Coordonnées polaires

(Partie 2 de 2)

© Pierre Lantagne

Enseignant retraité du Collège de Maisonneuve

La première version de ce document est parue en février 2001. Ce document est le second d'une série de deux portant sur les coordonnées polaires. L'objectif de cette seconde partie est de montrer la manière dont s'effectue le tracé d'une courbe définie en coordonnées polaires. On y trouvera une procédure permettant d'animer, étape par étape, la création d'une courbe définie en coordonnées polaires permettant de **bien voir le début de la courbe** ainsi que **le sens dont s'effectue le tracé**.

L'objectif de la première partie est de détailler le système de coordonnées polaires. Après avoir défini ce qu'est le système de coordonnées polaires, nous présenterons les équations de transformation du système polaire au système cartésien. Nous illustrerons ensuite la réprésentation graphique de quelques lieux géométriques remarquables défines en coordonnées polaires avec un quadrillage polaire et avec une quadrillage cartésien. Pour terminer ce premier document, nous verrons les précautions à prendre dans la résolution de systèmes d'équations qui sont à poser lorsqu'il faut déterminer les points d'intersection de courbes définies en coordonnées polaires.

Bonne lecture à tous !

* Ce document Maple est exécutable avec la version 2020.2

Initialisation

```
> restart;
> with(plots,coordplot,display,polarplot,setoptions,textplot):
  setoptions(size=[300,300],axesfont=[TIMES,ROMAN,8],labelfont=[TIMES,ROMAN,
  8],titlefont = ["ROMAN", 12]);
  with(plottools,disk):
```

Macro-commande app

La macro-commande

animepola5plot(Équation, Intervalle, Nombre d'images, Mode, Système,Options)

possède cinq arguments obligatoires:

- **Équation** est une équation polaire de la forme $r = \mathrm{f}(\theta)$.
- **Intervalle** est l'intervalle de la forme $\theta = a\,..b$ dans lequel variera l'angle θ.
- **Nombre d'images** est un entier positif spécifiant le nombre de séquences qui seront générées.
- **Mode** est le type d'affichage de la courbe: fixe ou anime. Dans le cas où le mode est fixe, la procédure ignorera la valeur de l'argument Nombre d'images.
- **Système** est une des valeurs suivantes: polaire, cartésien, normal, frame, boxed, none.

Cette macro-commande sera active pour la session seulement après avoir exécuté la procédure **app** de la sous-section Initialisation. Cette macro-commande n'appartient donc ni à la bibliothèque principale ni à la bibliothèque plots.

Il est toujours possible d'abréger le nom de cette macro-commande en créant un alias de son nom.

```
> alias(app=animepolarplot):
```

Initialisation

```
> animepolarplot:=proc(Lieu::equation,Intervalle::equation,Sequences::posint,Mode::name,
  Axes::name)
  local a,Animation,b,Courbe,d,k,r,Opts,P,Pas,Rayon,Var;
  option `Copyright février 2001 Pierre Lantagne`;
  Rayon:=lhs(Lieu);
  Opts:=[args[6..nargs]];
  Var:=lhs(Intervalle);
  a:=evalf(lhs(rhs(Intervalle)));
  b:=evalf(rhs(rhs(Intervalle)));
  r:=unapply(rhs(Lieu),Var);
  Digits:=15;
  Pas:=seq(a+((b-a)/Sequences)*(k+1e-10),k=0..Sequences);
  d:=max(seq(abs(evalf(r(i))),i=Pas));
  if Axes=polaire then
   P:=plots[coordplot](polar,[0..ceil(d),0..2*Pi],
            grid=[10,25],
            color=[grey,grey],
            linestyle=[0,0],
            thickness = 0,
            labelling=false,
            font=[TIMES,ROMAN,8],
            scaling=constrained,
            view=[-ceil(d)..ceil(d),-ceil(d)..ceil(d)])
   elif Axes=cartésien then
   P:=plots[coordplot](cartesian,[-ceil(d)..ceil(d),-ceil(d)..ceil(d)],
            grid=[21,21],
            color=[grey,grey],
            linestyle=[0,0],
            thickness = 0,
            labelling=false,
            font=[TIMES,ROMAN,8],
            scaling=constrained,
            view=[-ceil(d)..ceil(d),-ceil(d)..ceil(d)])
   else
   P:=plot([[0,0]],axes=Axes,op(Opts))
  fi;
  if Mode=anime then
   Animation:=seq(plots[display](
        [P,plottools[line]([0,0],[-1*d*cos(k),-1*d*sin(k)],color="Niagara 4",linestyle=2),
         plottools[line]([0,0],[d*cos(k),d*sin(k)],color="Niagara 1",linestyle=2),
         plottools[point]([Re(r(k))*cos(k),Re(r(k))*sin(k)],color="Niagara 1",symbol=circle),
         plot(Re(r(t)),t=a..k,coords=polar,color="Niagara 5",thickness=2,op(Opts),numpoints=
  ceil(480+240*abs(k)))]),k=Pas);
   plots[display](Animation,insequence=true,scaling=constrained,op(Opts));
   else
   Courbe:=plot(Re(r(t)),t=a..b,coords=polar,thickness=2,scaling=constrained,op(Opts)):
   plots[display]([P,Courbe],op(Opts))
  fi
  end:
```

Droites

Avec une équation linéaire de la forme $a x + b y + c = 0$, les équations $x = r\cos(\theta)$ et $y = r\sin(\theta)$ nous conduisent à l'équation

$$r = -\frac{c}{a\cos(\theta) + b\sin(\theta)}$$

Obtenons l'équation polaire de la droite d'équation cartésienne $2x + 3y + 2 = 0$. Tracez ensuite cette équation polaire.

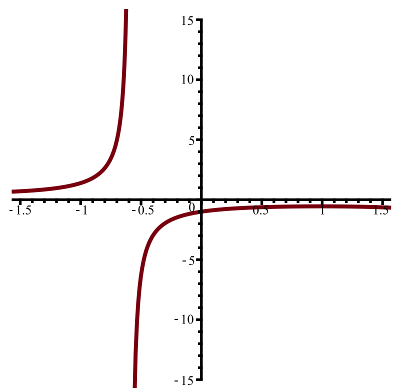
```
> Éq:=2*x+3*y+2 = 0;
```

$$\acute{E}q := 2x + 3y + 2 = 0 \quad \textbf{(3.1)}$$

```
> Éq_polaire:=isolate(subs(x=r*cos(theta),y=r*sin(theta),Éq),r);
```

$$\acute{E}q\_polaire := r = -\frac{2}{2\cos(\theta) + 3\sin(\theta)} \quad \textbf{(3.2)}$$

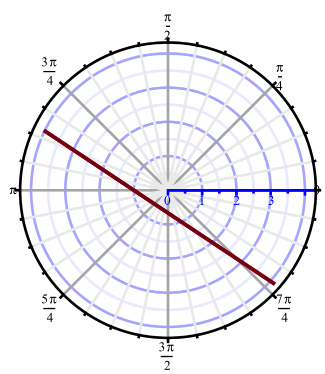
```
> plot([theta,-2/(2*cos(theta)+3*sin(theta)),theta=-Pi/2..Pi/2],discont=true);
```



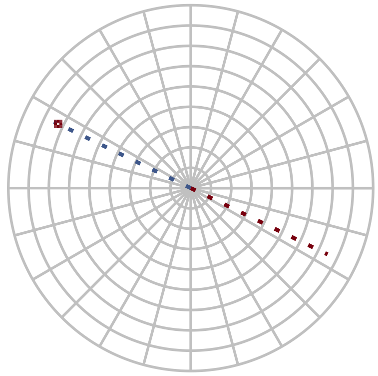

```
> Pi/3.;
```

$$1.0471975510 \quad \textbf{(3.3)}$$

```
> polarplot(-2/(2*cos(theta)+3*sin(theta)), theta = -0.45..2.42, axis[radial]
  = [color = "Blue"]);
```

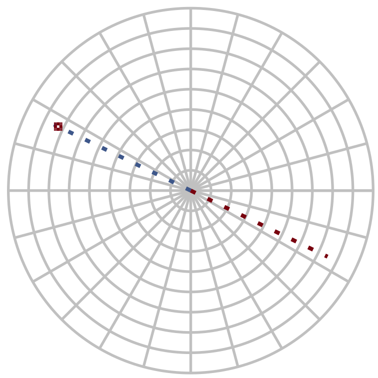



```
> app(Éq_polaire,theta=-0.45..2.42,18,anime,polaire,grid=[25,25]);
```



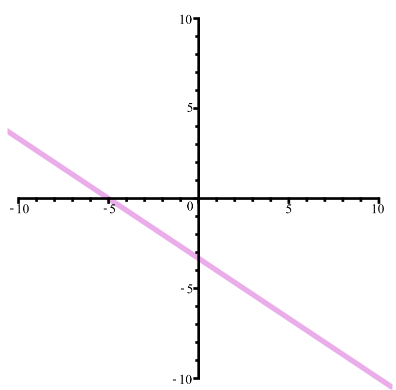
Animons le tracé de cette droite. Vous allez mieux voir la manière dont sont engendrés les points de cette droite.

```
> app(Éq_polaire,theta=-0.45..2.42,18,anime,polaire);
```



L'animation suivante est composée d'une séquence de droites parallèles à la droite précédente. Observons ce qui se passe à l'origine.

```
> A:=seq(app(r=k/(2*cos(theta)+3*sin(theta)),theta=-0.45..2.42,12,fixe,normal,
  color=plum),k=-10..10):
> display(A,insequence=true,view=[-10..10,-10..10]);
```



Lorsque $c = 0$ dans l'équation $a\,x + b\,y + c = 0$, la droite passe par l'origine. Puisque de telles droites ne peuvent s'exprimer par la forme $r = -\dfrac{c}{a\cos(\theta) + b\sin(\theta)}$, il n'y a donc pas eu de tracé passsant par l'origine. C'est ce qui explique le clignotement dans l'exécution de l'animation précédente.

Les équations polaires de droites passant par l'origine sont de la forme $\theta = \textit{constante}$. Pour déterminer la pente

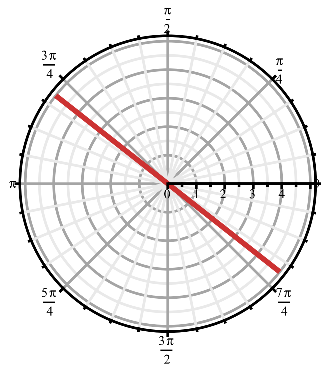
θ de la droite, il suffit de mettre en évidence l'inclinaison de la droite d'équation cartésienne $2x + 3y + 2 = 0$.

```
> isolate(Éq,y);
```

$$y = -\frac{2x}{3} - \frac{2}{3} \qquad \mathbf{(3.4)}$$

Alors, $\tan(\theta) = -\frac{2}{3}$. Utilisons alors la macro-commande `polarplot` de l'extension `plots`.

```
> polarplot([r,-2/3, r=-5..5],color=orange,thickness=2);
```

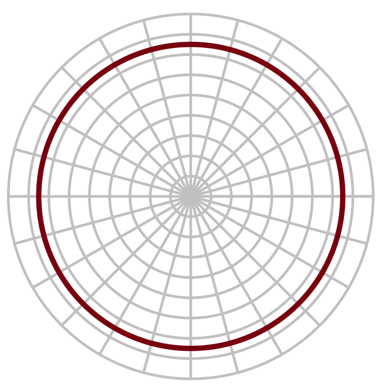


Cercles

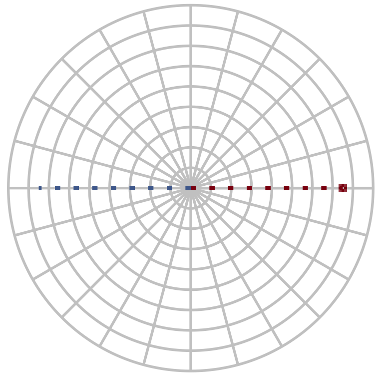
Les coordonnées polaires d'un cercle ont plusieurs formes. Cela dépend de la situation du centre:

$r = a$	centre: (0,0);	rayon: a
$r = 2a\cos(\theta)$	centre: (a,0);	rayon: a
$r = 2a\sin(\theta)$	centre: (0,a);	rayon: a

```
> app(r=2.5,theta=0..2*Pi,24,fixe,polaire);
```

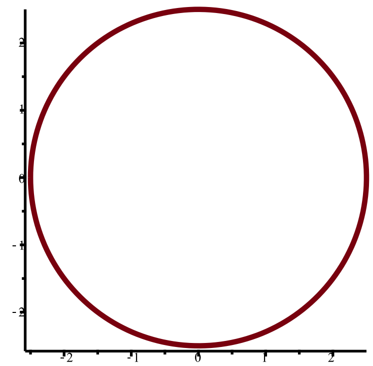


```
> app(r=2.5,theta=0..2*Pi,12,anime,polaire);
```



Remplaçons $r = 2.5$ par $r = -2.5$ pour constater que ces deux équations polaires représentent le même cercle.

```
> app(r=-2.5,theta=-2*Pi..0,12,fixe,frame);
```



REMARQUE: Lorsque l'on fait afficher aucun système de repère, on voit mieux le bras traceur du lieu géométrique. La partie orange de ce bras nous indique que la valeur de ***r*** est positive tandis dans la partie grise indique que la valeur de ***r*** est négative. Rappelons que si ***r*** < 0, les coordonnées polaires $[r, \theta]$ et $[-r, \theta + \pi]$ désigne le même point. La macro-commande app trace donc le point $[-r, \theta + \pi]$ si ***r*** < 0 et, dans ce cas, la partie grise du bras traceur nous rappelle que ***r*** est négatif.

Soit le cercle d'équation polaire $r = 2\sin(\theta)$. Ce cercle est entièrement tracé avec $\theta \in [0, \pi[$. Ici, $\boldsymbol{r} > \mathbf{0}$.

```
> app(r=2*sin(theta),theta=0..Pi,18,anime,none);
```


Ce cercle peut aussi être entièrement tracé avec $\theta \in [\pi, 2\pi[$. Dans ce cas-ci, $r < 0$.

```
> app(r=2*sin(theta),theta=Pi..2*Pi,18,anime,none);
```

Dans le cas du tracé avec $\theta \in [0, 2\pi[$, le **<u>cercle sera</u>** évidemment **<u>tracé deux fois</u>**: une fois avec $\boldsymbol{r} > 0$ et une fois avec $\boldsymbol{r} < 0$.

```
> app(r=2*sin(theta),theta=0..2*Pi,36,anime,none);
```

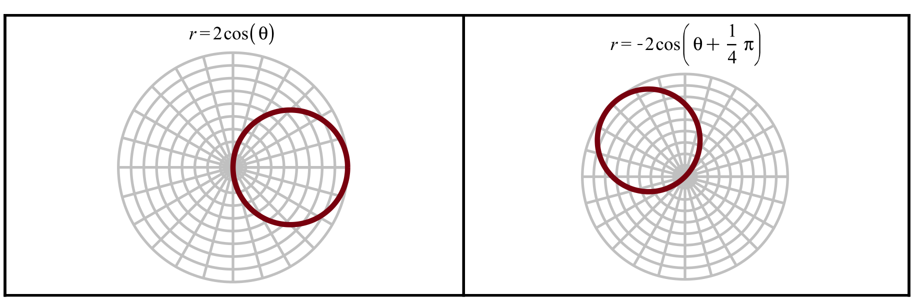

Rappelons que le graphique polaire de $r = \mathrm{f}(\theta - \alpha)$ est celui du graphique polaire de $r = \mathrm{f}(\theta)$ ayant subit une rotation d'un angle α

– si l'angle α est positif, le graphique initial subit une rotation dans le sens anti-horaire, et

– si l'angle α est négatif, la rotation est dans le sens horaire.

Considérons un angle positif $\alpha = \dfrac{3\pi}{4}$.

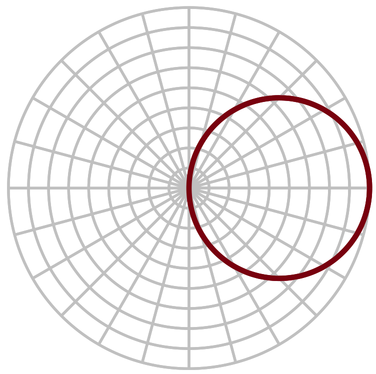
```
> A:=app(r=2*cos(theta),theta=0..Pi,18,fixe,polaire,title=typeset(r=2*cos
  (theta))):
  B:=app(r=2*cos(theta-3*Pi/4),theta=0..Pi,18,fixe,polaire,title=typeset(r=2*
  cos(theta-3*Pi/4))):
> display(Matrix(1,2,[A,B]));
```



Remarquer la simplification automatique de $r = 2\cos\left(\theta - \frac{3\pi}{4}\right)$ dans le titre de la figure de droite précédente.

Animons cet exemple de rotation.

```
> a:=0:
  b:=3*Pi/4:
  Sequences:=20:
  A:=app(r=2*cos(theta),theta=0..Pi,1,fixe,polaire):
  Pas:=seq(a+((b-a)/Sequences)*k,k=0..Sequences):
  B:=seq(app(r=2*cos(theta-k),theta=0..Pi,2,fixe,polaire),k=Pas):
  display(seq(display([A,B[k]]),k=1..nops([Pas])),insequence=true);
```



Dans les exemples ci-dessous, les animations comportent 18 séquences mais, on peut spécifier un nombre positif de séquences de son choix.

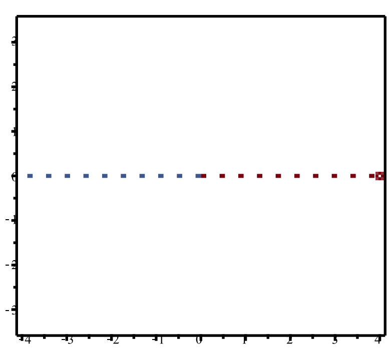
Cardioïdes et limaçons

Formes générales:
$r = a + b\cos(\theta)$ et $r = a + b\sin(\theta)$

En général:
|**a**| ≥ |**b**| **cardioïde**
|**a**| < |**b**| **limaçon**

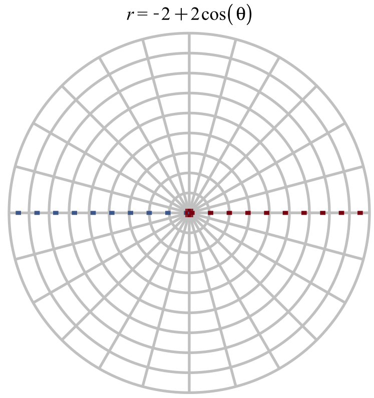
```
> app(r=2+2*cos(theta),theta=0..2*Pi,6,anime,boxed,title=typeset(r=2+2*cos
  (theta)));
```

$r = 2 + 2\cos(\theta)$

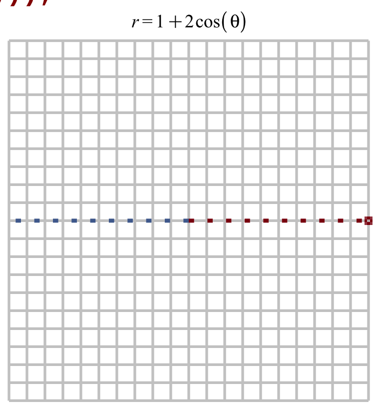


L'animation suivante trace à nouveau cette cardoïde mais montre une origine du tracé différente.

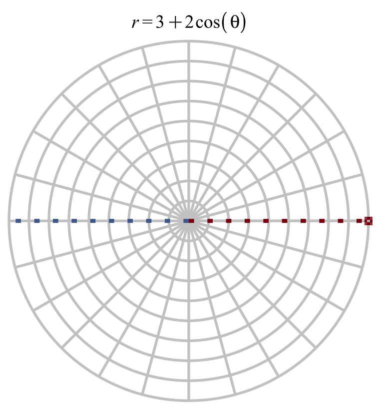
```
> app(r=-2+2*cos(theta),theta=0..2*Pi,18,anime,polaire,color="Niagara 3",
  title=typeset(r=-2+2*cos(theta)));
```




```
> app(r=1+2*cos(theta),theta=0..2*Pi,18,anime,cartésien,color="Navy",title=
  typeset(r=1+2*cos(theta)));
```



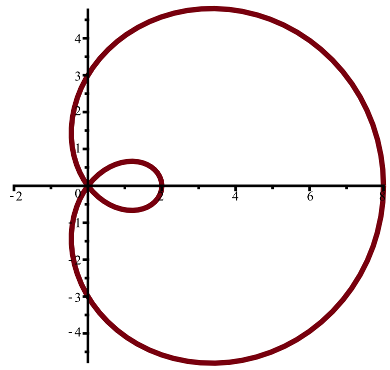

```
> app(r=3+2*cos(theta),theta=0..2*Pi,18,anime,polaire,title=typeset(r=3+2*cos
  (theta)));
```




```
> b:=1:
  a:=2+b:
  Sequences:=36:
  Pas_1:=[seq(a+((b-a)/Sequences)*k,k=0..Sequences)]:
  Pas_2:=[seq(a-((b-a)/Sequences)*k,k=0..Sequences)]:
  Critère:=(x,y)->evalb(x>y):
  Pas:=sort([op(Pas_1),op(2..nops(Pas_2),Pas_2)],Critère):
```

```
> A:=seq(app(r=a+k*cos(theta),theta=0..2*Pi,18,fixe,normal,numpoints=120),k=
  Pas):
```

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

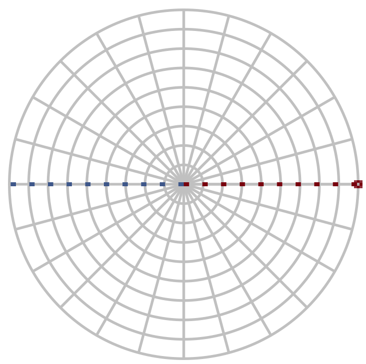


Rosaces

Forme générale: $r = a\cos(n\theta)$ et $r = a\sin(n\theta)$

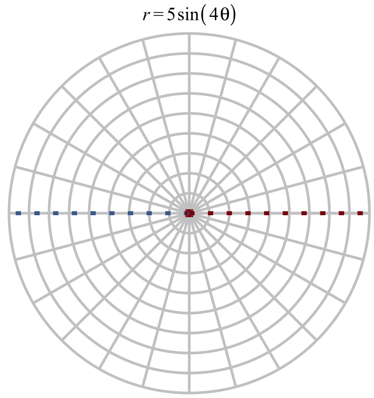
```
> app(r=4*cos(2*theta),theta=0..2*Pi,18,anime,polaire,color="Niagara
  13",title=typeset(r=4*cos(2*theta)));
```

$r=4\cos(2\theta)$

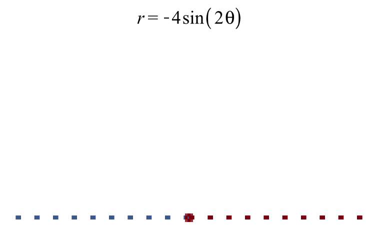


```
> app(r=5*sin(4*theta),theta=0..2*Pi,18,anime,polaire,color="Niagara 7",title=
  typeset(r=5*sin(4*theta)));
```

$r=5\sin(4\theta)$



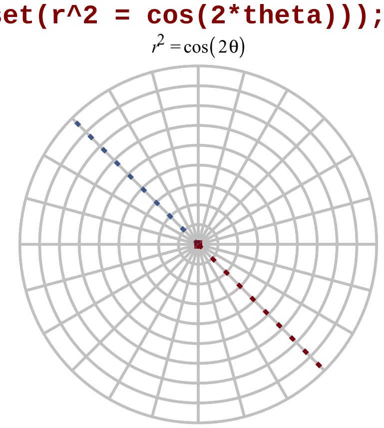
```
> app(r=4*cos(2*(theta+Pi/4)),theta=0..2*Pi,18,anime,none,color="Niagara
  4",title=typeset(r=4*cos(2*(theta+Pi/4))))
```



Lemniscate

Forme générale: $r^2 = \cos(2\theta)$. Il est nécessaire, pour cette équation, d'effectuer la superposition de deux tracés obtenus séparément (r^2 oblige).

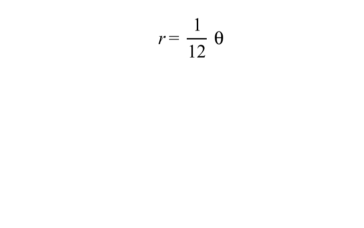
```
> A:=app(r=cos(2*theta),theta=-Pi/4..Pi/4,18,anime,polaire):
  B:=app(r=-cos(2*theta),theta=-Pi/4..Pi/4,18,anime,polaire):
> display(A,B,title=typeset(r^2 = cos(2*theta)));
```



Courbes diverses

Spirale équiangulaire

```
> app(r=theta/12,theta=0..18*Pi,18,anime,none,thickness=2,title=typeset(r=
  theta/12));
```



Spirale logarithmique

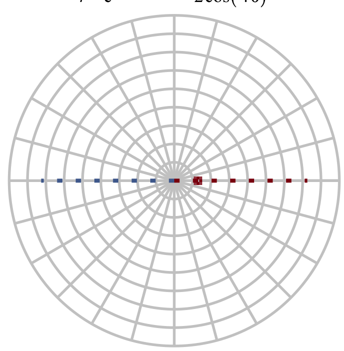
```
> app(r=4*exp(-theta/12),theta=0..18*Pi,18,anime,none,thickness=2,title=
  typeset(r=4*exp(-theta/12)));
```

$r=4\mathrm{e}^{-\frac{1}{12}\theta}$

Papillon

```
> app(r=exp(cos(theta))-2*cos(4*theta),theta=0..2*Pi,18,anime,polaire,title=
  typeset(r=exp(cos(theta))-2*cos(4*theta)));
```

$r=\mathrm{e}^{\cos(\theta)}-2\cos(4\theta)$



Rotation dans le sens anti-horaire d'un angle de $\frac{\pi}{2}$ radian.

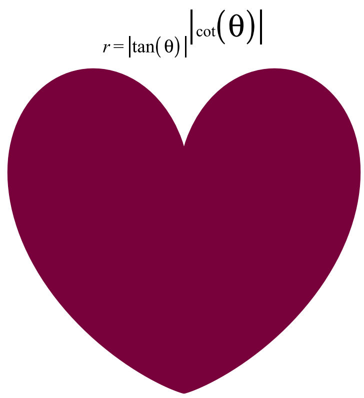
```
> app(r=exp(cos(theta-Pi/2))-2*cos(4*(theta-Pi/2)),theta=-Pi/2..3*Pi/2,18,
  anime,none,color="Niagara 9",title=typeset(r=exp(cos(theta-Pi/2))-2*cos(4*
  (theta-Pi/2))));
```

$r=\mathrm{e}^{\sin(\theta)}-2\cos(4\theta)$

Autres courbes intéressantes

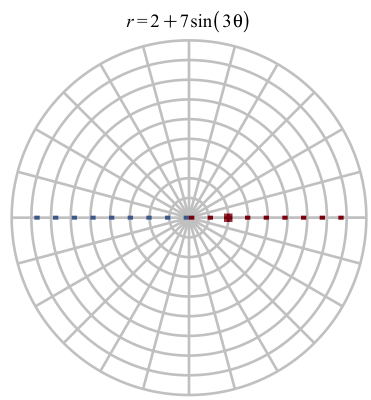
Le coeur de Dwight Baddorf (2008) $:r=\left|\tan(\theta)\right|^{\left|\cot(\theta)\right|}$

```
> app(r=abs(tan(theta))^abs(cot(theta)),theta=0..Pi,24,fixe,none,color=
  "Niagara 12",filled=[color="Niagara 12"],
       title=typeset(r=abs(tan(theta))^abs(cot(theta))),
       view=[-0.8..0.8,0..1.4]);
```



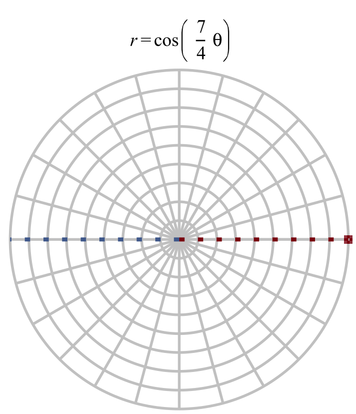
$r=2+7\sin(3\,\theta)$

```
> app(r=2+7*sin(3*theta),theta=0..2*Pi,18,anime,polaire,title=typeset(r=2+7*
  sin(3*theta)));
```



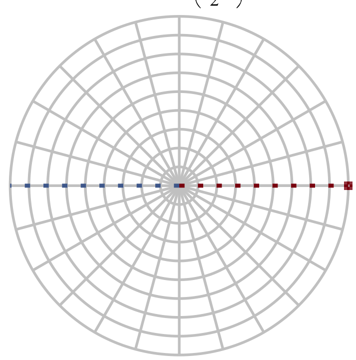
$r=\cos\left(\frac{7\,\theta}{4}\right)$

```
> app(r=cos(7*theta/4),theta=0..8*Pi,36,anime,polaire,title=typeset(r=cos(7*
  theta/4)));
```



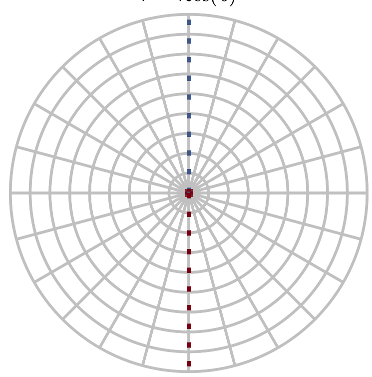
$r = 1 + \cos\left(\frac{\theta}{2}\right)$

```
> app(r=1+cos(theta/2),theta=0..4*Pi,18,anime,polaire,title=typeset(r=1+cos
  (theta/2)));
```



$r^2 = 4\cos(\theta)$

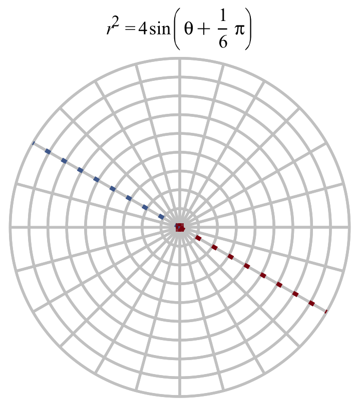
```
> A:=app(r=sqrt(4*cos(theta)),theta=-Pi/2..Pi/2,18,anime,polaire):
  B:=app(r=-sqrt(4*cos(theta)),theta=-Pi/2..Pi/2,18,anime,polaire):
  display(A,B,title=typeset(r^2=4*cos(theta)));
```



Rotation dans le sens anti-horaire d'un angle de $\frac{\pi}{3}$ radian.

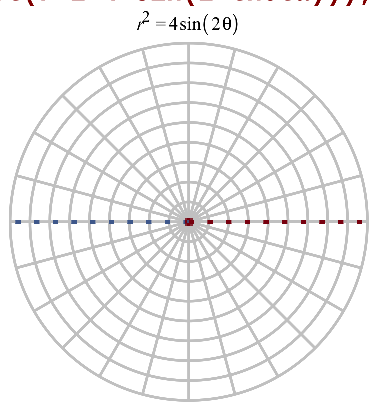
$r^2 = 4\cos\left(\theta - \frac{\pi}{3}\right)$

```
> A:=app(r=sqrt(4*cos(theta-Pi/3)),theta=-Pi/6..5*Pi/6,18,anime,polaire):
  B:=app(r=-sqrt(4*cos(theta-Pi/3)),theta=-Pi/6..5*Pi/6,18,anime,polaire):
  plots[display](A,B,title=typeset(r^2=4*cos(theta-Pi/3)));
```



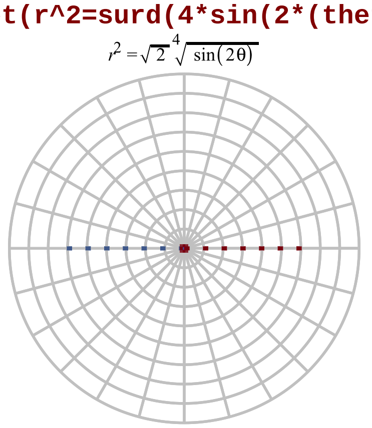
$r^2 = 4\sin(2\theta)$

```
> A:=app(r=sqrt(4*sin(2*theta)),theta=0..Pi/2,18,anime,polaire):
  B:=app(r=-sqrt(4*sin(2*theta)),theta=0..Pi/2,18,anime,polaire):
  display(A,B,title=typeset(r^2=4*sin(2*theta)));
```



$r^4 = 4\sin(2\theta)$

```
> A:=app(r=surd(4*sin(2*(theta)),4),theta=0..Pi/2,18,anime,polaire):
  B:=app(r=-surd(4*sin(2*(theta)),4),theta=0..Pi/2,18,anime,polaire):
  display(A,B,title=typeset(r^2=surd(4*sin(2*(theta)),4)));
```



Rotation dans le sens horaire d'un angle de $\frac{\pi}{4}$ radian.

$$r^4 = 4\sin\left(2\left(\theta+\frac{\pi}{4}\right)\right)$$

```
> A:=app(r=surd(4*sin(2*(theta+Pi/4)),4),theta=-Pi/4..Pi/4,18,anime,polaire):
  B:=app(r=-surd(4*sin(2*(theta+Pi/4)),4),theta=-Pi/4..Pi/4,18,anime,polaire):
  display(A,B,title=typeset(r^2=surd(4*sin(2*(theta+Pi/4)),4)));
```

$r^2 = \sqrt{2}\,\sqrt[4]{\cos(2\theta)}$