Numerical Methods

Lecture 2.

Analysis of errors in numerical methods

Why represent numbers in floating point format?

- Example 1. How a number 256.78 can be represented using 5-places?

2	5	6	·	7	8

What is the smallest number that can be represented in this format?

0	0	0	.	0	0

What is the largest number that can be represented in this format?

9	9	9	.	9	9

Why represent numbers in floating point format?

- Example 2. How a number 256.786 can be represented using 5-places?

rounded off

2	5	6	.	7	9

chopped

2	5	6	.	7	8

Conclusion: The error is smaller than 0.01

Calculation of errors

True Error $\left|x - x_o\right|$

accurate or true value of x_o

Relative True Error $\dfrac{\left|x - x_o\right|}{x_o}$

Calculations:

$$\varepsilon_t = \frac{\left|x - x_o\right|}{x_o} \times 100\% = \frac{\left|256.79 - 256.786\right|}{256.786} \times 100\% = 0.001558\%$$

Calculation of errors

Relative errors of small numbers are large.

Compare:

$$\varepsilon_t = \frac{|x - x_o|}{x_o} \times 100\% = \frac{|256.79 - 256.786|}{256.786} \times 100\% = 0.001558\%$$

$$\varepsilon_t = \frac{|x - x_o|}{x_o} \times 100\% = \frac{|3.55 - 3.546|}{3.546} \times 100\% = 0.11280\%$$

Absolute true errors are the same:

$$|x - x_o| = |256.786 - 256.79| = |3.546 - 3.55| = 0.004$$

How to keep relative true errors at a constant level?

The number can be represented as:

$$\text{sign} \times \text{mantissa} \times 10^{\text{exponent}}$$

or

$$\text{sign} \times \text{mantissa} \times 2^{\text{exponent}}$$

$$256.78 \text{ is written as } +2.5678\times10^{2}$$

$$0.003678 \text{ is written as } +3.678\times10^{-3}$$

$$-256.78 \text{ is written as } -2.5678\times10^{2}$$

Floating point arithmetic – decimal representation

Form of a number

an integer exponent

$$\sigma \times m \times 10^{e}$$

sign (-1 or +1)

mantissa $(1)_{10} \leq m < (10)_{10}$

Example

$$-2.5678 \times 10^{2}$$

$$\sigma = -1$$

$$m = 2.5678$$

$$e = 2$$

Floating point arithmetic – binary representation

Form of a number

an integer exponent

$$\sigma \times m \times 2^{e}$$

sign (0 – positive or 1 – negative)

mantisses $(1)_2 \leq m < (2)_2$

Example:

$$\left(1.1011011\right)_2 \times 2^{(101)_2}$$

$$\sigma = 0$$

$$m = 1011011$$

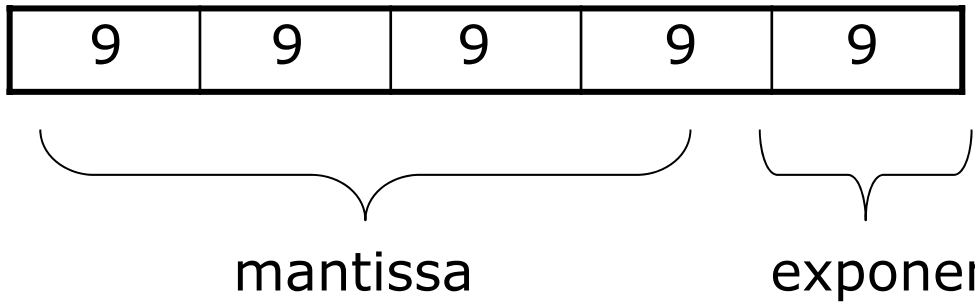
$$e = 101$$

1 is not represented

What do we gain by using the floating point representation?

The rage of the numbers that can be expressed has increased

If we use only 5 places to represent a positive number with a positive exponent, the smallest number that can be expressed is 1 and the largest is $9.999 \cdot 10^9$



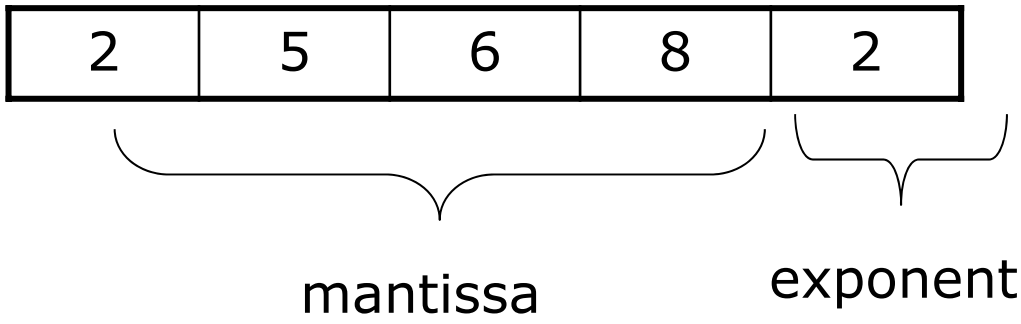
The range of numbers that can be represented has increased from 999.99 to $9.999 \cdot 10^9$.

What do we lose by using a floating point representation?

Precision

Why?

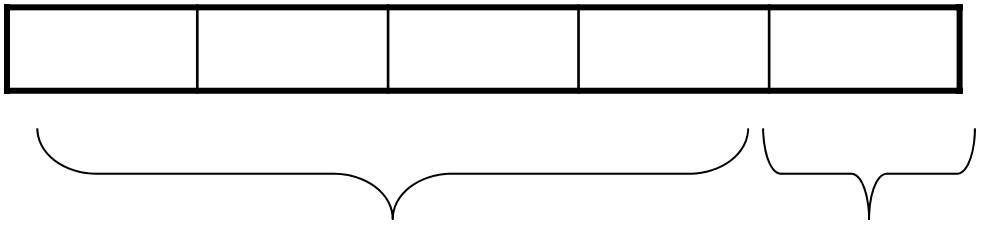
The number 256.78 will be represented as $2.5678 \cdot 10^2$ on five places:



There will be a round-off error.

Test problem 1

1. Write down the number 576329.78 on five places in a similar manner as that discussed in the previous example:



2. Calculate the true error and the relative true error of round off

3. Compare the results with the previous example (256.78).

Test problem 2

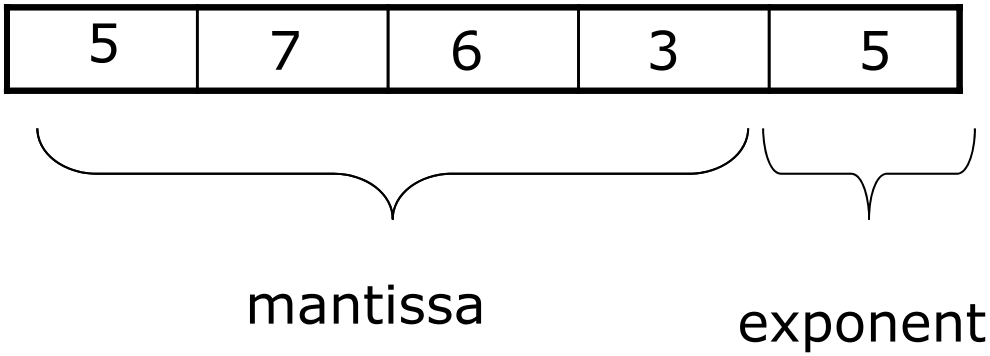
We have 9-bits

- first bit represents the sign of the number,
- second bit represents the sign of the exponent,
- the next four bits represent the mantissa,
- the last three bits represent the exponent

Find the number (in decimal), which is presented above

Solution of the problem 1

1. The number 576329.78 written on five places is:

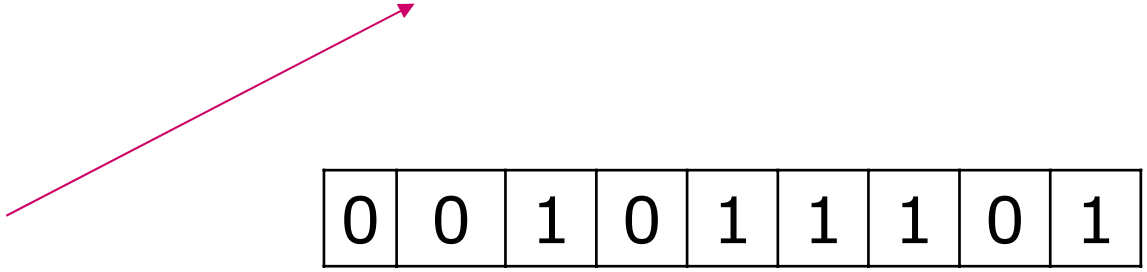


2. The true error is 29.78 and the relative true error is 0.0051672%

3. For the number 256.78 these errors are: 0.02 (smaller) and 0.0077888% (comparable)

Solution of the problem 2

$$(54.75)_{10} = (110110.11)_2 = (1.1011011)_2 \times 2^5$$
$$\cong (1.1011)_2 \times (101)_2$$



is not represented

$$(54)_{10}$$

What is the accuracy ε?

For each digital machine epsilon ε is defined as a parameter determining the accuracy of the calculations:

$$\varepsilon = N^{-t}$$

$$(54.75)_{10} = (110110.11)_2 = (1.1011011)_2 \times 2^5$$
$$\cong (1.1011)_2 \times (101)_2$$

where: N=2 (in binary), N=10 (in decimal), t is the number of bits representing the mantissa

ε is lowered when more bits are allocated to represent the mantissa M

Epsilon ε can be regarded as a parameter characterizing the accuracy of the computing machine (small ε = more accurate calculations).

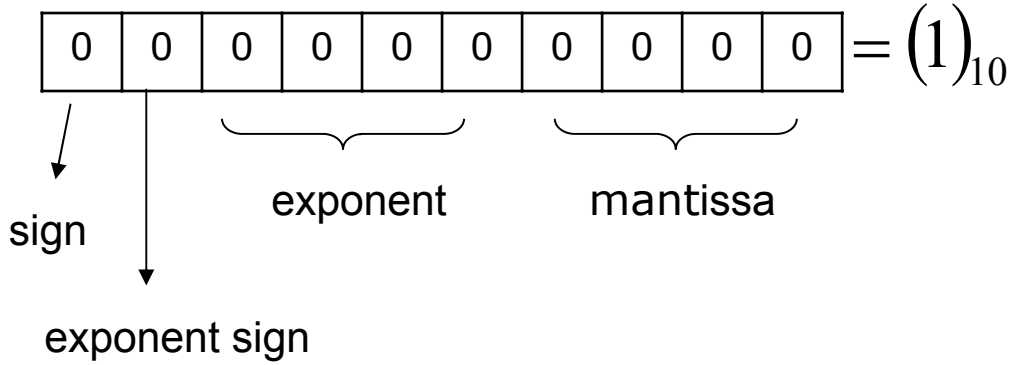
Double precision (Fortran) $\varepsilon_{DP} = \varepsilon^2$

What is the accuracy ε?

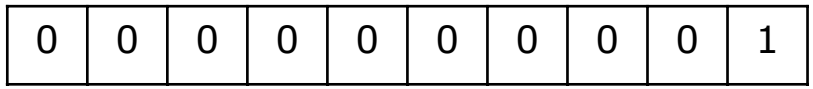
Epsilon ε is the smallest number that when added to 1.000 produces a number that can be represented as different from 1.000.

Example: 10-bit word

$$x = M \times N^{w}$$



the next number

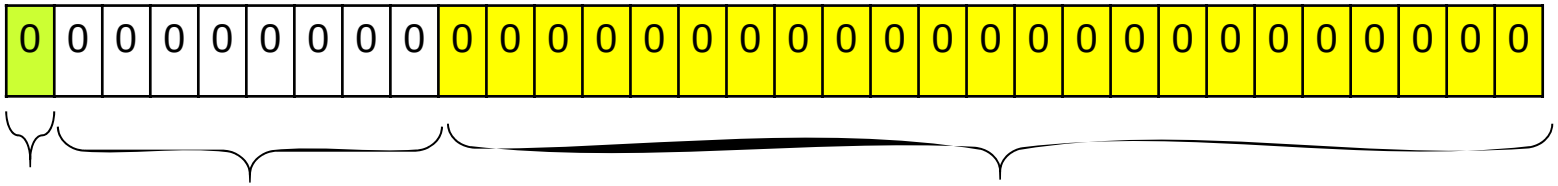


$$= (1.0001)_2 = (1.0625)_{10}$$

$$\in_{mach} = 1.0625 - 1 = 2^{-4}$$

Single precision in IEEE-754 format (Institute of Electrical and Electronics Engineers)

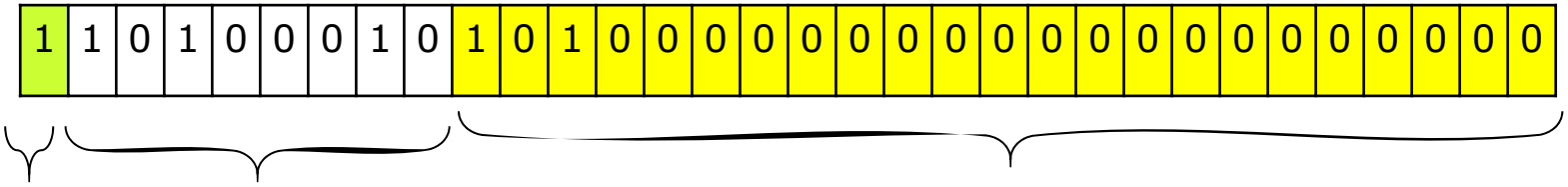
32 bits for single precision



sign (s) exponent (e') mantissa (m)

$$Number = (-1)^s \times (1.m)_2 \times 2^{e'-127}$$

Example



$$\text{Value} = (-1)^s \times (1.m)_2 \times 2^{e'-127}$$

$$= (-1)^1 \times (1.10100000)_2 \times 2^{(10100010)_2 - 127}$$

$$= (-1) \times (1.625) \times 2^{162-127}$$

$$= (-1) \times (1.625) \times 2^{35} = -5.5834 \times 10^{10}$$

The exponent for the 32-bit IEEE -754 standard

8-bit exponent means $0 \le e' \le 255$

The fixed exponent shift is 127 and therefore

$$-127 \le e \le 128$$

In fact, $1 \le e' \le 254$

Numbers $e' = 0$ and $e' = 255$ are reserved for special cases

Exponent range $-126 \le e \le 127$

Representation of special numbers

$e' = 0$ — all zeros

$e' = 255$ — all ones

s	e'	m	Represents
0	zeros	zeros	0
1	zeros	zeros	-0
0	ones	zeros	∞
1	ones	zeros	$-\infty$
0 or 1	ones	nonzero	NaN

IEEE-754 Format

The largest number

$$(1.1.....1)_2 \times 2^{127} = 3.40 \times 10^{38}$$

The smallest number

$$(1.00.....0)_2 \times 2^{-126} = 2.18 \times 10^{-38}$$

Epsilon

$$\varepsilon_{mach} = 2^{-23} = 1.19 \times 10^{-7}$$

Analysis of errors

If we do not know the exact value of x_o we calculate the approximate error as the difference in the values obtained in the consecutive approximations:

$$x_n - x_{n-1}$$

Relative error ε_a :

$$\varepsilon_a = \frac{x_n - x_{n-1}}{x_n}$$

Example

For $f(x) = 7e^{0.5x}$ at $x = 2$ find

a) $f'(2)$ for $h = 0.3$

b) $f'(2)$ for $h = 0.15$

c) approximate error

Solution

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

a) $h = 0.3$

$$f'(2) \approx \frac{f(2+0{,}3) - f(2)}{0{,}3} = \frac{7e^{0{,}5(2{,}3)} - 7e^{0{,}5(2)}}{0{,}3} = 10{,}265$$

Example (cont.)

b) $h = 0.15$

$$f'(2) \approx \frac{f(2+0{,}15) - f(2)}{0{,}15} = \frac{7e^{0{,}5(2{,}15)} - 7e^{0{,}5(2)}}{0{,}15} = 9{,}880$$

c) $$\varepsilon_a = \frac{x_n - x_{n-1}}{x_n}$$

$$\varepsilon_a = \frac{9{,}880 - 10{,}265}{9{,}8800} \approx -0{,}0389$$

Error 3,89%

The relative error as a criterion for ending the iterative procedure

If the relative error is less than or equal to a predetermined number then further iterations are no longer required

$$|\varepsilon_a| \leq \varepsilon_s$$

If we require at least m significant digits in the result then

$$|\varepsilon_a| \leq 0.5 \times 10^{2-m}$$

Relative error and significant digits

$$|\varepsilon_a| \leq 0.5 \times 10^{2-m}$$

h	$f'(2)$	$\lvert\varepsilon_a\rvert$	m
0.3	10.265	N/A	0
0.15	9.8800	0.03894	1
0.10	9.7559	0.01271	1
0.01	9.5378	0.02286	1
0.001	9.5164	0.00225	2

The exact value is 9.514

Sources of errors in numerical calculations

1. Input errors (input data errors)
2. Truncation error
3. Round off error

Input errors occur when the input data entered into the computer memory are different from the exact values.

Truncation errors are errors due to numerical procedures caused by reducing the number of operations.

Round off errors are errors that usually can not be avoided. They arise in the course of the calculations, and can be reduced by setting skillfully the method and sequence of tasks.

Input errors

Sources of input errors:

- the input data are the result of measurement of physical quantities
- a finite length of binary words and therefore pre-rounding is needed
- preliminary rounding of irrational numbers

Rounding of numbers that cannot be expressed exactly is accomplished by:

- chopping
- rounding off

Example:

$$\pi \approx 3{,}14159265359$$

$$\pi \approx 3{,}1415$$

chopping

$$\pi \approx 3{,}1416$$

rounding off

Rounding off introduces smaller error than chopping.

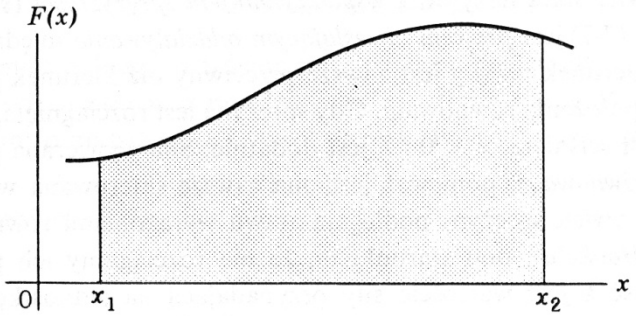
Truncation error

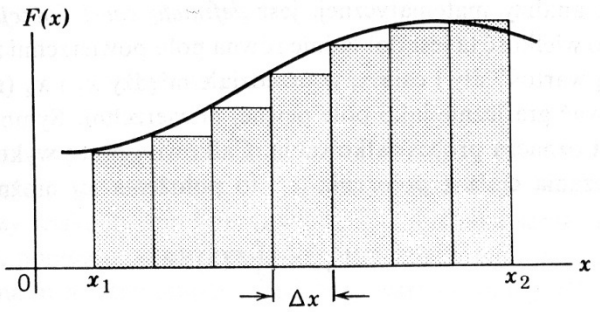
Caused by the use of the approximate formula instead of a full mathematical operation:

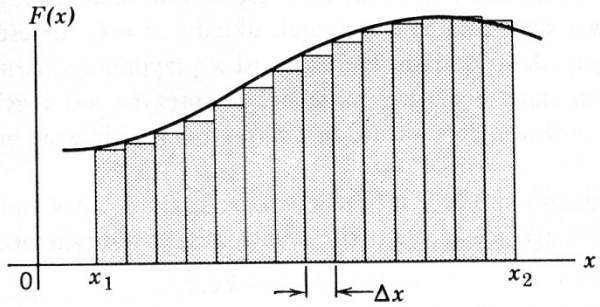
- when calculating the sum of the infinite series
- when calculating integral, derivative

$$W = \lim_{\Delta x \to 0} \sum_{x_1}^{x_2} F \Delta x = \int_{x_1}^{x_2} F dx$$

work







Taylor series

If the function is continuous and all derivatives f ', f'', ... f^n exist within the interval [x, x + h] the value of the function at the point x + h can be calculated as:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \cdots$$

The Maclaurin series is simply the Taylor series about the point x=0

$$f(0+h) = f(0) + f'(0)h + f''(0)\frac{h^2}{2!} + f'''(0)\frac{h^3}{3!} + \cdots +$$

Examples

Some examples of Taylor series which you must have seen

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Truncation error in the Taylor series

$$f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2!} + \cdots + f^{(n)}(x)\frac{h^n}{n!} + R_n(x)$$

remainder

$$R_n(x) = \frac{(x-h)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

$$x < c < x + h$$

Example

The Taylor series for e^x at point x=0 is given by

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots$$

It can be seen that as the number of terms used increases, the error decreases and hence a better estimate can be found.

Question: How many terms would it require to get an approximation of e^1 with a true error of less than 10^{-6}?

$$e^1 = 1 + 1 + \frac{1^2}{2!} + \frac{1^3}{3!} + \frac{1^4}{4!} + \frac{1^5}{5!} + \cdots$$

$$\approx 2 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120}$$

Solution

$$x = 0, h = 1, f(x) = e^x \qquad R_n(x) = \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

$$R_n(0) = \frac{1^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

$$= \frac{(1)^{n+1}}{(n+1)!} e^c$$

since

$$x < c < x + h$$

$$0 < c < 0 + 1$$

$$0 < c < 1$$

$$\frac{1}{(n+1)!} < |R_n(0)| < \frac{e}{(n+1)!}$$

Solution

$$\frac{e}{(n+1)!} < 10^{-6}$$ true error

$$(n+1)! > 10^6 e$$

$$(n+1)! > 10^6 \times 3$$

$$n \geq 9$$

It means that 9 terms or more are needed to get a true error less than 10^{-6}

Arithmetic operations

1. Addition and subtraction

To add or subtract two standard floating-point numbers, the exponents should be made equal with an adequate shift of the mantissa.

Example: Add $0{,}4546 \cdot 10^5$ to $0{,}5433 \cdot 10^7$

shift

$$0{,}0045 \cdot 10^7 + 0{,}5433 \cdot 10^7 = 0{,}5478 \cdot 10^7$$

Answer: We loose some significant digits

Arithmetic operations

2. Multiplication

Example: Multiply $0{,}5543 \cdot 10^{12}$ by $0{,}4111 \cdot 10^{-15}$

Multiply mantissas and add the exponents.

$$0{,}5543 \cdot 10^{12} \cdot 0{,}4111 \cdot 10^{-15} = 0{,}2278273 \cdot 10^{-3} = 0{,}2278 \cdot 10^{-3}$$

3. Division

Example: Divide $0{,}1000 \cdot 10^{5}$ by $0{,}9999 \cdot 10^{3}$

$$0{,}1000 \cdot 10^{5} / 0{,}9999 \cdot 10^{3} = 0{,}1000 \cdot 10^{2}$$

All the time we loose some significant digits which is the source of the error

Sequence of operations

(a+b)-c≠(a-c)+b lack of associativity

a(b-c) ≠(ab-ac) lack of distributive property

Example: a= $0{,}5665 \cdot 10^{1}$, b=$0{,}5556 \cdot 10^{-1}$,
c=$0{,}5644 \cdot 10^{1}$

$(a+b)=0{,}5665 \cdot 10^{1}+0{,}5556 \cdot 10^{-1}$

$=0{,}5665 \cdot 10^{1}+0{,}0055 \cdot 10^{1}=0{,}5720 \cdot 10^{1}$

$(a+b)-c=0{,}5720 \cdot 10^{1}-0{,}5644 \cdot 10^{1}=0{,}7600 \cdot 10^{-1}$

$(a-c)=0{,}5665 \cdot 10^{1}-0{,}5644 \cdot 10^{1}=0{,}0021 \cdot 10^{1}=0{,}2100 \cdot 10^{-1}$

$(a-c)+b=0{,}2100 \cdot 10^{-1}+0{,}5556 \cdot 10^{-1}=0{,}7656 \cdot 10^{-1}$

Wilkinson Lemma

Rounding off errors occurring during floating point operations are equivalent to substitution of accurate numbers by slightly perturbed numbers on which we act exactly.

For individual arithmetical operations:

$$fl(x_1 \pm x_2) = x_1(1+\varepsilon_1) \pm x_2(1+\varepsilon_2)$$

$$fl(x_1 \cdot x_2) = x_1(1+\varepsilon_3) \cdot x_2 = x_1 \cdot x_2(1+\varepsilon_3)$$

$$fl(x_1 / x_2) = x_1(1+\varepsilon_4) / x_2 = x_1 /(x_2(1+\varepsilon_5))$$

symbol of operation performed on floating-point data

and

$$\varepsilon_i \leq \varepsilon$$

Tragic example of the rounding off error

On February 25th, 1991 in Dhahran, Saudi Arabia, 28 American soldiers were killed in the attack of the Iraqi Scud missiles. The defense system Patriot did not detect any assault. Why?

The system calculates the area, which should be scanned based on the speed of the object and the last detection. The internal clock was set to measure every 1/10 second and 24-bit word length was assumed. Due to rounding off, the absolute error was $9.5\ 10^{-8}$ s which after 100 hours amounted to:

$$9.5 \cdot 10^{-8} \times 10 \times 60 \times 60 \times 100 = 0.34 \text{ sec}$$

Based on this, the calculated displacement is 687 m. Object is considered outside the range when the displacement is 137 m

Useful hints:

In the numerical calculations it is advantageous:

- To solve the same problem by another method, or by the same method, but with a different order of operations
- To solve the problem on slightly perturbed input data

Problems for individual studies

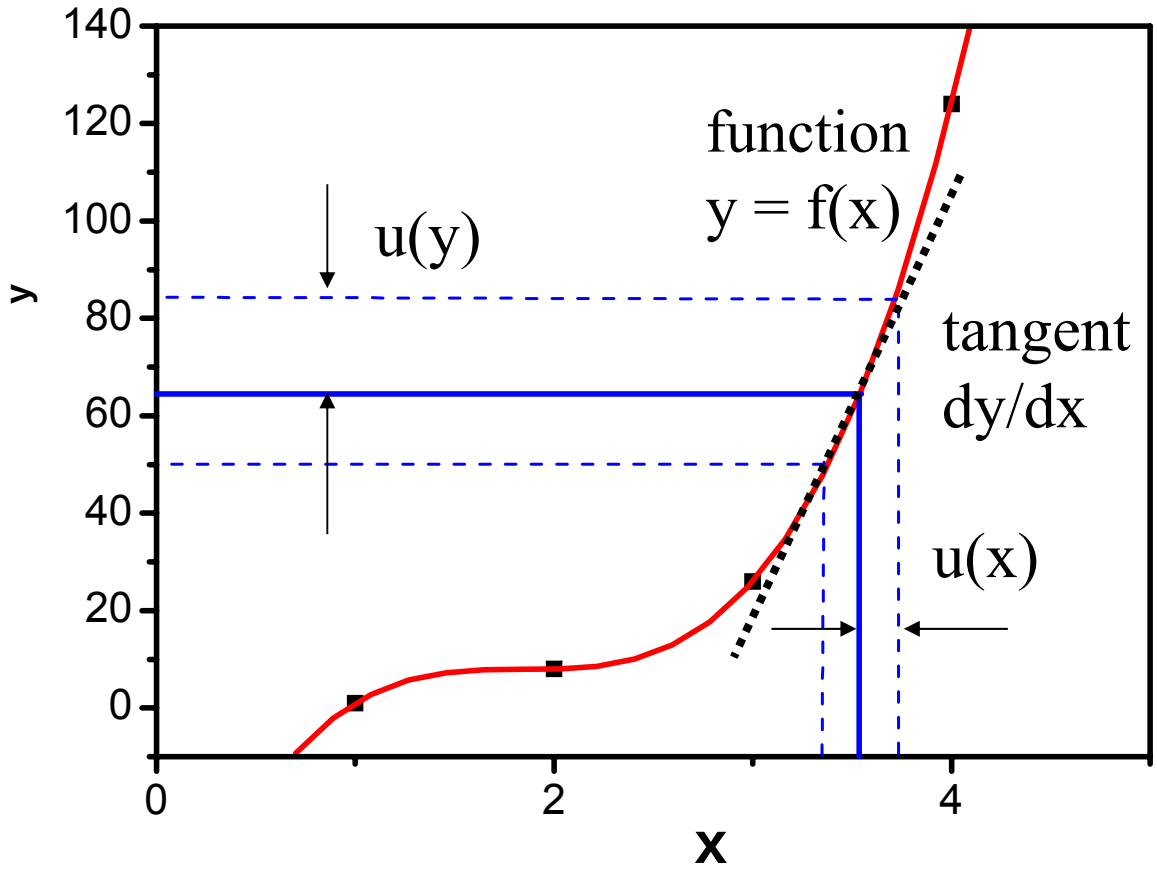
1. The series $\sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} 4^n$

is a Maclaurin series for the following function:

(A) cos(x); B) cos (2x); C) sin (x); D) sin (2x)

2. The number 1/10 is registered in a fixed 6 bit-register with all bits used for the fractional part. The difference gets accumulated every $1/10^{th}$ of a second for one day. Find the magnitude of accumulated difference.

Propagation of errors



$$u(y) = \frac{dy}{dx} u(x)$$

Total differential in application to errors

For $y=f(x_1,x_2,...x_n)$ where maximum uncertainties Δx_1, Δx_2, ... Δx_n are small compared with the values of the variables $x_1,x_2,$ ... x_n the maximum uncertainty of y can be expressed as:

$$\Delta y = \left|\frac{\partial y}{\partial x_1}\right|\left|\Delta x_1\right| + \left|\frac{\partial y}{\partial x_2}\right|\left|\Delta x_2\right| + \ldots + \left|\frac{\partial y}{\partial x_n}\right|\left|\Delta x_n\right|$$

Example

Estimate the uncertainty of density ρ of a sphere of mass m and radius R

$$\rho(m,R)=\frac{m}{(4/3)\pi R^3}$$

absolute error

$$\Delta\rho=\left|\frac{\partial\rho}{\partial m}\right|\left|\Delta m\right|+\left|\frac{\partial\rho}{\partial R}\right|\left|\Delta R\right|$$

but

$$\frac{\partial\rho}{\partial m}=\frac{1}{(4/3)\pi R^3}\qquad\frac{\partial\rho}{\partial R}=\frac{-3m}{(4/3)\pi R^4}$$

relative error

$$\varepsilon_\rho=\varepsilon_m+3\varepsilon_R$$

Errors of arithmetic operations

Addition

$$A = a \pm \Delta a \qquad B = b \pm \Delta b$$

absolute errors

$$A + B = a + b \pm \Delta a \pm \Delta b = a + b \pm \Delta(a + b)$$

the addition absolute error

Therefore, the addition (subtraction) absolute error is equal to the sum of absolute errors of components.

$$\Delta(a \pm b) = \Delta a + \Delta b$$

Errors of arithmetic operations

the addition relative error

$$\varepsilon_{a+b} = \frac{\Delta a + \Delta b}{a + b}$$

the subtraction relative error

$$\varepsilon_{a-b} = \frac{\Delta a + \Delta b}{a - b}$$

Subtraction relative error can be large even if the relative errors of minuend and subtrahend are small. Subtracting of nearly equal numbers should be avoided !

This phenomenon is called **reduction** of significant digits

It is important for the calculation of Newton's difference quotients approximating derivatives of functions, roots of a quadratic equation with the dominant factor of the first power, etc.

The concept of zero

We lose the precise meaning of the number 0 if we perform numerical calculations

$$x^2 + 2x - 2 = 0$$

Exact roots are

$$-1 \pm \sqrt{3}$$

Approximate solutions

$0{,}7320 \cdot 10^0$

$-0.2732 \cdot 10^1$

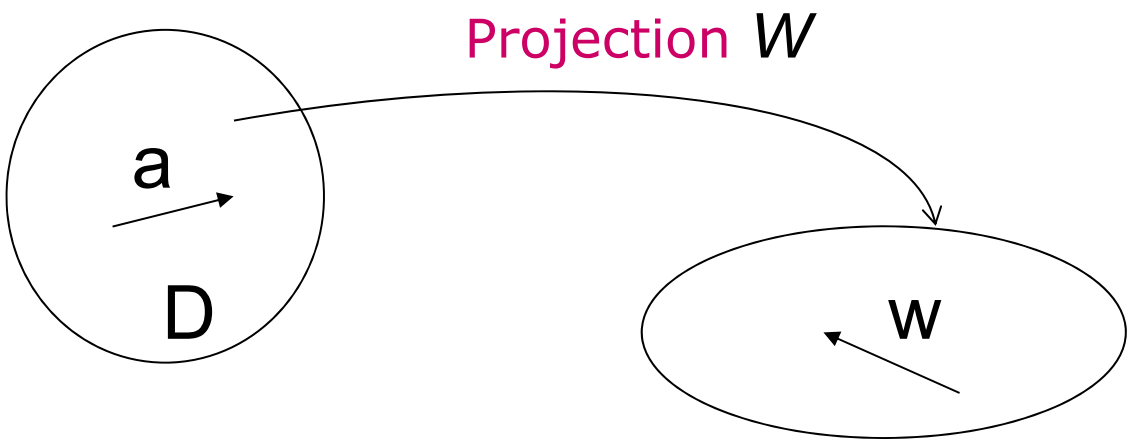
Verify that upon substitution of the approximate solutions to x^2+2x-2 you will not get 0!!!

You should therefore avoid subtraction and the loop condition should not be set to "zero",

$$\text{if } a-b<\varepsilon$$

Tasks and numerical algorithms

- **Numerical task** requires a clear and unambiguous description of the functional relationship between *the input* or "independent variables" of the task and *output data*, i.e. searched results.
- Numerical task is a problem determining the results vector $\vec{\mathbf{w}}$ on the basis of data vector $\vec{\mathbf{a}}$



task well-defined

$$\vec{\mathbf{w}} = W(\vec{\mathbf{a}})$$

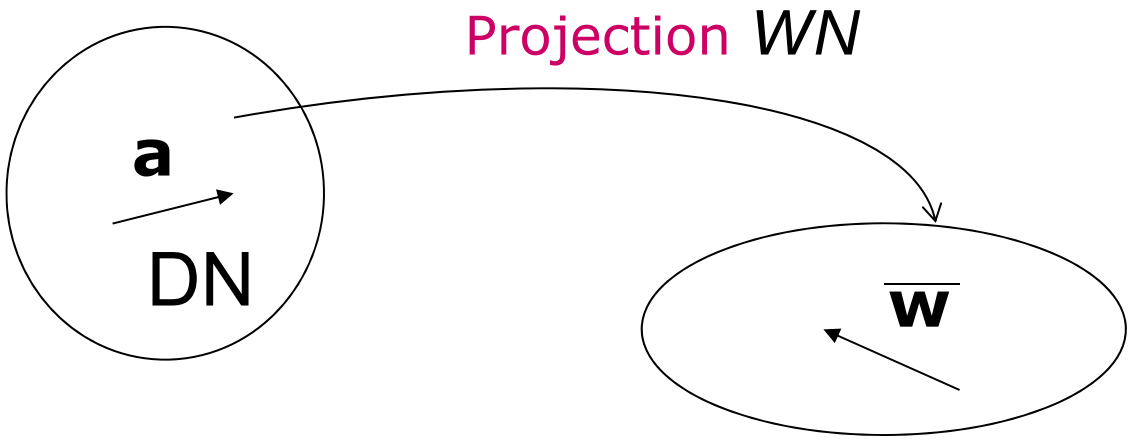
unambiguous assignment

Tasks and numerical algorithms

- **Numerical algorithm** is a full description of the *operations* correctly transforming the input data vector to the vector of output data.
- The algorithm is formulated correctly when the number of necessary actions is finite

$$DN \cap D \neq \varnothing$$

$$\overline{\mathbf{w}} = WN(\mathbf{a}, \varepsilon)$$



Resulting vector depends on the computing accuracy ε of the machine

Examples of algorithms

Given a complex number a=x+iy. Calculate $1/a^2$

Algorithm I:

1. $t = y / x$ (tangent phase of number a)
2. $|a|^2 = x^2 + y^2$ (module of number a)
3. $\mathrm{Re}\left(\frac{1}{a^2}\right) = \frac{1}{/a/^2}\frac{1-t^2}{1+t^2}$ $\mathrm{Im}\left(\frac{1}{a^2}\right) = \frac{1}{/a/^2}\frac{-2t^2}{1+t^2}$

The task is well-defined if: $x^2 + y^2 \neq 0$

that is: $D = R^2 - \{(0,0)\}$

This algorithm is formulated correctly (11 necessary steps)

Examples of algorithms

Not for each data pair (x, y) ≠ 0 the solution to the problem can be found using the algorithm I.

1. There will be an overflow of floating point numbers (for x = 0, also because of rounding to zero)

2. The overflow may happen even in the first step when

$x = 10^{-25}$ and $y = 10^{25}$ because of the division y / x

3. For x = 0, and y ≠ 0 the solution cannot be determined using this algorithm. The increase in the accuracy of the calculation does not change this fact.

Algorithm I is not numerically stable

Examples of algorithms

Given a complex number a=x+iy. Calculate $1/a^2$

Algorithm II:

1. $r = \mathrm{Re}\left(\frac{1}{a^2}\right) = \frac{x^2 - y^2}{x^2 + y^2}$

2. $u = \mathrm{Im}\left(\frac{1}{a^2}\right) = \frac{-2xy}{x^2 + y^2}$

Algorithm II is formulated correctly (9 necessary steps)

Algorithm II is numerically stable due to the continuity of equations for

$$x^2 + y^2 \neq 0$$

Conditioning and stability

The calculation algorithm is numerically stable if for any selected data vector

$$\boldsymbol{a_0} \in \boldsymbol{D}$$

there exists an accuracy of the calculations ε_0, that for $\varepsilon<\varepsilon_0$ we have

$$\boldsymbol{a_0} \in \boldsymbol{DN(\varepsilon)}$$

and

$$\lim_{\varepsilon \to 0} WN(\boldsymbol{a}_0, \varepsilon) = W(\boldsymbol{a}_0)$$

The algorithm is numerically stable when increasing the accuracy of the calculations any existing solution to the problem can be found.

Conditioning and stability

Due to the rounding off error

$$WN(\boldsymbol{a}, \varepsilon) \neq W(\boldsymbol{a})$$

By lemma of Wilkinson disturbed data vector can be selected

$$\boldsymbol{a} + \delta\boldsymbol{a} \in \boldsymbol{D}$$

for which $WN(\boldsymbol{a}, \varepsilon) = W(\boldsymbol{a} + \delta\boldsymbol{a})$

Size of the disturbance $\delta\boldsymbol{a}$ depends on:

-the data vector **a**

-the number of performed operations

-the accuracy of calculations

Conditioning and stability

Based on Wilkinson's Lemma:

$$\frac{\|\delta \boldsymbol{a}\|}{\|\boldsymbol{a}\|} \to 0$$

when $\varepsilon \to 0$

Conditioning tells us how much the result of the disturbed vector data differs from the exact result for vector data that is:

$$\boldsymbol{W(a+\delta a)} \qquad \boldsymbol{W(a)}$$

Condition number B(a) is the number for which the following formula is satisfied:

$$\frac{\|\delta w\|}{\|w\|} \le B(a)\frac{\|\delta \boldsymbol{a}\|}{\|\boldsymbol{a}\|} \qquad \delta w = WN(a,\varepsilon) - W(a)$$

Condition number

- Let us assume the relative error of x

$$\frac{x-\tilde{x}}{\tilde{x}}$$

- The relative error of f(x)

$$\frac{f(x)-f(\tilde{x})}{f(\tilde{x})} \approx \frac{f'(\tilde{x})(x-\tilde{x})}{f(\tilde{x})}$$

- Condition number:

$$\frac{\tilde{x}f'(\tilde{x})}{f(\tilde{x})}$$

Condition number

- Example

$$f(x) = \sqrt{x}$$

- Condition number:

$$\frac{\tilde{x} f'(\tilde{x})}{f(\tilde{x})} = \frac{x \dfrac{1}{2\sqrt{x}}}{\sqrt{x}} = \frac{1}{2}$$

task is well-conditioned when condition number is small

Condition number

- Example

$$f(x) = \frac{10}{1 - x^2}$$

- Condition number:

$$\frac{\tilde{x} f'(\tilde{x})}{f(\tilde{x})} = \frac{2x^2}{\left|1 - x^2\right|}$$

ill-conditioned task near x=1 i x=-1