate twice.

(1) Prediction step:

Get $y_{n+1}$ using an explicit scheme.

(2) Correction step:

Recalculate $y_{n+1}$, using an implicit scheme with $f_{n+1}$ gotten from the prediction step.

In practice:

Adams - Bashforth - Moulton scheme:

(1) $y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}) + O(h^4)$

(2) $y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1}) + O(h^4)$

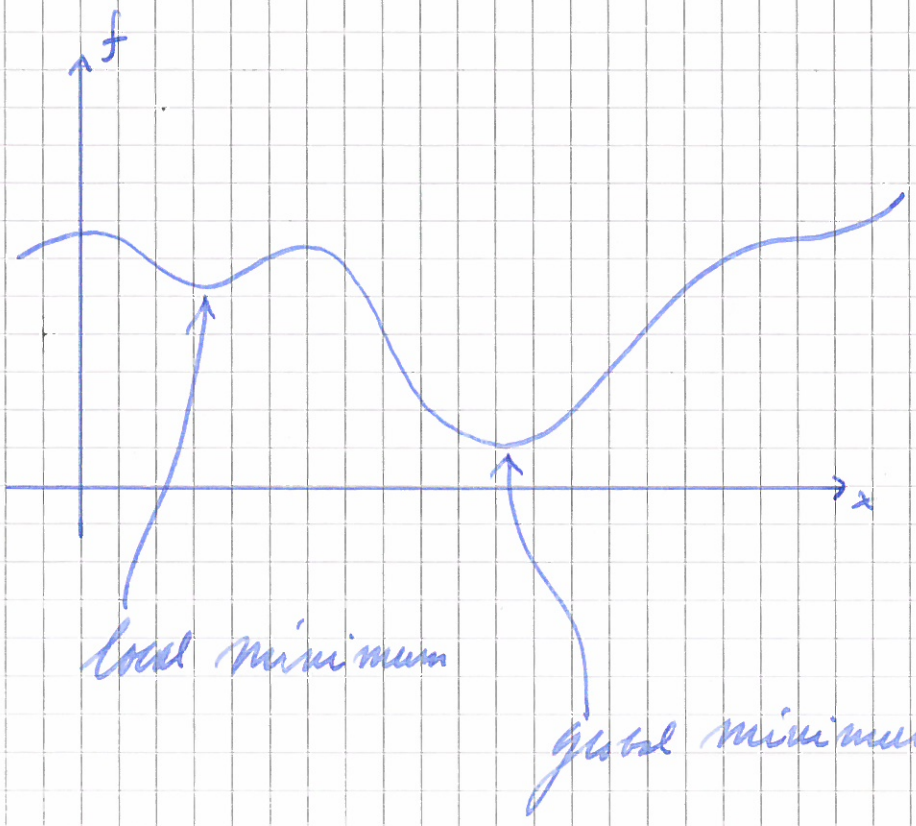
---

## Optimization

$\min(f) = -\max(-f)$

$\Downarrow$

Same problem.




It is <u>easy</u> to find <u>local minima.</u>

It is <u>extremely difficult</u> to find <u>global minima.</u>

$f(x)$

- $x$ – 1 dimensional / N dimensional
- $f$ – linear / nonlinear
- Constraints / No constraints

## When $x$ is one dimensional:

### Local minima:

Find $x$ such that

$$f(x) = \min_{x' \in [a,b]} f(x')$$

---

## Simplest method: Bracketing.

$$a < b < c$$

If $f(b) < f(a)$ and

$f(b) < f(c)$,

then there is a minimum

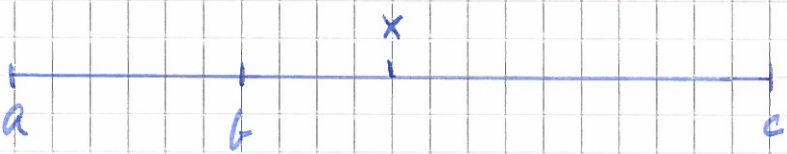
in the interval $[a, c]$.

## Bisection method

We start with $a, b, c$ such that

$f(b) < f(a)$ and $f(b) < f(c)$.

Define a new point $x = \frac{1}{2}(a+c)$

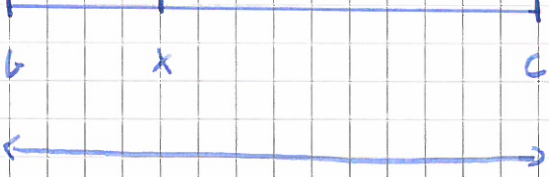



If $b < x < c$.

$(a, b, c) \rightarrow (a, b, x)$

New interval if

$f(b) < f(x)$




New interval if

$f(b) > f(x)$.

$(a, b, c) \rightarrow (b, x, c)$.

Likewise if $a < x < b$.

Repeat - and the minimum will be inside an ever-decreasing interval.

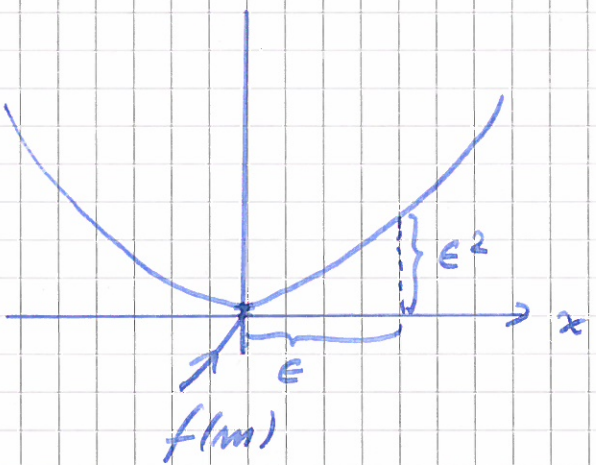
---

There is a problem with accuracy:

Close to the minimum:

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

$m$ minimum - $x$.




One cannot determine with $f(x) \geq f(m)$ with precision better than $\epsilon^2$:

Machine precision: $10^{-8}$ ⇒

accuracy in $x$: $10^{-4}$.

Midpoint method is not optimal

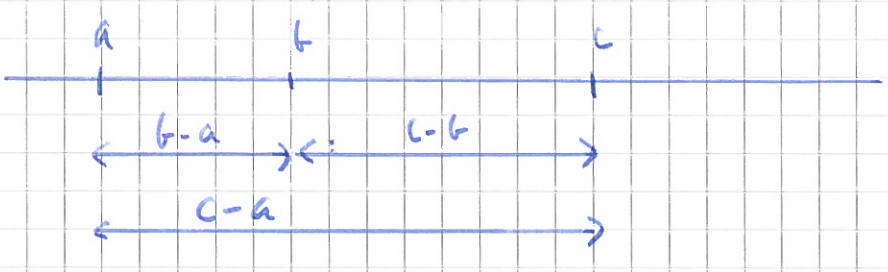
$(x = \frac{1}{2}(a+c))$.

Optimal choice: Golden mean!

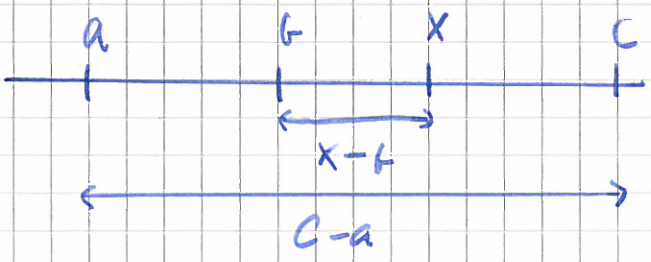
Golden mean search algorithm.

Given $(a, b, c)$.

Define $w = \frac{b-a}{c-a} \Rightarrow \frac{c-b}{c-a} = 1-w$.

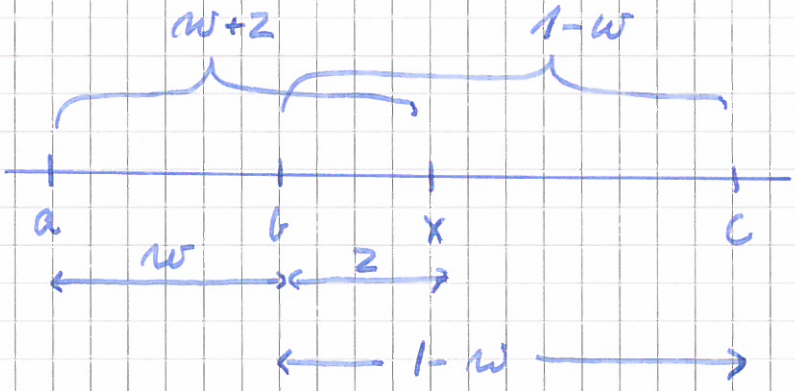



Assume that $x$ is placed such that $\frac{x-b}{c-a} = z$.




Next bracket will be

$(a, b, x)$ or $(b, x, c)$




We choose x such that the two relative lengths are equal.

Equal probability of finding minimum in either interval.

$$w + z = 1 - w$$

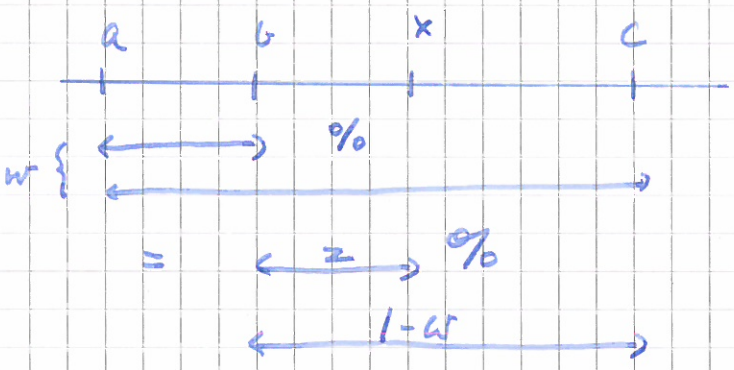
$$\Rightarrow z = 1 - 2w. \quad (1)$$

We repeat this strategy.

w is chosen in the same way as z.

$\Rightarrow$ self similarity $\Rightarrow$ obtain here that

$$\frac{z}{1-w} = w \qquad (2)$$




Solve (1) and (2): $w = \frac{3-\sqrt{5}}{2}$

which is the golden mean.

Local minimum, $x$ is N-dimensional:

## Conjugate gradient method

Minimization in N dimension

$\Rightarrow$

Sequence of 1⊕ minimization.

$$\min_{\lambda} f(\vec{p} + \lambda \vec{n})$$

$\vec{p}$: Starting point

$\vec{n}$: direction

How to choose this direction?

This defines the method.

We define conjugate gradient direction

$$f(\vec{p} + \vec{x}) = f(\vec{p}) + \sum_{i=1}^{N} \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{(i,j)}^{N} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \dots$$

$\vec{x} = \lambda \vec{n}$

$$= \vec{c} - b\vec{x} + \frac{1}{2} \vec{x}^T \cdot \vec{A} \cdot \vec{x}$$

$\vec{c} = f(\vec{p})$, $\vec{b} = -\nabla f(\vec{p})$, $\vec{A} \ni \vec{\nabla} \otimes \vec{\nabla} f(\vec{p})$.

In the following, we work to second order:

$$\vec{\nabla} f = \vec{A}\vec{x} - \vec{b}$$

Changing $\vec{x}$ : $\vec{x} \to \vec{x} + \delta\vec{x}$

How does this change $\vec{\nabla} f$?

$$\delta \vec{\nabla} f = [A(\vec{x} + \delta\vec{x}) - \vec{b}] - [\vec{A}\vec{x} - \vec{b}]$$
$$= \vec{A} \cdot \delta\vec{x}.$$

Assume that $\vec{x}$ has been found after n.-D minimization in the direction $\vec{u}_0$.

This implies $\vec{u}_0 \cdot \vec{\nabla} f(\vec{x}) = 0$.

How to choose a new direction $\vec{u}_1$ that does not destroy the minimization in the $\vec{u}_0$ direction?

We must have that $\delta\vec{x} = \|\delta\vec{x}\|\,\vec{u}_1$

so that $\vec{u}_0^{\,T} \cdot \delta\vec{\nabla} f = 0 \Rightarrow$

$$\vec{u}_0^{\,T} \cdot \vec{\vec{A}} \cdot \delta\vec{x} = 0 \quad \text{or}$$

$$\boxed{\vec{u}_0^{\,T}\, \vec{\vec{A}} \cdot \vec{u}_1 = 0}$$

This leads to the <u>conjugate gradient method</u> that we have already met.

Problem with the conjugate gradient method: don't know the derivatives of $f$ explicitly.

<u>The Powell algorithm</u> avoids this.

Start with an orthonormal basis

$$\{\vec{e}_i\}, \quad i = 1, \ldots, N.$$