



Tracés par enveloppes

© Pierre Lantagne

Enseignant retraité du Collège de Maisonneuve

Voici des éléments de programmation pour le tracé de motifs qu'on retrouve dans les pages de mon site Internet. Ces éléments de programmation sont des éléments de travail et ne sont donc pas documentés.

Bonne lecture à tous !

* Ce document Maple est exécutable avec la version 2020.2

Initialisation

```
> restart;
> with(plots,pointplot,display,setoptions):
  setoptions(titlefont=[Times,bolditalic,12],
             axesfont=[Times,italic,8]);
with(plottools,rotate):
> ColorTools[RGB24ToRGB]([255, 253, 247]):
ColorTools[RGB24ToRGB]([128, 0, 128]):
Fond:=ColorTools:-Color([1,0.99215686,0.96862745]);
Tracé:=ColorTools:-Color([0.50196078, 0., 0.50196078]);
      Fond := <RGB : 1 0.992 0.969>
      Tracé := <RGB : 0.502 0 0.502>
```

(1.1)

Astroïde par enveloppe

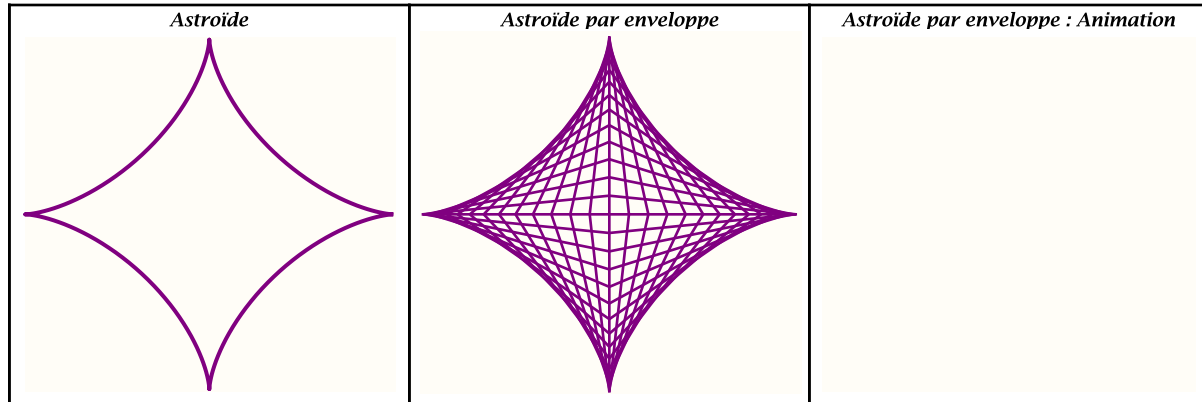
```
> nbrDeDroites:=60:
Longueur:=10:
Centre:=[10,10]:
extremA:=[10,20]:
extremB:=[20,10]:

> Astroïde:=plot([10*(cos^3)(t)+10,10*(sin^3)(t)+10,t=0..2*Pi],color=Tracé,
linestyle=0):
A:=display(Astroïde,axes=none,background=Fond,title="Astroïde"):
for i from 1 to nbrDeDroites do
  XA:=op(1,extremA)+Longueur*cos(2*Pi*i/nbrDeDroites):
  YB:=op(2,extremB)+Longueur*sin(2*Pi*i/nbrDeDroites):
  D||i:=plot([[evalf(XA),op(2,Centre)],[op(1,Centre),evalf(YB)]],style=line,
```

```

thickness=0,color=Tracé):
  F||i:=display(seq(D||k,k=1..i)):
od:
B:=display([D|(1..nbrDeDroites)],axes=None,background=Fond,title="Astroïde
par enveloppe"):
F0:=plot([10*(cos^3)(t)+10,10*(sin^3)(t)+10,t=0..2*Pi],color=Fond,
background=Fond,axes=None):
C:=display([F|(0..nbrDeDroites)],insequence=true,axes=None,title=typeset
("Astroïde par enveloppe : Animation"),background=Fond):
display(Matrix(1,3,[A,B,C]));

```



Épicycloïde $q = n$ par enveloppe

```

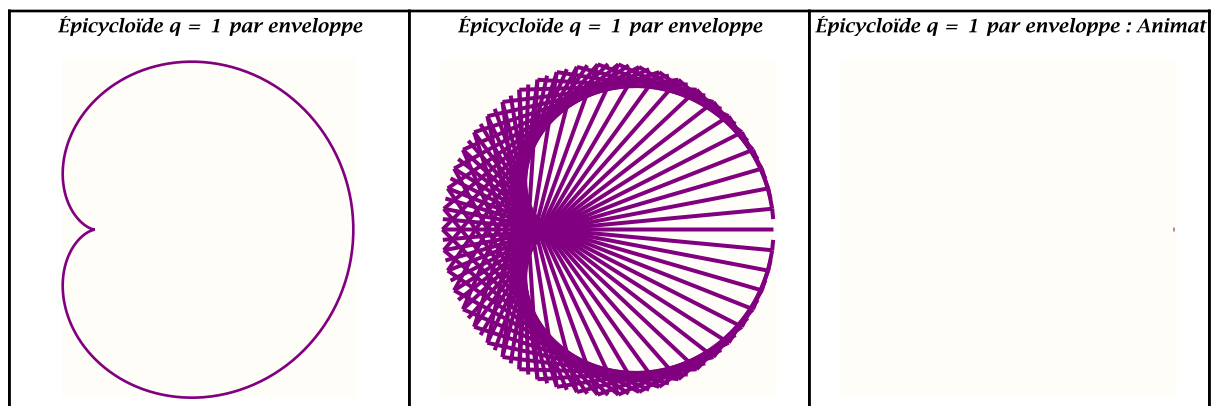
> Épicycloïde:=proc(q::posint)
  local a,A,B,C,Centre,i,F0,J,L,L_rot,nbrDeDroites,nbrDePoints,
nbrDeRebroussements,rayonDuCercle,TI,TJ:
  a:=1:
  rayonDuCercle:=10:
  Centre:=[0,0]:
  if q <=5 then nbrDePoints:=100+(q-1)*20 elif
    q <=10 then nbrDePoints:=100+(q-1)*10 elif
    q <=15 then nbrDePoints:=100+(q-1)*5
  else nbrDePoints:=180
  fi:
  L:=plot([-a*((q+1)*cos(t)-cos((q+1)*(t))),-a*((q+1)*sin(t)-sin((q+1)*(t))),
t=0..2*Pi],color=Tracé,thickness=0,linestyle=1,scaling=constrained):
  L_rot:=rotate(L,Pi/q,Centre,axes=None,background=Fond):
  if type(q,even) then
    L_rot:=rotate(L,Pi/q,Centre,axes=None,background=Fond):
    L:=L_rot fi:
  A:=display(L,axes=None,background=Fond,title=typeset("Épicycloïde q = ",q,"
par enveloppe\n")):
  nbrDeRebroussements:=q+1:

```

```

for i from 0 to nbrDePoints-1 do
  TI:=2*Pi*i/nbrDePoints:
  J:=i*nbrDeRebroussements-nbrDePoints*iquo(i*nbrDeRebroussements,
nbrDePoints):
  TJ:=2*Pi*J/nbrDePoints:
  D||i:=plot([[op(1,Centre)+rayonDuCercle*cos(TI),op(2,Centre)+
rayonDuCercle*sin(TI)],
              [op(1,Centre)+rayonDuCercle*cos(TJ),op(2,Centre)+
rayonDuCercle*sin(TJ)]]),
            style=line,
            color=Tracé):
  F||i:=display(seq(D||k,k=0..i));
od:
nbrDeDroites:=nbrDePoints-1:
B:=display( D||(1..nbrDeDroites),axes=None,background=Fond,title=typeset
("Épicycloïde q = ",q, " par enveloppe\n"));
F0:=plot([-a*((q+1)*cos(t)-cos((q+1)*(t))),-a*((q+1)*sin(t)-sin((q+1)*(t))
),t=0..2*Pi],color=Fond,thickness=0,linestyle=1, scaling=constrained):
C:=plots[display]([F||(0..nbrDeDroites)],insequence=true,
                  scaling=constrained,axes=None,background=Fond,
                  title=typeset("Épicycloïde q = ",q, " par
enveloppe : Animation\n"));
display(Matrix(1,3,[A,B,C]),scaling=constrained);
end proc:
> Épicycloïde(1);

```

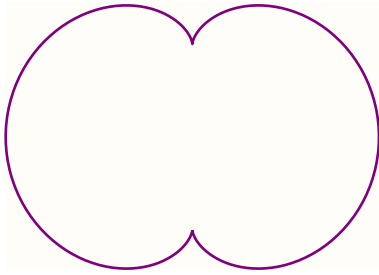


```

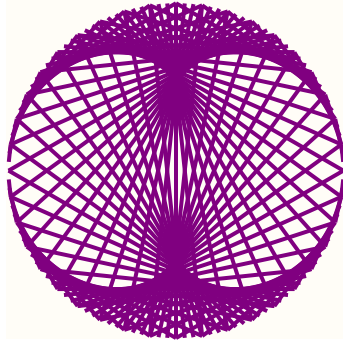
> Épicycloïde(2);

```

Épicycloïde $q = 2$ par enveloppe



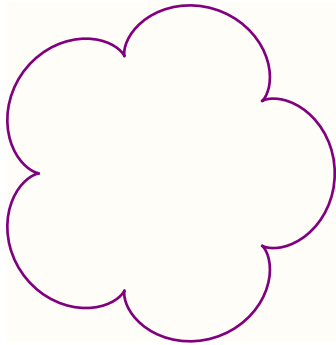
Épicycloïde $q = 2$ par enveloppe



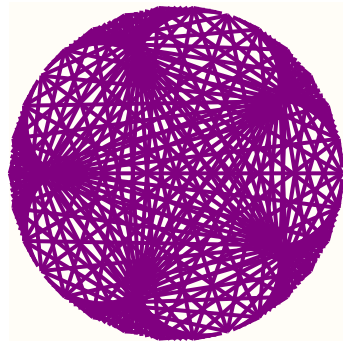
Épicycloïde $q = 2$ par enveloppe : Animat

> Épicycloïde(5);

Épicycloïde $q = 5$ par enveloppe



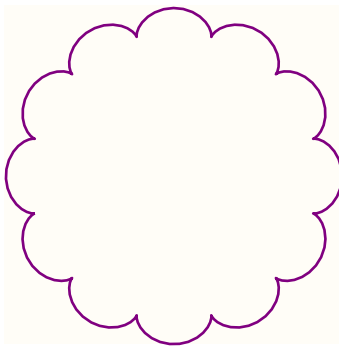
Épicycloïde $q = 5$ par enveloppe



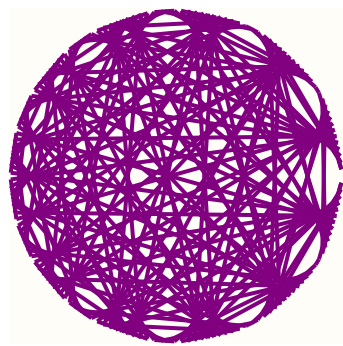
Épicycloïde $q = 5$ par enveloppe : Animat

> Épicycloïde(12);

Épicycloïde $q = 12$ par enveloppe



Épicycloïde $q = 12$ par enveloppe



Épicycloïde $q = 12$ par enveloppe : Anima

▼ Napperons par enveloppe

```

> Napperons:=proc(m::posint)
  local A,B,C,Centre,F0,J,i,j,n,nbrDeDroites,nombreDesommets,Points,
  rayonDuCercle,TI,TJ:
  if m<3 then ERROR(`Vous devez donner un nombre de sommets égale à
  trois au minimum`) fi;
  rayonDuCercle:=30:

```

```

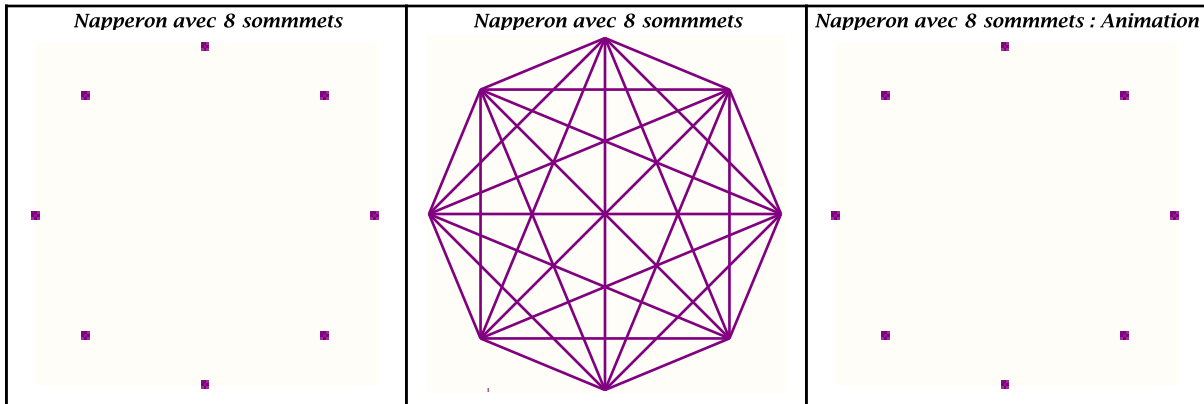
Centre:=[30,30]:
nombreDeSommets:=13:
Points:=NULL:
nombreDeSommets:=m:
n:=0;
for i from 0 to nombreDeSommets-2 do
  TI:=2*Pi*i/nombreDeSommets:
  for j from i+1 to nombreDeSommets-1 do
    TJ:=2*Pi*j/nombreDeSommets:
    n:=n+1:
    D|n:=plot([[op(1,Centre)+rayonDuCercle*cos(TI),op(2,Centre)+
rayonDuCercle*sin(TI)],
              [op(1,Centre)+rayonDuCercle*cos(TJ),op(2,
Centre)+rayonDuCercle*sin(TJ)]]),
              style=line,thickness=0,color=Tracé):
    Points:=Points,[op(1, Centre) + rayonDuCercle*cos(TI), op(2,
Centre) + rayonDuCercle*sin(TI)],
                  [op(1, Centre) + rayonDuCercle*cos(TJ), op(2,
Centre) + rayonDuCercle*sin(TJ)]:
  od:
od:
nbrDeDroites:=sum('k', 'k'=1..nombreDeSommets-1):

A:=display(pointplot({Points},symbol=solidcircle,color=Tracé,axes=
none,
  background=Fond,title=typeset("Napperon avec ",m," sommets"))):
F0:=pointplot({Points},symbol=solidcircle,color=Tracé,background=
Fond):
for i from 1 to nbrDeDroites do
  F|i:=display(A,seq(D|k,k=1..i))
od:
B:=display([D|(0..nbrDeDroites)],axes=none,background=Fond,
  title=typeset("Napperon avec ",m," sommets")):#print(rotate(B,
Pi/2/m+Pi,Centre));
C:=display([F0,F|(1..nbrDeDroites)],
  axes=none,
  scaling=constrained,
  insequence=true,
  title=typeset("Napperon avec ",m," sommets : Animation"),
  background=Fond):
if type(m,odd) then
  display(Matrix(1,3,map(x->rotate(x,Pi/2/m+Pi,Centre),[A,B,C])),
scaling=constrained);
else display(Matrix(1,3,[A,B,C]),scaling=constrained)
fi;

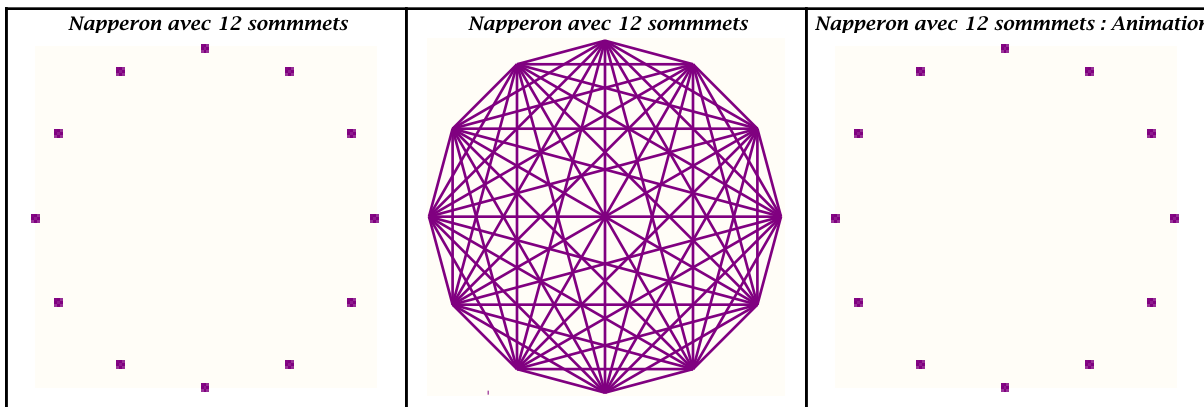
```

end proc:

> Napperons(8);



> Napperons(12);



> Napperons(17);

