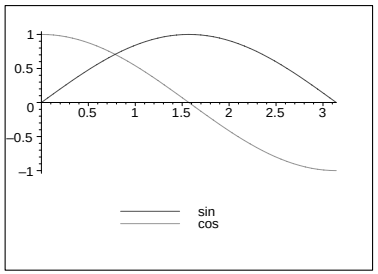
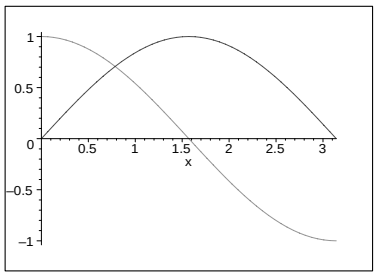
Correction - Maple - Chapitre 3

La commande plot - exemple

```
> plot([sin,cos],0..Pi, legend=["sin","cos"]);
```




```
> plot([sin(x),cos(x)],x=0..Pi);
```



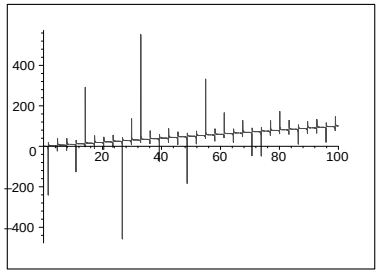
Exercice 11

```
> f :=x->x-tan(x);
```

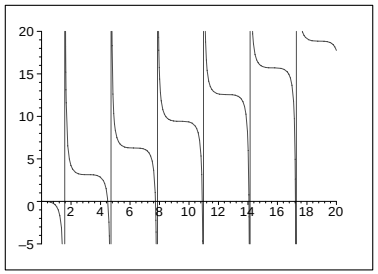
$$f := x \to x - \tan(x)$$

Remarque : f est strictement décroissante sur chaque intervalle maximal. Il existe donc au plus qu'un seul zéro par intervalle maximal.

```
> plot(f,0..100);
```

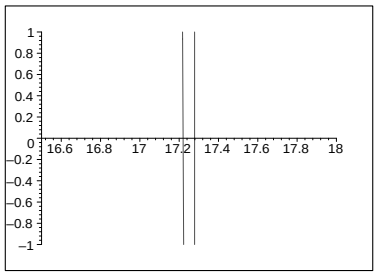


```
> plot(f,0..20,-5..20);
```

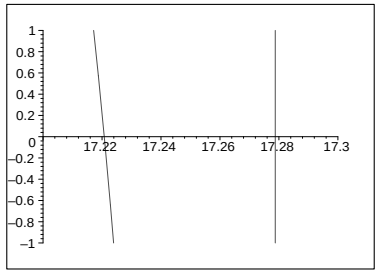


0 est le zéro du premier intervalle maximal. Le cinquième zéro se trouve donc dans le sixième intervalle, entre 16 et 18

```
> plot(f,16.5..18,-1..1);
```



```
> plot(f,17.2..17.3,-1..1);
```

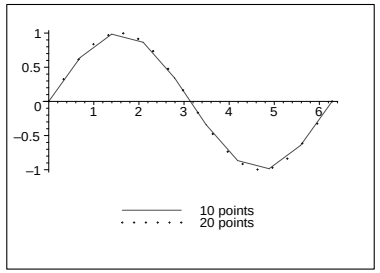


```
> fsolve(f(x),x,17..18);
```

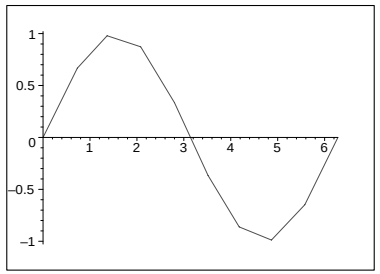
17.22075527

Exercice 12

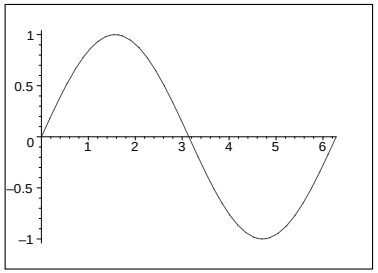
```
> seq10 :=[seq([2*Pi*k/9,sin(2*Pi*k/9)],k=0..9)] : se
> sin10 :=plot(seq10,color=red,legend="10 points") :
> with(plots) : display({sin10,sin20});
```




```
> plot(sin,0..2*Pi,adaptive=false,numpoints=10);
```



```
> plot(sin,0..2*Pi,numpoints=10);
```

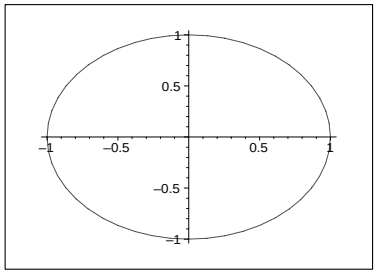


Exercice 13

```
> with(plots) :
```

Warning, the name changecoords has been redefined

1.
```
> plot([cos(s),sin(s),s=0..2*Pi]);
```

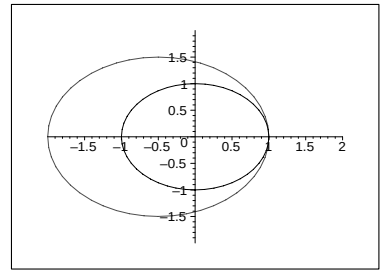


2.
```
> N :=20 :
  R :=-1.5 :
  t :=0 :

> dt :=2*Pi/N :

> C :=[seq(0,k=1..N)] :
  G0 :=plot([cos(s),sin(s),s=0..2*Pi],color=black) :
  for k to N do
  G1 :=plot( [(R+1)*cos(t)+abs(R)*cos(s),
     (R+1)*sin(t)+abs(R)*sin(s),s=0..2*Pi] ) :
  C[k] :=display(G0,G1) :
  t :=t+dt :
  od :

> display(C, insequence = true);
```

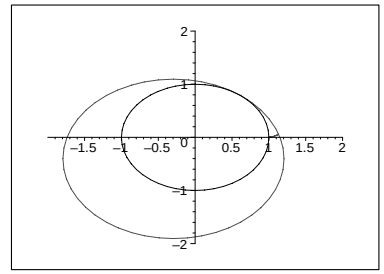


3.
```
> N :=20 :
  R :=-1.5 :
  t :=0 :
  nb_tours :=3 :

> dt :=2*nb_tours*Pi/N :

> C :=[seq(0,k=1..N)] :
  G0 :=plot([cos(s),sin(s),s=0..2*Pi],
       color=black) :
  for k to N do
  t :=t+dt :
  G1 :=plot( [(R+1)*cos(t)+abs(R)*cos(s),
    (R+1)*sin(t)+abs(R)*sin(s),s=0..2*Pi] ) :
  G2 :=plot([ (1+R)*cos(s)-R*cos((1/R+1)*s),
    (1+R)*sin(s)-R*sin((1/R+1)*s), s=0..t]) :
  C[k] :=display(G0,G1,G2) :
  od :

> display(C, insequence = true);
```

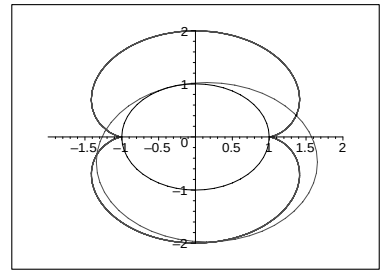


4.
```
> N :=20 :
  R :=-1.5 :
  t :=0 :
  nb_tours :=3 :

> dt :=2*nb_tours*Pi/N :

> C :=[seq(0,k=1..N)] :
  G0 :=plot([cos(s),sin(s),s=0..2*Pi],color=black) :
  for k to N do
  t :=t+dt :
  G1 :=plot( [(R+1)*cos(t)+abs(R)*cos(s),
    (R+1)*sin(t)+abs(R)*sin(s),s=0..2*Pi] ) :
  G2 :=plot([ (1+R)*cos(s)-R*cos((1/R+1)*s),
    (1+R)*sin(s)-R*sin((1/R+1)*s), s=0..t]) :
  G3 :=plot([[(1+R)*cos(t)-R*cos((1/R+1)*t),
  (1+R)*sin(t)-R*sin((1/R+1)*t) ]],
    thickness=6,style=point,color=black) :
  C[k] :=display(G0,G1,G2,G3) :
  od :

> display(C, insequence = true);
```



Exercice 14

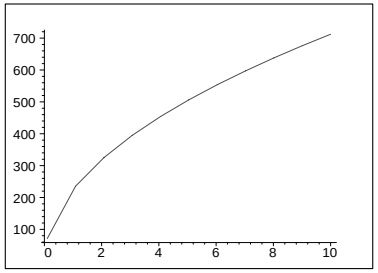
1.
```
> dicho :=proc(A,B,eps,Nmax,fonction)
  local fa,fb,err,i,c,a,b,tmp,f;
  a :=A : b :=B : f :=fonction :
  fa :=evalf(f(a)); fb :=evalf(f(b));
  err :=fa;
  i :=0 :
  #if (fa<0) then f :=-fonction; end;
  while ((abs(err)>eps) and (i<Nmax)) do
  c :=0.5*(a+b) :
  err :=evalf(f(c)) :
  if (err<0) then b :=c :
  fb :=err :
  else a :=c :
  fa :=err :
  fi :
  i :=i+1 :
  od :
  if i=Nmax then
  error "pas de convergence\ n" : fi;
  c;
  end proc :
```

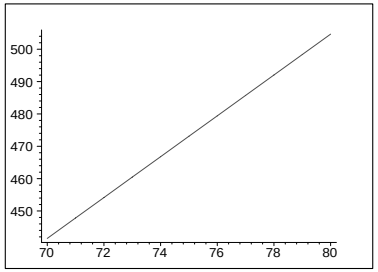
```
> g :=9.81 :
  NT :=10 : Tmin :=70 : Tmax :=80 :
  Nd :=10 : dmin :=0.1 : dmax :=10 :
```

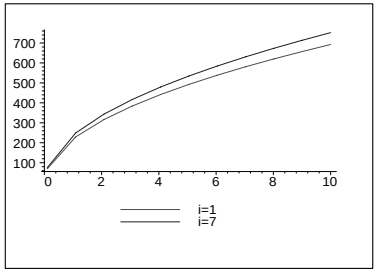
```
> T :=[seq(Tmin+(i-1)*(Tmax-Tmin)/NT,i=1..NT+1)] :
  d :=[seq(dmin+(i-1)*(dmax-dmin)/Nd,i=1..Nd+1)] :
  L :=array(1..NT+1,1..Nd+1) :
  maxi :=0 :
```

```
> for i to NT+1 do
  for j to Nd +1 do
  tmp :=fsolve(g*T[i]^2/(2*Pi)*
  tanh(2*Pi*d[j]/x) =x,
  x=0..infinity);
  a :=50 : b :=10000 : eps :=1e-5 :
  f :=x->g*T[i]^2/(2*Pi)*tanh(2*Pi*d[j]/x)-x;
  L[i,j] :=dicho(a,b,eps,100,f);
  if (abs(L[i,j]-tmp)>maxi) then maxi :=abs(L[i,j]-tmp);
  fi :
  od :od :
  eval(L);
  maxi;
```

```
> i :=3 :
  plot([seq([d[k],L[i,k]],k=1..Nd+1)]);
  i :=5 :
  plot([seq([T[k],L[k,i]],k=1..Nd+1)]);
  i :=1 :
  G1 :=plot([seq([d[k],L[i,k]],k=1..Nd+1)],
       legend="i=1") :
  i :=7 :
  G2 :=plot([seq([d[k],L[i,k]],k=1..Nd+1)],
       color=blue,legend="i=7") :
  with(plots) :
  display([G1,G2]);
```







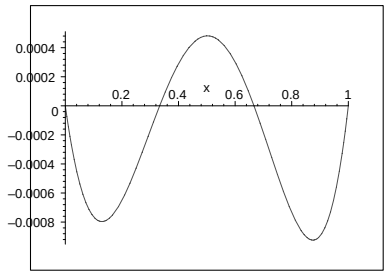
Exercice 15

```
> restart :
  f :=exp(x); N :=3 : a :=0 : b :=1 :
```

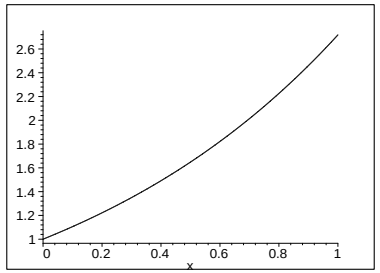
$$f := e^x$$

```
> h :=(b-a)/N :
  L :=[seq(a+k*h,k=0..N)] :
  g :=unapply(f,x) :
  Pf :=interp(L,map(g,L),x) :
  seq(subs(x=L[k],Pf-f),k=1..N+1);
  plot(f-Pf,x=a..b);
```

$$1 - e^0, 0, 0, 0$$




```
> with(plots) : G1 :=plot(f,x=a..b,color=blue) : G2 :
```



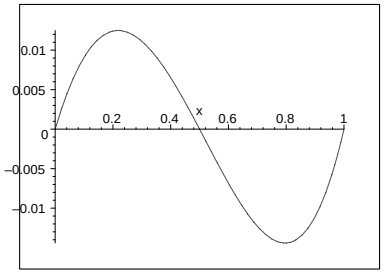
Modifications

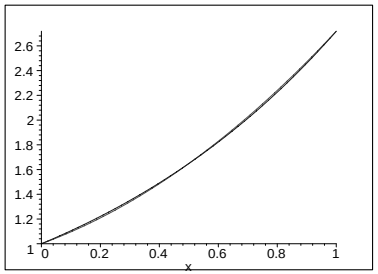
```
> f :=exp(x); Nmax :=10 : a :=0 : b :=1 :
```

$$f := e^x$$

```
> G1 :=plot(f,x=a..b,color=blue) :
  Erreur :=[seq(0,k=1..Nmax-1)] :
  superposition :=[seq(0,k=1..Nmax-1)] :

  for N from 2 to Nmax do
  h :=(b-a)/N :
  L :=[seq(a+k*h,k=0..N)] :
  g :=unapply(f,x) :
  Pf :=interp(L,map(g,L),x) :
  Erreur[N-1] :=plot(f-Pf,x=a..b);
  G2 :=plot(Pf,x=a..b,color=red) :
  superposition[N-1] :=display([G1,G2]) :
  od :
  display(Erreur,insequence=true);
  display(superposition,insequence=true);
```





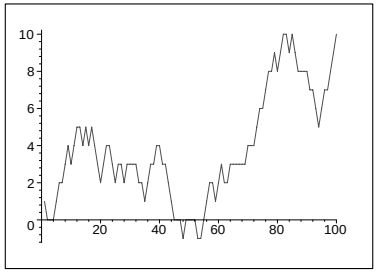
Exercice 16

Marche dans la rue :

```
> hasard :=rand(-1..1) :
> position_ivrogne :=0 :
  for k from 1 to 100 do
  a :=hasard() :
  position_ivrogne :=position_ivrogne+a :
  position[k] :=position_ivrogne :
  od :
> position_ivrogne;
```

10

```
> l :=[seq([n,position[n]],n=1..100)] :
> plot(l,0..100);
```

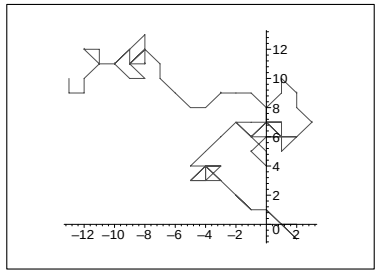


Marche dans le plan :

```
> nb_pas :=100 :
> hasard :=[rand(-1..1),rand(-1..1)] :

> position_ivrogne :=[0,0] :
  for k from 1 to nb_pas do
  pas :=hasard() :
  position_ivrogne :=position_ivrogne+pas :
  position[k] :=position_ivrogne :
  od :

> l :=[seq(position[n],n=1..nb_pas)] :
> plot(l,scaling=constrained);
```



distance totale

```
> distance_totale :=proc(position,nb_pas)
  local dist,k,x,y;
  dist :=0;
  for k to nb_pas do
  x :=position[k][1] :
  y :=position[k][2] :
  dist :=dist + sqrt(x^2+y^2) :
  od;
  evalf(dist);
  end proc :
> distance_totale(position,nb_pas);
```

832.3370383

```
> distance_origine :=proc(position,nb_pas) local x,y
> distance_origine(position,nb_pas);
```

16.40121947