Integer, Rational,
Floating Point,
Arbitrary Precision

REDUCE 3.2

REDUCE EXAMPLES

The following REDUCE examples are typeset with comments in the Helvetical Bold font, user typed input in the Times Roman font and algebra system responses in the Courier font and slightly indented. User typed input may be in upper or lower case.

Integer arithmetic is exact

```
A := (for i := 1:50 product i);
```

```
   A := 30414093201713378043612608166064768844377641568960512000000000000
```

and so is rational arithmetic.

```
A/2^60;
```

```
   216105129892080882169214875191192738017616943359375/8192
```

Floating point numbers can also be as accurate as you wish.

```
on bigfloat, numval;
precision 25;
```

```
   25
```

This looks like an integer ...

```
e**(pi*sqrt(163));
```

```
   26253 74126 40768 744.0
```

... but higher precision shows the truth.

```
precision 35;
```

```
   35
```

```
e**(pi*sqrt(163));
```

```
   26253 74126 40768 743.9 99999 99999 92500 7
```

```
off bigfloat;
```

Polynomials can be factored.

```
on factor;
(x^20-1);
```

$$(X^8 - X^6 + X^4 - X^2 + 1)*(X^4 + X^3 + X^2 + X + 1)*(X^4 - X^3 + X^2 - X + 1)*$$

$$(X^2 + 1)*(X + 1)*(X - 1)$$

```
off factor;
```

Differential and Integral calculus examples.

```
int(1/(x^8-1),x);
```

$$(SQRT(2)*LOG(-SQRT(2)*X + X^2 + 1) - SQRT(2)*LOG(SQRT(2)*X + X^2 + 1) - 2*SQRT(2)*ATAN((-SQRT(2) + 2*X)/SQRT(2)) - 2*SQRT(2)*ATAN((SQRT(2) + 2*X)/SQRT(2)) + 2*LOG(X - 1) - 2*LOG(X + 1) - 4*ATAN(X))/16$$

The answer is available in the REDUCE work space variable called ws.

```
df(ws,x);
```

$$1/(X^8 - 1)$$

Other examples ... ...

```
df( (x*(cos(log(x)) + sin(log(x))))/2, x);
```

$$cos(log(x))$$

```
int(cos(log(x)),x);
```

$$(X*(COS(LOG(X)) + SIN(LOG(X))))/2$$

```
int(1/cos(x),x);
```

$$-LOG(TAN(X/2) - 1) + LOG(TAN(X/2) + 1)$$

```
int((a+b*x)^n,x);
```

$$((A+B*X)^N*(A+B*X))/(B*(N+1))$$

```
df( (a+b*x)^n * (a+b*x) / b*(n+1), x);
```

$$(A+B*X)^N$$

... again using the variable ws.

```
df(exp exp exp exp x,x);
```

$$E^{(E^{(E^{(E^X)})} + E^{(E^X)} + E^X + X)}$$

```
int(ws,x);
```

$$E^{(E^{(E^{(E^X)})})}$$

Inversion of Hilbert matrices.

```
mata := mat( (1/(x-1), 1/(x-2), 1/(x-3)),
             (1/(x-2), 1/(x-3), 1/(x-4)),
             (1/(x-3), 1/(x-4), 1/(x-5)) )$
```

```
1/mata;
```

$$MAT(1,1) := (X^5 - 11*X^4 + 47*X^3 - 97*X^2 + 96*X - 36)/4$$

$$MAT(1,2) := (- X^5 + 13*X^4 - 65*X^3 + 155*X^2 - 174*X + 72)/2$$

$$MAT(1,3) := (X^5 - 15*X^4 + 85*X^3 - 225*X^2 + 274*X - 120)/4$$

$$MAT(2,1) := (- X^5 + 13*X^4 - 65*X^3 + 155*X^2 - 174*X + 72)/2$$

$$MAT(2,2) := X^5 - 15*X^4 + 88*X^3 - 252*X^2 + 352*X - 192$$

$$MAT(2,3) := (- X^5 + 17*X^4 - 113*X^3 + 367*X^2 - 582*X + 360)/2$$

$$MAT(3,1) := (X^5 - 15*X^4 + 85*X^3 - 225*X^2 + 274*X - 120)/4$$

$$MAT(3,2) := (- X^5 + 17*X^4 - 113*X^3 + 367*X^2 - 582*X + 360)/2$$

$$MAT(3,3) := (X^5 - 19*X^4 + 143*X^3 - 533*X^2 + 984*X - 720)/4$$

1/mata might be useful in a FORTRAN program.

```
on fort;

ws;

      MAT(1,1)=(X**5-11.*X**4+47.*X**3-97.*X**2+96.*X-36.)/
     . 4.
      MAT(1,2)=(-X**5+13.*X**4-65.*X**3+155.*X**2-174.*X+
     . 72.)/2.
      MAT(1,3)=(X**5-15.*X**4+85.*X**3-225.*X**2+274.*X-
     . 120.)/4.
      MAT(2,1)=(-X**5+13.*X**4-65.*X**3+155.*X**2-174.*X+
     . 72.)/2.
      MAT(2,2)=X**5-15.*X**4+88.*X**3-252.*X**2+352.*X-192.
      MAT(2,3)=(-X**5+17.*X**4-113.*X**3+367.*X**2-582.*X+
     . 360.)/2.
      MAT(3,1)=(X**5-15.*X**4+85.*X**3-225.*X**2+274.*X-
     . 120.)/4.
      MAT(3,2)=(-X**5+17.*X**4-113.*X**3+367.*X**2-582.*X+
     . 360.)/2.
      MAT(3,3)=(X**5-19.*X**4+143.*X**3-533.*X**2+984.*X-
     . 720.)/4.

off fort;
```

REDUCE knows about sine, cosine, exponentials and logarithms, and many other functions. If that is not enough users can define their own functions.

```
operator myfn;
myfn(x)+y;
```

```
   MYFN(X) + Y
```

You can give rules for differentiation for example

```
for all x let df(myfn(x),x) = exp(x^2);
```

and now use it to simplify a second derivative,

```
df(myfn(x)^3,x,2);
```

$$6*E^{(X^2)}*MYFN(X)*(E^{(X^2)} + MYFN(X)*X)$$

and some integrals.

```
int(x*myfn(x),x);
```

$$(- E^{(X^2)}*X + 2*MYFN(X)*X^2 + MYFN(X))/4$$

REDUCE can handle matrices as well.

```
matrix rx(3,3), ry(3,3), mata(3,3);
```

Product of rotation matrices

```
rx := mat( ( 1,         0,         0),
           ( 0,    cos(th), sin(th)),
           ( 0,   -sin(th), cos(th)) )$

ry := mat( ( cos(phi),   0, -sin(phi)),
           (        0,   1,         0),
           ( sin(phi),   0,  cos(phi)) )$

rx*ry;
```

```
   MAT(1,1) := COS(PHI)
   MAT(1,2) := 0
   MAT(1,3) :=  - SIN(PHI)
   MAT(2,1) := SIN(TH)*SIN(PHI)
   MAT(2,2) := COS(TH)
   MAT(2,3) := COS(PHI)*SIN(TH)
   MAT(3,1) := COS(TH)*SIN(PHI)
   MAT(3,2) :=  - SIN(TH)
   MAT(3,3) := COS(TH)*COS(PHI)
```