Nonlinear Algebraic Equations

Nonlinear algebraic equations

When solving the system

$$u'(t) = g(u), \qquad u(0) = u_0, \tag{1}$$

with an implicit Euler scheme we have to solve the nonlinear algebraic equation

$$u_{n+1} - \Delta t\, g(u_{n+1}) \;=\; u_n, \tag{2}$$

at each time step. Here u_n is known and u_{n+1} is unknown. If we let c denote u_n and v denote u_{n+1}, we want to find v such that

$$v - \Delta t\, g(v) \;=\; c, \tag{3}$$

where c is given.

Nonlinear algebraic equations

First consider the case of $g(u) = u$, which corresponds to the differential equation

$$u' = u, \quad u(0) = u_0. \tag{4}$$

The equation (3) for each time step, is now

$$v - \Delta t\, v = c, \tag{5}$$

which has the solution

$$v = \frac{1}{1-\Delta t} c. \tag{6}$$

The time stepping in the Euler scheme for (4) is written

$$u_{n+1} = \frac{1}{1-\Delta t} u_n. \tag{7}$$

Nonlinear algebraic equations

Similarly, for any linear function g, i.e., functions on the form

$$g(v) = \alpha + \beta v \tag{8}$$

with constants α and β, we can solve equation (3) directly and get

$$v = \frac{c + \alpha \Delta t}{1 - \beta \Delta t}. \tag{9}$$

Nonlinear algebraic equations

Next we study the nonlinear differential equation

$$u' = u^2, \tag{10}$$

which means that

$$g(v) = v^2. \tag{11}$$

Now (3) reads

$$v - \Delta t\, v^2 = c. \tag{12}$$

Nonlinear algebraic equations

This second order equation has two possible solutions

$$v_+ = \frac{1+\sqrt{1-4\Delta t\, c}}{2\Delta t} \tag{13}$$

and

$$v_- = \frac{1-\sqrt{1-4\Delta t\, c}}{2\Delta t}. \tag{14}$$

Note that

$$\lim_{\Delta t \to 0} \frac{1+\sqrt{1-4\Delta t\, c}}{2\Delta t} = \infty.$$

Since Δt is supposed to be small and the solution is not expected to blow up, we conclude that v_+ is not correct.

Nonlinear algebraic equations

Therefore the correct solution of (12) to use in the Euler scheme is

$$v = \frac{1-\sqrt{1-4\Delta t\, c}}{2\Delta t}. \tag{15}$$

We can now conclude that the implicit scheme

$$u_{n+1} - \Delta t\, u_{n+1}^2 = u_n \tag{16}$$

can be written on computational form

$$u_{n+1} = \frac{1-\sqrt{1-4\Delta t\, u_n}}{2\Delta t}. \tag{17}$$

Nonlinear algebraic equations

We have seen that the equation

$$v - \Delta t\, g(v) = c \qquad (18)$$

can be solved analytically when

$$g(v) = v \qquad (19)$$

or

$$g(v) = v^2. \qquad (20)$$

Generally it can be seen that we can solve (18) when g is on the form

$$g(v) = \alpha + \beta v + \gamma v^2. \qquad (21)$$

Nonlinear algebraic equations

- For most cases of nonlinear functions g, (18) can not be solved analytically
- A couple of examples of this is

$$g(v) = e^v \quad \text{or} \quad g(v) = \sin(v)$$

Nonlinear algebraic equations

Since we work with nonlinear equations on the form

$$u_{n+1} - u_n = \Delta t \, g(u_{n+1}) \tag{22}$$

where Δt is a small number, we know that u_{n+1} is close to u_n. This will be a useful property later.
In the rest of this lecture we will write nonlinear equations on the form

$$f(x) = 0, \tag{23}$$

where f is nonlinear. We assume that we have available a value x_0 close to the true solution x^* (, i.e. $f(x^*) = 0$).
We also assume that f has no other zeros in a small region around x^*.

The bisection method

Consider the function

$$f(x) = 2 + x - e^x \tag{24}$$

for x ranging from 0 to 3, see the graph in Figure 1.

- We want to find $x = x^*$ such that

$$f(x^*) = 0$$

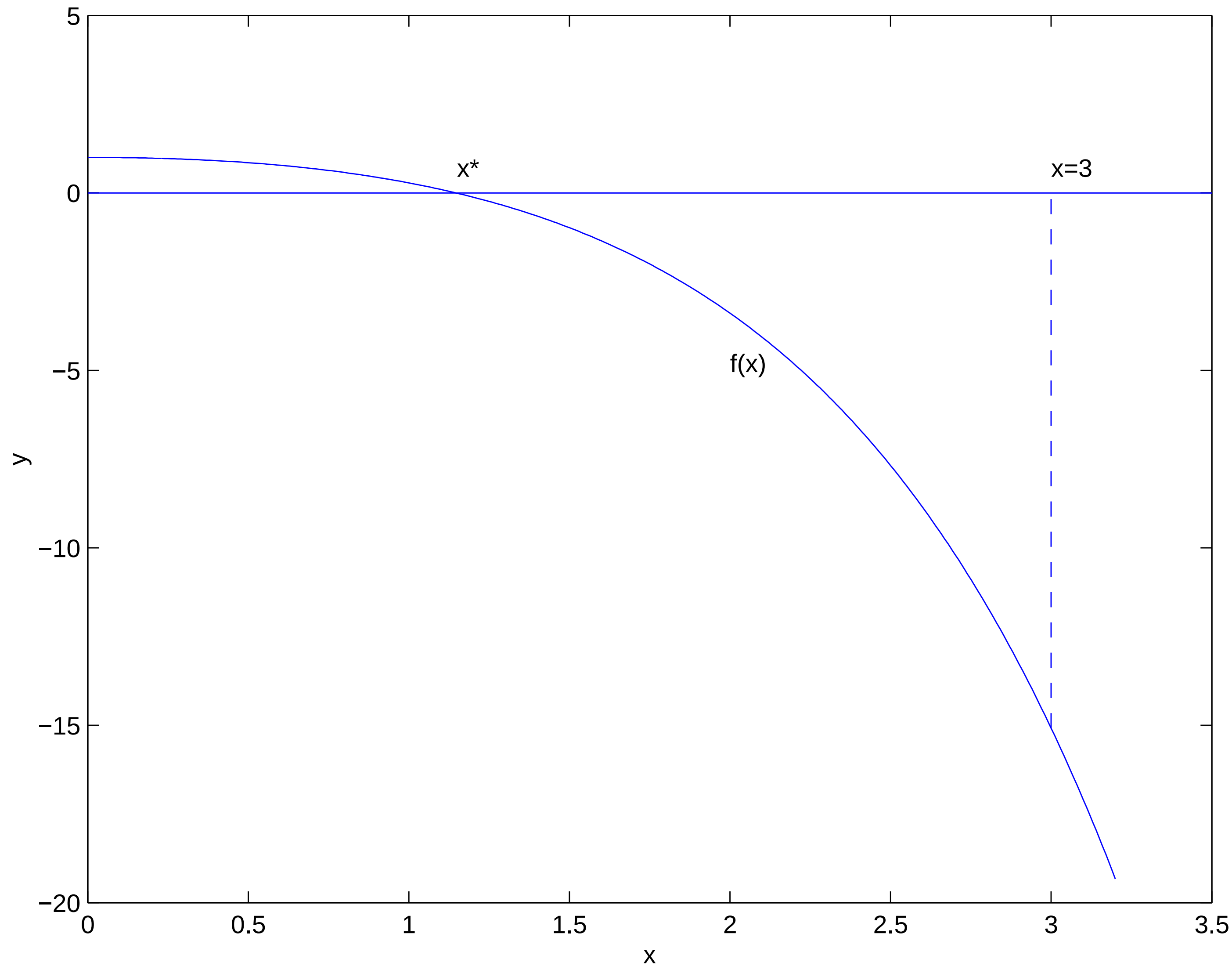

Figure 1: The graph of $f(x) = 2 + x - e^x$.

The bisection method

- An iterative method is to create a series $\{x_i\}$ of approximations of x^*, which hopefully converges towards x^*
- For the Bisection Method we choose the two first guesses x_0 and x_1 as the endpoints of the definition domain, i.e.

$$x_0 = 0 \quad \text{and} \quad x_1 = 3$$

- Note that $f(x_0) = f(0) > 0$ and $f(x_1) = f(3) < 0$, and therefore $x_0 < x^* < x_1$, provided that f is continuous
- We now define the mean value of x_0 and x_1

$$x_2 = \frac{1}{2}(x_0 + x_1) = \frac{3}{2}$$

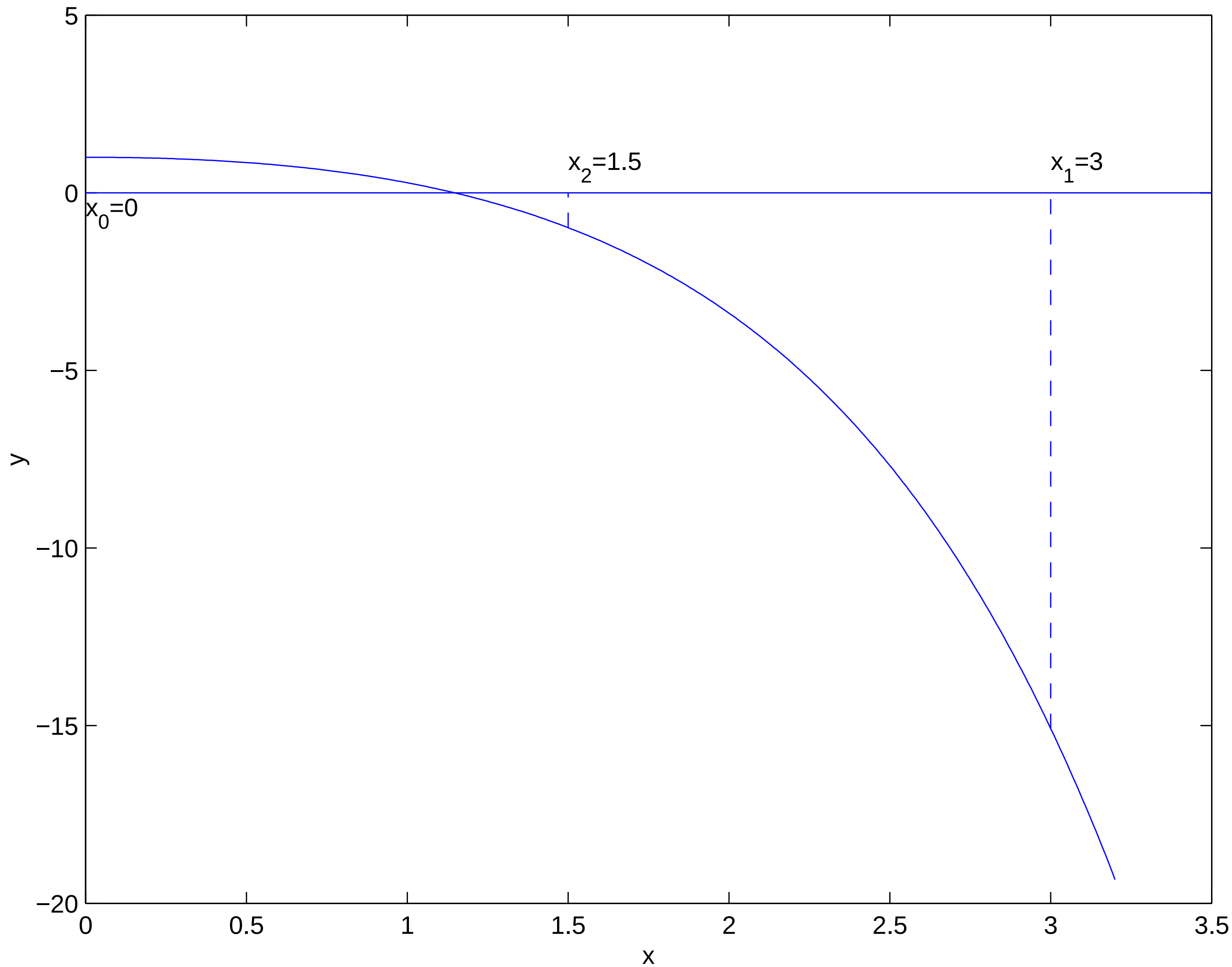

Figure 2: The graph of $f(x) = 2 + x - e^x$ and three values of f: $f(x_0)$, $f(x_1)$ and $f(x_2)$.

The bisection method

- We see that

$$f(x_2) = f(\frac{3}{2}) = 2 + 3/2 - e^{3/2} < 0,$$

- Since $f(x_0) > 0$ and $f(x_2) < 0$, we know that $x_0 < x^* < x_2$
- Therefore we define

$$x_3 = \frac{1}{2}(x_0 + x_2) = \frac{3}{4}$$

- Since $f(x_3) > 0$, we know that $x_3 < x^* < x_2$ (see Figure 3)
- This can be continued until $|f(x_n)|$ is sufficiently small

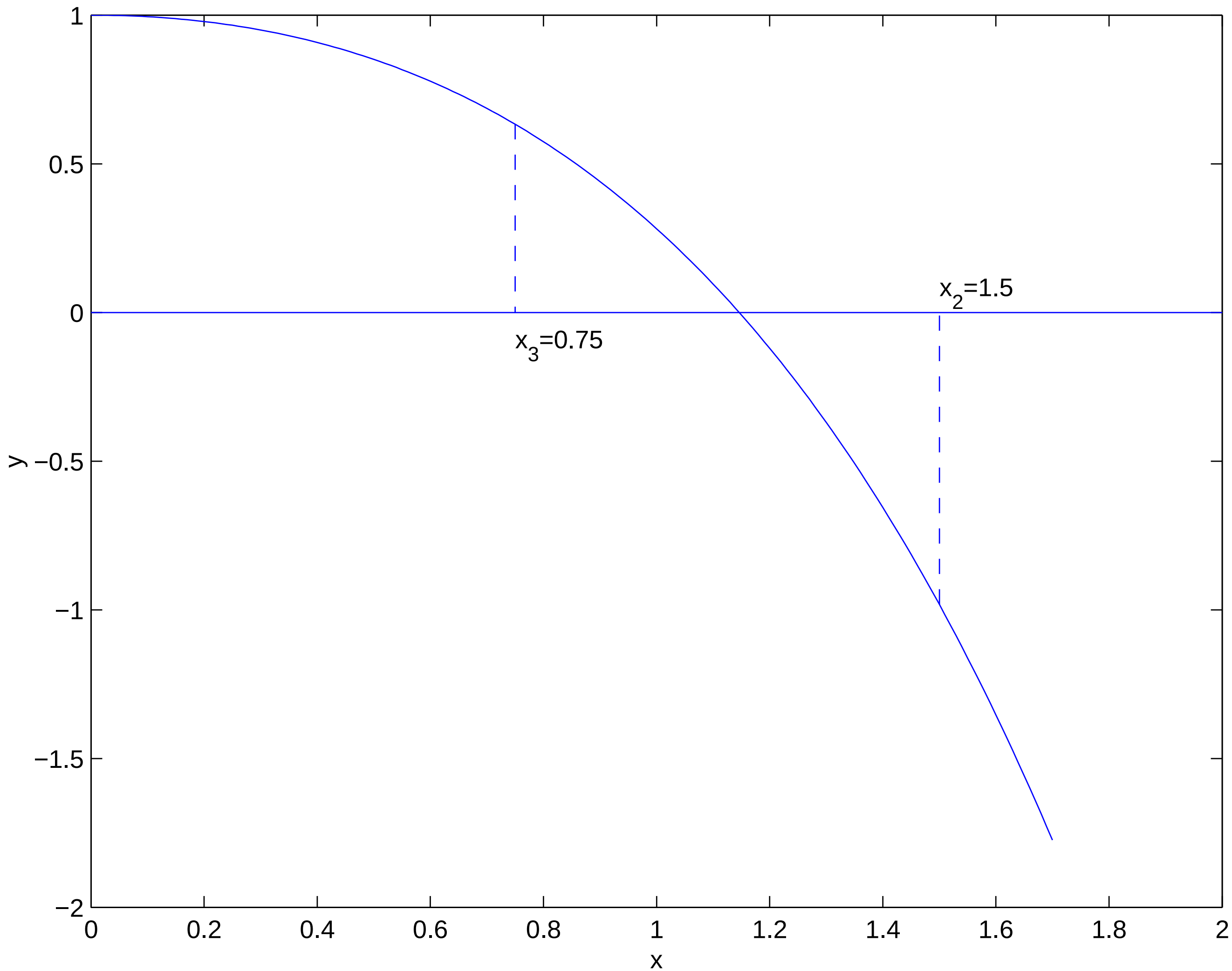

Figure 3: The graph of $f(x) = 2 + x - e^x$ and two values of f: $f(x_2)$ and $f(x_3)$.

The bisection method

Written in algorithmic form the Bisection method reads:

Algorithm 1. Given a, b such that $f(a)\cdot f(b) < 0$ and given a tolerance ε. Define $c = \frac{1}{2}(a+b)$.

while $|f(c)| > \varepsilon$ **do**

    if $f(a)\cdot f(c) < 0$

    then $b = c$

    else $a = c$

    $c := \frac{1}{2}(a+b)$

end

Example 11

Find the zeros for

$$f(x) = 2 + x - e^x$$

using Algorithm 1 and choose $a = 0$, $b = 3$ and $\varepsilon = 10^{-6}$.

- In Table 1 we show the number of iterations i, c and $f(c)$
- The number of iterations, i, refers to the number of times we pass through the while-loop of the algorithm

i	c	$f(c)$
1	1.500000	-0.981689
2	0.750000	0.633000
4	1.312500	-0.402951
8	1.136719	0.0201933
16	1.146194	$-2.65567 \cdot 10^{-6}$
21	1.146193	$4.14482 \cdot 10^{-7}$

Table 1: Solving the nonlinear equation $f(x) = 2 + x - e^x = 0$ by using the bisection method; the number of iterations i, c and $f(c)$.

Example 11

- We see that we get sufficient accuracy after 21 iterations
- The next slide show the C program that is used to solve this problem
- The entire computation uses $5.82 \cdot 10^{-6}$ seconds on a Pentium III 1GHz processor
- Even if this quite fast, we need even faster algorithms in actual computations
  - In practical applications you might need to solve billions of nonlinear equations, and then “every micro second counts”

```c
#include <stdio.h>
#include <math.h>

double f (double x) { return 2.+x-exp(x); }
inline double fabs (double  r) { return ( (r >= 0.0) ? r : -r ); }

int main (int nargs, const char** args)
{
  double epsilon = 1.0e-6; double a, b, c, fa, fc;
  a = 0.; b = 3.; fa = f(a); c = 0.5*(a+b);
  while (fabs(fc=(f(c))) > epsilon) {
    if ((fa*fc) < 0) {
      b = c;
    }
    else {
      a = c;
      fa = fc;
    }
    c = 0.5*(a+b);
  }
  printf("final c=%g, f(c)=%g\n",c,fc);
  return 0;
}
```

Newton's method

- Recall that we have assumed that we have a good initial guess x_0 close to x^* (where $f(x^*) = 0$)
- We will also assume that we have a small region around x^* where f has only one zero, and that $f'(x) \neq 0$
- Taylor series expansion around $x = x_0$ yields

$$f(x_0 + h) = f(x_0) + hf'(x_0) + O(h^2) \tag{25}$$

- Thus, for small h we have

$$f(x_0 + h) \approx f(x_0) + hf'(x_0) \tag{26}$$

Newton's method

- We want to choose the step h such that $f(x_0+h)\approx 0$
- By (26) this can be done by choosing h such that

$$f(x_0)+hf'(x_0)=0$$

- Solving this gives

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

- We therefore define

$$x_1 \stackrel{\text{def}}{=} x_0+h = x_0-\frac{f(x_0)}{f'(x_0)} \tag{27}$$

Newton's method

- We test this on the example studied above with $f(x) = 2 + x - e^x$ and $x_0 = 3$
- We have that

$$f'(x) = 1 - e^x$$

- Therefore

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 3 - \frac{5 - e^3}{1 - e^3} = 2.2096$$

- We see that

$$|f(x_0)| = |f(3)| \approx 15.086 \quad \text{and} \quad |f(x_1)| = |f(2.2096)| \approx 4.902$$

i.e, the value of f is significantly reduced

Newton's method

We can now repeat the above procedure and define

$$x_2 \stackrel{\text{def}}{=} x_1 - \frac{f(x_1)}{f'(x_1)}, \tag{28}$$

and in algorithmic form Newton's method reads:

Algorithm 2. Given an initial approximation x_0 and a tolerance ε.

$k = 0$
while $|f(x_k)| > \varepsilon$ **do**

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

$k = k + 1$
end

Newton's method

In Table 2 we show the results generated by Newton's method on the above example.

k	x_k	$f(x_k)$
1	2.209583	-4.902331
2	1.605246	-1.373837
3	1.259981	-0.265373
4	1.154897	$-1.880020 \cdot 10^{-2}$
5	1.146248	$-1.183617 \cdot 10^{-4}$
6	1.146193	$-4.783945 \cdot 10^{-9}$

Table 2: Solving the nonlinear equation $f(x) = 2 + x - e^x = 0$ by using Algorithm 25 and $\varepsilon = 10^{-6}$; the number of iterations k, x_k and $f(x_k)$.

Newton's method

- We observe that the convergence is much faster for Newton's method than for the Bisection method
- Generally, Newton's method converges faster than the Bisection method
- This will be studied in more detail in Project 1

Example 12

Let

$$f(x) = x^2 - 2,$$

and find x^* such that $f(x^*) = 0$.

- Note that one of the exact solutions is $x^* = \sqrt{2}$
- Newton's method for this problem reads

$$x_{k+1} = x_k - \frac{x_k^2 - 2}{2x_k}$$

- or

$$x_{k+1} = \frac{x_k^2 + 2}{2x_k}$$

Example 12

If we choose $x_0 = 1$, we get

$$x_1 = 1.5,$$
$$x_2 = 1.41667,$$
$$x_3 = 1.41422.$$

Comparing this with the exact value

$$x^* = \sqrt{2} \approx 1.41421,$$

we see that a very accurate approximation is obtained in only 3 iterations.

An alternative derivation

- The Taylor series expansion of f around x_0 is given by

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + O((x - x_0)^2)$$

- Let $F_0(x)$ be a linear approximation of f around x_0:

$$F_0(x) = f(x_0) + (x - x_0)f'(x_0)$$

- $F_0(x)$ approximates f around x_0 since

$$F_0(x_0) = f(x_0) \quad \text{and} \quad F_0'(x_0) = f'(x_0)$$

- We now define x_1 to be such that $F(x_1) = 0$, i.e.

$$f(x_0) + (x_1 - x_0)f'(x_0) = 0$$

An alternative derivation

- Then we get

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

which is identical to the iteration obtained above

- We repeat this process, and define a linear approximation of f around x_1

$$F_1(x) = f(x_1) + (x - x_1) f'(x_1)$$

- x_2 is defined such that $F_1(x_2) = 0$, i.e.

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

An alternative derivation

- Generally we get

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

- This process is illustrated in Figure 4

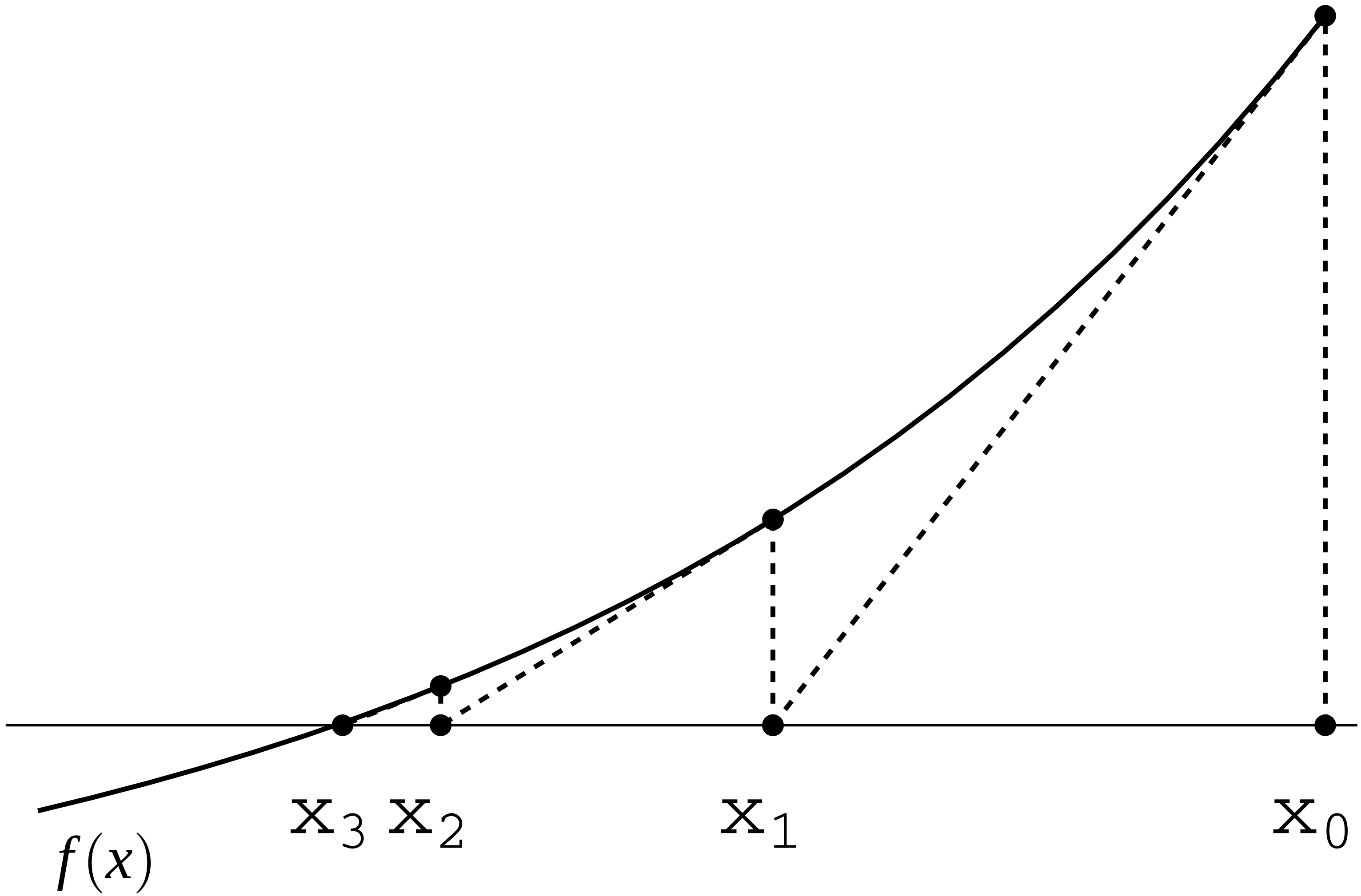

Figure 4: Graphical illustration of Newton's method.

The Secant method

- The secant method is similar to Newton's method, but the linear approximation of f is defined differently
- Now we assume that we have two values x_0 and x_1 close to x^*, and define the linear function $F_0(x)$ such that

$$F_0(x_0) = f(x_0) \quad \text{and} \quad F_0(x_1) = f(x_1)$$

- The function $F_0(x)$ is therefore given by

$$F_0(x) = f(x_1) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_1)$$

- $F_0(x)$ is called the linear interpolant of f

The Secant method

- Since $F_0(x) \approx f(x)$, we can compute a new approximation x_2 to x^* by solving the linear equation

$$F(x_2) = 0$$

- This means that we must solve

$$f(x_1) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_1) = 0,$$

with respect to x_2 (see Figure 5)

- This gives

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

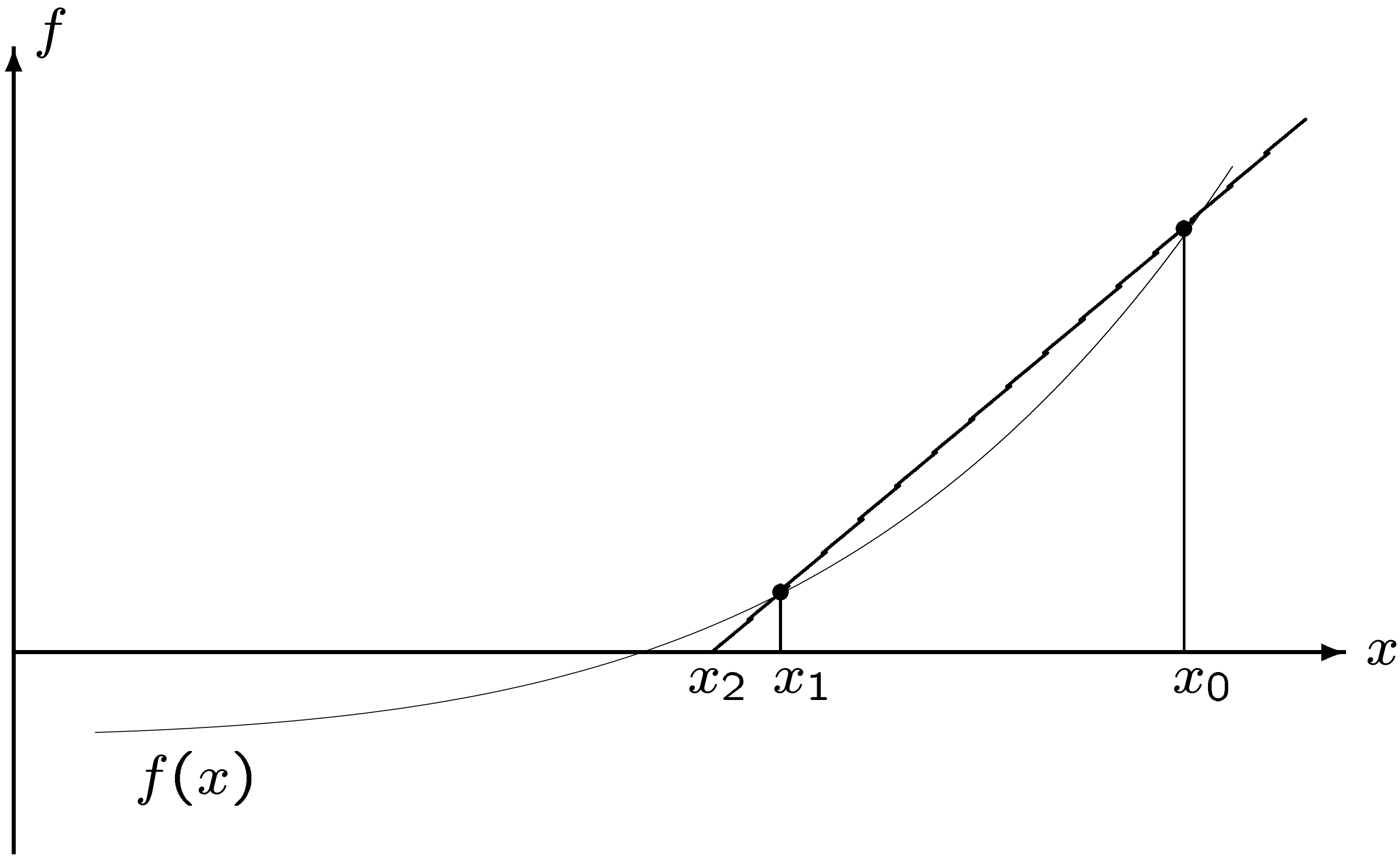

Figure 5: The figure shows a function $f = f(x)$ and its linear interpolant F between x_0 and x_1.

The Secant method

Following the same procedure as above we get the iteration

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

and the associated algorithm reads

Algorithm 3. Given two initial approximations x_0 and x_1 and a tolerance ε.

$k = 1$

while $|f(x_k)| > \varepsilon$ **do**

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

$k = k + 1$

end

Example 13

Let us apply the Secant method to the equation

$$f(x) = 2 + x - e^x = 0,$$

studied above. The two initial values are $x_0 = 0$, $x_1 = 3$, and the stopping criteria is specified by $\varepsilon = 10^{-6}$.

- Table 3 show the number of iterations k, x_k and $f(x_k)$ as computed by Algorithm 3
- Note that the convergence for the Secant method is slower than for Newton's method, but faster than for the Bisection method

k	x_k	$f(x_k)$
2	0.186503	0.981475
3	0.358369	0.927375
4	3.304511	-21.930701
5	0.477897	0.865218
6	0.585181	0.789865
7	1.709760	-1.817874
8	0.925808	0.401902
9	1.067746	0.158930
10	1.160589	$-3.122466 \cdot 10^{-2}$
11	1.145344	$1.821544 \cdot 10^{-3}$
12	1.146184	$1.912908 \cdot 10^{-5}$
13	1.146193	$-1.191170 \cdot 10^{-8}$

Table 3: The Secant method applied with $f(x) = 2 + x - e^x$=0.

Example 14

Find a zero of

$$f(x) = x^2 - 2,$$

which has a solution $x^* = \sqrt{2}$.

- The general step of the secant method is in this case

$$\begin{aligned}
x_{k+1} =& x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \\
=& x_k - (x_k^2 - 2)\frac{x_k - x_{k-1}}{x_k^2 - x_{k-1}^2} \\
=& x_k - \frac{x_k^2 - 2}{x_k + x_{k-1}} \\
=& \frac{x_k x_{k-1} + 2}{x_k + x_{k-1}}
\end{aligned}$$

Example 14

- By choosing $x_0 = 1$ and $x_1 = 2$ we get

$$x_2 = 1.33333$$
$$x_3 = 1.40000$$
$$x_4 = 1.41463$$

- This is quite good compared to the exact value

$$x^* = \sqrt{2} \approx 1.41421$$

- Recall that Newton's method produced the approximation 1.41422 in three iterations, which is slightly more accurate

Fixed-Point iterations

Above we studied implicit schemes for the differential equation $u' = g(u)$, which lead to the nonlinear equation

$$u_{n+1} - \Delta t\, g(u_{n+1}) = u_n,$$

where u_n is known, u_{n+1} is unknown and $\Delta t > 0$ is small. We defined $v = u_{n+1}$ and $c = u_n$, and wrote the equation

$$v - \Delta t\, g(v) = c.$$

We can rewrite this equation on the form

$$v = h(v), \tag{29}$$

where

$$h(v) = c + \Delta t\, g(v).$$

Fixed-Point iterations

The exact solution, v^*, must fulfill

$$v^* = h(v^*).$$

This fact motivates the Fixed Point Iteration:

$$v_{k+1} = h(v_k),$$

with an initial guess v_0.

- Since h leaves v^* unchanged; $h(v^*) = v^*$, the value v^* is referred to as a *fixed-point* of h

Fixed-Point iterations

We try this method to solve

$$x = \sin(x/10),$$

which has only one solution $x^* = 0$ (see Figure 6)
The iteration is

$$x_{k+1} = \sin(x_k/10). \tag{30}$$

Choosing $x_0 = 1.0$, we get the following results

$$\begin{aligned} x_1 &= 0.09983, \\ x_2 &= 0.00998, \\ x_3 &= 0.00099, \end{aligned}$$

which seems to converge fast towards $x^* = 0$.

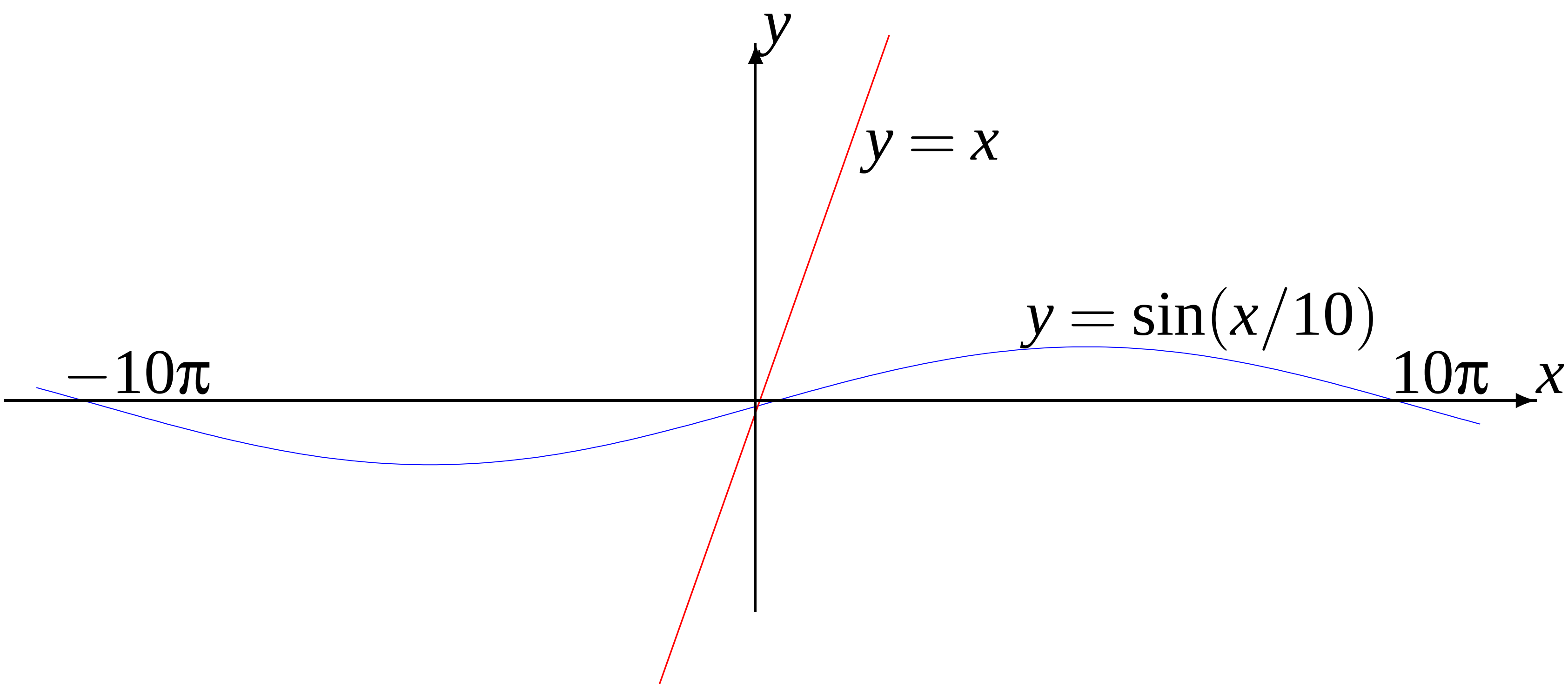

Figure 6: The graph of $y = x$ and $y = \sin(x/10)$.

Fixed-Point iterations

We now try to understand the behavior of the iteration. From calculus we recall for small x we have

$$\sin(x/10) \approx x/10.$$

Using this fact in (30), we get

$$x_{k+1} \approx x_k/10,$$

and therefore

$$x_k \approx (1/10)^k.$$

We see that this iteration converges towards zero.

Convergence of Fixed-Point iterations

We have seen that $h(v) = v$ can be solved with the Fixed-Point iteration

$$v_{k+1} = h(v_k)$$

We now analyze under what conditions the values $\{v_k\}$ generated by the Fixed-Point iterations converge towards a solution v^* of the equation.

Definition: $h = h(v)$ is called a contractive mapping on a closed interval I if

(i) $|h(v) - h(w)| \leq \delta |v - w|$ for any $v, w \in I$, where $0 < \delta < 1$,

and

(ii) $v \in I \Rightarrow h(v) \in I$.

Convergence of Fixed-Point iterations

The Mean Value Theorem of Calculus states that if f is a differentiable function defined on an interval $[a,b]$, then there is a $c \in [a,b]$ such that

$$f(b) - f(a) = f'(c)(b-a).$$

- It follows from this theorem that h in is a contractive mapping defined on an interval I if

$$|h'(\xi)| < \delta < 1 \quad \text{for all } \xi \in I, \tag{31}$$

and $h(v) \in I$ for all $v \in I$

Convergence of Fixed-Point iterations

Let us check the above example

$$x = \sin(x/10)$$

We see that $h(x) = \sin(x/10)$ is contractive on $I = [-1, 1]$ since

$$|h'(x)| = \left| \frac{1}{10} \cos(x/10) \right| \leq \frac{1}{10}$$

and

$$x \in [-1, 1] \quad \Rightarrow \quad \sin(x/10) \in [-1, 1].$$

Convergence of Fixed-Point iterations

For a contractive mapping h, we assume that for any v, w in a closed interval I we have

$$|h(v) - h(w)| \leq \delta \, |v - w| \,, \qquad \text{where } 0 < \delta < 1,$$
$$v \in I \quad \Rightarrow \quad h(v) \in I$$

The error, $e_k = |v_k - v^*|$, fulfills

$$\begin{aligned} e_{k+1} &= |v_{k+1} - v^*| \\ &= |h(v_k) - h(v^*)| \\ &\leq \delta \, |v_k - v^*| \\ &= \delta e_k. \end{aligned}$$

Convergence of Fixed-Point iterations

It now follows by induction on k, that

$$e_k \leq \delta^k e_0.$$

Since $0 < \delta < 1$, we know that $e_k \longrightarrow 0$ as $k \longrightarrow \infty$. This means that we have convergence

$$\lim_{k \longrightarrow \infty} v_k = v^*.$$

We can now conclude that the Fixed-Point iteration will converge when h is a contractive mapping.

Speed of convergence

We have seen that the Fixed-Point iterations fulfill

$$\frac{e_k}{e_0} \leq \delta^k.$$

Assume we want to solve this equation to the accuracy

$$\frac{e_k}{e_0} \leq \varepsilon.$$

- We need to have $\delta^k \leq \varepsilon$, which gives

$$k \ln(\delta) \leq \ln(\varepsilon)$$

- Therefore the number of iterations needs to satisfy

$$k \geq \frac{\ln(\varepsilon)}{\ln(\delta)}$$

Existence and Uniqueness of a Solution

For the equations on the form $v = h(v)$, we want to answer the following questions

a) Does there exist a value v^* such that

$$v^* = h(v^*)?$$

b) If so, is v^* unique?

c) How can we compute v^*?

We assume that h is a contractive mapping on a closed interval I such that

$$|h(v) - h(w)| \leq \delta |v - w|, \qquad \text{where } 0 < \delta < 1, \tag{32}$$

$$v \in I \Rightarrow h(v) \in I \tag{33}$$

for all v, w.

Uniqueness

Assume that we have two solutions v^* and w^* of the problem, i.e.

$$v^* = h(v^*) \quad \text{and} \quad w^* = h(w^*) \tag{34}$$

From the assumption (32) we have

$$|h(v^*) - h(w^*)| \leq \delta |v^* - w^*|,$$

where $\delta < 1$. But (34) gives

$$|v^* - w^*| \leq \delta |v^* - w^*|$$

which can only hold when $v^* = w^*$, and consequently the solution is unique.

Existence

We have seen that if h is a contractive mapping, the equation

$$h(v) = v \tag{35}$$

can only have one solution.

- If we now can show that there exists a solution of (35) we have answered (a), (b) and (c) above
- Below we show that assumptions (32) and (33) imply existence

Cauchy sequences

First we recall the definition of Cauchy sequences.

- A sequence of real numbers, $\{v_k\}$, is called a Cauchy sequence if, for any $\varepsilon > 0$, there is an integer M such that for any $m, n \geq M$ we have

$$\left| v_m - v_n \right| \quad < \quad \varepsilon \tag{36}$$

- **Theorem:** A sequence $\{v_k\}$ converges if and only if it is a Cauchy sequence
- Under we shall show that the sequence, $\{v_k\}$, produced by the Fixed-Point iteration, is a Cauchy series when assumptions (32) and (33) hold

Existence

- Since $v_{n+1} = h(v_n)$, we have

$$|v_{n+1} - v_n| = |h(v_n) - h(v_{n-1})| \leq \delta |v_n - v_{n-1}|$$

- By induction, we have

$$|v_{n+1} - v_n| \quad \leq \quad \delta^n |v_1 - v_0|$$

- In order to show that $\{v_n\}$ is a Cauchy sequence, we need to bind $|v_m - v_n|$
- We may assume that $m > n$, and we see that

$$v_m - v_n = (v_m - v_{m-1}) + (v_{m-1} - v_{m-2}) + \ldots + (v_{n+1} - v_n)$$

Existence

- By the triangle-inequality, we have

$$|v_m - v_n| \leq |v_m - v_{m-1}| + |v_{m-1} - v_{m-2}| + \ldots + |v_{n+1} - v_n|$$

- (37) gives

$$\begin{aligned} |v_m - v_{m-1}| &\leq \delta^{m-1} |v_1 - v_0| \\ |v_{m-1} - v_{m-2}| &\leq \delta^{m-2} |v_1 - v_0| \\ &\vdots \\ |v_{n+1} - v_n| &\leq \delta^{n} |v_1 - v_0| \end{aligned}$$

- consequently

$$\begin{aligned} |v_m - v_n| &\leq |v_m - v_{m-1}| + |v_{m-1} - v_{m-2}| + \ldots + |v_{n+1} - v_n| \\ &\leq \left(\delta^{m-1} + \delta^{m-2} + \ldots + \delta^{n}\right) |v_1 - v_0| \end{aligned}$$

Existence

- We can now estimate the power series

$$\begin{aligned} \delta^{m-1}+\delta^{m-2}+\ldots+\delta^{n} &= \delta^{n-1}\left(\delta+\delta^{2}+\ldots+\delta^{m-n}\right) \\ &\leq \delta^{n-1}\sum_{k=1}^{\infty}\delta^{k} \\ &= \delta^{n-1}\frac{1}{1-\delta} \end{aligned}$$

- So

$$|v_m - v_n| \leq \frac{\delta^{n-1}}{1-\delta}|v_1 - v_0|$$

- δ^{n-1} can be as small as you like, if you choose n big enough

Existence

This means that for any $\varepsilon > 0$, we can find an integer M such that

$$|v_m - v_n| \;<\; \varepsilon$$

provided that $m,n \geq M$, and consequently $\{v_k\}$ is a Cauchy sequence.

- The sequence is therefore convergent, and we call the limit v^*
- Since

$$v^* = \lim_{k\to\infty} v_k = \lim_{k\to\infty} h(v_k) = h(v^*)$$

  by continuity of h, we have that the limit satisfies the equation

Systems of nonlinear equations

We start our study of nonlinear equations, by considering a linear system that arises from the discretization of a linear 2×2 system of ordinary differential equations,

$$\begin{aligned} u'(t) &= -v(t), \qquad & u(0) = u_0, \\ v'(t) &= u(t), \qquad & v(0) = v_0. \end{aligned} \tag{37}$$

An implicit Euler scheme for this system reads

$$\frac{u_{n+1} - u_n}{\Delta t} = -v_{n+1}, \quad \frac{v_{n+1} - v_n}{\Delta t} = u_{n+1}, \tag{38}$$

and can be rewritten on the form

$$\begin{aligned} u_{n+1} + \Delta t\, v_{n+1} &= u_n, \\ -\Delta t\, u_{n+1} + v_{n+1} &= v_n. \end{aligned} \tag{39}$$

Systems of linear equations

We can write this system on the form

$$\mathbf{A}\mathbf{w}_{n+1} = \mathbf{w}_n, \tag{40}$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{w}_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix}. \tag{41}$$

In order to compute $\mathbf{w}_{n+1} = (u_{n+1}, v_{n+1})^T$ from $\mathbf{w}_n = (u_n, v_n)$, we have to solve the linear system (40). The system has a unique solution since

$$\det(\mathbf{A}) = 1 + \Delta t^2 > 0. \tag{42}$$

Systems of linear equations

And the solution is given by $\mathbf{w}_{n+1} = \mathbf{A}^{-1}\mathbf{w}_n$, where

$$\mathbf{A}^{-1} = \frac{1}{1+\Delta t^2}\begin{pmatrix} 1 & -\Delta t \\ \Delta t & 1 \end{pmatrix}. \tag{43}$$

Therefore we get

$$\begin{pmatrix} u_{n+1} \\ v_{n+1} \end{pmatrix} = \frac{1}{1+\Delta t^2}\begin{pmatrix} 1 & -\Delta t \\ \Delta t & 1 \end{pmatrix}\begin{pmatrix} u_n \\ v_n \end{pmatrix} \tag{44}$$

$$= \frac{1}{1+\Delta t^2}\begin{pmatrix} u_n - \Delta t\, v_n \\ \Delta t\, u_n + v_n \end{pmatrix}. \tag{45}$$

Systems of linear equations

We write this as

$$\begin{aligned} u_{n+1} &= \frac{1}{1+\Delta t^2}(u_n - \Delta t\, v_n), \\ v_{n+1} &= \frac{1}{1+\Delta t^2}(v_n + \Delta t\, u_n). \end{aligned} \tag{46}$$

By choosing $u_0 = 1$ and $v_0 = 0$, we have the analytical solutions

$$u(t) = \cos(t), \quad v(t) = \sin(t). \tag{47}$$

In Figure 7 we have plotted (u, v) and (u_n, v_n) for $0 \leq t \leq 2\pi$, $\Delta t = \pi/500$. We see that the scheme provides good approximations.

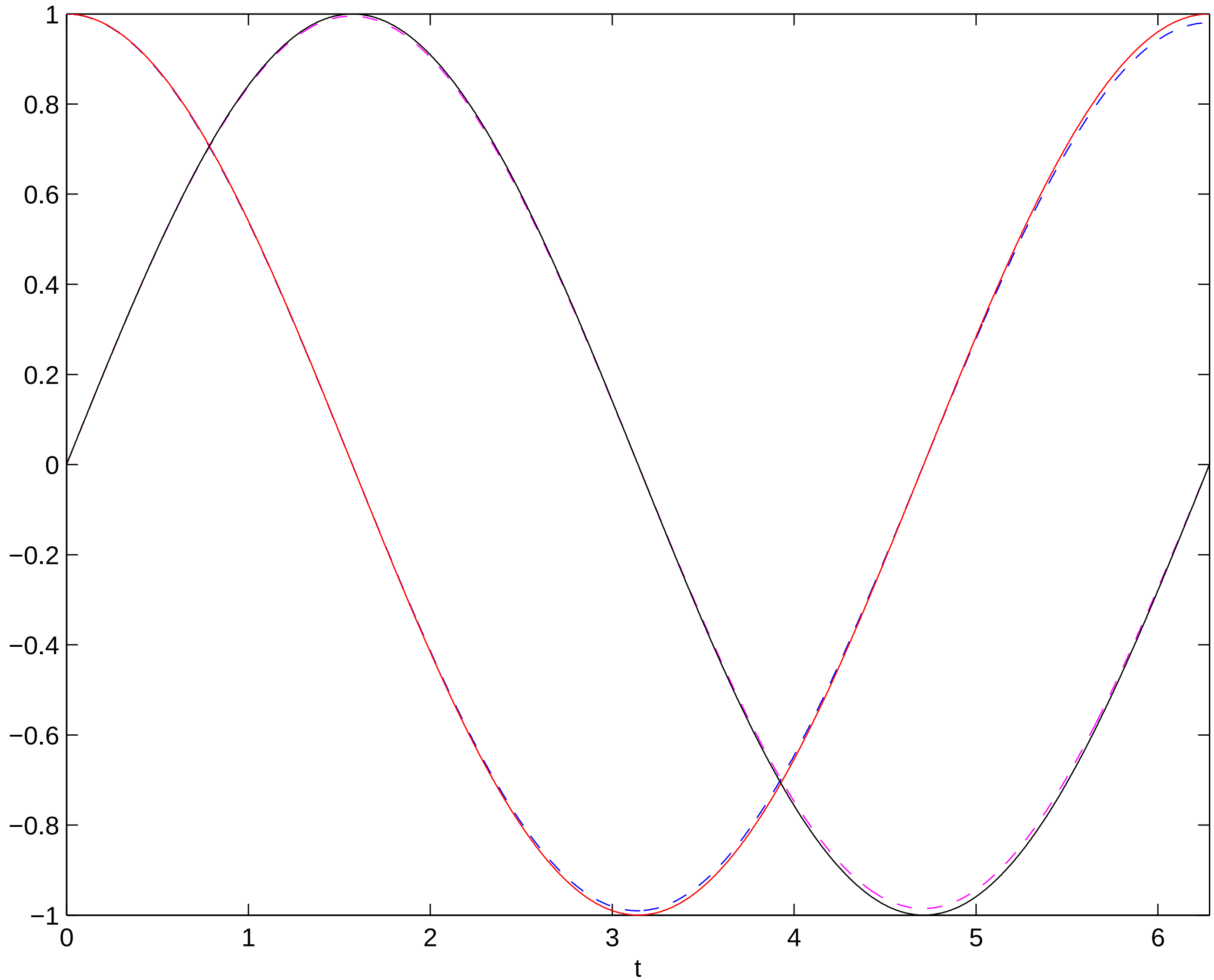

Figure 7: The analytical solution $(u = \cos(t), v = \sin(t))$ and the numerical solution (u_n, v_n), in dashed lines, produced by the implicit Euler scheme.

A nonlinear system

Now we study a nonlinear system of ordinary differential equations

$$
\begin{aligned}
u' &= -v^3, \qquad & u(0) = u_0, \\
v' &= u^3, \qquad & v(0) = v_0.
\end{aligned}
\tag{48}
$$

An implicit Euler scheme for this system reads

$$
\frac{u_{n+1} - u_n}{\Delta t} = -v_{n+1}^3, \quad \frac{v_{n+1} - v_n}{\Delta t} = u_{n+1}^3, \tag{49}
$$

which can be rewritten on the form

$$
\begin{aligned}
u_{n+1} + \Delta t\, v_{n+1}^3 - u_n &= 0, \\
v_{n+1} - \Delta t\, u_{n+1}^3 - v_n &= 0.
\end{aligned}
\tag{50}
$$

A nonlinear system

- Observe that in order to compute (u_{n+1}, v_{n+1}) based on (u_n, v_n), we need to solve a nonlinear system of equations

We would like to write the system on the generic form

$$\begin{aligned} f(x,y) &= 0, \\ g(x,y) &= 0. \end{aligned} \tag{51}$$

This is done by setting

$$\begin{aligned} f(x,y) &= x + \Delta t\, y^3 - \alpha, \\ g(x,y) &= y - \Delta t\, x^3 - \beta, \end{aligned} \tag{52}$$

$\alpha = u_n$ and $\beta = v_n$.

Newton's method

When deriving Newton's method for solving a scalar equation

$$p(x) = 0 \tag{53}$$

we exploited Taylor series expansion

$$p(x_0 + h) = p(x_0) + hp'(x_0) + O(h^2), \tag{54}$$

to make a linear approximation of the function p, and solve the linear approximation of (53). This lead to the iteration

$$x_{k+1} = x_k - \frac{p(x_k)}{p'(x_k)}. \tag{55}$$

Newton's method

We shall try to extend Newton's method to systems of equations on the form

$$\begin{aligned} f(x,y) &= 0, \\ g(x,y) &= 0. \end{aligned} \tag{56}$$

The Taylor-series expansion of a smooth function of two variables $F(x,y)$, reads

$$\begin{aligned} F(x+\Delta x, y+\Delta y) &= F(x,y) + \Delta x \frac{\partial F}{\partial x}(x,y) + \Delta y \frac{\partial F}{\partial y}(x,y) \\ &\quad + O(\Delta x^2, \Delta x \Delta y, \Delta y^2). \end{aligned} \tag{57}$$

Newton's method

Using Taylor expansion on (56) we get

$$
\begin{aligned}
f(x_0+\Delta x, y_0+\Delta y) &= f(x_0,y_0)+\Delta x\frac{\partial f}{\partial x}(x_0,y_0)+\Delta y\frac{\partial f}{\partial y}(x_0,y_0) \\
&\quad +O(\Delta x^2, \Delta x\Delta y, \Delta y^2), \qquad (58)
\end{aligned}
$$

and

$$
\begin{aligned}
g(x_0+\Delta x, y_0+\Delta y) &= g(x_0,y_0)+\Delta x\frac{\partial g}{\partial x}(x_0,y_0)+\Delta y\frac{\partial g}{\partial y}(x_0,y_0) \\
&\quad +O(\Delta x^2, \Delta x\Delta y, \Delta y^2). \qquad (59)
\end{aligned}
$$

Newton's method

Since we want Δx and Δy to be such that

$$\begin{aligned} f(x_0+\Delta x, y_0+\Delta y) &\approx 0, \\ g(x_0+\Delta x, y_0+\Delta y) &\approx 0, \end{aligned} \tag{60}$$

we define Δx and Δy to be the solution of the linear system

$$\begin{aligned} f(x_0,y_0)+\Delta x\frac{\partial f}{\partial x}(x_0,y_0)+\Delta y\frac{\partial f}{\partial y}(x_0,y_0) &= 0, \\ g(x_0,y_0)+\Delta x\frac{\partial g}{\partial x}(x_0,y_0)+\Delta y\frac{\partial g}{\partial y}(x_0,y_0) &= 0. \end{aligned} \tag{61}$$

Remember here that x_0 and y_0 are known numbers, and therefore $f(x_0,y_0)$, $\frac{\partial f}{\partial x}(x_0,y_0)$ and $\frac{\partial f}{\partial y}(x_0,y_0)$ are known numbers as well. Δx and Δy are the unknowns.

Newton's method

(61) can be written on the form

$$\begin{pmatrix} \frac{\partial f_0}{\partial x} & \frac{\partial f_0}{\partial y} \\ \frac{\partial g_0}{\partial x} & \frac{\partial g_0}{\partial y} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -\begin{pmatrix} f_0 \\ g_0 \end{pmatrix}. \tag{62}$$

where $f_0 = f(x_0, y_0)$, $g_0 = g(x_0, y_0)$, $\frac{\partial f_0}{\partial x} = \frac{\partial f}{\partial x}(x_0, y_0)$, etc. If the matrix

$$\mathbf{A} = \begin{pmatrix} \frac{\partial f_0}{\partial x} & \frac{\partial f_0}{\partial y} \\ \frac{\partial g_0}{\partial x} & \frac{\partial g_0}{\partial y} \end{pmatrix} \tag{63}$$

is nonsingular. Then

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -\begin{pmatrix} \frac{\partial f_0}{\partial x} & \frac{\partial f_0}{\partial y} \\ \frac{\partial g_0}{\partial x} & \frac{\partial g_0}{\partial y} \end{pmatrix}^{-1} \begin{pmatrix} f_0 \\ g_0 \end{pmatrix}. \tag{64}$$

Newton's method

We can now define

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - \begin{pmatrix} \frac{\partial f_0}{\partial x} & \frac{\partial f_0}{\partial y} \\ \frac{\partial g_0}{\partial x} & \frac{\partial g_0}{\partial y} \end{pmatrix}^{-1} \begin{pmatrix} f_0 \\ g_0 \end{pmatrix}.$$

And by repeating this argument we get

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} \frac{\partial f_k}{\partial x} & \frac{\partial f_k}{\partial y} \\ \frac{\partial g_k}{\partial x} & \frac{\partial g_k}{\partial y} \end{pmatrix}^{-1} \begin{pmatrix} f_k \\ g_k \end{pmatrix}, \quad (65)$$

where $f_k = f(x_k, y_k)$, $g_k = g(x_k, y_k)$ and $\frac{\partial f_k}{\partial x} = \frac{\partial f}{\partial x}(x_k, y_k)$ etc.
The scheme (65) is Newton's method for the system (56).

A Nonlinear example

We test Newton's method on the system

$$\begin{aligned} e^x - e^y &= 0, \\ \ln(1+x+y) &= 0. \end{aligned} \tag{66}$$

The system have analytical solution $x = y = 0$. Define

$$\begin{aligned} f(x,y) &= e^x - e^y, \\ g(x,y) &= \ln(1+x+y). \end{aligned}$$

The iteration in Newton's method (65) reads

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} e^{x_k} & -e^{y_k} \\ \frac{1}{1+x_k+y_k} & \frac{1}{1+x_k+y_k} \end{pmatrix}^{-1} \begin{pmatrix} e^{x_k} - e^{y_k} \\ \ln(1+x_k+y_k) \end{pmatrix}. \tag{67}$$

A Nonlinear example

The table below shows the computed results when $x_0 = y_0 = \frac{1}{2}$.

k	x_k	y_k
0	0.5	0.5
1	-0.193147	-0.193147
2	-0.043329	-0.043329
3	-0.001934	-0.001934
4	$-3.75 \cdot 10^{-6}$	$-3.75 \cdot 10^{-6}$
5	$-1.40 \cdot 10^{-11}$	$-1.40 \cdot 10^{-11}$

We observe that, as in the scalar case, Newton's method gives very rapid convergence towards the analytical solution $x = y = 0$.

The Nonlinear System Revisited

We now go back to nonlinear system of ordinary differential equations (48), presented above. For each time step we had to solve

$$\begin{aligned} f(x,y) &= 0, \\ g(x,y) &= 0, \end{aligned} \tag{68}$$

where

$$\begin{aligned} f(x,y) &= x + \Delta t\, y^3 - \alpha, \\ g(x,y) &= y - \Delta t\, x^3 - \beta. \end{aligned} \tag{69}$$

We shall now solve this system using Newton's method.

The Nonlinear System Revisited

We put $x_0 = \alpha$, $y_0 = \beta$ and iterate as follows

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} \frac{\partial f_k}{\partial x} & \frac{\partial f_k}{\partial y} \\ \frac{\partial g_k}{\partial x} & \frac{\partial g_k}{\partial y} \end{pmatrix}^{-1} \begin{pmatrix} f_k \\ g_k \end{pmatrix}, \qquad (70)$$

where

$$\begin{aligned} f_k &= f(x_k, y_k), & g_k &= g(x_k, y_k), \\ \frac{\partial f_k}{\partial x} = \frac{\partial f}{\partial x}(x_k, y_k) &= 1, & \frac{\partial f_k}{\partial y} = \frac{\partial f}{\partial y}(x_k, y_k) &= 3\Delta t\, y_k^2, \\ \frac{\partial g_k}{\partial x} = \frac{\partial g}{\partial x}(x_k, y_k) &= -3\Delta t\, x_k^2, & \frac{\partial g_k}{\partial y} = \frac{\partial g}{\partial y}(x_k, y_k) &= 1. \end{aligned}$$

The Nonlinear System Revisited

The matrix

$$\mathbf{A} = \begin{pmatrix} \frac{\partial f_k}{\partial x} & \frac{\partial f_k}{\partial y} \\ \frac{\partial g_k}{\partial x} & \frac{\partial g_k}{\partial y} \end{pmatrix} = \begin{pmatrix} 1 & 3\Delta t\, y_k^2 \\ -3\Delta t\, x_k^2 & 1 \end{pmatrix} \tag{71}$$

has its determinant given by: det$(\mathbf{A}) = 1 + 9\Delta t^2 x_k^2 y_k^2 > 0$. So $\mathbf{A}^{-1}$ is well defined and is given by

$$\mathbf{A}^{-1} = \frac{1}{1 + 9\Delta t^2 x_k^2 y_k^2} \begin{pmatrix} 1 & -3\Delta t\, y_k^2 \\ 3\Delta t\, x_k^2 & 1 \end{pmatrix}. \tag{72}$$

For each time-level we can e.g. iterate until

$$|f(x_k, y_k)| + |g(x_k, y_k)| < \varepsilon = 10^{-6}. \tag{73}$$

The Nonlinear System Revisited

- We have tested this method with $\Delta t = 1/100$ and $t \in [0,1]$
- In Figure 8 the numerical solutions of u and v are plotted as functions of time, and in Figure 9 the numerical solution is plotted in the (u,v) coordinate system
- In Figure 10 we have plotted the number of Newton's iterations needed to reach the stopping criterion (73) at each time-level
- Observe that we need no more than two iterations at all time-levels

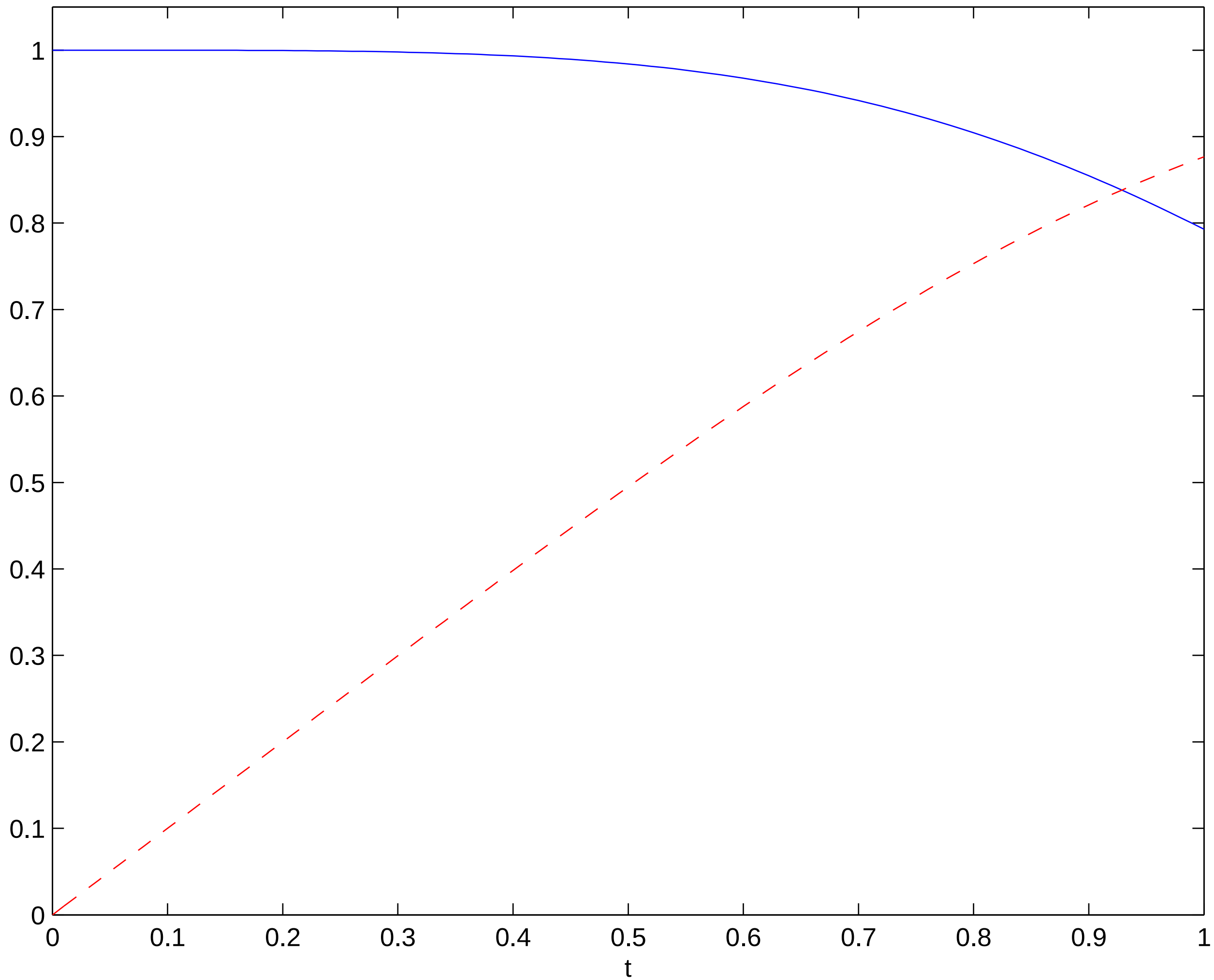

Figure 8: The numerical solutions $u(t)$ and $v(t)$ (in dashed line) of (48) produced by the implicit Euler scheme (49) using $u_0 = 1$, $v_0 = 0$ and $\Delta t = 1/100$.

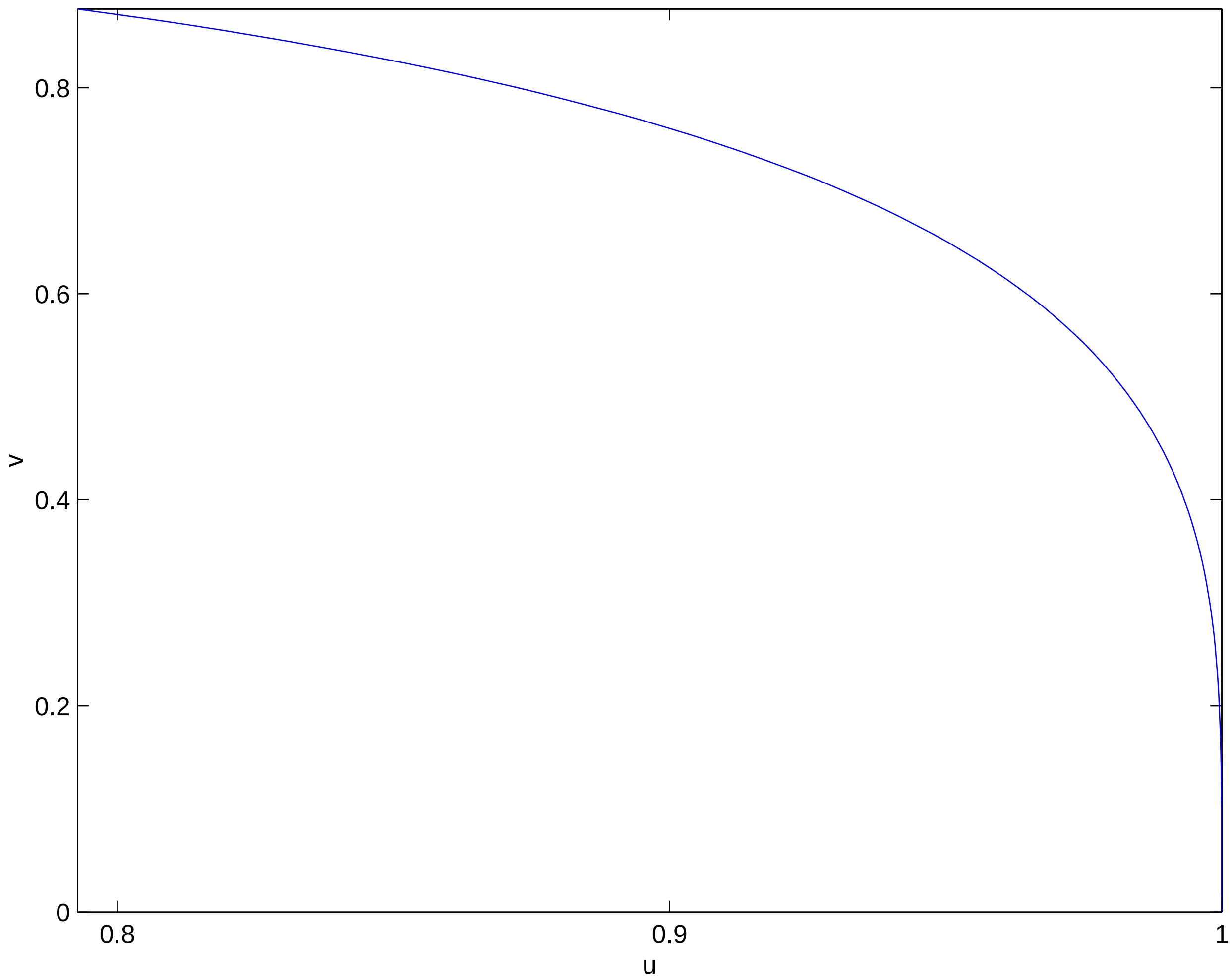

Figure 9: The numerical solutions of (48) in the (u,v)-coordinate system, arising from the implicit Euler scheme (49) using $u_0 = 1$, $v_0 = 0$ and $\Delta t = 1/100$.

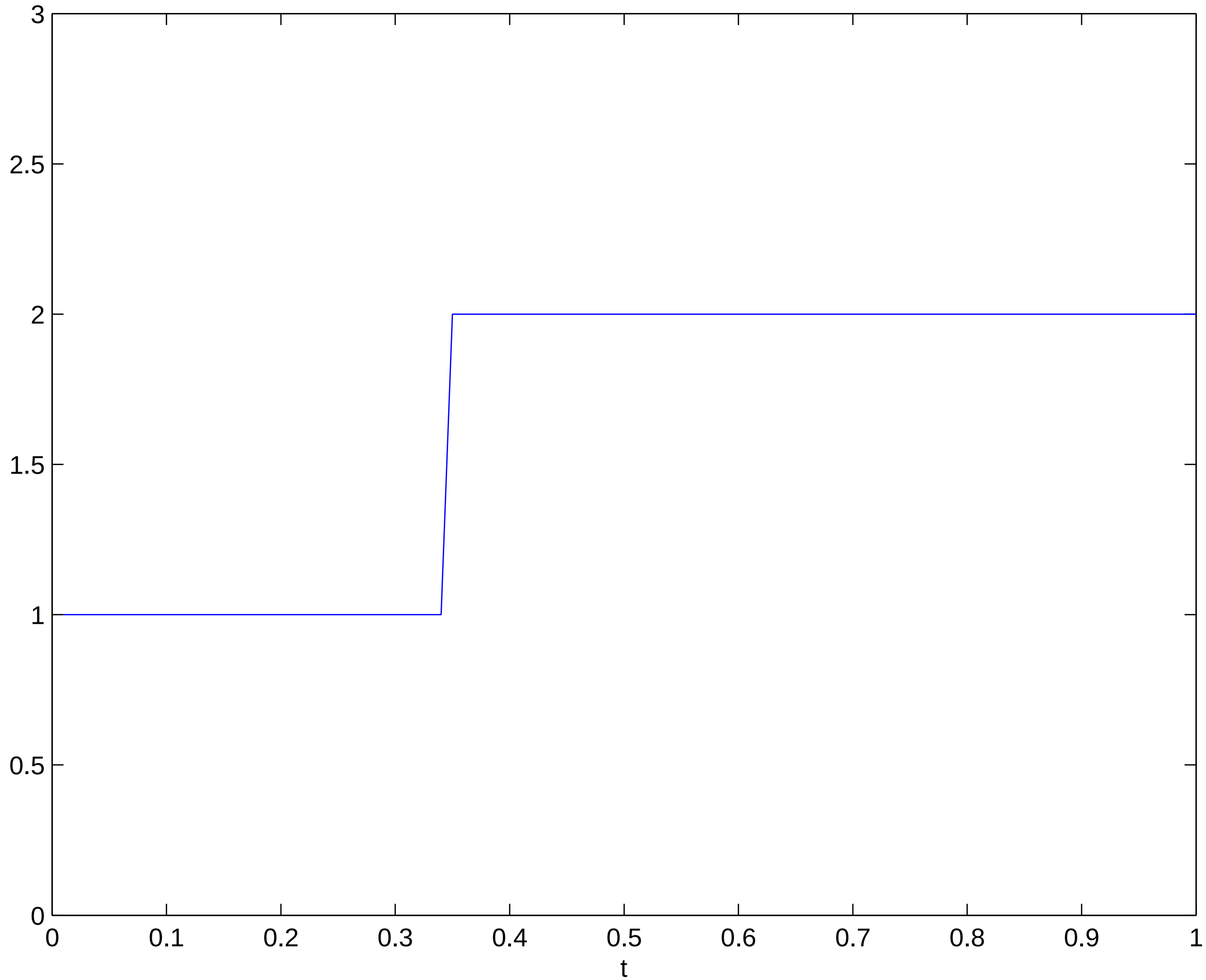

Figure 10: The graph shows the number of iterations used by Newton's method to solve the system (50) at each time-level.