

CORRECTION - MAPLE - CHAPITRE 2

Exercice 7

► Utilisation de fonctions :

```
> N :=10 :
f :=x->1/(1+x^2); a :=-1 : b :=1 :
# f :=x->exp(-x^2); a :=-1 : b :=1 :
```

```
int(f(x),x=a..b) : Iexact :=evalf(%) :
a :=evalf(a) : b :=evalf(b) :
dx :=(b-a)/N :
```

méthode des rectangles

```
0. : a :
for i to N do %f(%) : %dx : od :
Irect :=%dx :
Irect-Iexact;
```

méthode des trapèzes

```
0.5*(f(a)+f(b)) : a+dx :
for i to N-1 do %f(%) : %dx : od :
Itrap :=%dx :
Itrap-Iexact;
```

$$f := x \rightarrow \frac{1}{1+x^2}$$

-0.003333269
-0.003333269

► Utilisation d'expressions

```
> N :=10;
f :=1/(1+x^2); a :=-1 : b :=1 :
#f :=exp(-x^2); a :=-1 : b :=1 :
```

```
int(f,x=a..b) : Iexact :=evalf(%) ;
a :=evalf(a) : b :=evalf(b) :
dx :=(b-a)/N :
```

méthode des rectangles

```
0. : a :
for i to N do %f(%) : %dx : od :
Irect :=%dx :
Irect-Iexact;
```

méthode des trapèzes

```
0.5*(subs(x=a,f)+subs(x=b,f)) : a+dx :
for i to N-1 do %f(%) : %dx :
od :
Itrap :=%dx :
Itrap-Iexact;
```

$$N := 10$$

$$f := \frac{1}{1+x^2}$$

$$Iexact := 1.570796327$$

-0.003333269
-0.003333269

Exercice 8

► Méthode des rectangles :

```
> rect :=proc(f,a,b,N)
local dx,i,A,B :
A :=evalf(a) : B :=evalf(b) :
dx :=(B-A)/N :
0. : A :
for i to N do %f(%) : %dx : od :
%dx :
end proc :
```

► Méthode des trapèzes :

```
> trap :=proc(f,a,b,N)
local dx,A,B,i :
A :=evalf(a) : B :=evalf(b) :
dx :=(B-A)/N :
0.5*(f(A)+f(B)) : A+dx :
for i to N-1 do %f(%) : %dx : od :
%dx :
end proc :
> f :=x->exp(-x^2);
a :=-1;
b :=1;
N :=10;
rect(f,a,b,N); trap(f,a,b,N);
```

$$f := x \rightarrow e^{-x^2}$$

```
a := -1
b := 1
N := 10
1.488736679
1.488736679
```

Exercice 9

```
> restart :
```

```
> Digits :=3 :
```

```
> N :=100 : L :=array(1..N) :
initial :=[1/7,1./7] :
resultat := initial :
for i from 1 to 2 do
L[1] :=initial[i] :
for k from 1 to N do
S :=add(L[j],j=1..k-1) :
T :=add(j*L[j]*L[k-j],j=1..k-1) :
L[k] :=(6/(5*k+9))*(T-S+1/3);
od :
L[N] :resultat[i] :=evalf(%) :
od :
resultat;
```

[0.2989229284 10⁻⁹, 0.2989229284 10⁻⁹]

Exercice 10

```
> restart :
```

```
dicho :=proc(aa,bb,eps,Nmax,f)
local fa,fb,err,i,c,a,b;
a :=evalf(aa) :
b :=evalf(bb) :
```

```
# valeur de f en a,b
fa :=evalf(subs(x=a,f));
fb :=evalf(subs(x=b,f));
```

```
# initialisation
err :=fa : i :=0 :
```

```
# boucle
if (fa*fb>0) then error
  "Mauvais Intervalle\ n"; fi :
while ((abs(err)>eps) and (i<Nmax)) do
  c := 0.5*(a+b) :
  err := evalf(subs(x=c,f)) :
  if (err<0) then b :=c :
  fb :=err :
  else a :=c :
  fa :=err :
  fi :
  i :=i+1 :
od :
i,c,evalf(subs(x=c,f)); end proc :
```

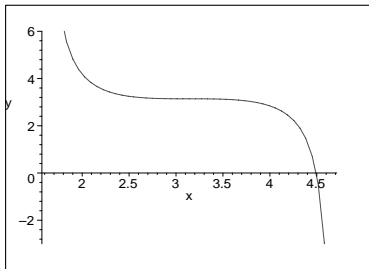
```
> f :=x-tan(x);
aa :=Pi/2+0.1 :
bb :=3*Pi/2-0.1;
dicho(aa,bb,1e-10,50,f);
```

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

$$bb := \frac{3\pi}{2} - 0.1$$

50, 4.493409457, 0.18 10⁻⁷

À titre d'information, voici le graphe de l'application $x \rightarrow x - \tan(x)$ sur l'intervalle $]\frac{\pi}{2}, \frac{3\pi}{2}[$:



► Modifications pour le cas où $f(a) < 0$

↪ 1ère version :

```
> dicho :=proc(aa,bb,eps,Nmax,f)
  local fa,fb,err,i,c,a,b;
  a :=evalf(aa) :
  b :=evalf(bb) :

  # valeur de f en a,b
  fa :=evalf(subs(x=a,f));
  fb :=evalf(subs(x=b,f));

  # initialisation
  err :=fa : i :=0 :

  # boucle
  if (fa*fb>0) then error
    "Mauvais Intervalle\ n"; fi :
  while ((abs(err)>eps) and (i<Nmax)) do
    c := 0.5*(a+b) :
    err := evalf(subs(x=c,f)) :
```

```
Cas où  $f(a) < 0$ 
if fa>0 then
  if (err<0) then b :=c :
  fb :=err :
  else a :=c :
  fa :=err :
  fi :
```

```
Cas où  $f(a) > 0$ 
else
  if (err>0) then b :=c :
  fb :=err :
  else a :=c :
  fa :=err :
  fi :
  i :=i+1 :
od :
i,c,evalf(subs(x=c,f));
end proc :
```

```
> dicho(aa,bb,0.01,20,-f);
```

11, 4.493174043, -0.004747927

↪ 2ème version

```
> dicho :=proc(aa,bb,eps,Nmax,ff)
  local fa,fb,err,i,c,a,b,f,signe_fa;
  a :=evalf(aa) :
  b :=evalf(bb) :
```

```
# valeur de f en a,b
fa :=evalf(subs(x=a,ff));
fb :=evalf(subs(x=b,ff));
if fa<0 then signe_fa :=-1;
f :=-ff;
else signe_fa :=1;
fi;
```

```
# initialisation
err :=fa : i :=0 :
```

```
# boucle
if (fa*fb>0) then error
  "Mauvais Intervalle\ n"; fi :
while ((abs(err)>eps) and (i<Nmax)) do
  c := 0.5*(a+b) :
  err := evalf(subs(x=c,f)) :
  if (err<0) then b :=c :
  fb :=err :
  else a :=c :
  fa :=err :
  fi :
  i :=i+1 :
od :
i,c,signe_fa*evalf(subs(x=c,f));
end proc :
```

```
> dicho(aa,bb,0.01,20,-f);
```

11, 4.493174043, -0.004747927