Mathematical variables, parameters and placeholders

Equations and linear equation systems

e.g.: $2 \cdot x^2 + 5 \cdot x - 2 = 0$ is equivalent to $x^2 + \frac{5}{2} \cdot x - 1 = 0$.

Application of pq-formula results in $x_{1,2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$.

With $p = \frac{5}{2}$ and $q = -1$, we get

$x_1 = -\frac{\frac{5}{2}}{2} + \sqrt{\left(\frac{\frac{5}{2}}{2}\right)^2 + 1}$; and $x_2 = -\frac{\frac{5}{2}}{2} - \sqrt{\left(\frac{\frac{5}{2}}{2}\right)^2 + 1}$;

$$-\frac{5}{4} - \frac{1}{4}\sqrt{41} \qquad \textbf{(1)}$$

'Nice to have' is something re-usable.
##show

$pq := \textbf{proc}(p, q)$

$\quad \textbf{return} -\frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - q}, -\frac{p}{2} - \sqrt{\left(\frac{p}{2}\right)^2 - q}$; *#returns 2 comma-separated expressions*

end proc;

```
proc(p, q)                                                            (2)
    return  -1/2*p + sqrt(1/4*p^2 - q),  -1/2*p - sqrt(1/4*p^2 - q)
end proc
```

$pq\left(\frac{5}{2}, -1\right)$;

$$-\frac{5}{4} + \frac{1}{4}\sqrt{41}, -\frac{5}{4} - \frac{1}{4}\sqrt{41} \qquad \textbf{(3)}$$

##re-usable, time-dependent variables. Here place-holder a:

$a := 2$;

$$2 \qquad \textbf{(4)}$$

$a := x^2 + 1$;

$$x^2 + 1 \qquad \textbf{(5)}$$

$a := a + 1$;

$$x^2 + 2 \qquad \textbf{(6)}$$

Simplification and Evaluation (numeric vs. symbolic, algorithmic vs. heuristic)

$restart;$

Numbers: $\frac{18}{6}, \frac{18.01}{6.03}, \sqrt{2}, 6\cdot\sqrt{2}, \sqrt{2}^2;$

$$3, 2.986733002, \sqrt{2}, 6\sqrt{2}, 2 \quad \textbf{(7)}$$

$evalb(4 < 3);$

$$false \quad \textbf{(8)}$$

$evalb(2 < 3);$

$$true \quad \textbf{(9)}$$

$evalb(\sqrt{2} < 3);$

$$\sqrt{2} < 3 \quad \textbf{(10)}$$

Symbolic expressions: $\frac{a\cdot(b+1)}{a};$

$$b + 1 \quad \textbf{(11)}$$

$factor\left(x^2 + \frac{2\cdot p}{2}\cdot x + \left(\frac{p}{2}\right)^2\right);$ *#symbolic, exact, algorithmic*

$$\frac{1}{4}(p + 2x)^2 \quad \textbf{(12)}$$

$x := 2;$

$$2 \quad \textbf{(13)}$$

> $\mathrm{sqrt}(a^2);$

$$\sqrt{a^2} \quad \textbf{(14)}$$

> $simplify(\mathrm{sqrt}(a^2));$

$$\mathrm{csgn}(a)\, a \quad \textbf{(15)}$$

> $\mathrm{sqrt}(a^2)$ assuming $a < 0;$

$$-a \quad \textbf{(16)}$$

> $simplify(\mathrm{sqrt}(a^2))$ assuming $a :: real, a > 0;$

$$a \quad \textbf{(17)}$$

> $simplify(\mathrm{sqrt}(a^2))$ assuming $a :: real;$

$$|a| \quad \textbf{(18)}$$

>

The following expression leads to a surprising answer. Why? Thus: be careful!

> $simplify(\sin(x)^2 \cdot x^4 + \cos(x)^2 \cdot x^4);$

$$16 \quad \textbf{(19)}$$

> $simplify(\sin(y)^2 \cdot y^4 + \cos(y)^2 \cdot y^4);$

$$y^4 \quad \textbf{(20)}$$

> $restart;$

> $simplify(\sin(x)^2 \cdot x^4 + \cos(x)^2 \cdot x^4);$

$$x^4 \quad \textbf{(21)}$$

Complex Numbers

- a complex number z is of the form a + bi, with i^2= -1 and a,b ∈ℝ. a = Re(z) is the real part of z and b=Im(z)
  is the imaginary part of z. An equivalent definition is via a two dimensional vector (a,b).
- two complex numbers are equal if and only if their real parts and their imaginary parts are equal

- Complex numbers are added, subtracted, multiplied, and divided by formally applying the associative,
  commutative and distributive laws of algebra, together with the equation i^2 = -1.
  - Addition : (a+bi) + (c+di) = (a+c) + (b+d)i [in vector notation: (a,b) + (c,d) = (a+c, b+d)]
  - Substraction : (a+bi) - (c+di) = (a-c) + (b-d)i
  - Multiplication: $(a+bi)\cdot(c+di) = (ac-bd) + (bc+ad)i$
  - Division : $\frac{a+bi}{c+di} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}i$, with c or d not equal to 0

- with the given definitions of addition, substraction, multiplication, division, and
  - the additive identity (zero-element) 0 + 0i,
  - the multiplicative identity (one-element) 1 + 0i,
  - the addidive inverse of a number a + bi: -a - bi, and
  - the multiplicative inverse of a + bi: $\frac{a}{a^2+b^2} + \frac{-b}{a^2+b^2}i$,

  the complex numbers ℂ are a *field* (dt: Körper)

Numeric complex computations

> $\frac{(3+3\cdot I)}{(2+6\cdot I)};$

$$\frac{3}{5} - \frac{3}{10}\,\mathrm{I} \quad \textbf{(22)}$$

> $\left(\frac{3}{3^2+5^2} + \frac{(-5)}{3^2+5^2}\cdot I\right)\cdot(3+5\cdot I);$

$$1 \quad \textbf{(23)}$$

Symbolic complex computations
Simplifying an expression

> $restart;$

> $\left(\frac{a}{a^2+b^2}+\frac{-b}{a^2+b^2}\cdot I\right)\cdot(a+b\cdot I)$ assuming $a>0$;

$$\left(\frac{a}{a^2+b^2}-\frac{\mathrm{I}\,b}{a^2+b^2}\right)(a+\mathrm{I}\,b) \tag{24}$$

>

> $simplify(\%);$

$$-\frac{-a^2-b^2}{a^2+b^2} \tag{25}$$

> $solve(x^2+1=0);$

$$\mathrm{I},\ -\mathrm{I} \tag{26}$$

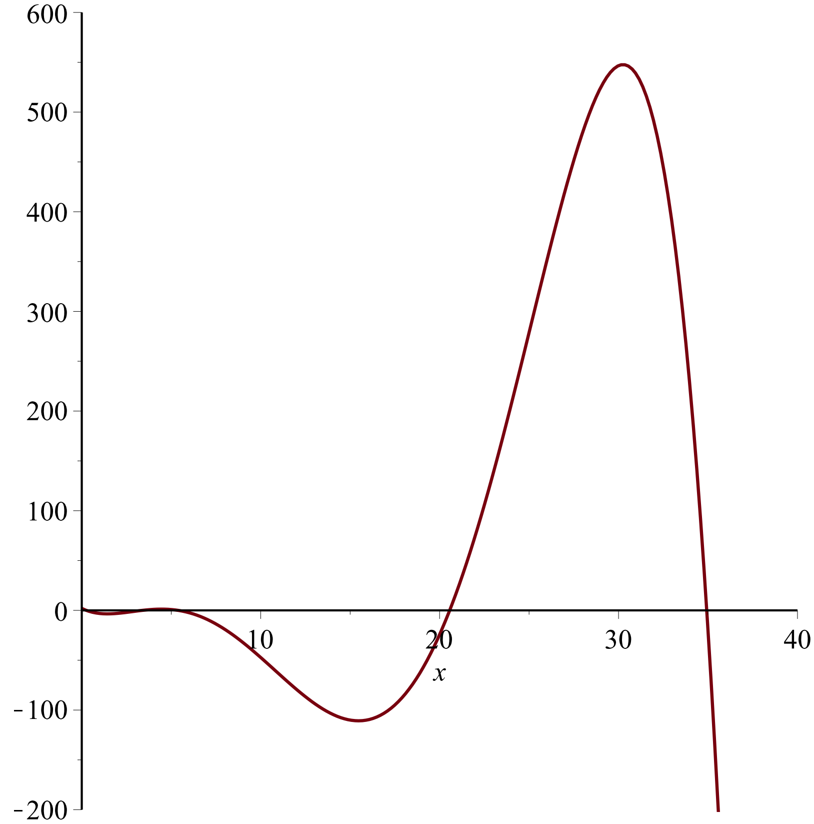
Programming with proc, for and if

Find all local maxima of a polynomial f

$f := x \rightarrow -\frac{683161}{1133371470}x^5+\frac{11752043}{302232392}x^4-\frac{112862553}{151116196}x^3+\frac{198166575}{43176056}x^2-\frac{416037877}{46260060}x+2;$

$$x \rightarrow -\frac{683161}{1133371470}x^5+\frac{11752043}{302232392}x^4-\frac{112862553}{151116196}x^3+\frac{198166575}{43176056}x^2-\frac{416037877}{46260060}x+2 \tag{27}$$

$plot(f(x), view=[0..40,-200..600], x=0..40);$



$fsolve(f'(x) = 0, x);$

$$1.431724935, 4.453992057, 15.46691845, 30.25471480 \quad \textbf{(28)}$$

$evalf(eval(diff(diff(f(x), x), x), x = 1.431724935));$

$$3.684775852 \quad \textbf{(29)}$$

$evalf(f''(4.453992057)); evalf(f''(15.46691845)); evalf(f''(30.25471480));$

$$-2.588142884$$

$$6.888828816$$

$$-33.14325482 \quad \textbf{(30)}$$

```
maxima := proc(f)
    local c, z, el;
    c := 0;
    z := [fsolve(f'(x) = 0, x)];
    for el in z do
        if f''(el) < 0 then c := c + 1;
        end if;
    end do;
    return c;
```

```
 end proc;
proc(f)
    local c, z, el;
    c := 0;
    z := [fsolve(diff(f(x), x) = 0, x)];
    for el in z do if eval(diff(f(x), x, x), x = el) < 0 then c := c + 1 end if end do;
    return c
end proc                                                                    (31)
maxima(f);
                                        2                                   (32)
```

analyze procedure, list, set, sequence

```
z := [fsolve(f'(x) = 0, x)];
              [1.431724935, 4.453992057, 15.46691845, 30.25471480]          (33)
```

```
for el in z do
  print(el)
end do;
                                  1.431724935
                                  4.453992057
                                  15.46691845
                                  30.25471480                               (34)
```

sequences

```
s := 3, 5, 7;
t := 2, 4, 6;
                                    2, 4, 6                                 (35)
t2 := s, t;
                                 3, 5, 7, 2, 4, 6                           (36)
```

lists

```
l1 := [3, 5, 7];
                                   [3, 5, 7]                                (37)
l2 := [2, 4, 6];
                                   [2, 4, 6]                                (38)
l3 := [l2, l1];
                              [[2, 4, 6], [3, 5, 7]]                        (39)
```

sets

```
s1 := {1, 2, 3, 4}; s2 := {3, 4, 5};
                                  {1, 2, 3, 4}
```

$$\{3, 4, 5\} \quad (40)$$

s3 := *s1* **union** *s2*;

$$\{1, 2, 3, 4, 5\} \quad (41)$$

Syntactical description of control structures:

Flow Control (if, for, while, ...)

if <conditional expression> **then** <statement sequence>
| **elif** <conditional expression> **then** <statement sequence> |
| **else** <statement sequence> |
end if

(Note: Phrases located between | | are optional.)

The **for ...while ... do** loop
| **for** <name> | | **from** <expr> | | **by** <expr> | | **to** <expr> | | **while** <expr> |
do <statement sequence> **end do**;

OR

| **for** <name> | | **in** <expr> | | **while** <expr> |
do <statement sequence> **end do**;

(**Note:** Clauses shown between | | above are optional, and can appear in any order, except that the for clause, if used, must appear first.)

```
> restart; for i from 2 to 4 do print(i); end do;
```

2
3
4 (42)

1) Print even numbers from 6 to 10.

```
> for i from 6 by 2 to 10 do print(i) end do;
```

6
8
10 (2.1)

2) Find the sum of all two-digit odd numbers from 11 to 99.

```
> mysum := 0;
  for i from 11 by 2 while i < 100 do
```

```
    mysum := mysum + i
  end do:
  mysum;
```

$$mysum := 0$$

$$2475 \quad \textbf{(2.2)}$$

3) Multiply the entries of an expression sequence.

```
> restart;
  total := 1 :
 for z in 1, x, y, q^2, 3  do
    total := total·z
  end do:
  total;
  x := 2 :
  q := 3 :
 total;
```

$$3\,x\,y\,q^2$$

$$54\,y \quad \textbf{(2.3)}$$

3) Add together the contents of a list.

```
>
```

```
> restart;
 y := 3;
  myconstruction := "";
 for z in [1, "+", y, "·", "q^2", "·", 3] do
    myconstruction := cat(myconstruction, z)
 end do;
 myconstruction;
```

y := 3

myconstruction := ""

myconstruction := "1"

myconstruction := "1+"

myconstruction := "1+3"

myconstruction := "1+3*"

myconstruction := "1+3*q^2"

myconstruction := "1+3*q^2*"

myconstruction := "1+3*q^2*3"

"1+3*q^2*3" **(2.4)**

```
> ?parse
```

```
> q := 4;
```

$$q := 4 \quad \textbf{(2.5)}$$

```
> qq := parse(myconstruction);
```

$$qq := 1 + 9\,q^2 \quad \textbf{(2.6)}$$

```
> qq;
```

$$145 \quad \textbf{(2.7)}$$

```
>
```

Procedures

Flow control constructions, simple commands and comparison operators can be bound together; in a so called
procedure. The simplest possible procedure looks as follow.

proc(parameter sequence)
   statements;
end proc: