```
    |\^/|     Maple 9 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2003
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
>  #astuce, a retenir
>  f:=x^2+1; g:=y->eval(f,x=y);expand(g(x+1));
                                        2
                                  f := x  + 1

                               g := y -> f|
                                          |x = y

                                   2
                                  x  + 2 x + 2

>  ##############################
>  f:=x->(x^2-2);
                                           2
                                f := x -> x  - 2

>  df:=y->eval(diff(f(x),x),x=y);
                                      /d      \|
                           df := y -> |-- f(x)||
                                      \dx     /|x = y

>  Digits:=1;# un seul chiffre significatif:
                                  Digits := 1

>  13*2.;evalf(13*2,1);
                                      20.

                                      30.

>  Digits:=3;13.26*2;
                                  Digits := 3

                                      26.6

>  interface(screenwidth=120);
>  ##############################
>  f:=x->(x^2-2);
                                                          2
                                                f := x -> x  - 2

>  df:=y->eval(diff(f(x),x),x=y);
                                                     /d      \|
                                          df := y -> |-- f(x)||
                                                     \dx     /|x = y

>  Digits:=10000;U:=1.:
                                                Digits := 10000

>  ######################################################
>  # On fait 14 iterations de la methode de Newton, et on
>  # affiche la difference avec racine de 2
>  ######################################################
>  x-f(x)/df(x);#On trouve x/2 +1/x c'est donc Heron.
                                                       2
                                                      x  - 2
                                                  x - ------
                                                       2 x

>  r:=evalf(sqrt(2)):#on ne le calcule qu'une fois.
>  for i from 1 to 14 do
>  U:=U/2+1/U:a:=U-r:print(evalf(a,10));
>  end do:
                                                 0.08578643763

                                                 0.002453104294

                                                              -5
                                               0.2123901415 10

                                                              -11
                                               0.1594861825 10

                                                              -24
                                               0.8992928322 10

                                                              -48
                                               0.2859283843 10

                                                              -97
                                               0.2890477193 10

                                                             -195
                                              0.2953888517 10

                                                             -391
                                              0.3084915038 10

                                                             -783
                                              0.3364661831 10

                                                             -1567
                                              0.4002559988 10

                                                             -3135
                                              0.5664097307 10
```

```
                                                              -6270
                                                0.1134269928 10

                                                         0.

>  ###
>  Digits:=10;
                                                     Digits := 10

#############################
>  #1<sqrt(2)<2. x,y dans [1,2] |f''(y)/2f'(x)|<1/2=C convient.
>  #On verifie qu'\'a chaque \'etape la precision choisie suffit.
>  Digits:=2;U:=1;
                                                     Digits := 2

                                                        U := 1

>  for i from 1 to 18 do
>  U:=U/2+1/U:a:=U-evalf(sqrt(2)):print(evalf(a,10));Digits:=2*Digits-1;
>  end do:
                                                         0.1

                                                         0.01

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

                                                          0.

bytes used=4117180, alloc=2489912, time=0.30
                                                          0.

bytes used=8529864, alloc=3079628, time=0.77
                                                          0.

bytes used=16217140, alloc=5635064, time=1.85
                                                          0.

>  ###
>  Digits:=10;
                                                     Digits := 10

>  #pour on ne calcule pas df a chaque fois.
>  n:=3;d:=8;f:=x->x^n-d;df:=y->eval(diff(f(x),x),x=y);
                                                        n := 3

                                                        d := 8

                                                              n
                                                   f := x -> x  - d

                                                        /d      \|
                                             df := y -> |-- f(x)||
                                                        \dx     /|x = y

>  #p =nbre de tours
>  testconv:=proc(z,p)
>  prec:=20;U:=z;
>  for i from 1 to p do U:=evalf(U-f(U)/df(U),prec) od;
>  k:=round(argument(U)*n/(2*Pi));[k,abs(evalf(U-exp(2*I*Pi*k/n)*root(d,n),p))];
>  end proc;
testconv := proc(z, p)
local prec, U, i, k;
    prec := 20;
    U := z;
    for i to p do U := evalf(U - f(U)/df(U), prec) end do;
    k := round(1/2*argument(U)*n/Pi);
    [k, abs(evalf(U - exp(2*I*Pi*k/n)*root(d, n), p))]
end proc

> quit
bytes used=16224296, alloc=5635064, time=1.85
```