Example 7: $f(x) = \ln(1+x)$ and $c = 0$.

We find: $f^{(k)}(x) = (-1)^{k-1}(k-1)! \frac{1}{(1+x)^k}$

Thus:

$$\ln(1+x) = \sum_{k=1}^{n} (-1)^{k-1} \frac{(k-1)!}{k!} x^k + E_n(x)$$

$$= \underbrace{\sum_{k=1}^{n} \frac{(-1)^{k-1}}{k} x^k}_{\text{Polynomial}} + \underbrace{(-1)^n \frac{1}{n+1} \frac{1}{(1+\xi_x)^{n+1}} x^{n+}}_{\text{remainder}}$$

Question: For which x does $\lim E_n(x) \to 0$ as $n \to \infty$?

Here: $\lim_{n \to \infty} E_n(x) = 0$ iff $0 \leq x \leq 1$

This means that the Taylor series represents $\ln(1+x)$ only for $x \in [0,1]$.

Putting it into practice: Compute $\cos(0.1)$ given its Taylor series approximation at $c=0$:

$$\cos(x) = \underbrace{1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}}_{} \pm \ldots + \text{remainder}$$

thus

$$\underbrace{\left| \cos(x) - \sum_{k=0}^{n} (-1)^k \frac{x^{2k}}{(2k)!} \right|}_{} = \left| (-1)^{n+1} \cos(\xi_x) \frac{x^{2(n+1)}}{(2(n+1))!} \right|$$

$$\leq \frac{(0.1)^{2(n+1)}}{(2(n+1))!}$$

n	Taylor poly	$\lvert\text{error}\rvert \le$
0	1	$\frac{(0.1)^2}{2!} = \frac{0.01}{2} = 0.005$
1	0.995	$\frac{(0.1)^4}{4!} = \frac{0.0001}{24}$
2	$0.9950041\overline{6}$	$0.000001/6!$

Theorem 8: Reformulation of Taylor's theorem

Let $f \in C^{n+1}([a,b])$, then letting $c = x$ and $x = c + h = x + h$ in the previous version of Taylor's theorem, we find: For $x, x+h \in [a,b]$

$$f(x+h) = \sum_{k=0}^{n} \frac{f^{(k)}(x)}{k!} h^k + \frac{f^{(n+1)}(\xi_h)}{(n+1)!} h^{n+1}$$

where ξ_h is between x and $x+h$.

We write the error term as

$$f(x+h) - \sum_{k=0}^{n} \frac{f^{(k)}(x)}{k!} h^k = \mathcal{O}(h^{n+1})$$

Landau symbol

Recall: $a(h) = \mathcal{O}(b(h))$

iff $\exists\, c > 0$ s.th. $\frac{a(h)}{b(h)} \le c$ as $h \to 0$

So, for $n = 1$ the error decreases with h^2 which is called quadratic convergence

For $n = 2$ the error decreases cubically, i.e with h^3, etc.

Summary:

- Problem: Evaluate $f(x)$ with a given error bound e.
- Required: $f \in C^{n+1}$, values of derivatives $f^{(k)}($
- Check interval of convergence, does Taylor series expansion work?
- Estimate the maximum error for n terms of the Taylor poly.
- choose n such that the error bound e is met.
- Evaluate the Taylor poly to get the result

I.1 Number representations:

Remember that for any $b \neq 1$, $b \in \mathbb{N}$ every natural number $x \in \mathbb{N}$ can be represented as

$$x = a_0 b^0 + a_1 b^1 + a_2 b^2 + \dots + a_n b^n$$
$$= \sum_{i=0}^{n} a_i b^i$$

where $a_i \in \{0, \dots, b-1\}$ and
- b is called the base
- a_i are called digits

Real number can also be expressed in this way, $x \in \mathbb{R}$

$$x = \sum_{i=0}^{n} a_i b^i + \sum_{i=1}^{\infty} \alpha_i b^{-i}$$

$$= a_n a_{n-1} \dots a_0 \,.\, \alpha_1 \alpha_2 \alpha_3 \dots$$

Examples 8:

(1) Base $b = 10$: $37294 = 4 \cdot 10^0 + 9 \cdot 10^1 + 2 \cdot 10^2 + 7 \cdot 10^3 + 3 \cdot 10^4$

(2) Base $b = 2$: $1011 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2$

$$= (1)_{10} + (2)_{10} + (8)_{10}$$
$$= (11)_{10}$$

to avoid confusion, the base is indicated by a subscript.

There are algorithms that convert between number systems, e.g. Euclid's algorithm for converting $(x)_{10}$ to $(x)_b$:

1) Input $(x)_{10}$
2) Determine smallest n s.th. $x < b^{n+1}$
3) For $i = n$ to 0 do
4) $\quad a_i := x \text{ div } b^i$ integer division
5) $\quad x := x \bmod b^i$ rest
6) end for
7) output result $a_n a_{n-1} \dots a_n = (x)_b$

Example: $(x)_{10} = (13)_{10} \to (x)_2 = ?$

step 2: smallest n is 3, because $13 < 2^4$.

loop: $i = 3$: $a_3 = 13 \text{ div } 2^3 = 13 \text{ div } 8 = 1$

rest $x = 13 \bmod 2^3$
$= 13 \bmod 8 = 5$

$\underline{i=2:}$ $a_2 = 5 \text{ div } 2^2 = 5 \text{ div } 4 = 1$
rest $x = 5 \bmod 2^2$
$= 5 \bmod 4 = 1$

$\underline{i=1:}$ $a_1 = 1 \text{ div } 2^1 = 1 \text{ div } 2 = 0$
rest $x = 1 \bmod 2 = 1$

$\underline{i=0:}$ $a_0 = 1 \text{ div } 2^0 = 1 \text{ div } 1 = 1$

output: $(1\,1\,0\,1)_2 = (13)_{10}$

Euclid's algorithm is intuitive but has 2 Problems:

(a) step 2 is inefficient
(b) division by large numbers b^i can be problematic.

Horner's scheme is better:

$$(a_n a_{n-1} \cdots a_0)_b = a_0 + b(a_1 + b(a_2 + \cdots + b(a_n))$$