AN INTRODUCTION TO MAPLE:

SAMPLE INTERACTIVE SESSION

Bruce W. Char
Keith O. Geddes
Gaston H. Gonnet

University of Waterloo

Research Report CS-84-04
January, 1984

An Introduction to Maple: Sample Interactive Session

Bruce W. Char
Keith O. Geddes
Gaston H. Gonnet

Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

Brief Facts about Maple

What is Maple?

Maple is a system for symbolic mathematical computation which has been under development at the University of Waterloo since December, 1980.

Implementation

The basic system is written in a BCPL-derivative language called Margay, which is then macro-processed into other languages of the BCPL family. Maple has been brought up in versions of C and in B, and there are plans to implement it in other languages such as WSL (Waterloo Systems Language) and Waterloo Port.

Machines

Maple is currently available on VAX computers, Honeywell 6000 series computers, and on MC68000-based microcomputers which support Unix-like operating systems.

Size

The basic Maple system occupies about 100K bytes on a VAX 11/780. Library functions are automatically loaded as required, so the data space grows at a rate that depends on the user's application.

References

B.W. Char, K.O. Geddes, G.H. Gonnet, and S.M. Watt, *Maple User's Manual, third edition,* Univ. of Waterloo Res. Rep. CS-83-41, December 1983.

B.W. Char, K.O. Geddes, W.M. Gentleman, and G.H. Gonnet, *The Design of Maple: A compact, portable, and powerful computer algebra system,* Univ. of Waterloo Res. Rep. CS-83-06, April 1983.

Sample Interactive Session

In the interactive Maple session demonstrated on the following pages, lines beginning at the left margin are user input lines and centered lines are system responses.

The Maple system is initiated by a command such as

/u/maple/bin/maple	on many Unix systems
maple/maple	on the Honeywell GCOS system.

When Maple is initiated, it displays a maple leaf and the version number of the Maple system that has been loaded.

```
    |\^/|
._|\|   |/|_.
\  MAPLE  /   Version 3.1 --- October 1983
<____ ____>
     |
```

One form of arithmetic in Maple is exact rational arithmetic.
'**' is the exponentiation operator.

1 + 1/4 + 1/16 + 1/64 + 1/256;

$$\frac{341}{256}$$

d := (3**50 + 5**20) / 2**80;

$$d := \frac{358948993893610010205437}{604462909807314587353088}$$

Constants in Maple may be approximated by floating-point numbers and the user has control
over the number of digits carried. The 'evalf' function causes 'evaluation to a real'.

evalf(d);

.59383129728

Digits := 50;

Digits := 50

evalf(d);

.593831297288418637242948488166265736624277460364851

h := tan(3*Pi/10);

h := tan(3/10 *Pi*)

evalf(h);

1.3763819204711735382072095819108876795258993360082

Restore the global variable 'Digits' to its default value.

Digits := 10;

Digits := 10

Maple supports rational expressions both unexpanded and expanded. The double-quote symbol
refers to the latest expression.

```
(x+1)**7;
```

$$(x+1)^7$$

```
expand(");
```

$$x^7+7x^6+21x^5+35x^4+35x^3+21x^2+7x+1$$

```
(y-x)*(y**4+y**3*x+y**2*x**2+y*x**3+x**4);
```

$$(y-x)(y^4+y^3x+y^2x^2+yx^3+x^4)$$

```
expand(");
```

$$y^5-x^5$$

```
(x*y/2 - (y**2)/3) * (x-y)**2;
```

$$(1/2\ xy\ -\ 1/3\ y^2)(x-y)^2$$

```
expand(");
```

$$1/2\ x^3y\ -\ 4/3\ x^2y^2\ +\ 7/6\ xy^3\ -\ 1/3\ y^4$$

Maple does not force rational expressions into a canonical form.

```
" * (3*x+y) / (x-y);
```

$$\frac{(1/2\ x^3y\ -\ 4/3\ x^2y^2\ +\ 7/6\ xy^3\ -\ 1/3\ y^4)(3x+y)}{x-y}$$

However, normalizing facilities are available for simplification.

```
normal(");
```

$$1/6\ (2y-3x)y(3x+y)(y-x)$$

```
" / (x**3 - x**2*y - x*y + y**2);
```

$$1/6\ \frac{(2y-3x)y(3x+y)(y-x)}{x^3-yx^2-xy+y^2}$$

```
expr := normal(");
```

$$expr\ :=\ 1/6\ \frac{(2y-3x)y(3x+y)}{y-x^2}$$

```
# There are explicit functions for gcd (greatest common divisor) and lcm (least common multiple)
# computations with polynomials.

p := 143*x**3*y-39*x**2*y**2-11*x*y+11*x+3*y**2-3*y;
```

$$p := 143x^3y-39x^2y^2-11xy+11x+3y^2-3y$$

```
q := 55*x*y**3+11*x*y**2+11*x*y+11*x-15*y**4-3*y**3-3*y**2-3*y;
```

$$q := 55xy^3+11y^2x+11xy+11x-15y^4-3y^3-3y^2-3y$$

```
gcd(p, q);
```

$$-11x+3y$$

```
lcm(15*(x-5)*y, 9*(x**2-10*x+25));
```

$$45yx^2-450xy+1125y$$

```
# Maple has facilities for differentiation of expressions.

f := sin(x) * cos(x);
```

$$f := \sin(x)\cos(x)$$

```
fp := diff(f, x);
```

$$fp := \cos(x)^2 - \sin(x)^2$$

```
diff(f, x, x);
```

$$-4\sin(x)\cos(x)$$

```
diff(sin(x) * x**(x**x), x)  -  cos(x) * x**(x**x);
```

$$\sin(x)\, x^{(x^x)}\, (x^x\, (\ln(x) + 1)\ln(x) + \frac{x^x}{x})$$

```
# The "subs" command is used below to substitute 1 for x in an expression.

subs(x=1, fp);
```

$$\cos(1)^2 - \sin(1)^2$$

```
evalf(");
```

$$-.4161468366$$

The "subs" command also can do more general substitutions.

```
subs(sin(x)**2 = 1 - cos(x)**2, fp);
```

$$2\cos(x)^2 - 1$$

```
expand( (sin(x) + 1)**2 );
```

$$\sin(x)^2 + 2\sin(x) + 1$$

```
subs(sin(x)**2 = 1 - cos(x)**2, ");
```

$$2 - \cos(x)^2 + 2\sin(x)$$

In Maple, equations may be manipulated as expressions.

```
eqn1 := 3*x + 5*y = 13;
```

$$eqn1 := 3x + 5y = 13$$

```
eqn2 := 4*x - 7*y = 30;
```

$$eqn2 := 4x - 7y = 30$$

```
3*eqn2 - 4*eqn1;
```

$$-41y = 38$$

```
y := 38/(-41);
```

$$y := -\frac{38}{41}$$

```
eqn1;
```

$$3x - \frac{190}{41} = 13$$

```
x := (13 + 190/41)/3;
```

$$x := \frac{241}{41}$$

```
eqn1; eqn2;
```

$$13 = 13$$

$$30 = 30$$

Unassign the names x and y so that their values will be their own names.

```
x := 'x'; y := 'y';
```

$$x := x$$

$$y := y$$

```
# Maple has a solve function which can solve many kinds of equations, including systems of linear
# equations, single equations involving elementary transcendental functions, and polynomial equations.

solve( {eqn1, eqn2}, {x, y} );
```

$$\{y = -\frac{38}{41}, x = \frac{241}{41}\}$$

```
# Note that the use of a colon instead of a semicolon suppresses maple's output in the following.

eqn1 := 3*r + 4*s - 2*t + u = -2:

eqn2 := r - s + 2*t + 2*u = 7:

eqn3 := 4*r - 3*s + 4*t - 3*u = 2:

eqn4 := -r + s + 6*t - u = 1:

solve( {eqn.(1..4)}, {r, s, t, u} );
```

$$\{t = 3/4, u = 2, r = 1/2, s = -1\}$$

```
solve( cos(x) + y = 9, x );
```

$$\arccos(-y+9)$$

```
solve( 2**u + G = 0, u );
```

$$\frac{\ln(-G)}{\ln(2)}$$

```
solve( x**2 - 46*x + 529 = 0, x );
```

$$23, 23$$

```
solve( 1/2*a*x**2 + b*x + c = 0, x );
```

$$\frac{-b+(b^2-2ac)^{1/2}}{a}, \frac{-b-(b^2-2ac)^{1/2}}{a}$$

```
{x+y+z = a, x+2*y-a*z = 0, sin(a)*z+a*y = 0}:

solve( " , {x, y, z} );
```

$$\{z = \frac{-a^2}{-\sin(a) - a^2 - a}, y = \frac{-(-\sin(a)-a^2-a)a + (-a-2)a^2 + a^2}{-\sin(a) - a^2 - a}, x = \frac{2(-\sin(a)-a^2-a)a - (-a-2)a^2}{-\sin(a) - a^2 - a}\}$$

```
# Maple has arrays and also a general "table" facility.

S := array(1..10);
```

$$S := \text{array}(1..10, [\,])$$

```
S[9] := x**9;
```

$$S[9] := x^9$$

```
T := array(0..3, 0..3, symmetric);
```

$$T := \text{array}(\text{symmetric}, 0..3, 0..3, [\,])$$

```
T[1,2] := G12;
```

$$T[1, 2] := G12$$

```
T[1,2] + T[2,1];
```

$$2\,G12$$

```
R[sin(x)] := cos(x):   R[cos(x)] := - sin(x):

sin(x) * R[sin(x)] + cos(x) * R[cos(x)];
```

$$0$$

Maple has facilities for computing series expansions of expressions. The series representation
includes an O( ) term to indicate the order of truncation of the series. The order of truncation
can be controlled by the user; the default value is order 6.

```
expr;
```

$$1/6\ \frac{(2y-3x)y(3x+y)}{y-x^2}$$

```
taylor(expr, x=0);
```

$$1/3\ y^2+1/2\ yx+1/6\ \frac{-9y+2y^2}{y}x^2+1/2\ x^3+1/6\ \frac{-9y+2y^2}{y^2}x^4+1/2\frac{1}{y}x^5+O(x^6)$$

```
r := (x**2 + 6*x - 1) / (2*x**2 + 1)**2;
```

$$r := \frac{x^2+6x-1}{(2x^2+1)^2}$$

```
s1 := taylor(r, x=0);
```

$$s1 := -1+6x+5x^2-24x^3-16x^4+72x^5+O(x^6)$$

```
s2 := taylor(r, x=1, 2);
```

$$s2 := 2/3 - 8/9\ (x-1) + \frac{11}{27}(x-1)^2 + O((x-1)^3)$$

```
s3 := taylor( exp(3*x**2 + x), x=0, 4 );
```

$$s3 := 1+x+7/2\ x^2 +19/6\ x^3 +\frac{145}{24}x^4 +O(x^5)$$

```
taylor(s1*s3, x=0);
```

$$-1+5x+15/2\ x^2 - 7/6\ x^3 - \frac{229}{24}x^4 + O(x^5)$$

Maple knows how to compute with asymptotic series as well as Taylor series.

```
f := n*(n+1) / (2*n-3);
```

$$f := \frac{n(n+1)}{2n-3}$$

```
asympt(f, n);
```

$$1/2\ n+5/4 +15/8\ n^{-1}+\frac{45}{16}n^{-2}+\frac{135}{32}n^{-3}+\frac{405}{64}n^{-4}+\frac{1215}{128}n^{-5}+O(n^{-6})$$

 # There is a limit function to compute the limiting value of an expression as a specified variable
 # approaches a specified value.

```
limit( (tan(x)-x)/x**3, x=0 );
```

1/3

```
r := (x**2 - 1) / (11*x**2 - 2*x - 9);
```

$$r := \frac{x^2-1}{11x^2-2x-9}$$

```
limit(r, x=0);
```

1/9

```
limit(r, x=infinity);
```

1/11

```
limit(r, x=1);
```

1/10

 # Maple integers can be arbitrarily long. The following is an example using the factorial operator.

```
720!;
```

```
26012189435657951002049032270810436111915218750169457857275418378508356311569473
82240678577958130457082619920575892247259536641565162052015873791984587740832529
10524469038881188412376434119195104550534665861624327194019711390984553672727853
70993456298555867193697740700037004307837589974206767840169672078462806292290321
07161669867260548988445514257193985499448939594496064045132362140265986193073249
36977047760606768067017649166940303481996188145562519559256691883082551494294759
65372748456246288242345265977897377408964665539924359287862125159674832209760295
05696699927284670563747137533019248313587076125412683415860129447566011455420749
58995256354306828863463108496565068277155299625679084523570255218622235813001670
08345234432368219357931847019565107297818043541738905607274280485839959197290217
26612291298420516067579036232337699453964191475175567557695392233803056825308599
97744167578435281591346134039460490126954202883834710136373382448450666009334848
44407119312925376946573543373757247722301815340326471775319845373414786743270484
57983786618703257405938924215709695994630557521063203263493209220738320923356309
92326750440170176057202601082928804233560664308988871029738079757801305604957634
28386830571906622052911748225105366977566030295740433879834715185526028053338663
57139101046336419769097397432285994219837046979109956303389604675889865795711176
56667003915674815311594398004362539939973120306649060132531130471902889849185620
37666691644687911252491937544258458950003115616829743046411425380748972817233759
55380661719801404677935614793635266265683339509760000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
```

Maple can represent and manipulate sets. The set operators are + (union), * (intersection), and
− (set difference).

a := {x, 2*y+1/3};

$$a := \{x, 2y+1/3\}$$

b := {z−4, x};

$$b := \{x, z-4\}$$

c := a + b;

$$c := \{x, z-4, 2y+1/3\}$$

d := a − b;

$$d := \{2y+1/3\}$$

e := a * b;

$$e := \{x\}$$

f := b * d;

$$f := \{\,\}$$

Selection of elements from a set (and more generally, selection of operands from any expression)
is accomplished using the "op" function. op(i,expr) yields the i-th operand. Also, nops(expr)
yields the "number of operands" in expr.

nops(c);

$$3$$

g := op(2,c);

$$g := z-4$$

h := {op(3,c),z};

$$h := \{z, 2y+1/3\}$$

```
# Another data structure in Maple is the list, represented using square brackets.  Selection of
# elements from a list is accomplished using the "op" function.  For composition of lists, it is
# convenient to use the form op(expr) which yields a sequence of all the operands separated by
# commas.  Also, op(i..j, expr) yields the sequence  op(i,expr), op(i+1,expr), . . ., op(j,expr) .

list1 := [x, 2*y+1/3];
```

$$list1 := [x,\ 2y+1/3]$$

```
list2 := [z-4, x];
```

$$list2 := [z-4,\ x]$$

```
new_list := [op(list1), op(list2)];
```

$$new\_list := [x,\ 2y+1/3,\ z-4,\ x]$$

```
a := [op(1, list1), op(3..4, new_list)];
```

$$a := [x,\ z-4,\ x]$$

```
F( op(list2) );
```

$$F(z-4,\ x)$$

```
# Maple can do indefinite summations as well as definite summations.

sum(i**2, i = 1 .. n-1);
```

$$1/3\ n^3 - 1/2\ n^2 + 1/6\ n$$

```
sum( (5*i-3)*(2*i+9), i = 1 .. 9876543210 );
```

$$3211394431368198288556209751725$$

```
sum( (5*i-3)*(2*i+9), i = 1 .. n-1 );
```

$$10/3\ n^3 + 29/2\ n^2 - 269/6\ n + 27$$

```
sum(i**4 * 7**i, i = 1 .. n-1);
```

$$-\frac{91}{54}\,7^n\,n + \frac{70}{81}\,7^n + 14/9\ 7^n\,n^2 - 7/9\ 7^n\,n^3 + 1/6\ n^4\,7^n - \frac{70}{81}$$

```
sum((i-1)/(i+1), i = 1 .. n);
```

$$n - 2\,Psi(n+2) + 2\,Psi(2)$$

```
n := 100:   evalf(" ");
```

$$91.60544298$$

An important facility in Maple is analytic integration.

```
f := 1/2*x**(-2) + 3/2*x**(-1) + 2 - 5/2*x + 7/2*x**2;
```

$$f := 1/2 \, \frac{1}{x^2} + 3/2 \, \frac{1}{x} + 2 - 5/2 \, x + 7/2 \, x^2$$

```
int(f, x);
```

$$-1/2 \, \frac{1}{x} + 3/2 \ln(x) + 2\,x - 5/4\,x^2 + 7/6\,x^3$$

```
int(f, x = 1..2);
```

$$20/3 + 3/2 \ln(2)$$

```
evalf(");
```

$$7.706387438$$

```
int( (x-7) / (3*x+11)**2 ,  x );
```

$$-1/3 \, \frac{x-7}{3x+11} + 1/9 \ln(3x+11)$$

```
g := (x**2 - 3*x + 2) * sin(x);
```

$$g := (x^2 - 3x + 2) \sin(x)$$

```
int(g, x);
```

$$-(x^2 - 3x + 2) \cos(x) + (2x - 3) \sin(x) + 2 \cos(x)$$

```
expand(");
```

$$-x^2 \cos(x) + 3\,x \cos(x) + 2\,x \sin(x) - 3 \sin(x)$$

```
h := sin(t) * cos(t);
```

$$h := \sin(t) \cos(t)$$

```
int(h, t);
```

$$1/2 \sin(t)^2$$

```
int(exp(x**2), x);
```

$$\mathrm{int}(\exp(x^2), x)$$

```
# An important component of the Maple system is the Maple programming language which may be
# used to write procedures. Following are some examples of procedures written in Maple.

fibonacci := proc (n)
      option remember;
      if nargs<>1 or not type(n,integer) or n<0 then
            ERROR(`wrong number or type of parameters in fibonacci`)
      else
            if n<2 then n else fibonacci(n-1) + fibonacci(n-2) fi
      fi
end;


# The "option remember" statement at the beginning of the above procedure tells the Maple
# system to store each computed result in a table. Then the system will retrieve the result rather
# than re-compute whenever the procedure is called with a previously-computed argument. In this
# example of a recursive procedure, "option remember" will be very advantageous.
#
# This procedure also includes error-checking on the arguments. The value of the special name
# "nargs" is the number of arguments with which the procedure was called, <> is the "not equal"
# operator, and "type" is Maple's type-checking function.

# Some examples invoking procedure fibonacci:

fibonacci(101);

                              573147844013817084101

fibonacci(-1);

ERROR: wrong number or type of parameters in fibonacci
```

```
# Tschebysheff(n) :  Computes the Tschebysheff polynomials of degrees 0 through n into a table.

Tschebysheff := proc (n)
      local p,k;
      p[0] := 1;  p[1] := x;
      for k from 2 to n do
            p[k] := expand( 2*x*p[k-1] - p[k-2] )
      od;
      RETURN(p)
end;

# An example invoking procedure Tschebysheff:

a := Tschebysheff(5):

a[0], a[1], a[2], a[3], a[4], a[5];
```

$$1,\ x,\ 2x^2-1,\ 4x^3-3x,\ 8x^4-8x^2+1,\ 16x^5-20x^3+5x$$

```
x := 1/4:

a[0], a[1], a[2], a[3], a[4], a[5];
```

$$1,\ 1/4,\ -7/8,\ -\frac{11}{16},\ \frac{17}{32},\ \frac{61}{64}$$

```
# member:  Test for membership in a list.

member := proc (element, List)
      local i;
      false;  for i to nops(List) while not " do
                  evalb( element = op(i,List) )  od;
      "
end;

# The above is an example of a Boolean procedure -- the value returned will be true or false.  The
# Maple function "evalb" causes "evaluation as a Boolean", "nops" yields the "number of operands",
# and the double-quote symbol refers to the latest expression.

# Some examples invoking procedure member follow.

member( x*y, [1/2, x*y, x, y] );
```

true

```
member( x, [1/2, x*y] );
```

false

```
member( x, [ ] );
```

false

```
# max:  Compute the maximum of a sequence of numbers.

max := proc ( )
      local i, M, p;
      if nargs = 0 then
            ERROR(`function max called with no parameters`)
      else
            M := args[1];
            for i to nargs do
                  p := args[i];
                  if not type(p, rational) and not type(p, real) then FAIL fi;
                  if M < p then M := p fi
            od;
            M
      fi
end;
```

```
# The above procedure is taken from the Maple system library which defines Maple's "max"
# procedure to compute the maximum of a sequence of numbers.  Unlike the previous procedures,
# no names are specified for the formal parameters.  Rather, the parameters are accessed by the
# special array args.  This procedure may be called with any number of parameters.

# Some examples invoking procedure max follow.

max(3/2, 1.49);
```

3/2

```
max(3/5, evalf(ln(2)), 9/13);
```

.6931471805

```
max(5);
```

5

```
max(-1001, 1/2, -1/2, -9);
```

1/2

```
max(x, y);
```

$\max(x,y)$