Les pavages avec la tortue ou la géométrie de xcas

Renée De Graeve

1er mars 2006

1 Généralités

On veut paver le plan avec un pavé `R`. À partir du pavé initial `R`, on va définir le pavé `P` qui pavera le plan par des translations dont les vecteurs sont donnés par deux nombres complexes `v` et `w`.
Ce pavé `P` sera défini à partir du pavé initial `R` à l'aide des transformations de `xcas` comme par exemple `rotation` et `symetrie`.

2 Comment faire un pavage comme un puzzle

On peut grâce à la souris translater des polygônes : pour cela on clique sur un côté du polygône, le trait devient bleu et sans relacher le bouton de la souris on emmene la figure avec la souris, puis on relache le bouton lorsqu'on est arrivé là ou l'on voulait aller. Si le pavé `P` est composé des pavés `R,R1,R2,R3` où `R1,R2,R3` sont des exemplaires de `R` obtenus par rotation ou symétrie, il suffit de définir sans les dessiner `R1,R2,R3` pour cela on fait des affectations que l'on termine par `:;` (par exemple `R :=rectangle(0,1,2) :;`), puis on dessine plusieurs de ces pavés (par exemple `R enter`, `R enter` etc...). Ainsi, on obtient une superposition de `R` et il suffit de déplacer ces différents exemplaires avec la souris.
Voici un premier exemple de pavage avec des rectangles.
On tape pour définir :

```
R :=rectangle(0,1,2) :;
R1 :=rotation(0,pi/2,R) :;
P :=[R,R1] :;
```

On peut alors paver le plan en faisant subir à `R` et à `R1` des translations de vecteurs `v=-1+i` et `w=2+2*i`.
On tape :

```
R enter R enter R enter R enter
R1 enter R1 enter R1 enter R1 enter
```

On a alors un puzzel de 8 pièces que l'on peut déplacer à sa guise.

Remarque
On ne peut pas déplacer `P` d'un seul coup car `P` est la juxtaposition de 2 polygones : on ne peut déplacer `P` que si on le définit comme polygone à savoir :

```
P :=polygone(0,1,1+2*i,2*i,0,i,-2+i,-2) :;
```

puis pour avoir 4 exemplaires de `P`, on tape :

```
P enter P enter P enter P enter
```

Après déplacement des 8 pièces `R` et `R1` ou des 4 pièces `P`, on obtient par exemple (on remarquera des petites imperfections dues à la mauvaise volonté de la souris !) :

3 Comment reproduire un pavage en géométrie

Nous allons prendre comme deuxième exemple un pavé `P` constitué de 4 rectangles obtenus par rotation et défini par :

```
R:=rectangle(0,1,2):;
R1:=rotation(0,pi/2,R):;
R2:=rotation(0,pi,R):;
R3:=rotation(0,-pi/2,R):;
P:=[R,R1,R2,R3]:;
PP:=polygone(0,1,1+2*i,2*i,0,i,-2+i,-2,0,-1,-1-2*i,-2*i,0,-i,-i+2,2):;
```

Voici le pavage que l'on veut reproduire

Nous allons créer ce pavage avec une fonction `LP` ayant 5 arguments :
`v` et `w` sont des nombres complexes définissant les vecteurs de 2 translations, `n` (resp `m`) est un nombre entier représentant le nombre de translations à faire avec `v` (resp `w`) et `Pa` représente le pavé à translater.
On dessine les transformées de `Pa` avec deux boucles `for` imbriquées, en faisant subir à `Pa`, `n` translations de vecteurs `v, 2*v, 3*v...n*v` où `v` est un nombre complexe et `m` translations de vecteurs `w, 2*w, 3*w...m*w` où `w` est un nombre complexe.

Soit par exemple le pavé défini au début formé des 4 rectangles obtenus par rotation (`P :=[R,R1,R2,R3]`).
On tape :

```
LP(v,w,n,m,Pa):={
 local j,k,PL;
 PL:=[];
 for (j:=0;j<m;j++) {
   for (k:=0;k<n;k++) {
      PL:=append(PL,translation(v*k,Pa));
   }
   Pa:=translation(w,Pa);
 }
 return PL;
}
```

puis on tape :
`LP(4,2*(1+i),4,5,P)`
On obtient bien le pavage désiré.

Ou encore, si on veut `P` en figure pleine (on fait alors des translations de vecteurs 4+4*i) :

```
PC :=polygone(1,1+2*i,2*i,i,-2+i,-2,-1,-1-2*i,-2*i,-i,2-i,2) :;
PCC :=couleur(PC,rempli) :;
couleur(LP(4,4*(1+i),4,3,PC),rempli);
```

ou
`LP(4,4*(1+i),4,3,PCC)`
On obtient alors :

4 Avec la tortue

On peut faire aussi des pavages avec la tortue. Pour réaliser le deuxième exemple, on dessine les transformées de `P` avec deux `repete` imbriqués, cf. la fonction `LP` ci-dessous : on a un `repete` dans `L(n,v,P)` qui fait `n` translations selon `v`, de `P` puis un autre `repete` qui fait `m` translations selon `w`, de `L(n,v,P)`. Dans la définition de `L` et de `LP` on fait revenir la tortue à son point de départ.
On peut aussi écrire une procédure récursive c'est `LPR`.

```
R():=repete 2, avance 10,tourne_gauche,avance 20,tourne_gauche;
P():=repete 4,R(),tourne_droite;
L(n,v,P):={
repete n,P(),saute(re(v)),pas_de_cote(im(v));
saute(-n*re(v))
};
LP(v,w,n,m,P):={
repete m,L(n,v,P),saute(re(w)),pas_de_cote(-im(w));
saute(-m*re(w));pas_de_cote(m*im(w));
};
LPR(v,w,n,m,P):={
L(n,v,P);
si (m>1) alors
  saute(re(w));pas_de_cote(-im(w));
  LPR(v,w,n,m-1,P);
  saute(-re(w));pas_de_cote(im(w));
fsi
}
```

Puis on tape :
`efface;saute(-120);LP(40,20*(1+i),4,4,P)`
ou
`efface;saute(-120);LPR(40,20*(1+i),4,4,P)`
ou si on veut `P` en figure pleine :
`RP() :=rectangle_plein(10,20);`
`PP() :=repete(4,RP(),tourne_droite 90);`
Puis on tape (on fait alors des translations de vecteurs 40+40*i) :
`efface;saute(-120);LP(40,40*(1+i),4,4,PP)`
ou
`efface;saute(-120);LPR(40,40*(1+i),4,4,PP)`

5 Exercice

On veut illustrer le théorème suivant avec `xcas` :

Un quadrilatère quelconque pave le plan.

Pour cela dessiner un quadrilatère, puis faites le pavage du plan avec ce quadrilatère avec la souris, ou/et avec les instructions de géométrie, ou/et avec la tortue. Bon courage ! ! ! ! ! !