Mathematik III

Prof. Dr. Michael Möckel

WS 2022/23

TH Aschaffenburg
university of applied sciences

Numerical Algorithms

What is an algorithm?

Mohammed ibn-Musa al-Khwarizimi (780-850)

Baghdad, was the royal mathematician

Algorithm	=	procedure that solves a recurrent problem in a (finite) number of well-defined steps

Mathematically, an **algorithm A is a function**, mapping inputs onto outputs:

A : set of all possible inputs -> set of all possible outputs:

output = A(specific input values)

Properties of the function A may be complicated (discontinuous, nonlinear, …)

Simple example algorithm

Find maximum element from a set of (real) numbers

```
max = S(1)
for i=1:1:length(S)
  if S(i) > max
    max = S(i);
  end
end
```

Iterative algorithms

Iterative algorithm: An algorithm A is called iterative, if

a core routine of the algorithm is repeated time and again

- to improve a temporary result
- until a certain criterion is reached
  (e.g. max. number of iterations or precision level)

An iterative algorithm is implemented by using a **loop** (for or while loop)

Iterative algorithms II

- Starting point: x_0
- Iterative procedure: $x_{n+1} = A(x_n)$
  (discrete map)

- Creates a chain of intermediate results:

  $x_1, \; x_2, \ldots \; x_n, \; x_{n+1}, \; \ldots$

Contracting map and fixed points

Definition: Let f: I -> R, x -> f(x) be a function and I = [l,u] a closed interval.
A function f is called a **contracting map** if

$$\forall x, y \in [l, u] : \qquad |f(x) - f(y)| \leq \alpha |x - y|$$

with $$0 \leq \alpha < 1$$

Theorem: Let f: I -> I, x -> f(x) be a contracting map and I a closed interval
Then there exists exactly one fixed point x* in I with

x* = f(x*) (Fixed point condition)

Fixed point algorithm

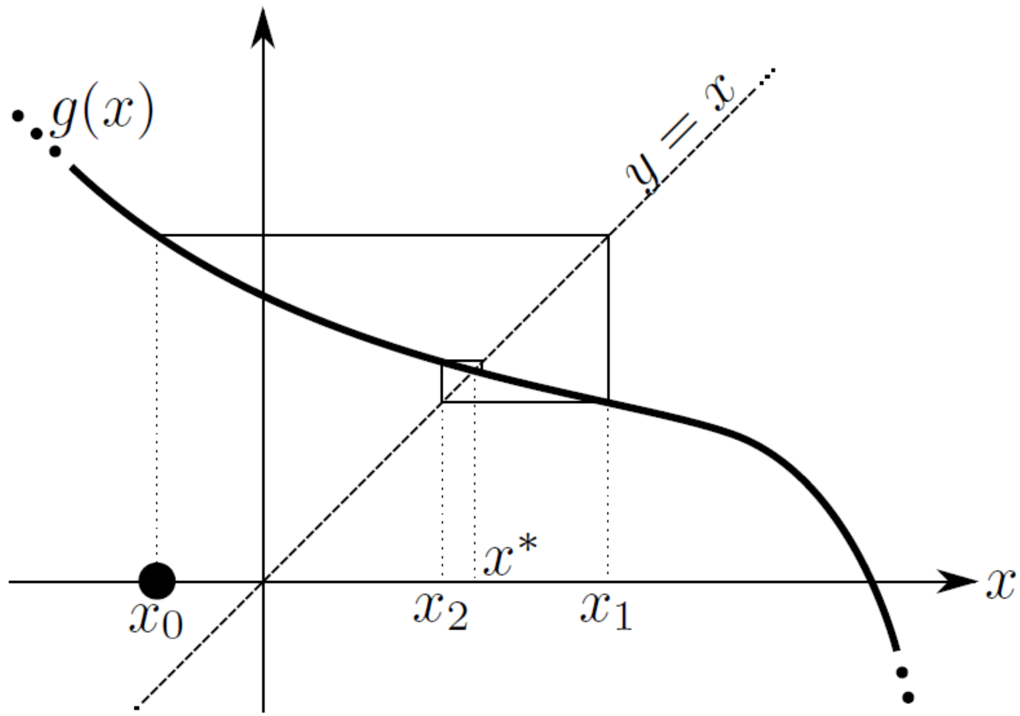
- Not all iterative algorithms are fixed point algorithms (there are diverging algorithms)

- There may be more than one fixed point or more complicated 'attractors'

- Iterative algorithms need to be designed such that the fixed point contains the solution of a given problem

The fixed point method: Convergence

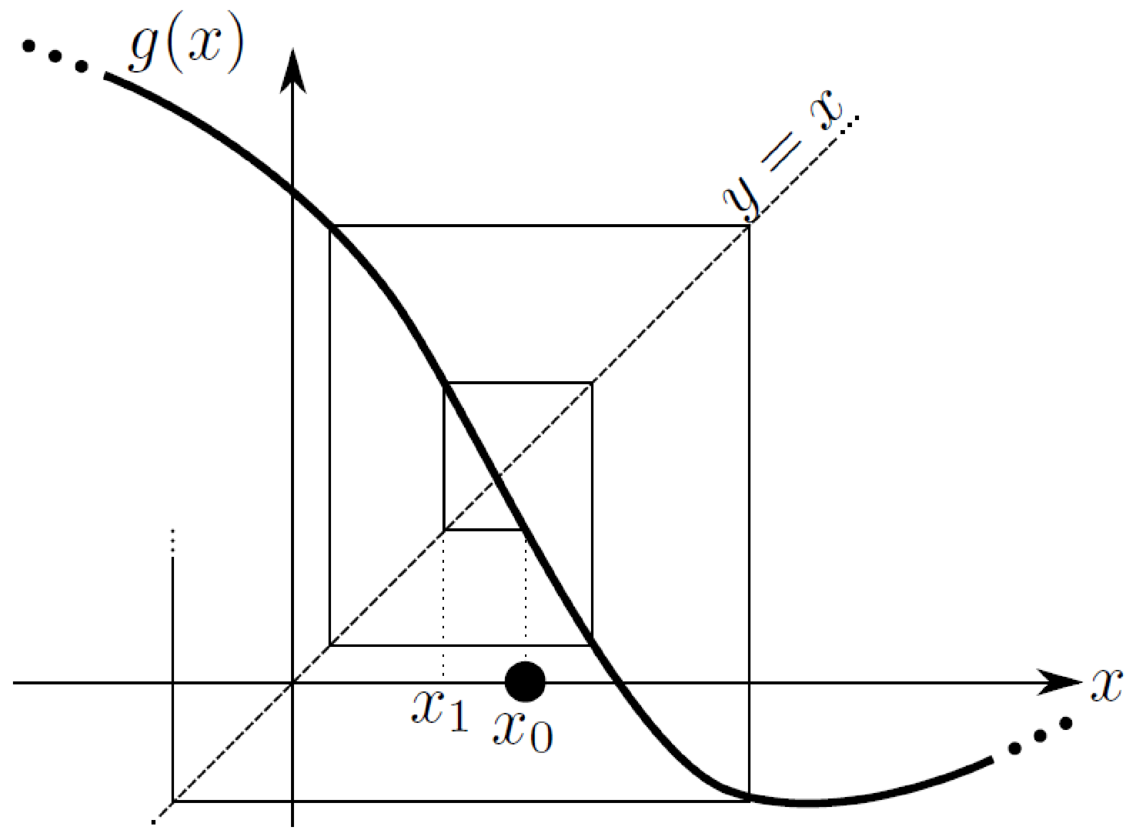
- Convergence of the fixed point method requires derivative |dA/dx| < 1 around the fixed point:



(a) Convergence

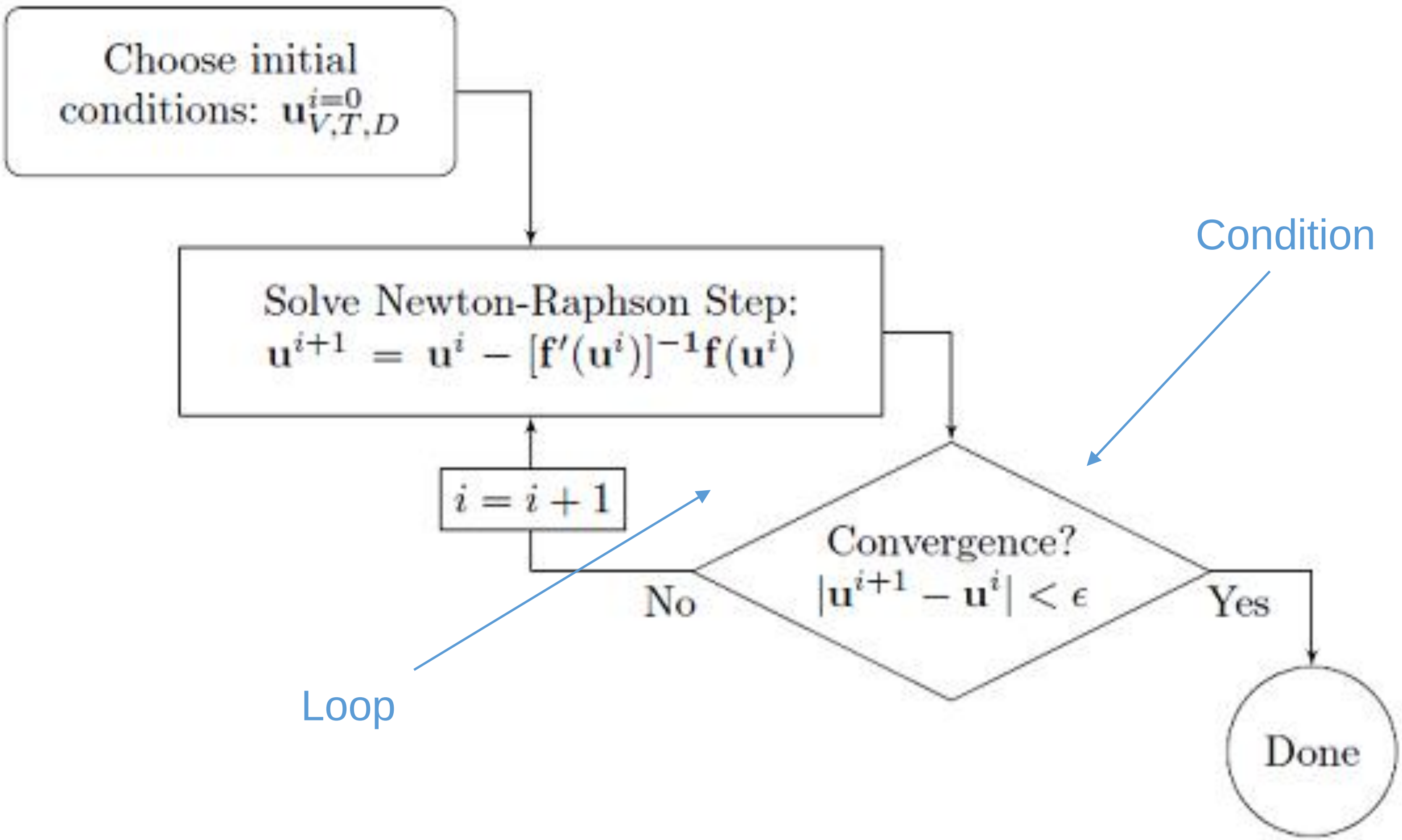
Fixed point method: Divergence

- Otherwise, the fixed point method diverges



(b) Divergence

Representation of algorithms by flow charts



Review:
Taylor expansion

Taylor expansion

Taylor polynomial of finite order n:

= approximate representation of a function f(x) as a polynomial

- Expansion around a given point x0
- f(x) is at least n times continuously differentiable
  (i.e. all derivatives must exist and must be continuous functions)

$$T[f, x_0, n](x) = \sum_{i=0}^{n} \frac{1}{i!} f^{(i)}(x_0)(x - x_0)^i;$$

Chapter 2

Root finding for nonlinear functions

Chapter 2.1

Root finding for nonlinear functions

Bisection method

The bisection method

Assume:

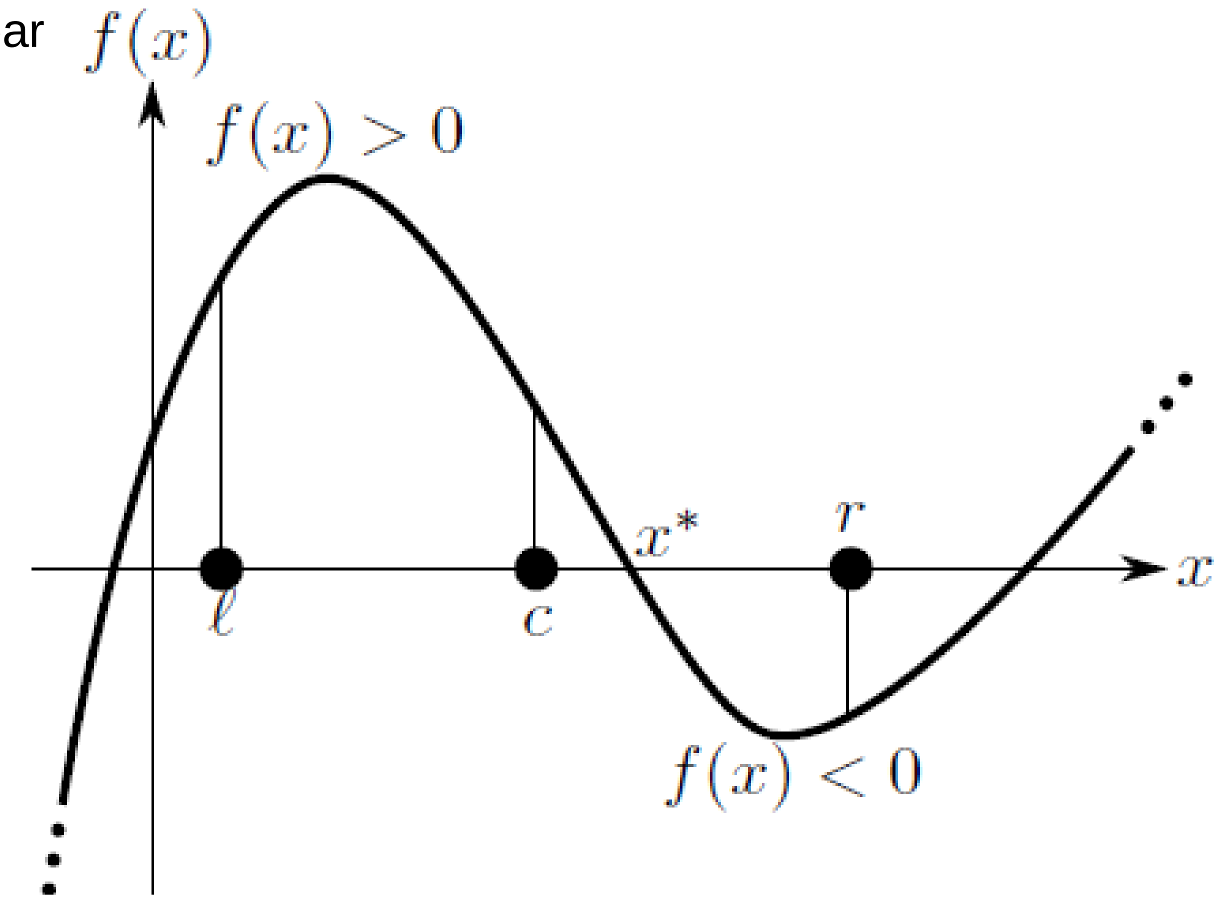
- f is continuous, nonlinear
- f has a single root

(1) Two initial values xl and xr are chosen on both sides of the root

(2) The center point c is calculated.

(3) Depending on the sign of f(c) either xl or xr Is updated

(4) the algorithm converges to x*



The bisection method

Assume:

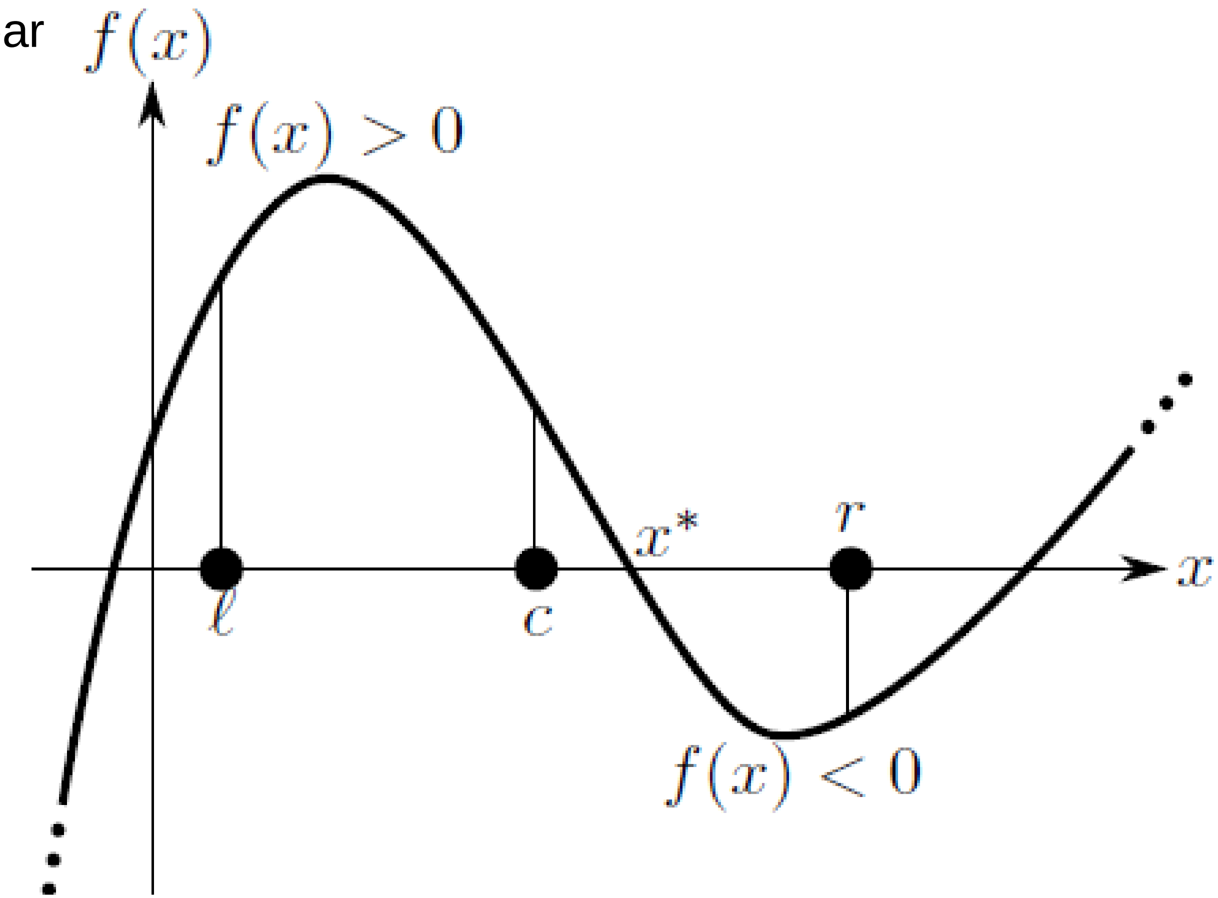
- f is continuous, nonlinear
- f has a single root

(1) Two initial values xl and xr are chosen on both sides of the root

(2) The center point c is calculated.

(3) Depending on the sign of f(c) either xl or xr Is updated

(4) the algorithm converges to x*



Bisection method: Pseudocode

function $\textsc{Bisection}(f(x), \ell, r)$
  for $k \leftarrow 1, 2, 3, \dots$
    $c \leftarrow {}^{\ell+r}/_{2}$
    if $|f(c)| < \varepsilon_f$ or $|r - \ell| < \varepsilon_x$ **then**
      return $x^* \approx c$
    else if $f(\ell) \cdot f(c) < 0$ **then**
      $r \leftarrow c$
    else
      $\ell \leftarrow c$

Using a secant: Regula falsi

Assume:

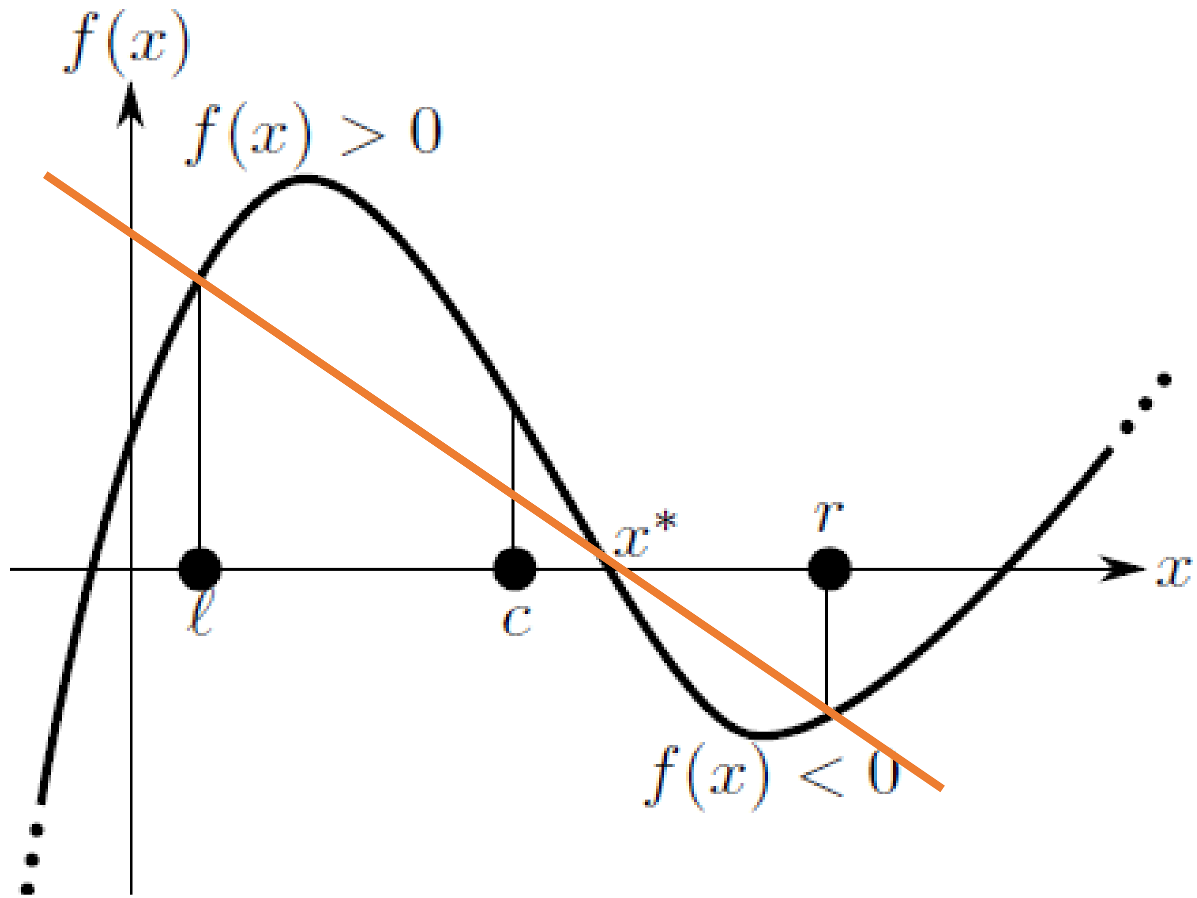
- f is continuous, nonlinear
- f has a single root

(1) Two initial values
xl and xr are chosen
on both sides of the root

(2) The center point c
is calculated.

(3) Depending on the
sign of f(c) either xl or xr
Is updated

(4) the algorithm
converges to x*



Chapter 2.2

Root finding for nonlinear functions

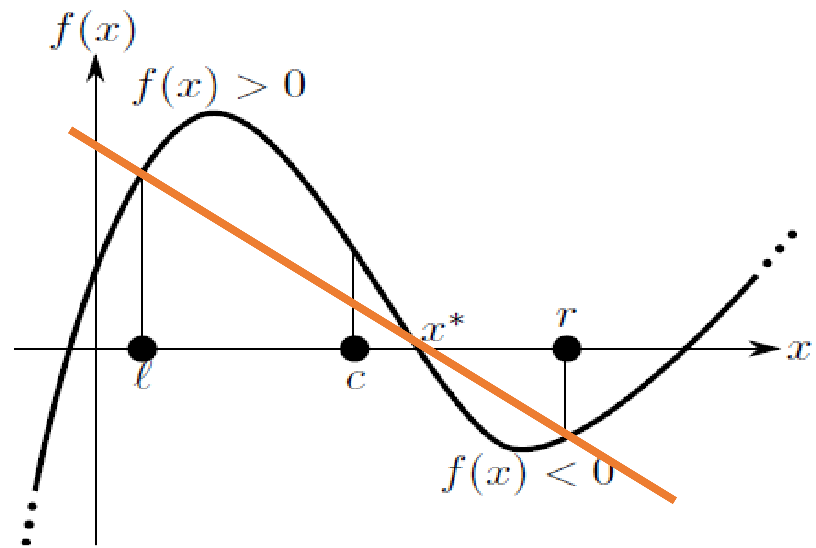
Regula falsi method

Using a secant: Regula falsi

(1) Two initial values
xl and xr are chosen
on both sides of the root

(2a) Define a line between
the points (xl, f(xl)) and (xr, f(xr))

$$y(x) = f(x_L) + \frac{f(x_R) - f(x_L)}{x_R - x_L}(x - x_L)$$



(2b) Calculate root of line xi by solving y(xi) = 0

(3) Depending on the sign of f(xi) either xl or xr is updated

(4) the algorithm converges to x* faster

Chapter 2.3.2

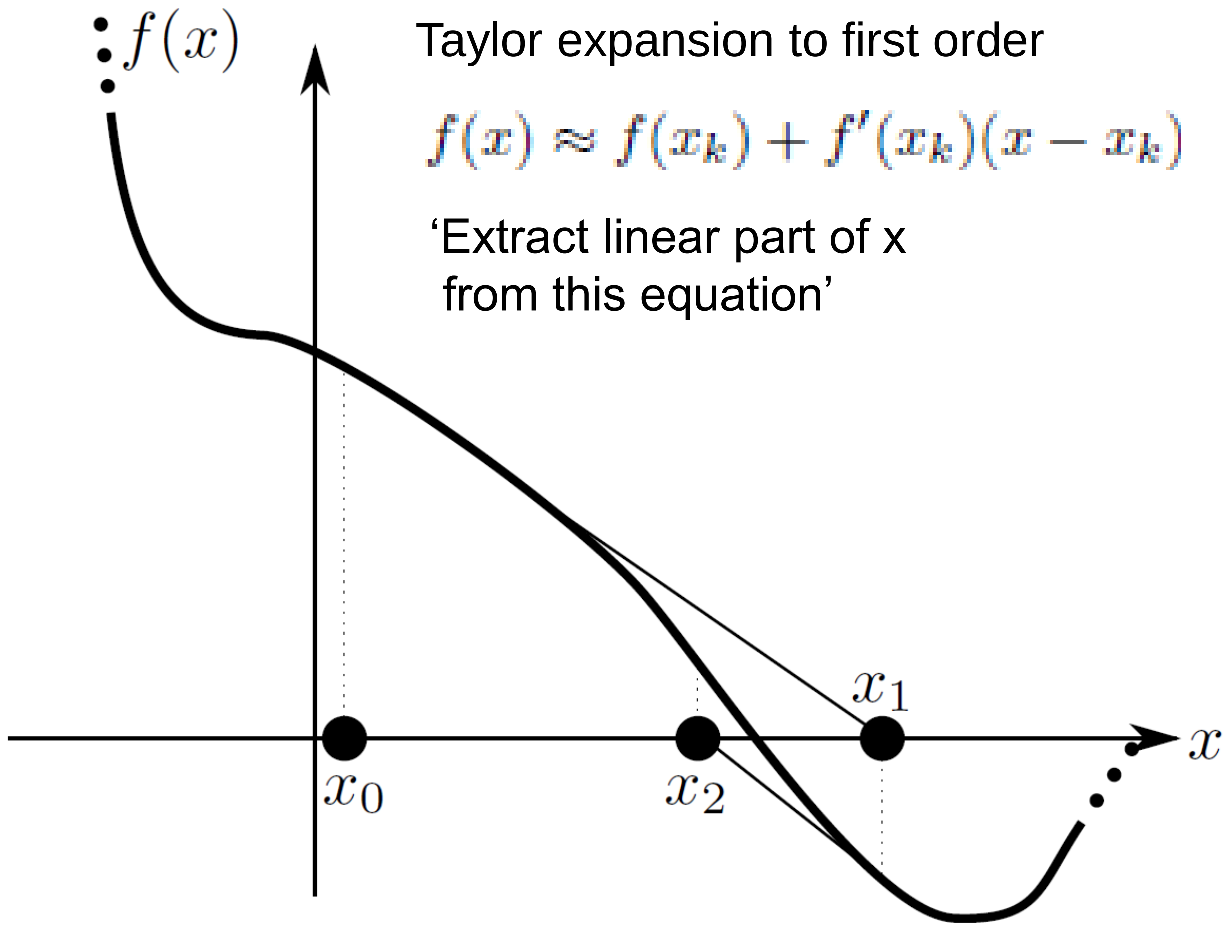
Root finding for nonlinear functions

Newton‘s method

Using a tangent: Newton's method

Fixed point algorithm:

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



Taylor expansion to first order

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

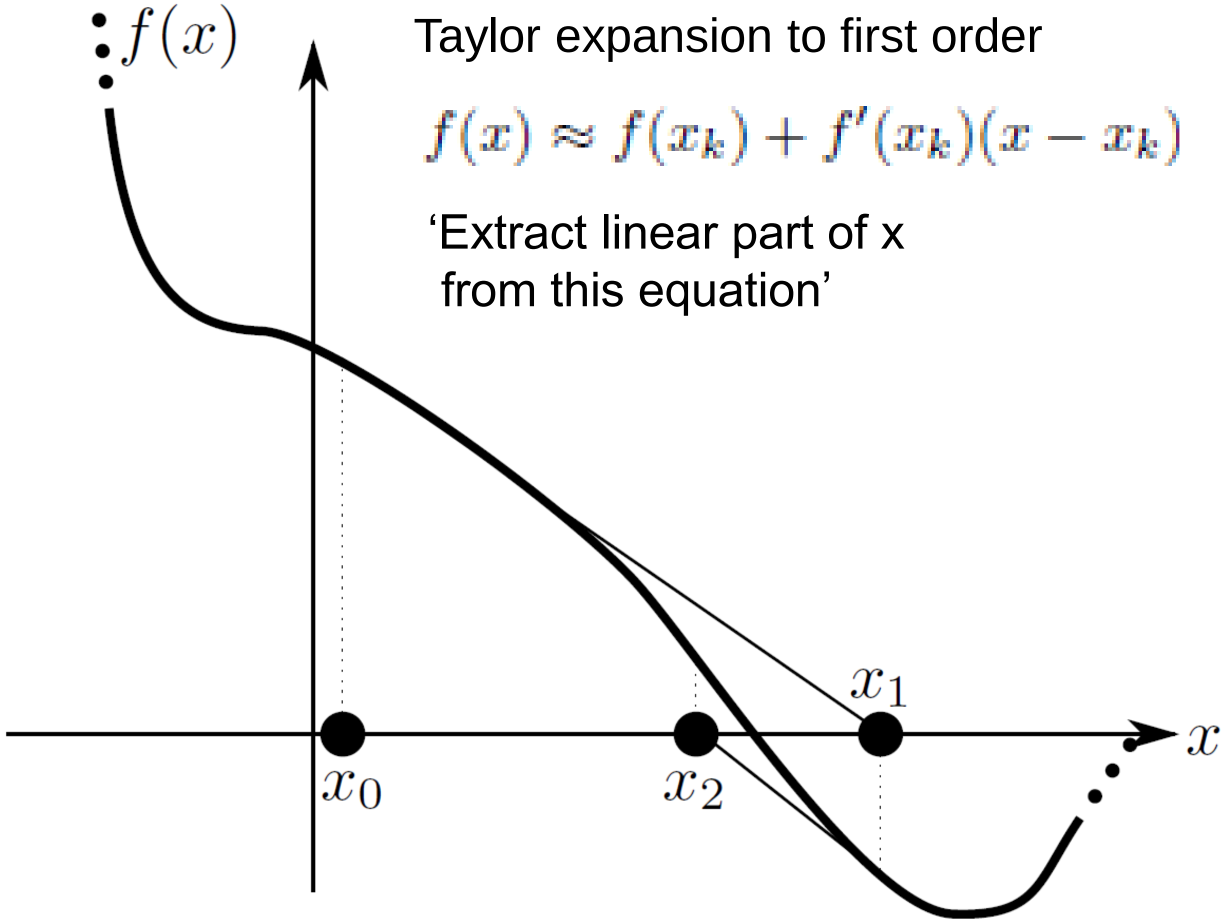
'Extract linear part of x from this equation'

Using a tangent: Newton's method

Fixed point algorithm:

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

Requires knowledge of derivative



Taylor expansion to first order

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

'Extract linear part of x from this equation'

Using a tangent: Newton's method

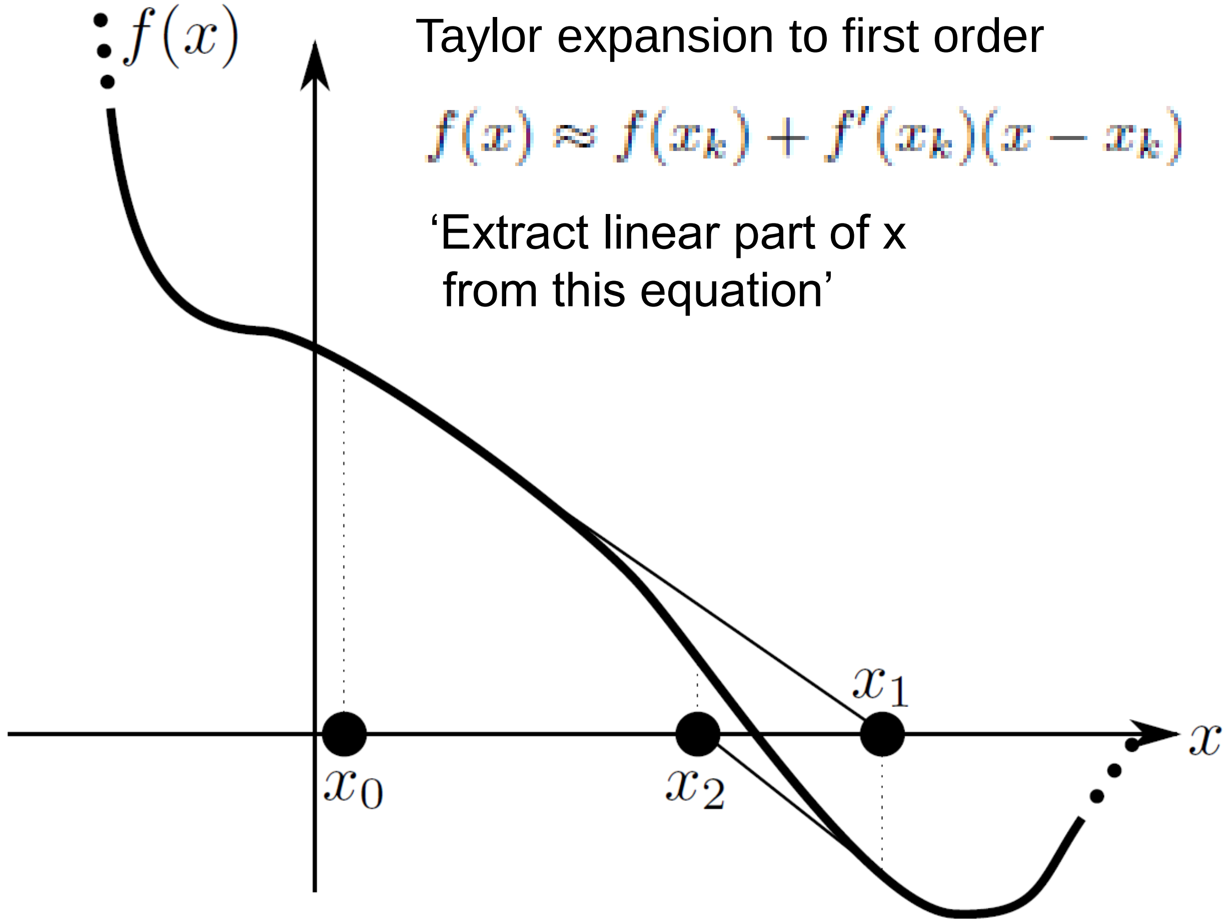
Fixed point algorithm:

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

Requires knowledge of derivative

Condition number:

$$\mathrm{cond}_{x^*} f = \frac{1}{|f'(x^*)|}.$$

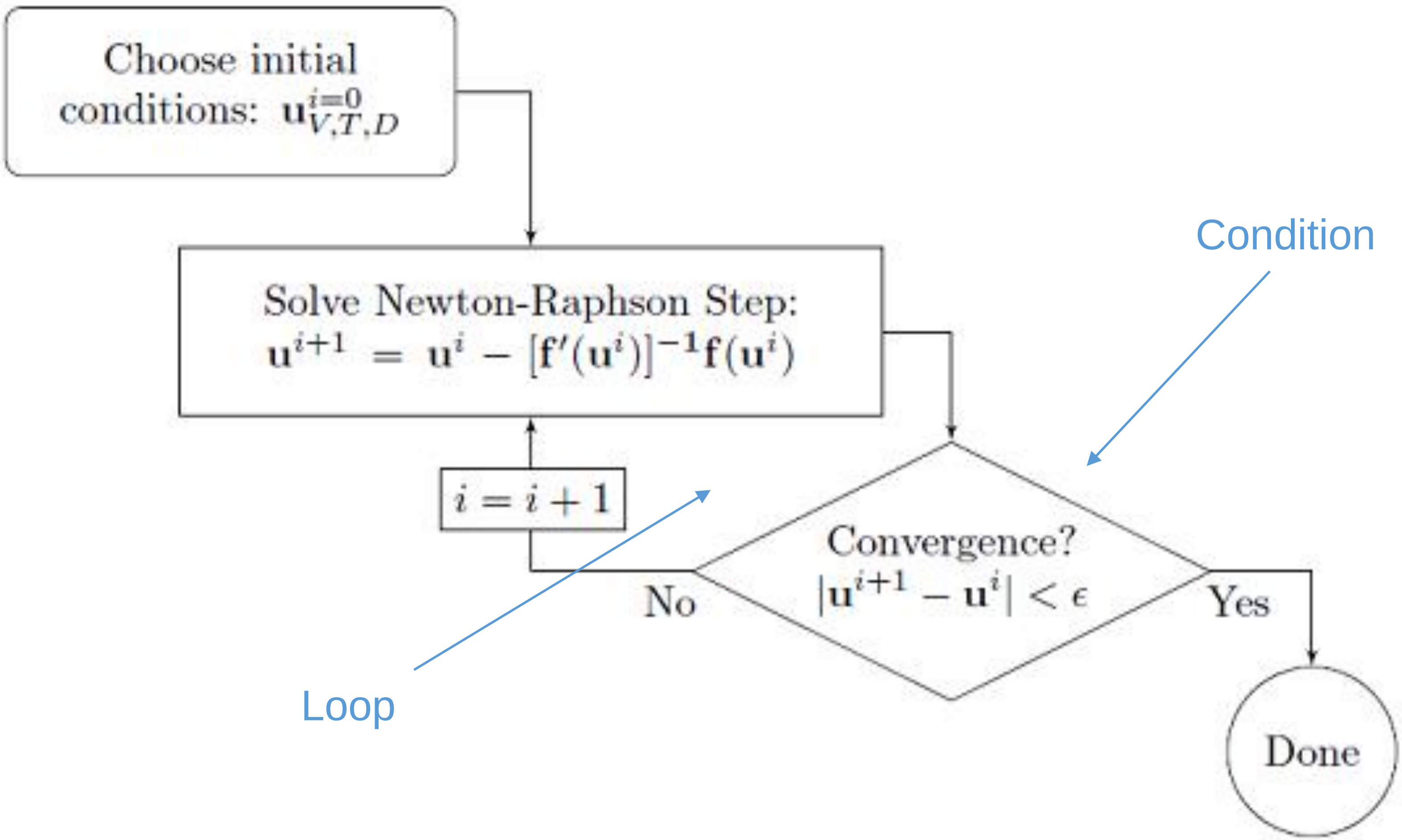


Taylor expansion to first order

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

'Extract linear part of x from this equation'

Representation of algorithms by flow charts



Root finding for nonlinear functions

Simple design of iterative algorithms

How to create a fixed point algorithm?

Create a one-point algorithm, i.e. a single series of points

‘Extract’ the linear variable x from the condition \$$f(x) = 0$

e.g. $$3x^2 - 0.5x + 5 = 0$$

$$x = 2(3x^2 + 5)$$

Turn this into an iterative algorithm.

There are several ways to do so. See problem set 4.