



Procédure Coloriage

© Pierre Lantagne

Enseignant retraité du Collège de Maisonneuve

Lors du tracé d'un graphique avec la macro-commande `plot`, l'option `filled` permet de remplir d'une couleur la région entre la courbe et l'axe des x . (Voir [?plot,option](#)). Cette option est valide avec `plot`, `contourplot`, `implicitplot`, `listcontplot`, `polarplot`, et `semilogplot`. La macro-commande proposée ici permet aussi de colorier de telles régions directement au traçage ou après coup mais permet aussi d'animer le coloriage d'une région plus correctement que l'on peut le faire avec l'option `filled`.

Bonne lecture à tous !

* Ce document Maple est exécutable avec la version 2020.2

Initialisation

```
> restart;
> with(plots,display,polarplot):
macro(dark_magenta=COLOR(RGB, 0.5, 0.0, 0.5),
      random=ColorTools:-Color([rand()/10^12, rand()/10^12, rand()
/10^12]),
      fond=ColorTools:-Color([1.00000000,0.99215686,0.96862745])):

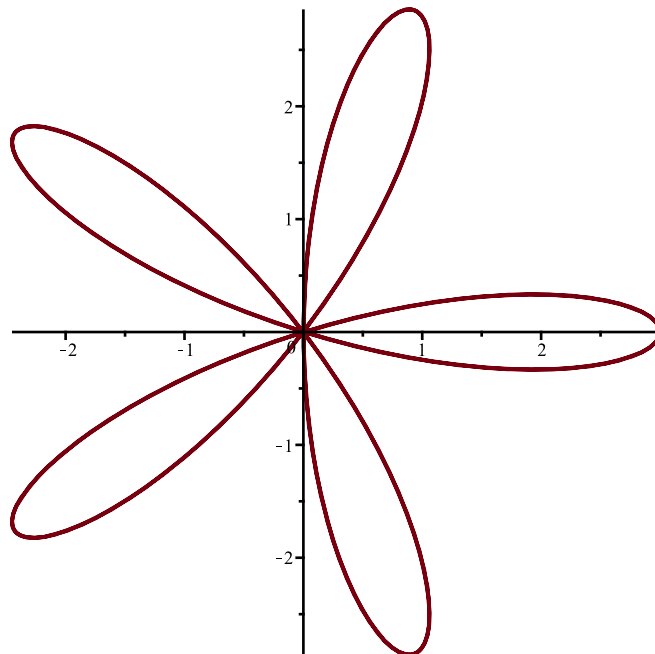
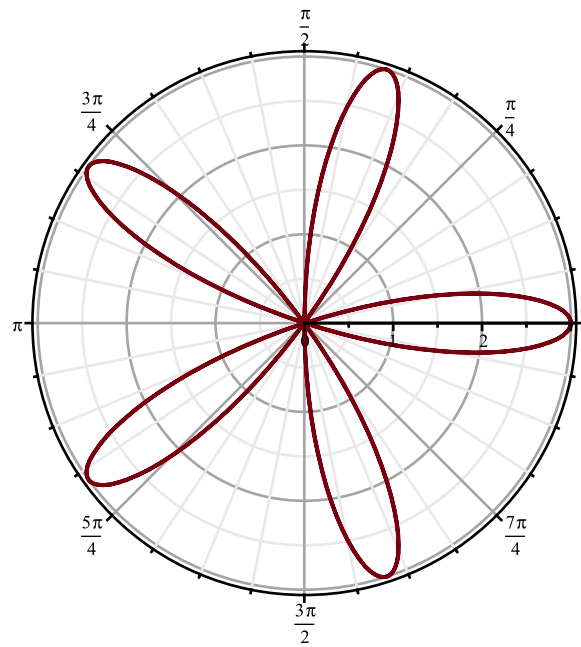
# Les options suivantes: axesfont=["TIMES", "TIMES", 8],\
      titlefont=[TIMES,ITALIC,10],\
      scaling=constrained,\
      size=[300,300]\
ont été lues dans un fichier texte maple.ini. Un tel fichier peut
être déposé le dossier Users de l'installation Maple sur votre
ordinateur ou placé au même niveau dans le dossier que celui de votre
document Maple. Ce fichier est lu au moment de l'ouverture de votre
document.
```

Remarque

Le choix de l'option globale `size=[300,300]` (setoptions du fichier `maple.ini`) est motivé par l'exportation en pdf de ce document d'une part, et d'autre part, par la nature même de l'affichage avec le système d'axe polaire qui laisse moins de place pour le tracé de la courbe que permet le système d'axes cartésiens. Voilà pourquoi, dans certains exemples, un autre choix pour l'option `size` a été arrêté.

Une image vaut mille mots: les deux graphiques ci-dessous ont la même taille d'affichage mais celui dans le système cartésien donne un tracé plus grand de la courbe.

```
> polarplot(3*cos(5*theta));
      polarplot(3*cos(5*theta),axiscoordinates=cartesian,axes=normal );
```



Procédure Coloriage

La procédure Coloriage effectue un remplissage de la couleur de son choix.

```
> Coloriage:=proc(C::function)
  local i,M,Opts,CouleurP,CouleurI,Dim_Ligne,Intérieur,
  Périmètre,Points_Courbe;
  Opts:=[args[2..nargs]];
  if evalb(op(0,C)=PLOT) then
  if hasfun(C,CURVES) then CouleurP:= op(selectfun(selectfun({op(C)},
  CURVES),COLOUR)) fi;
  if hasfun(C,POLYGONS) then CouleurI:=op(selectfun(selectfun({op(C)},
```

```

POLYGONS),COLOUR)) fi;
M:=op(1,op(1,C));
else M:=op(1,C);
fi;
Dim_Ligne:=op(1,M)[1];
if evalb(M[1,1]>M[2,1]) then Points_Courbe:=[[0,0],seq([M[i,1],M[i,2]
],i=1..Dim_Ligne)]
else Points_Courbe:=[seq([M[i,1],M[i,2]],i=1..Dim_Ligne)]
fi;
Périmètre:=(CURVES(convert(Points_Courbe,Matrix)));
Intérieur:=(POLYGONS(convert(Points_Courbe,Matrix)));
display([Périmètre,Intérieur],color=[CouleurP,CouleurI],op(Opts))
end proc:

```

Exemple 1

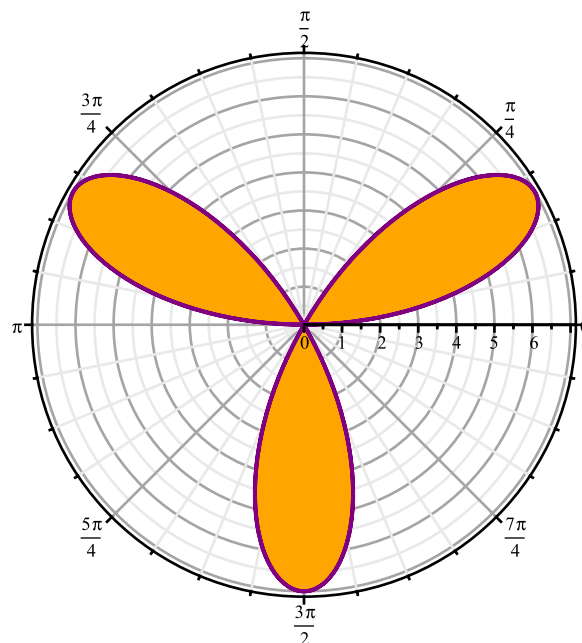
Ce premier exemple ne montre pas du tout l'intérêt à utiliser la macro-commande Coloriage, l'option filled faisant tout à fait le même travail.

Un premier tracé en utilisant l'option filled de $\rho = 7 \cos\left(3\theta - \frac{\pi}{2}\right)$

```

> polarplot(7*cos(3*theta-Pi/2),theta=0..2*Pi,
numpoints=200,
color=dark_magenta,
filled=[color="Orange",
transparency=.4]);

```

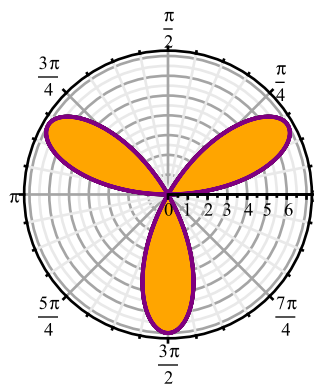
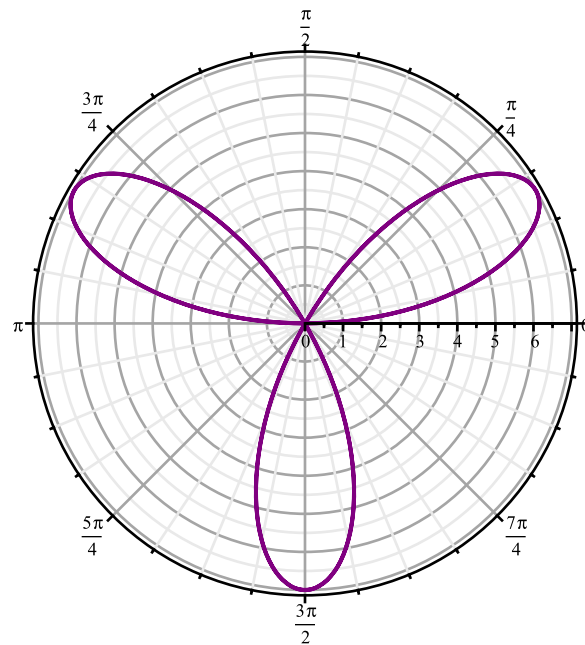


Les deux tracés suivants avec la macro-commande Coloriage: d'abord en coloriant après le tracé du graphique et ensuite, en imbriquant le tracé dans la macro-commande Coloriage.

```

> G:=polarplot(7*cos(3*theta-Pi/2),theta=0..2*Pi,
  numpoints=200,
  color=dark_magenta);
Coloriage(G,
  color=[dark_magenta,"Orange"],
  size=[300,300],
  transparency=.4,
  axiscoordinates=polar,
  axes=normal);

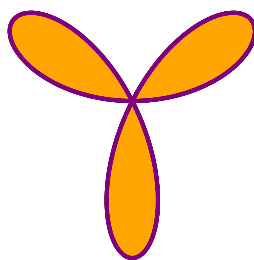
```



```

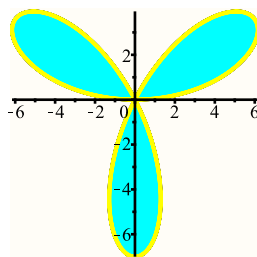
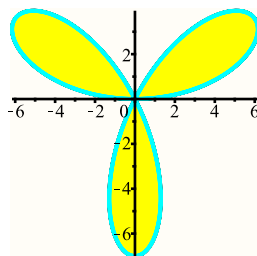
> Coloriage(polarplot(7*cos(3*theta-Pi/2),theta=0..2*Pi),
  color=[dark_magenta,"Orange"],
  size=[200,200],
  transparency=.4,
  axes=None);

```



Observons l'ordre de la liste color: la première couleur de la liste est pour colorier le périmètre et la seconde couleur est pour colorier la région intérieure.

```
> Coloriage(G,  
  color=[cyan,yellow],  
  transparency=.4,  
  size=[200,200],  
  background=fond);  
Coloriage(G,  
  color=[yellow,cyan],  
  size=[200,200],  
  transparency=.4,  
  background=fond);
```

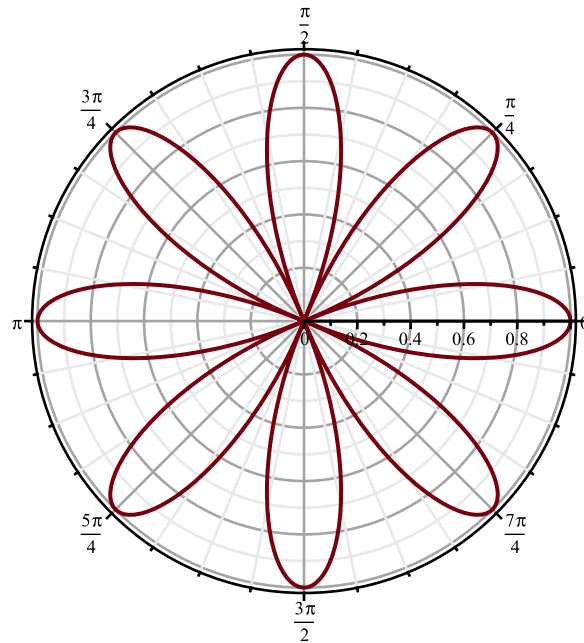


Exemple 2

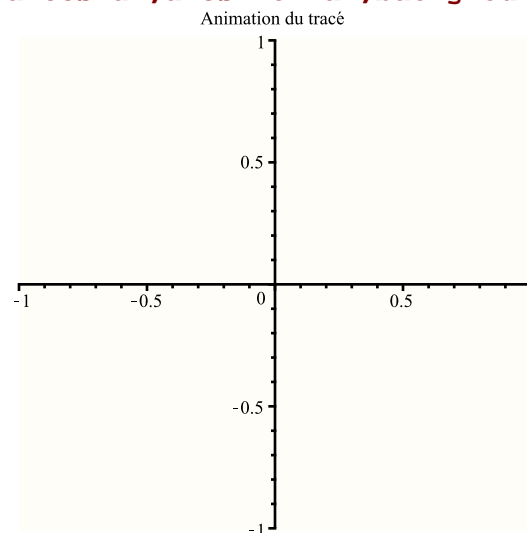
Pourquoi avoir créé la procédure Coloriage si l'option filled fait aussi bien le remplissage? La réponse apparaît clairement lorsqu'on désire animer la construction de la courbe. Réalisons d'abord une animation impliquant l'utilisation de l'option filled.

Traçons d'abord la région d'équation polaire $r \cos(4\theta)$.

```
> polarplot(cos(4*theta), numpoints=400);
```

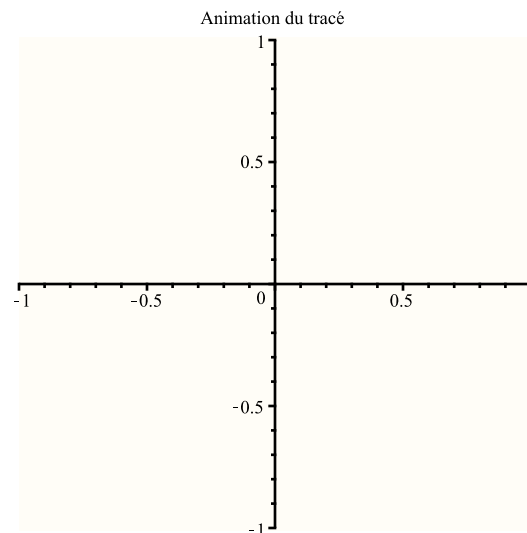


```
> Animation:=seq(polarplot(cos(4*theta),theta=0..2*k*Pi/80,filled=
[ color="Orange",transparency=0.5],numpoints=400),k=0..80):
display(Animation,insequence=true,title="Animation du tracé",
axiscoordinates=cartesian,axes=normal,background=fond);
```



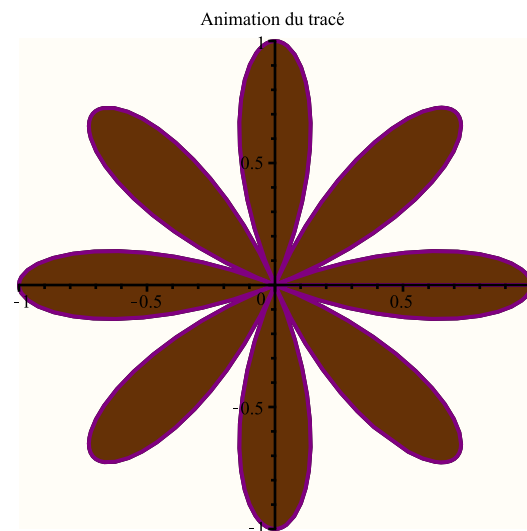
Réalisons maintenant la même animation avec Coloriage.

```
> Animation:=seq(Coloriage(polarplot(cos(4*theta),theta=0..2*k*Pi/80,
numpoints=200),color=["Burgundy","Orange"],transparency=0.5),k=0..80)
:
display(Animation,insequence=true,axiscoordinates=cartesian,axes=
normal,title="Animation du tracé",background=fond);
```

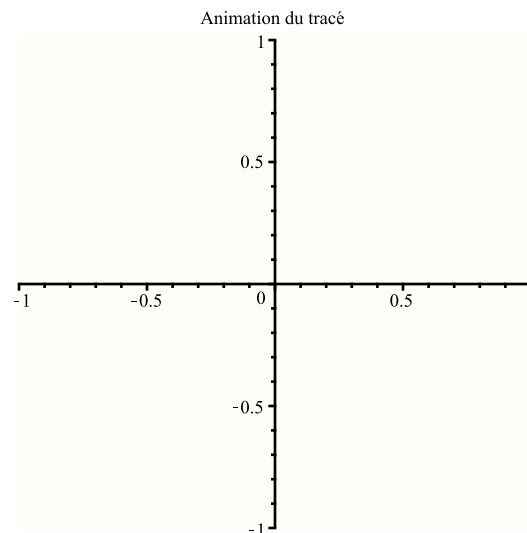


Voici deux animations qui "clignotent".

```
> Animation:=seq(Coloriage(polarplot(cos(4*theta),numpoints=100),color=[dark_magenta,random]),k=1..100):
display(Animation,insequence=true,title="Animation du tracé",
background=fond);
```



```
> Animation:=seq(Coloriage(polarplot(cos(4*theta),theta=0..2*k*Pi/100),
color=[dark_magenta,random]),k=0..100),
Coloriage(polarplot(cos(4*theta),theta=0..2*Pi,numpoints=
600),color=[dark_magenta,random]):
display(Animation,insequence=true,title="Animation du tracé",
background=fond);
```

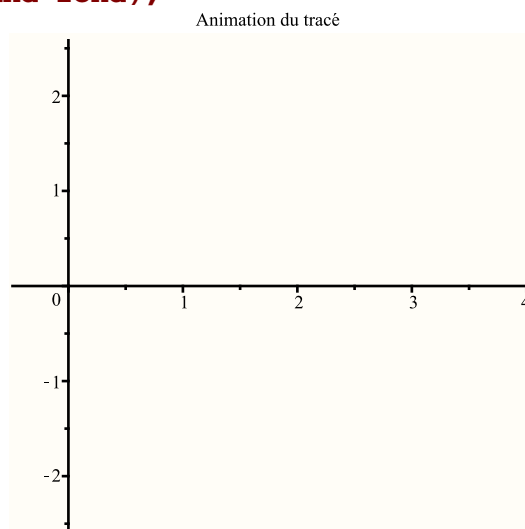


Exemple 3

La macro-commande Coloriage peut animer le tracé d'une courbe.

Cardioïde

```
> Animation:=seq(Coloriage(polarplot(2*(1+cos(theta)),theta=0..2*k*
  Pi/80),color=[dark_magenta,white],transparency=[0,1]),k=0..80):
  display(Animation,insequence=true,title="Animation du tracé",style=
  polygon,background=fond);
```

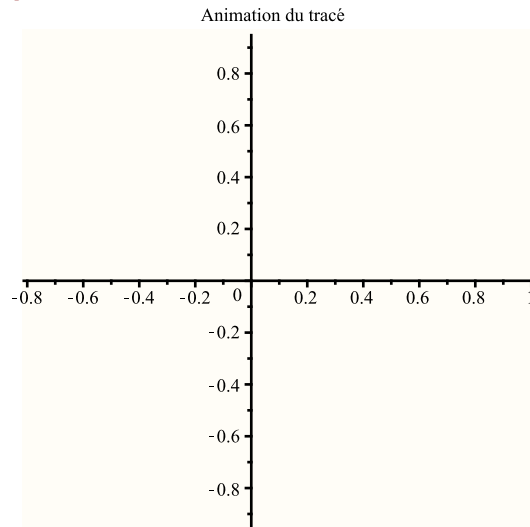


Rosace à 5 pétales

```
> Animation:=seq(Coloriage(polarplot(cos(5*theta),theta=0..2*k*
  Pi/100),color=[dark_magenta,white],transparency=[0,1]),k=0..100):
  display(Animation,insequence=true,title="Animation du tracé",
```



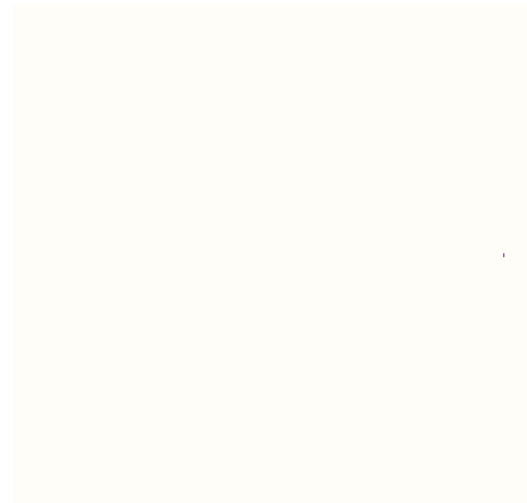
```
background=fond);
```



Trèfle de Habenicht

```
> Animation:=seq(Coloriage(polarplot(1+cos(5*theta)+(sin^2)(5*  
theta),theta=0..2*k*Pi/100),color=[dark_magenta,white],  
transparency=[0.,1.]),k=0..100):  
display(Animation,insequence=true,title="Animation du tracé",  
background=fond,axes=None);
```

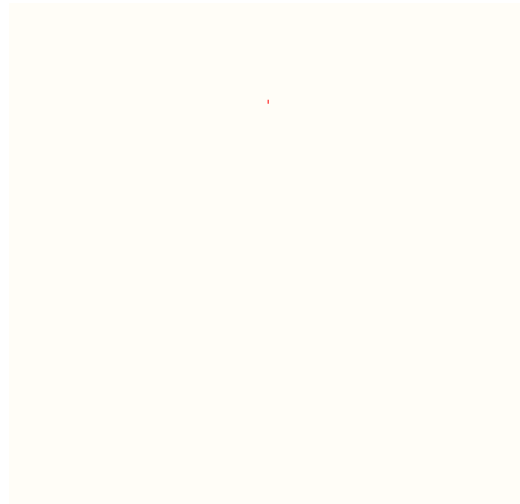
Animation du tracé



Coeur de Raphaël Laporte (1993)

Courbe trouvée par l'auteur à l'âge de 16 ans pour sa petite amie..

```
> Animation:=seq(Coloriage(plot([(sin^3)(t),cos(t)-(cos^4)(t)],t=0..  
.2*k*Pi/100]),color=[red,white],transparency=[0.,1.]),k=0..100):  
display(Animation,insequence=true,title="Animation du tracé",  
background=fond,axes=None);
```



Exemple 4

Quelques coloriages de régions très irrégulières en finissant avec un gros pétard...

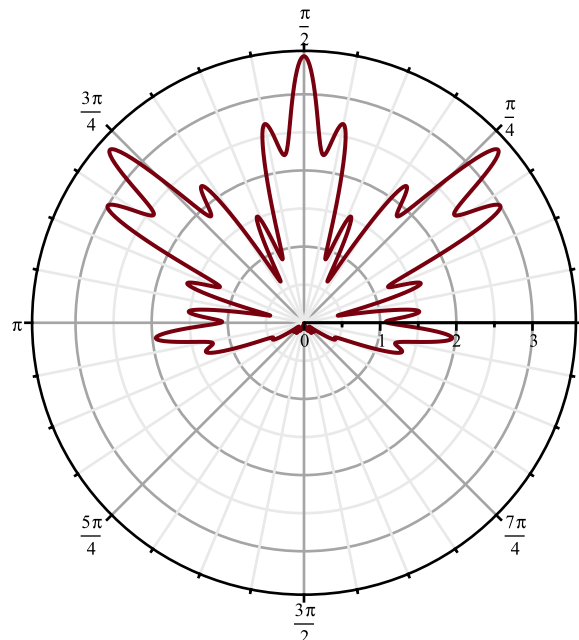
```
> S:=t->100/(100+(t-1/2*Pi)^8);
R:=t->S(t)*(2-sin(7*t)-1/2*cos(30*t));
```

$$S := t \mapsto \frac{100}{100 + \left(t - \frac{\pi}{2}\right)^8}$$

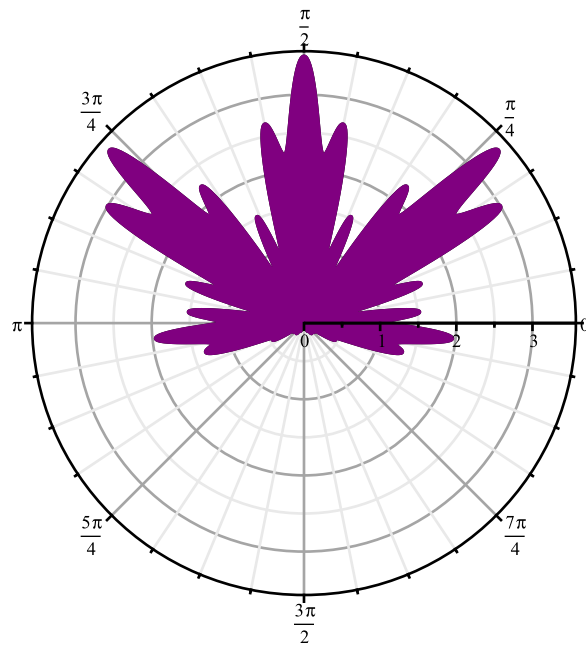
$$R := t \mapsto S(t) \cdot \left(2 - \sin(7 \cdot t) - \frac{\cos(30 \cdot t)}{2}\right)$$

(6.1)

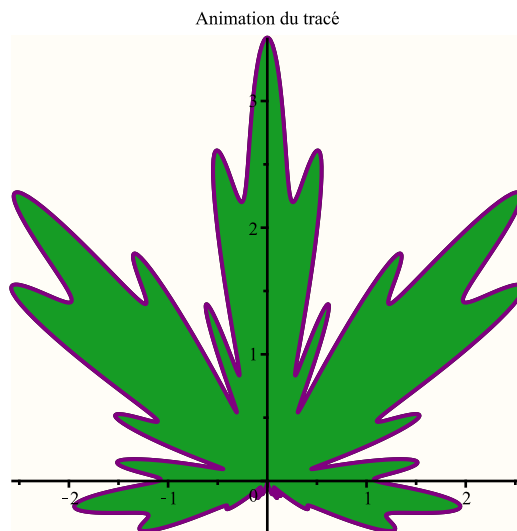
```
> C:=plots[polarplot]([R,t->t,-Pi/2..(3*Pi)/2],numpoints=1000);
```



```
> Coloriage(C,color=dark_magenta,axiscoordinates=polar,axes=normal);
```

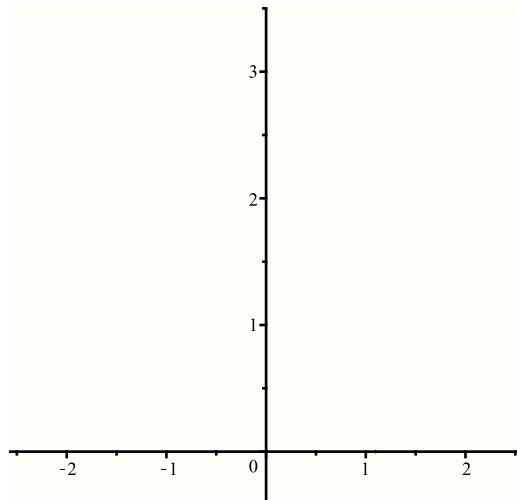


```
> Animation:=seq(Coloriage(polarplot([R,t->t,-Pi/2..(3*Pi)/2],
numpoints=1000),color=[dark_magenta,random]),k=1..80),
Coloriage(polarplot([R,t->t,-Pi/2..(3*Pi)/2],numpoints=1000,
thickness=3),color=[dark_magenta,yellow]):
plots[display](Animation,insequence=true,title="Animation du tracé",
background=fond);
```



```
> Animation:=seq(Coloriage(polarplot([R,t->t,-k*Pi/2..(3*k*Pi)/2/60],
numpoints=1000),color=[dark_magenta,random]),k=0..60),
Coloriage(polarplot([R,t->t,-Pi/2..(3*Pi)/2],numpoints=1000),color=
[dark_magenta,yellow]):
plots[display](Animation,insequence=true,title="Animation du tracé",
background=fond);
```

Animation du tracé



Et pour terminer... le gros pétard.

```
> r:=(1+0.9*cos(8*theta))*(1+0.1*cos(24*theta))*(0.9+0.1*cos(200*theta)
)* (1+sin(theta));
C:=polarplot(r,theta=0..2*Pi,axiscoordinates=cartesian,axes=none,
numpoints=2000);
r := (1 + 0.9000000000 cos(8θ)) (1 + 0.1000000000 cos(24θ)) (0.9000000000
+ 0.1000000000 cos(200θ)) (1 + sin(θ))
```



```
> C:=polarplot(r,theta=0..2*Pi,axiscoordinates=cartesian,axes=none,
color="LeafGreen",filled=[color="LeafGreen"],numpoints=2000);
```



```
> Coloriage(C,color=["LeafGreen","LeafGreen"],transparency=0.4,  
background=fond,axes=None);
```



```
> Animation:=seq(Coloriage(polarplot(r,theta=0..2*k*Pi/40),color=  
["LeafGreen","LeafGreen"],numpoints=2000,transparency=0.4),k=0..40),  
Coloriage(polarplot(r,theta=0..2*Pi,numpoints=2000),color=  
["LeafGreen","LeafGreen"],transparency=0.4):  
display(Animation,insequence=true,title="Animation du tracé",  
background=fond,axes=None);
```

Animation du tracé

