# File la laine, filent les jours…

# Filent les routes… et filent les tortues !

## ou comment rencontrer une clothoïde

*Alain MASCRET, collège la Champagne, Gevrey-Chambertin*

*Résumé :* La clothoïde est une solution trouvée pour raccorder des lignes de chemin de fer ou des portions de route afin d'éviter des problèmes d'accélération ou de décélération brutale dans les virages.
Les élèves d'un club informatique de collège rencontrent par hasard une clothoïde et d'autres courbes apparentées. Ils cherchent à expliquer les phénomènes observés.

*Mots clés :* Clothoïde ; raccordement ; langage *logo* ; représentation des courbes en informatique ; motivation à la recherche.

Le 25 novembre dernier, Xavier Lefort, professeur à l'IUT de Saint Nazaire est venu à l'IREM parler de la façon de dessiner les virages des voies de chemin de fer. Le principal problème qui se pose, en supposant la voie sur un plan horizontal, provient de ce que l'accélération normale d'un mobile est proportionnelle à la courbure de sa trajectoire.

Essayons de raccorder deux tronçons rectilignes par un arc de cercle (Figure 1). La courbure est nulle sur la première droite, passe brutalement à la courbure du cercle au point de raccordement et redevient, tout aussi brutalement, nulle au deuxième point de raccordement. Il s'en suit des désagréments pour le voyageur, pouvant aller, si la vitesse est suffisante, jusqu'au déraillement du train, l'accélération normale d'un mobile étant également proportionnelle au carré de sa vitesse.

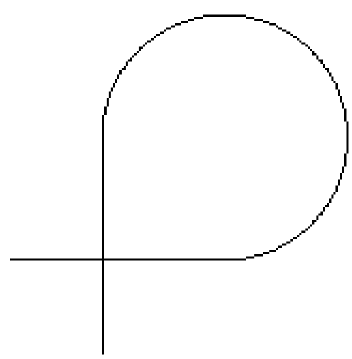


Figure 1

Il faut donc faire varier la courbure de la trajectoire de façon régulière. Une courbe dont la courbure est proportionnelle à l'abscisse curviligne ferait bien l'affaire. Cette courbe existe, c'est la clothoïde.

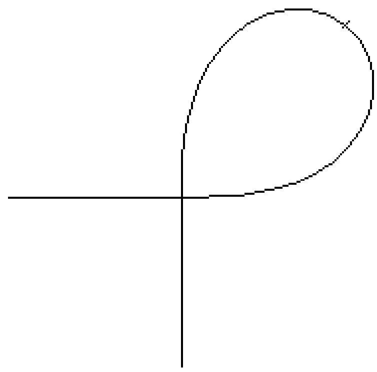


Figure 2

À titre d'exemple voici, figure 2, un virage à 270°, réalisé avec deux arcs de clothoïde. Ce tracé peut être utilisé dans un échangeur d'autoroute, le problème étant le même que pour les voies de chemin de fer.

Dans la pratique, seuls les arcs de clothoïde sont utilisés mais, bien sûr, il est intéressant de voir à quoi ressemble la courbe complète.

En voici (figure 3) une représentation.

Elle possède un centre de symétrie qui est aussi son point d'inflexion. C'est en ce point que se font les raccordements puisqu'en ce point la courbure est nulle. En s'éloignant de ce point, la courbure augmente et tend vers l'infini. La courbe s'enroule indéfiniment autour de deux points asymptotes ce qui fait ressembler ses extrémités à deux pelotes de laine.

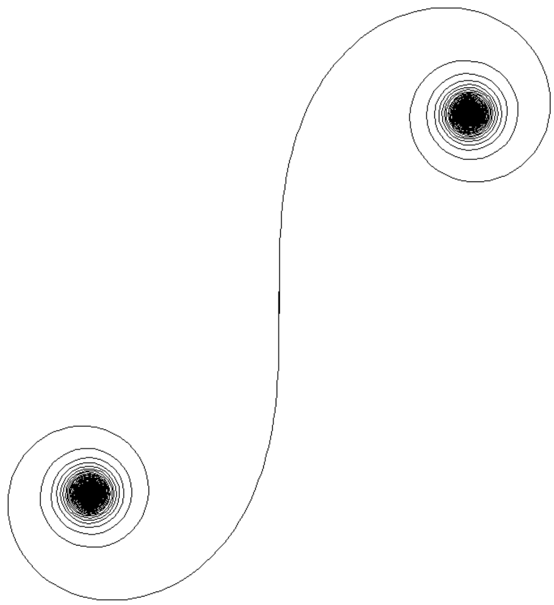


Figure 3

C'est de là que provient son nom. En effet, Clotho, *la fileuse*, était l'une des trois Moires de la mythologie grecque. Son rôle consistait à filer la vie des humains symbolisée par un fil. Elle présidait à leur naissance tandis que sa sœur Lachesis, *la destinée*, leur « dispensait les biens et les maux ». Quant à son autre sœur Atropos, *l'implacable*, elle coupait le fil le moment venu...

La conférence de Xavier Lefort m'a rappelé ma première rencontre avec une clothoïde et c'est ce dont je vais vous parler maintenant. Remontons le fil du temps, environ vingt ans en arrière. Nous sommes en club informatique. Les élèves utilisent le langage *logo* et veulent tracer un cercle.

Tout d'abord, présentons le langage *logo* en quelques mots.

Le *logo* permet, entre autre chose, de dessiner sur l'écran de l'ordinateur en pilotant une « tortue téléguidée » représentée par un petit triangle. La tortue est munie d'un crayon qu'on peut baisser ou lever. Si le crayon est levé, la tortue se déplace sans tracer.

Voici un minimum d'instructions de base pour comprendre la suite :

( *:n* désigne un nombre. Les deux points l'indiquent à *logo* )

| | |
|---|---|
| av *:n* | La tortue avance de *:n* pas ( *:n* pixels ). |
| re *:n* | La tortue recule de *:n* pas. |
| tg *:n* | La tortue tourne à gauche de *:n* degrés. |
| td *:n* | La tortue tourne à droite de *:n* degrés. |
| lc | La tortue lève son crayon. |
| bc | La tortue baisse son crayon. |

Programmer en *logo,* consiste à étendre le nombre d'instructions compréhensibles par la machine en utilisant l'instruction « pour ». En général, dès la première séance, un élève de sixième est capable de faire dessiner un rectangle à la tortue.

```
Pour rectangle
    av 150   tg 90
    av 50    tg 90    # Le dièse permet de commenter les
    av 150   tg 90    # instructions si nécessaire.
    av 50    tg 90    # Cette dernière instruction n'est pas
                      indispensable mais replace la tortue
                      dans sa position de départ.
fin
```

Au club informatique de mon collège, les élèves voulaient pouvoir dessiner des cercles. Pour cela, l'idée est de faire avancer « un peu » la tortue puis de la faire tourner « un peu » et de recommencer. Ce qui nous donne :

```
Pour cercle1
    av 1         # La tortue avance de 1 pas.
    tg 1         # La tortue tourne à gauche de 1°.
    cercle1      # La tortue refait la même chose.
fin
```

Il est tout à fait possible qu'une procédure s'appelle elle-même. Le langage *logo* accepte la récursivité. Les jeunes élèves aussi, à condition de ne pas leur dire que c'est difficile ! .... Bien sûr, il faut apprendre à l'utiliser. La procédure cercle1 va boucler indéfiniment car elle n'a pas de test d'arrêt. Il faudra l'arrêter « à la main ».

À partir de cette idée, en utilisant des variables et en introduisant un test d'arrêt, les élèves vont tenter d'améliorer cette procédure qui deviendra par exemple :

```
Pour cercle2 :d :a :c      #:d, :a et:c sont des variables locales
  si :c > 0                # :c sert de compteur
     [av :d                # pour le test d'arrêt.
      td :a                # :c est décrémenté à chaque
      cercle2 :d :a :c – 1] # appel de la procédure cercle2
fin                        # qui se termine quand :c = 0.
```

Ce travail d'amélioration va se faire à travers de nombreux essais, chacun cherchant de son côté et comparant ses résultats à ceux de ses camarades. Toujours est-il que sur l'un des écrans, au grand étonnement des élèves et de leur professeur apparaît une clothoïde ! Que s'est-il passé ?

La question intéresse beaucoup les élèves qui trouvent cette spirale bizarre plus esthétique qu'un banal cercle et voudraient bien la dessiner sur leur écran. L'erreur est vite repérée, non pour la corriger, mais pour la reproduire ! Au lieu de conserver à la variable *:a* sa valeur quand il rappelle récursivement la procédure, l'élève lui a ajouté 1. Autrement dit, l'avant-dernière ligne de la procédure est devenue :

cercle2 *:d :a* + ***1*** *:c – 1* ]

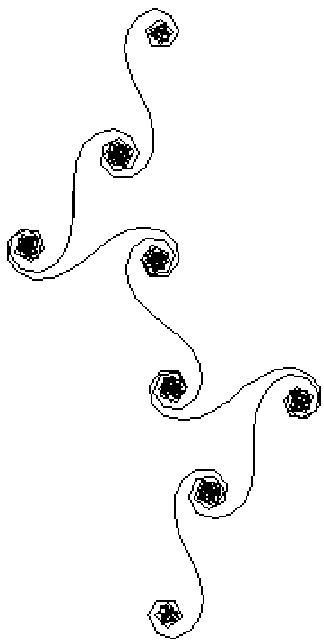


Figure 4 (spirale 10 1 7 720)

À chaque appel récursif de la procédure, la tortue tourne d'un angle plus grand, donc son virage est plus serré. C'est un peu comme si le « cercle » changeait à chaque étape, son rayon devenant de plus en plus petit. Plus étonnant encore, la tortue revient sur ses pas et passe indéfiniment d'une « pelote » à l'autre si la procédure n'a pas de test d'arrêt. C'est beaucoup plus motivant à dessiner qu'un cercle !

Aussi chacun se met-il à programmer sa spirale, en lui apportant sa petite touche personnelle, c'est-à-dire en ajoutant à la variable *:a* un autre nombre que 1. Si nous avons bien compris, nous allons obtenir des spirales plus ou moins grandes ou plus ou moins serrées. Mais ce n'est pas du tout ce qui apparaît ! Par exemple, en ajoutant 7, le dessin obtenu est celui de la figure 4, qui comporte 8 « pelotes » !

Aussitôt les élèves se mettent à la recherche de figures toutes plus étonnantes les unes que les autres. Pour faciliter cette recherche, nous ajoutons une nouvelle variable *:k* correspondant à la valeur ajoutée à l'angle *:a* à chaque appel de la procédure.

Celle-ci devient :

```
Pour spirale :d :a :k :c
      si :c > 0
         [av :d
         td :a
         spirale :d :a +:k :k :c – 1]
fin
```

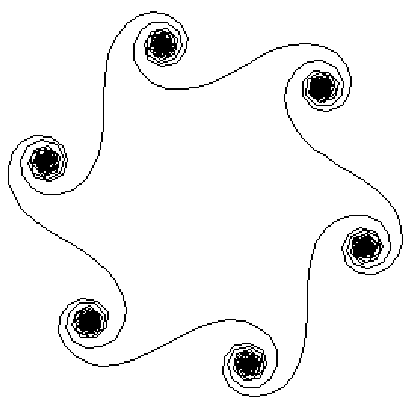


Figure 5 (spirale 10 1 3 720)

Il est possible de classer les formes qui apparaissent sur les écrans. Nous retrouvons, bien sûr, les cercles pour $:k = 0$ et les clothoïdes pour $:k = 1$. Parfois le motif est cyclique, comme sur la figure 5 ou périodique comme sur la figure 6. Nous comprenons bien que ces différentes formes dépendent des variables *:a* et *:k*, mais la façon dont elles en dépendent nous paraît mystérieuse. Il va falloir examiner de plus près ce qui se passe et reprendre notre étude depuis le début.

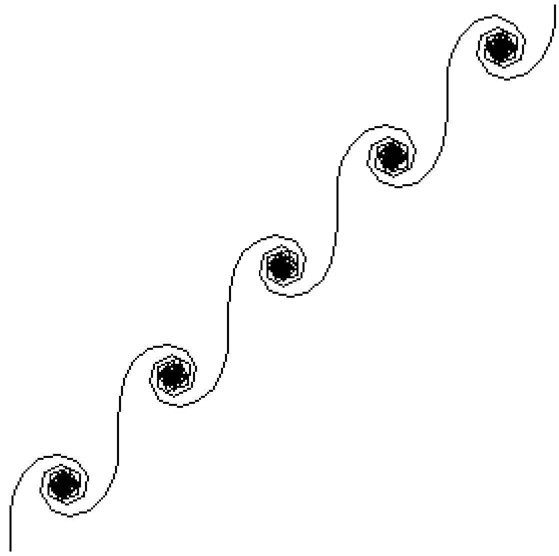


Figure 6 (spirale 10 3 6 720)

Repartons de notre procédure cercle2 et examinons ce qu'elle trace vraiment. L'appel cercle2 1 1 360, par exemple, demande à la tortue de faire un pas et de tourner à droite et de répéter cette opération 360 fois. La tortue a donc tracé, non pas un cercle, mais un polygone de 360 côtés. La preuve en image (figure 7) : cercle2 100 72 5 trace un polygone régulier de 5 côtés de longueur 100. La tortue revient à sa position de départ après avoir tourné de 5 × 72°.

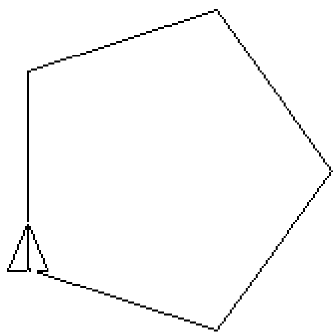


Figure 7 (cercle2 100 72 5)

Ceci n'est pas spécial à *logo*, tous les logiciels de dessins sur ordinateur font de même car les « points » des écrans sont des pixels et sont en nombre fini. S'ils sont assez nombreux, notre œil est trompé. Sur les figures qui illustrent cet article les pixels sont nettement visibles.

Revenons à notre clothoïde, elle possède deux points asymptotes, les centres des « pelotes ». Ce qui veut dire qu'elle se rapproche indéfiniment de ces points sans les atteindre. En ce qui concerne notre représentation, la tortue fait des pas de longueur constante quelle que soit sa proximité des points asymptotes. Finalement ce que trace la tortue n'est pas plus une clothoïde qu'un polygone régulier n'est un cercle ! Tant que la tortue est suffisamment loin des points asymptotes, notre représentation est acceptable, comme celle du cercle. Mais auprès de ces points c'est différent. D'ailleurs sur la figure 2, on voit seulement deux taches noires qui ne nous renseignent guère sur le comportement de la tortue en ces points.

Pour mieux voir, faisons comme pour le cercle, augmentons la longueur du segment tracé et la valeur de l'angle *:a* dont tourne la tortue à chaque appel récursif comme sur la figure 8 qui correspond à l'appel : spirale 50 0 15 48. La pelote se réduit sur cette figure à quelques segments.

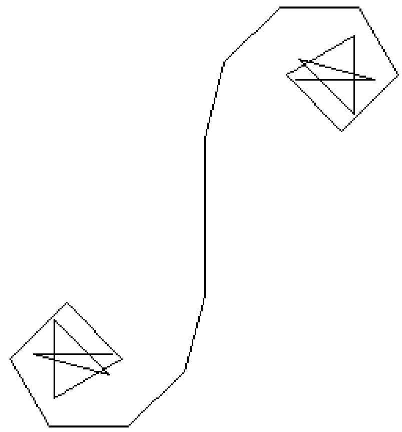


figure 8 (spirale 50 0 15 48)

Au douzième appel récursif, la tortue s'apprête à tracer un trait horizontal (figure 9). À la douzième étape, l'angle *:a* vaut 12 × 15° = 180°. Au treizième appel, le trait horizontal sera tracé et la tortue va faire demi-tour (figure 10). Elle va donc maintenant parcourir à nouveau les segments qu'elle a tracés, mais dans l'autre sens car tourner à droite de *:a*° revient à tourner à gauche de (180 - *:a*)°.

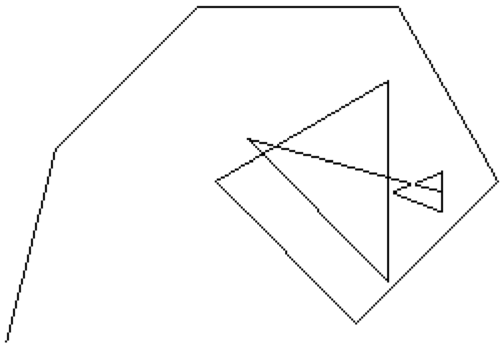


Figure 9 (spirale 100 0 15 12)

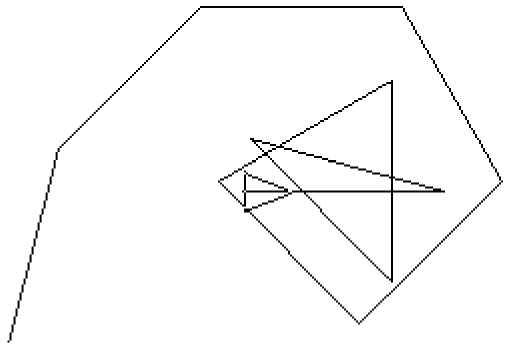


Figure 10 (spirale 100 0 15 13)

Bien entendu, si l'angle *:a* n'atteint jamais un multiple de 180°, d'autres phénomènes se produiront.

La figure 11 présente un motif périodique obtenu par l'appel : spirale 50 10 20 54. La tortue commence par tourner de 10°, puis à chaque étape de 20° de plus. À la neuvième étape, elle tourne à droite de :

10° + 8 × 20° = 170°

et à la dixième étape de :

10° + 9 × 20° = 190°,

c'est-à-dire de 170° à gauche. Elle va donc tracer un chemin symétrique de celui qu'elle a parcouru, par rapport au milieu du dixième segment.

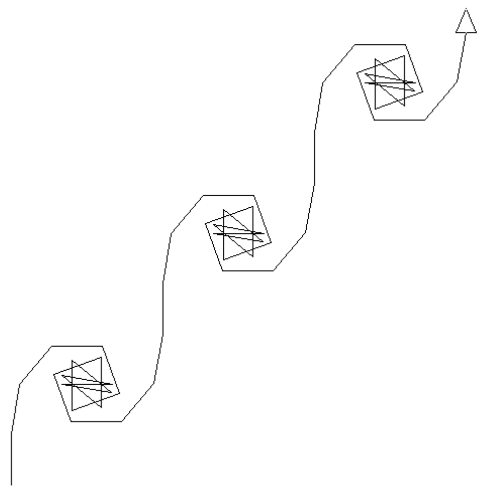


Figure 11 (spirale 50 10 20 54)

À la dix-neuvième étape, elle passera par le symétrique de son point de départ et tout pourra recommencer comme le montre la figure 12. Le tracé sera donc périodique. Si la tortue ne s'arrêtait pas, elle dessinerait une frise. En effet, la figure comporte deux centres de symétrie distincts : les milieux des dixième et dix-neuvième segments. (Pour avoir la frise complète, il suffit de faire reculer la tortue à partir de son point de départ, le milieu du premier segment est aussi un centre de symétrie).

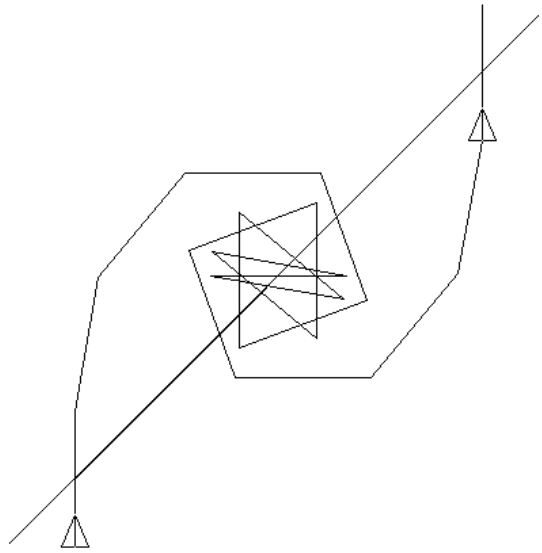


Figure 12 (spirale 100 10 20 19)

Examinons maintenant le cas d'un tracé cyclique comme celui de la figure 13 correspondant à l'appel : spirale 50 10 30 72. Après avoir dessiné six pelotes, la tortue revient à sa position de départ et tout recommence.

À la $n^{\text{ième}}$ étape, la tortue tourne de $10n+30(n\text{-}1)$ modulo 360. Depuis son départ, elle a tourné de $10n+30\dfrac{(n-1)n}{2}$ modulo 360. Ce dernier nombre est appelé cap de la tortue par *logo* et peut lui être demandé par la commande « cap ».

À la sixième étape la tortue tourne de 160° et à la septième étape de 190°, c'est-à-dire qu'elle commence à dérouler la première pelote, puisque *:a* vient de dépasser pour la première fois 180°.

À la douzième étape elle tourne de 340° et à la treizième étape de 370°, c'est-à-dire que *:a* dépasse 360° pour la première fois. Elle se trouve alors à mi-chemin entre deux pelotes. Son cap est de 300°, ce qui revient à dire qu'elle a tourné de 60° vers la gauche.

En répétant six fois ce processus, la tortue va arriver à sa soixante-douzième étape avec un cap de 0° et prête à tourner de 10°, comme au départ.

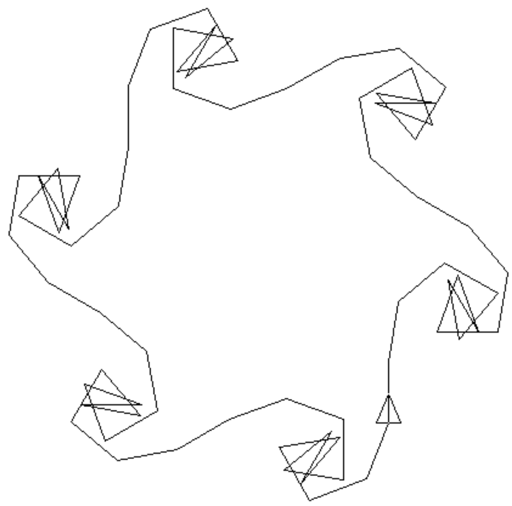


Figure 13 (spirale 50 10 30 72)

Il est possible d'expliquer de la même façon les motifs étranges comme ceux de la figure 3. Toutes les fois que *:a* dépasse 180° ou un multiple impair de 180°, la tortue commence à sortir de la pelote qu'elle était en train de tracer. Toutes les fois que *:a* dépasse un multiple de 360°, la tortue se trouve à mi-chemin entre deux pelotes. La tortue revient sur ses pas quand *:a* est un multiple impair de 180°.

Les élèves du club informatique n'ont pas forcément trouvé ce que je viens d'exposer, mais je peux dire qu'ils ont cherché, parce que les dessins obtenus leur posaient un problème concret. Il fallait tenter d'expliquer pourquoi les formes étaient si différentes. Ils ont procédé expérimentalement, par essai et erreur. J'entendais des phrases du genre : « Si j'ai bien compris je devrais avoir 6 pelotes.... » et peu de temps après « Et bien non ! Je n'en ai que trois ! ». L'essentiel pour moi est qu'ils aient cherché et surtout qu'ils aient ressenti le besoin de chercher une explication. D'autre part, il est assez facile de modifier les procédures pour qu'en plus du dessin, elles nous donnent des informations sur la position et le cap de la tortue. De cette façon le langage *logo* aide les élèves à résoudre les problèmes qu'ils se posent.

Si vous voulez, vous aussi, voir se créer sous vos yeux les figures de cet article et faire, comme mes élèves du club informatique, vos propres expériences, je vous invite à venir sur le site geowiki.u-bourgogne.fr. La page correspondante sera prête début mai 2011.

Examinons maintenant comment varie la courbure de la clothoïde. À chaque étape, il est possible de demander à la tortue de tracer le cercle qui a la même courbure et qui lui est tangent. Ce cercle s'appelle le cercle osculateur. Bien entendu, la tortue ne trace que des approximations de la clothoïde et de ses cercles osculateurs.

Comme d'habitude, pour mieux voir ce qui se passe, il faut lui faire tracer des segments assez longs et choisir un angle *:a* assez grand. Sur la figure 14, *:a* vaut, au départ, 10° et tous les segments tracés ont une longueur de 100 pas de tortue.

Pour obtenir un cercle de même courbure que la clothoïde, je fais tourner la tortue de l'angle *:a* après chaque tracé d'un segment, jusqu'à ce qu'elle ait fait un tour complet. Comme *:a* augmente à chaque étape, le nombre de segments diminue et mes cercles sont de moins en moins beaux. De cette façon, la figure obtenue est assez grossière mais l'approximation du cercle et celle de la clothoïde sont les mêmes.

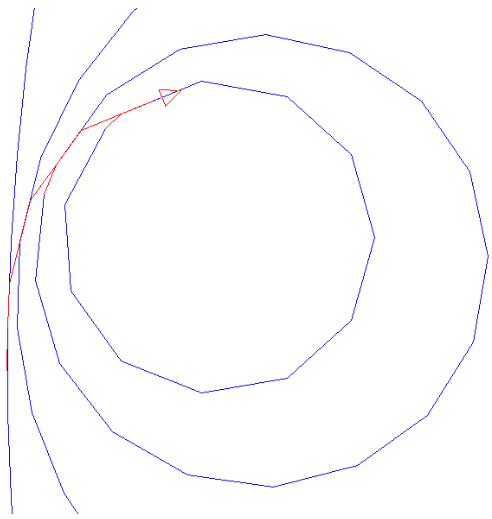


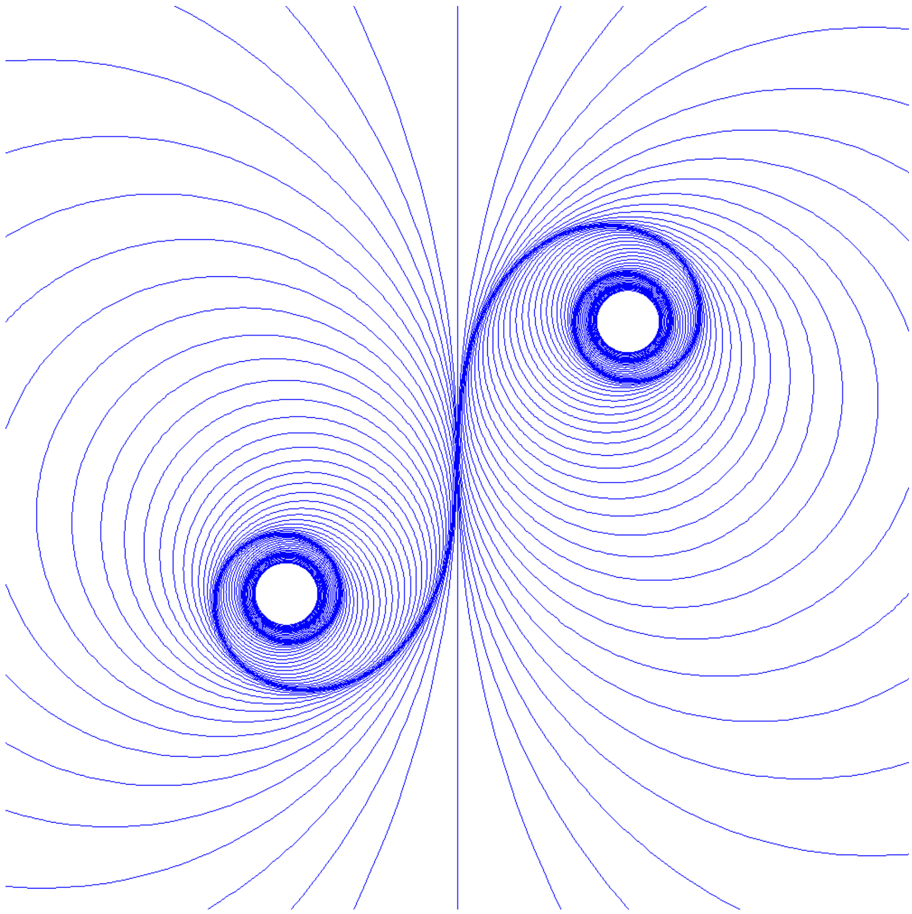
Figure 14

En se limitant à quatre cercles comme sur la figure 14, il est possible d'observer quelques propriétés des cercles osculateurs : ils sont tous deux à deux disjoints et la courbe les traverse au point de contact puisque ce contact est d'ordre trois comme pour la tangente en un point d'inflexion.

Pour rendre bien visible ce phénomène, j'ai fait en sorte que les cercles soient « tangents » au milieu de chaque segment de la « courbe ». La tortue avance de 50 pas, trace le cercle et termine ensuite le segment. Le segment de la « courbe » a donc une partie commune avec le « segment » du cercle, ce qui exprime, à la manière de *logo* le fait que le « cercle » est tangent à la « clothoïde ». Mais ces deux segments sont décalés. En conséquence, avant le point de contact, la courbe est à l'extérieur du cercle, tandis qu'après elle est à l'intérieur.

Pour améliorer la qualité de la figure, j'ai demandé à la tortue de tourner seulement de un demi-degré et d'adapter le segment qu'elle trace pour que la longueur du cercle tracé soit la même qu'avec la méthode précédente. De cette façon, le rayon du cercle est conservé mais son tracé est bien meilleur.

Le résultat devient nettement plus esthétique. Sur la couverture de ce numéro de la Feuille de Vigne, ainsi que sur la figure 15, la clothoïde n'est même pas tracée. Elle est cependant visible comme enveloppe de ses cercles osculateurs. J'ai arrêté la tortue bien avant les points asymptotes pour éviter les taches, encore plus gênantes que sur la figure 3.



N'oublions pas cependant qu'il ne s'agit, comme toujours, que d'une approximation. D'ailleurs, les cercles ne sont pas totalement lisses, ça se voit encore. Et même s'ils semblaient parfaits, il ne s'agirait que d'une meilleure approximation. Les logiciels de géométrie nous aident, mais nous devons essayer de comprendre leur façon de fonctionner. Sinon, nous risquons facilement de confondre le modèle informatique et la réalité.

## *Bibliographie - Sitographie :*


Sur le langage logo, les livres de son concepteur Seymour Papert , en particulier :

- Jaillissement de l'esprit. Ordinateurs et apprentissage, Flammarion, 1981.
- L'enfant et la machine à connaître. Repenser l'école à l'ère de l'ordinateur, Dunod, 1994.
- Harold Abelson et Andrea diSessa : Turtle geometry : The computer as Medium for Exploring Mathematics, The MIT Press Series in Artificial Intelligence, Cambridge, Massachusett ; London, England, 1981 (À ma connaissance, toujours non traduit en français).

Sur la géométrie :

- Marcel Berger : Géométrie vivante ou L'échelle de Jacob, Cassini, 2009. (Vous y trouverez une très jolie figure analogue à la figure 15, page 316. Elle est due à Étienne Ghys. Le livre parle de beaucoup d'autres choses de façon très intéressante).

Sur internet :

Les différentes versions actuelles du langage *logo*

Sur le site du groupe « logiciels de géométrie » de l'IREM de Dijon vous trouverez trois versions de ce langage, toutes libres et téléchargeables gratuitement (geowiki.u-bourgogne.fr ). Personnellement, j'utilise avec mes élèves *Xlogo* qui est écrit en *java*. Il est donc possible d'une part de s'en servir sans rien installer sur son ordinateur, d'autre part de l'intégrer à un site. Toutes les figures de cet article ont été programmées dans ce langage (sans utiliser la primitive cercle que les vieux logos n'avaient pas). Son auteur est Loïc Le Coq qui en assure aussi la maintenance et a écrit un manuel de référence en ligne, lisible et muni d'un index bien pratique.

De nombreux sites vous proposent de vous guider dans l'apprentissage de *logo*. Je me contenterai de vous inviter à venir sur geowiki.u-bourgogne.fr et de poser vos questions aux membres du groupe. Il suffit pour cela de vous enregistrer.

Sur la clothoïde :

Vous verrez une animation montrant l'accélération normale de deux mobiles dans le cas d'un raccord droite-cercle et d'un raccord droite-clothoïde sur la page :
http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/meca/clotho.html
Le site contient de nombreuses animations illustrant des lois physiques.

Pour les propriétés mathématiques voir :
http://www.mathcurve.com/courbes2d/cornu/cornu.shtml
Le site mathcurve est une véritable encyclopédie des courbes ! Il est dû à Robert Ferreol. La clothoïde est aussi appelée spirale de Cornu en mémoire du physicien français Alfred Cornu qui s'en servait pour calculer les intégrales de Fresnel. (Je précise, car un de mes élèves pensait que le nom de cette courbe provenait de sa forme qui rappelle une cornue !).