



Polygones emboîtés

© Pierre Lantagne

Enseignant retraité du Collège de Maisonneuve

La procédure Polygones_emboîtés a été écrite en 2005 pour illustrer l'énoncé d'un problème sur les séries géométriques pour mes élèves au volet enrichissement Maple du programme Sciences de la nature. Je pense que c'est un exemple intéressant de programmation Maple pour tous ceux et celles désirant avoir une plus grande maîtrise de Maple.

Bonne lecture à tous !

* Ce document Maple est exécutable avec la version 2020.2

Initialisation

```
> restart;  
> with(plots,display):  
with(plottools,rotate):
```

Polygones emboîtés

Le résultat de cette procédure est la construction d'une suite de polygones réguliers emboîtés tels que les sommets de chaque polygone de la suite sont sur les arêtes du polygone précédents décalés d'une fraction de la longueur de l'arête du polygone précédent.

Procédure Poly_Emboîtés

Poly_Emboîtés(Nombre_de_côtés, Nombres_d'emboîtements, Décalage, Mode)

La macro-commande Poly_Emboîtés construit une suite de polygones réguliers emboîtés. Le polygone P_{k+1} est construit à partir du polygone P_k . Chacun des sommets du polygone P_{k+1} sont sur les arêtes du polygone P_k situés à une fraction de la distance de chaque sommet du polygone P_k .

- **Nombres_de_côtés** est un nombre entier supérieur à 2. C'est le nombre de côtés du polygone régulier initial;
- **Nombres_d'emboîtements** est un nombre entier supérieur ou égal à 0. C'est le nombre de polygones semblables inscrits;
- **Décalage** est un nombre réel de l'intervalle $[0, 1]$;
- **Mode** est le type d'affichage des polygones: fixe ou anime.

Positionner le curseur clignotant sur une ligne quelconque du bloc de requêtes ci-dessous et pressez la touche Entrée.

```
> Poly_Emboîtés:=proc(Nbc,Nbe,Décal,Mode)  
# -----
```

```

# Création des variables locales
local Centre,Couleur,Couleur1,Couleur2,Décalage,j,k,Nbr_Côtés,
Nbr_Emboîtements,Options,Points,S,X0,X1,Y0,Y1:

# Validation des arguments de la procédure
if evalf(Nbc)<3 or not type(Nbc,posint) then ERROR(`Vous devez
donner un nombre entier de côté égale à trois au minimum`) fi;
if evalf(Nbe)<0 or not type(Nbc,posint) then ERROR(`Vous devez
donner un nombre entier d'emboîtements égale à 0 au minimum`) fi;
if evalf(Décal)<0 or evalf(Décal)>1 then ERROR(`Vous devez donner
une valeur de décalage comprise entre 0 et 1 inclusivement`) fi;
# -----

# Options d'affichage
Options:=scaling=constrained,
  title=`Polygones emboîtés`,
  titlefont=[TIMES,ITALIC,14],
  axesfont=[TIMES,ITALIC,8],args[5..nargs];
# -----

# Initialisation des variables
Nbr_Côtés:=Nbc:
Nbr_Emboîtements:=Nbe+1:
Décalage:=evalf(Décal):
Centre:=[10,10]:
X0:=10:
Y0:=0:
S:=evalf(2*Pi/Nbr_Côtés):
# -----
# Choix des couleurs par défaut
Couleur1:=color="MapleV white";
Couleur2:=color="MapleV aquamarine";
# -----
# Choix des couleurs utilisateurs
if hasoption([Options],'color') or hasoption([Options],'colour')
then
  hasoption([Options], 'color', 'CouleursUsagers', 'Options');
Options:=op(Options);
Couleur1:=color=op(1,CouleursUsagers);
Couleur2:=color=op(2,CouleursUsagers)
fi;
# -----
# Création des images
for j from 1 to Nbr_Emboîtements do
  Points:=[X0,Y0]:

```

```

for k from Nbr_Côtés by -1 to 1 do
  X1:=Centre[1]+(X0-Centre[1])*cos(S)+(Y0-Centre[2])*sin(S):
  Y1:=Centre[2]-(X0-Centre[1])*sin(S)+(Y0-Centre[2])*cos(S):
  if k<>1 then X0:=X1: Y0:=Y1 fi:
  Points:=Points,[X1,Y1]:
od:
# -----
if type(j,even) then Couleur:=Couleur1 else Couleur:=Couleur2 fi:
Figure||j:=plottools[polygon]([Points],Couleur,linestyle=0):
Polygone||(Nbr_Emboîtements-j+1):=plots[display](Figure||j):
Polyanime||j:=plots[display]([Polygone||((Nbr_Emboîtements-j+1).
.Nbr_Emboîtements)]):
X0:=X0+Décalage*(X1-X0):
Y0:=Y0+Décalage*(Y1-Y0):
od:
# -----
# Superposition des images
if Mode=fixe then
plots[display]([Polygone||(1..Nbr_Emboîtements)],Options)
elif Mode=anime then
plots[display]([Polyanime||(1..Nbr_Emboîtements)],insequence=true,
Options) fi
end proc:

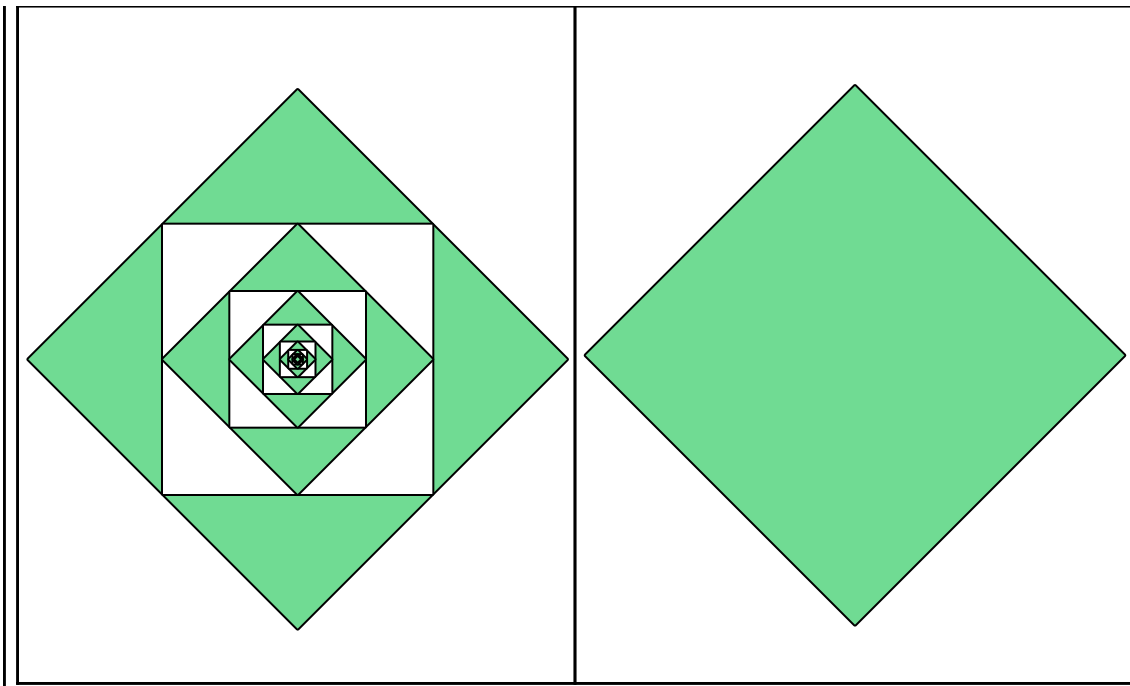
```

Exemples

```

> A:=rotate(Poly_Emboîtés(4,14,1/2,fixe),Pi): # Couleurs par
défaut
B:=rotate(Poly_Emboîtés(4,14,1/2,anime),Pi): # Couleurs par
défaut
display(Matrix(1,2,[A,B]),axes=none,title=``);

```

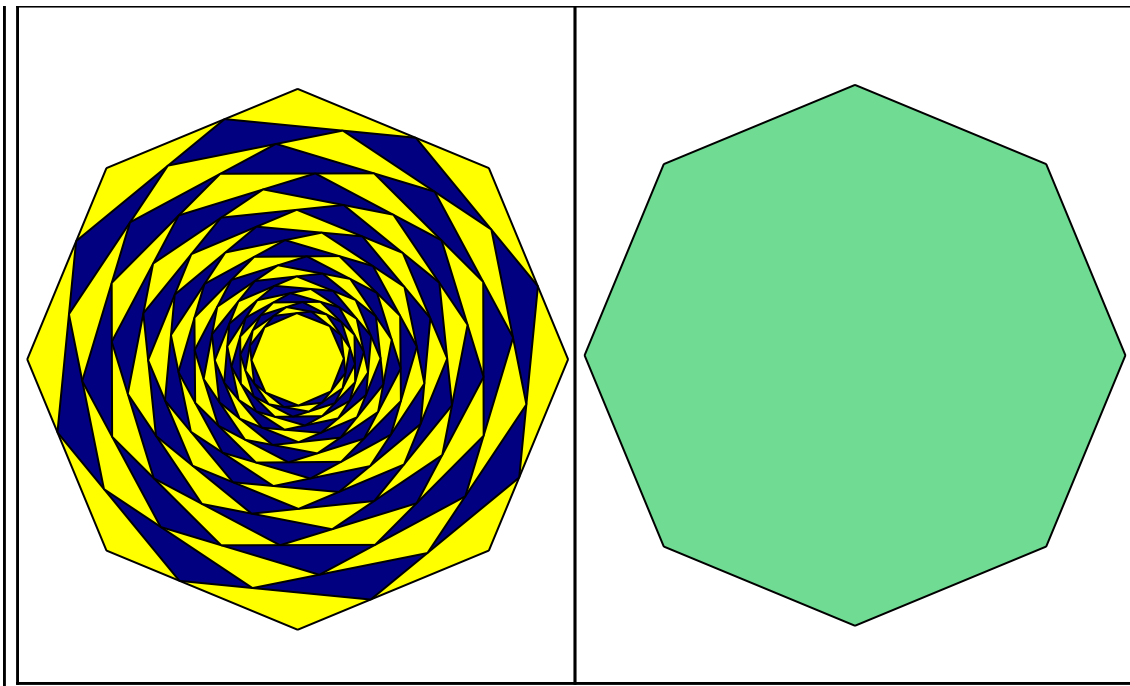


```
> Phi_inv:=1/(1/2+sqrt(5)/2);
```

$$\text{Phi_inv} := \frac{1}{\frac{1}{2} + \frac{\sqrt{5}}{2}}$$

(2.1.1.1)

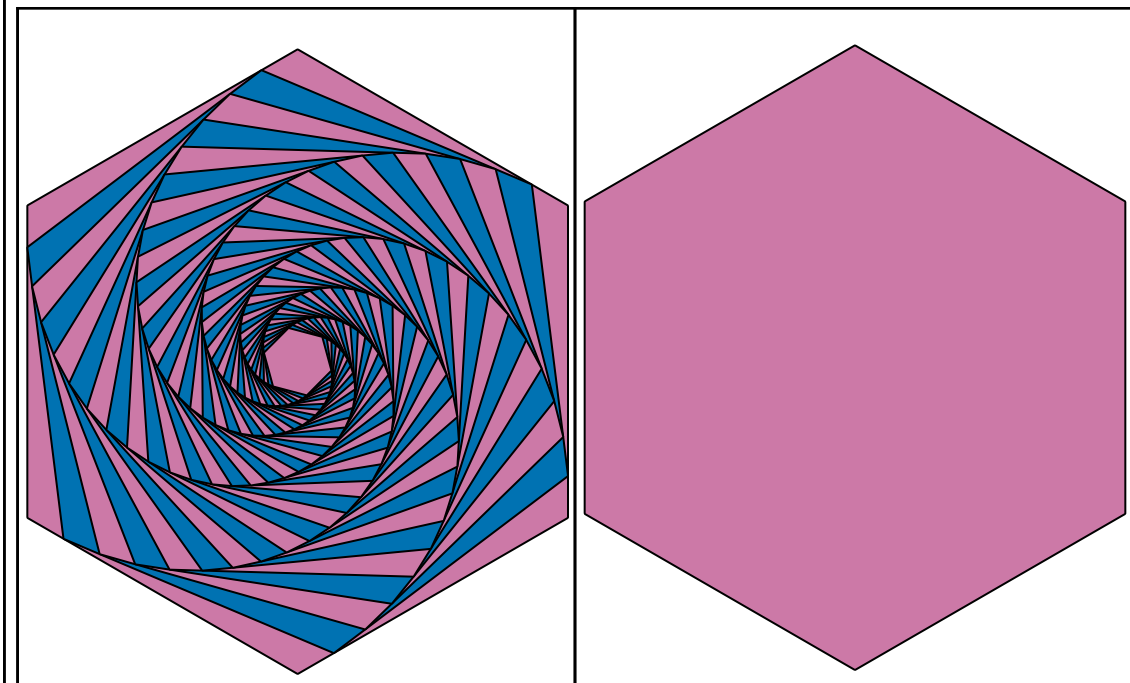
```
> A:=rotate(Poly_Emboîtés(8,24,Phi_inv,fixe,color=["CSS 12","CSS
5"]),Pi):
B:=rotate(Poly_Emboîtés(8,24,Phi_inv,anime),Pi): # Couleurs par
défaut
display(Matrix(1,2,[A,B]),axes=none,title=``);
```



```

> A:=Poly_Emboîtés(6,36,sqrt(3)/2,fixe,color=["Dalton 4","Dalton
6"]):
B:=Poly_Emboîtés(6,36,sqrt(3)/2,anime,color=["Dalton 4","Dalton
6"]):
display(Matrix(1,2,[A,B]),axes=none,title=``);

```



```

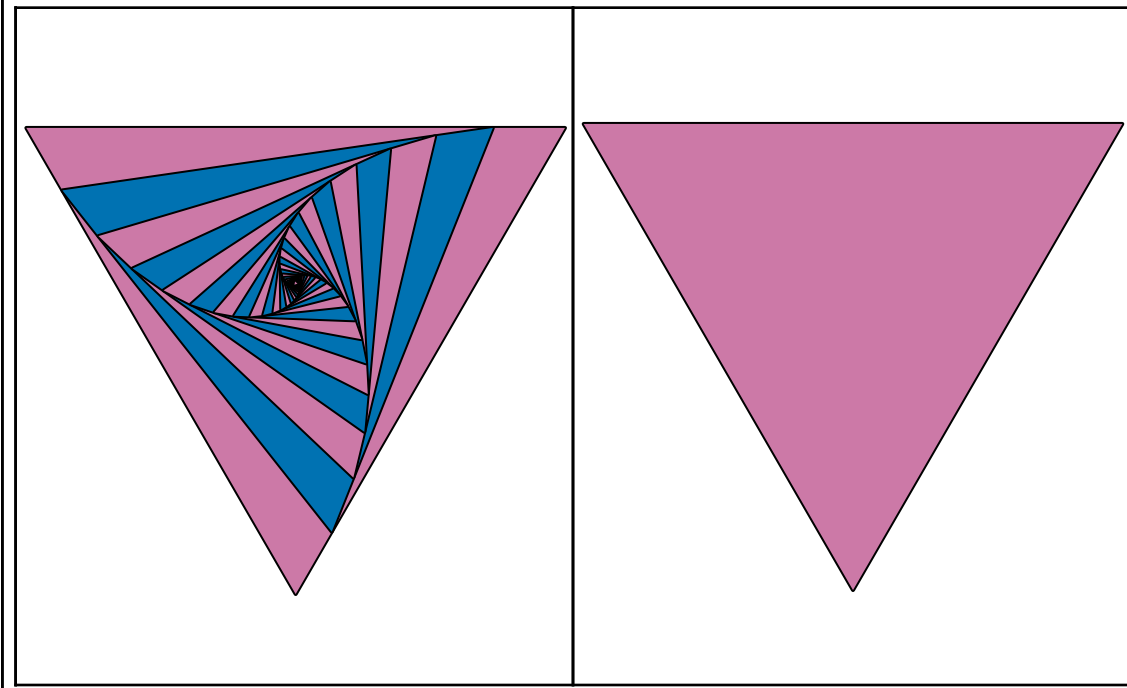
> A:=Poly_Emboîtés(3,24,sqrt(3)/2,fixe,color=["Dalton 4","Dalton
6"]):

```

```

B:=Poly_Emboîtés(3,24,sqrt(3)/2,anime,color=["Dalton 4","Dalton
6"]):
display(Matrix(1,2,[A,B]),axes=None,title="");

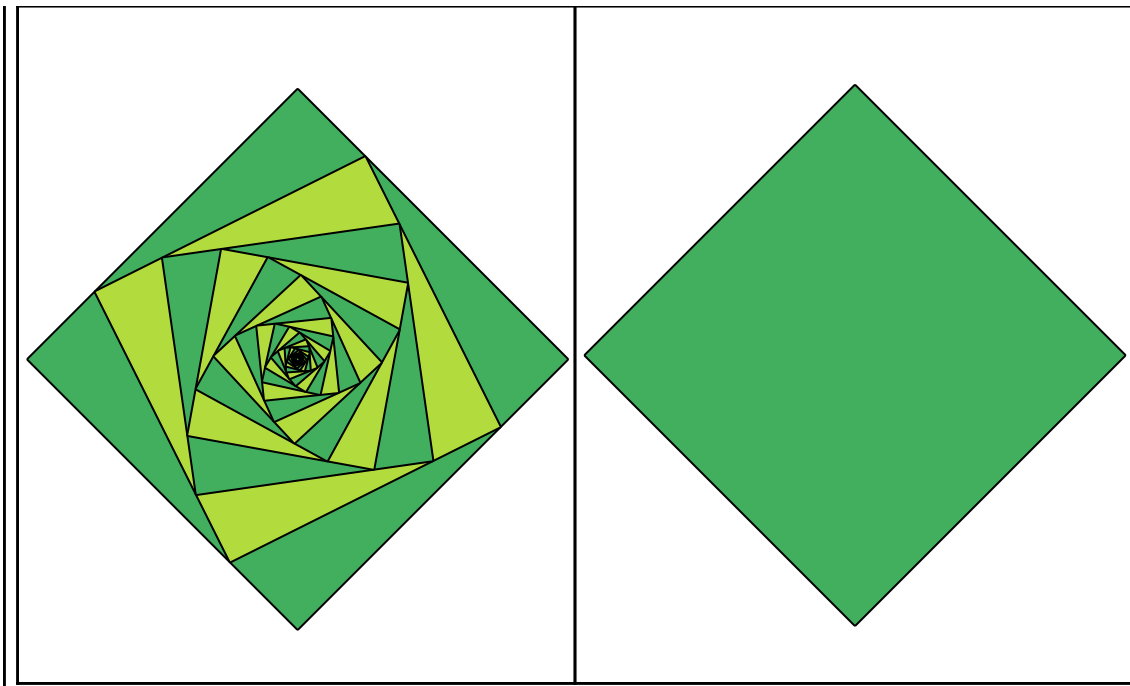
```



```

> A:=rotate(Poly_Emboîtés(4,24,1/4,fixe,color=["Executive 10",
"Executive 13"]),Pi):
B:=rotate(Poly_Emboîtés(4,24,1/4,anime,color=["Executive 10",
"Executive 13"]),Pi):
display(Matrix(1,2,[A,B]),axes=None,title="");

```

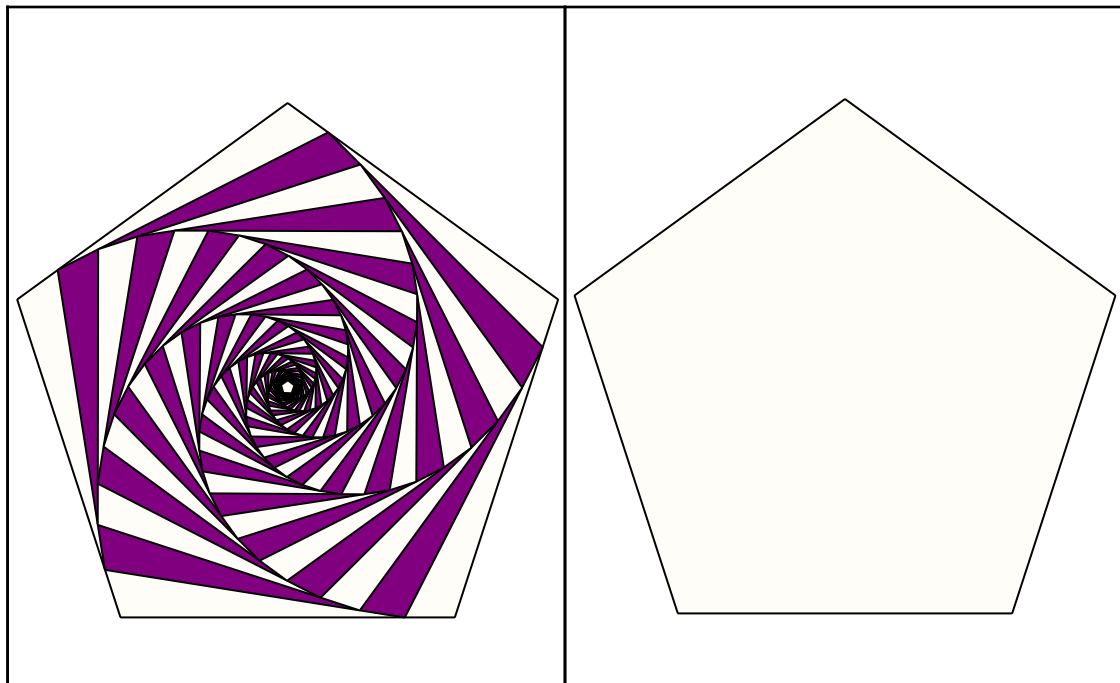


```

> Fond:=ColorTools:-Color([1,0.99215686,0.96862745]);
Tracé:=ColorTools:-Color([0.50196078, 0., 0.50196078]);
A:=rotate(Poly_Emboîtés(5,40,0.15,fixe,color=[Tracé,Fond]),Pi):
B:=rotate(Poly_Emboîtés(5,40,0.15,anime,color=[Tracé,Fond]),Pi):

display(Matrix(1,2,[A,B]),axes=none,title=``);
Fond := <RGB : 1 0.992 0.969>
Tracé := <RGB : 0.502 0 0.502>

```



```
> A:=Poly_Emboîtés(12,72,.5,fixe): # Couleurs par défaut  
B:=Poly_Emboîtés(12,72,.5,anime): # Couleurs par défaut  
display(Matrix(1,2,[A,B]),axes=None,title='');
```

