SOME USEFUL MAPLE COMMANDS

diff: Compute symbolic derivatives

> ex1 := exp(sin(4*x)) + ln(x^2 + sqrt(x));

$$ex1 := e^{\sin(4\,x)} + \ln\left(x^2 + \sqrt{x}\right)$$

> diff(ex1, x);

$$4\cos(4\,x)\,e^{\sin(4\,x)} + \frac{2\,x + \frac{1}{2}\frac{1}{\sqrt{x}}}{x^2 + \sqrt{x}}$$

> diff(ex1, x$2);

$$-16\sin(4\,x)\,e^{\sin(4\,x)} + 16\cos(4\,x)^2\,e^{\sin(4\,x)} + \frac{2 - \frac{1}{4}\frac{1}{x^{3/2}}}{x^2 + \sqrt{x}}$$

$$-\frac{\left(2\,x + \frac{1}{2}\frac{1}{\sqrt{x}}\right)^2}{\left(x^2 + \sqrt{x}\right)^2}$$

> diff(diff(cos(x/y) , x) , y);

$$\frac{\cos\left(\frac{x}{y}\right)x}{y^3} + \frac{\sin\left(\frac{x}{y}\right)}{y^2}$$

int: Compute symbolic integrals (anti-derivatives and definite integrals)

```
> ex2 := 1 / sqrt( x^2 - a^2 );
```

$$ex2 := \frac{1}{\sqrt{x^2 - 4}}$$

```
> int( ex2, x );
```

$$\ln\left(x + \sqrt{x^2 - 4}\right)$$

A definite integral

```
> int( ex2, x=1..2 );
```

$$\ln(2) - \ln\left(1 + I\sqrt{3}\right)$$

```
> int( ex1, x );
```

$$\int e^{\sin(4\,x)} + \ln\left(x^2 + \sqrt{x}\right)\,dx$$

```
> ex3 := (x^3+ y^3) / (x^2- y^2);
```

$$ex3 := \frac{x^3 + y^3}{x^2 - y^2}$$

```
> int(int(ex3,x),y);
```

$$\frac{1}{6} x^2 y - x^2 \ln(x - y)\, y + \frac{11}{18} x^3 + x\, y^2 \ln(x - y) - \frac{1}{6} x\, y^2 - \frac{1}{3} (x - y)^3 \ln(x - y) - \frac{1}{9} y^3$$

---

series AND taylor: Compute power series expansions

> series(exp(x),x=0);

$$1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5+\mathrm{O}(x^6)$$

> series(exp(x),x=0,11);

$$1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5+\frac{1}{720}x^6+\frac{1}{5040}x^7+\frac{1}{40320}x^8+\frac{1}{362880}x^9+\frac{1}{3628800}x^{10}+\mathrm{O}(x^{11})$$

> taylor(exp(x),x=0);

$$1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5+\mathrm{O}(x^6)$$

> taylor((1 + x)^(-1),x=0,7);

$$1-x+x^2-x^3+x^4-x^5+x^6+\mathrm{O}(x^7)$$

> convert(",'polynom');

$$1-x+x^2-x^3+x^4-x^5+x^6$$

> coeffs(",'x');

1, -1, 1, -1, 1, -1, 1

>

solve: Solve equations, including linear systems

```
>  eq1 := x + y + z = 6;  eq2 := 2*x + y + z = 2;  eq3 := x + y + 3 *z = 3;
```

$$eq1 := x + y + z = 6$$

$$eq2 := 2\,x + y + z = 2$$

$$eq3 := x + y + 3\,z = 3$$

```
> solve( {eq1,eq2,eq3} );
```

$$\left\{ y = \frac{23}{2},\, z = \frac{-3}{2},\, x = -4 \right\}$$

```
> solve( {eq1,eq2,eq3} , {x,y,z} );
```

$$\left\{ y = \frac{23}{2},\, z = \frac{-3}{2},\, x = -4 \right\}$$

```
> eq1 := c[1] + c[2] + c[3] = 6:
> eq2 := 2*c[1] + c[2] + c[3] = 2:
> eq3 := c[1] + c[2] + 3*c[3] = 3:
```

```
> solve( {eq1,eq2,eq3} , {c[1], c[2], c[3] });
```

$$\left\{ c_1 = -4,\, c_2 = \frac{23}{2},\, c_3 = \frac{-3}{2} \right\}$$

MAPLE FEATURES DESCRIBED IN THE "Maple V Programming Guide" , Chapter 1 NOT AVAILABLE IN VERSION V.3 (OUR VERSION)

(1) AUTOMATIC TYPE-CHECKING OF PROCEDURE ARGUMENTS

The ’::<type>’ construct is not recognized in Maple V.3 :

> foo := proc(n::numeric)

```
syntax error:
foo := proc(n::numeric)
              ^
```

But the ’: <type>’ construct IS recognized :

> foo := proc(n:numeric)
>   n^2
> end;

```
foo := proc(n:numeric) n^2 end
```

> foo(2);

$$4$$

> foo(‘bar‘);

```
Error, foo expects its 1st argument, n, to be of type numeric,
but received bar
```

---

(2) add COMMAND (see page 13)

> add(n, n = 1..100);

$$\mathrm{add}(n, n = 1 .. 100)$$

---

(3) SIMULTANEOUS (MULTIPLE) PLOTS USING plot COMMAND and '[..]' ('plotdiff' example on page 10 and following)

> plot([sin(x),cos(x)] , x = -4..4);
```
Error, (in plot) invalid argument, [sin(x), cos(x)]
```

USE '{ ..}' INSTEAD

> plot({sin(x),cos(x)} , x = -4..4);

COMMON MISTAKES; THINGS TO WATCH OUT FOR

(0) Not terminating statement with ’;’ or ’:’

> a := int(x^2 , x = 0..1)

> ;

$$a := \frac{1}{3}$$

---

(1) Using ’=’ rather than ’:=’ for assigment

> a = 2;

$$\frac{1}{3} = 2$$

> a;

$$\frac{1}{3}$$

> a := 2;

$$a := 2$$

> a;

$$2$$

---

(2) Using quotes (’) rather than backquotes (‘) to delimit strings

> string1 := ‘This is a Maple string‘;

string1 := This is a Maple string

> string2 := ’This is not a Maple string’;

```
syntax error:
string2 := ’This is not a Maple string’;
                  ^
```

---

(3) Unique syntax for programming constructs

> for i from 1 to 2 do
>   print(i)
> od;

1

2

> for i = 1 to 2 do

```
syntax error:
for i = 1 to 2 do
      ^
```

> if a = 2 then print(‘yes‘) else print(‘no‘) fi;

yes

> if a = 2 then print(‘yes‘) else print(‘no‘) endif;

```
syntax error:
if a = 2 then print(‘yes‘) else print(‘no‘) endif;
                                                ^
```