



**HAL**  
open science

# Fonctions de couplage et chaînage des courbes et metacourbes de Gray généralisant les courbes de Hilbert

Jean-François Rotgé

## ► To cite this version:

Jean-François Rotgé. Fonctions de couplage et chaînage des courbes et metacourbes de Gray généralisant les courbes de Hilbert. 2022. <hal-03773928>

**HAL Id: hal-03773928**

**<https://hal.science/hal-03773928v1>**

Preprint submitted on 9 Sep 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

---

# Fonctions de couplage et chaînage des courbes et metacourbes de Gray généralisant les courbes de Hilbert

Jean-François Rotgé<sup>1</sup>

<sup>1</sup> Université de Montréal, Montréal, Québec, Canada

---

9 septembre 2022

## Table des matières

<b>1</b>	<b>Rappels des principes généraux</b>	<b>2</b>	2.2	Codage des courbes de type uE . . . . .	5
1.1	Détermination des courbes de Gray . . . . .	2	2.2.1	Algorithme ALG.2 CD0uE . . . . .	5
<b>2</b>	<b>Génération par morceaux</b>	<b>3</b>	2.3	Chaînage des courbes de type uE . . . . .	6
2.1	Décodage des courbes de type uE . . . . .	3	2.3.1	Algorithme ALG.3 matE . . . . .	6
2.1.1	Algorithme Alg.1 DC0uE . . . . .	4	2.4	Décodage des courbes de type uEV . . . . .	6
			2.4.1	Algorithme ALG.4 DC0uEV . . . . .	7
			2.5	Codage des courbes de type uEV . . . . .	8
			2.5.1	Algorithme ALG.5 CD0uEV . . . . .	8

2.6	Chaînage des courbes de type uEV . . . . .	9
2.6.1	Algorithme ALG.6 matEV . . . . .	9
2.7	Décodage des courbes de type uEMS . . . . .	10
2.7.1	Algorithme ALG.7 DC0uEMS . . . . .	10
2.8	Codage des courbes de type EMS . . . . .	12
2.8.1	Algorithme ALG.8 CD0uEMS . . . . .	12
2.9	Chaînage des courbes de type uEMS . . . . .	13
2.9.1	Algorithme ALG.9 matEMS . . . . .	13
2.10	Décodage des courbes de type uEMSi . . . . .	13
2.10.1	Algorithme ALG.10 DC0uEMSi . . . . .	14
2.11	Codage des courbes de type uEMSi . . . . .	15
2.11.1	Algorithme ALG.11 CD0uEMSi . . . . .	15
2.12	Chaînage des courbes de type uEMSi . . . . .	16
2.12.1	Algorithme ALG.12 matEMSi . . . . .	17
2.13	Décodage des courbes de type uLMS . . . . .	18
2.13.1	Algorithme ALG.13 DC0uLMS . . . . .	18
2.14	Codage des courbes de type uLMS . . . . .	19
2.14.1	Algorithme ALG.14 CD0uLMS . . . . .	19
2.15	Chaînage des courbes de type uLMS . . . . .	20
2.15.1	Algorithme ALG.15 matLMS . . . . .	20
3	<b>Génération par rotors de niveau 1</b>	<b>21</b>
3.1	Fractalisation de Hilbert généralisée . . . . .	23
4	<b>Génération par rotors de niveau 3</b>	<b>23</b>
5	<b>Notes bibliographiques</b>	<b>24</b>

## Résumé

Cet article complète une série d'articles sur l'exploration des fonc-

tions de construction et couplage de courbes et metacourbes de Gray dans le plan. Son objectif est d'explorer plus particulièrement différentes méthodes de généralisation des courbes de Hilbert. Le mécanisme original de fractalisation récursive imaginé par Hilbert limite en effet les courbes éponymes à des résolutions  $2^m - 1$ . Cette famille de courbes a été formalisée sur la base d'un schéma récursif général construit à partir des deux premières courbes de la série correspondant à  $m = 1$  ou  $m = 2$ . Le problème de généralisation des courbes de Hilbert se ramène par conséquent à l'identification des familles de courbes ou metacourbes remplissantes qui génèrent des séquences inductives de courbes comprenant les deux premières courbes de la série originale d'Hilbert comme occurrences particulières. Les nouvelles familles mises en évidence ont des résolutions  $2m - 1$  ou  $m$  avec des propriétés symétriques ou asymétriques. Les méthodes de construction et les fonction de couplage sont présentées sous la forme d'une compilation algorithmique pour chaque méthode proposée.

## 1 Rappels des principes généraux

### 1.1 Détermination des courbes de Gray

La détermination des courbes de Gray peut-être réalisée à partir de deux méthodes principales qui peuvent être complémentaires. La première méthode est de nature purement combinatoire et revient à construire des séquences algorithmiques de codes de Gray  $n$ -aires à deux dimensions (**Bib.5**). La seconde méthode est de nature géométrique à partir d'algorithmes de force brute (**Bib.5**) qui génèrent l'ensemble des chemins hamiltoniens couvrant l'ensemble des points d'une grille carrée de résolution donnée. A partir de l'énumération des chemins dans des grilles de basse résolution  $a \leq 5$ , il convient d'extraire des schémas inductifs permettant de formaliser des fonc-

tions de couplage pour des résolutions quelconques. De manière générale, les courbes générées par la première méthode doivent évidemment appartenir à l'ensemble des courbes générées par la méthode géométrique.

## 2 Génération par morceaux

La génération des courbes de Hilbert par morceaux consiste à former une nouvelle courbe ayant les propriétés d'une courbe de Gray à partir de courbes élémentaires racordées ou chaînées les unes aux autres. Ces nouvelles courbes doivent satisfaire aux conditions de congruence des courbes de Hilbert pour  $m = 1$  et  $m = 2$ . Contrairement aux courbes de Hilbert classiques le chaînage permettant d'obtenir une nouvelle courbe n'est pas basé sur le principe exclusif de récursivité mais adopte le principe de partition ou dissection d'un carré du plan en un certain nombre de rectangles. La figure Fig.1 illustre différents types de partitionnements à connexité variable. Le partitionnement 3-connexe de type A-1 correspond au type **uE**, le partitionnement A-2 au type **uEV**, le partitionnement A-3 au type **uEMS**, le partitionnement B-1 au type **uEMSi**, le partitionnement B-2 est défini par une courbe de type **uLMS**. Le partitionnement B-3 correspond à la construction d'une metacourbe de Gray de niveau 6 et de chaînage 1 dont la description est faite en **Bib.5**. Ce dernier partitionnement B-3 peut également correspondre à un mode de génération par rotors. Les 3 courbes de la colonne de droite illustrent des courbes sans partitionnement inductif évident.

### 2.1 Décodage des courbes de type uE

Le premier type de courbe correspondant à la partition A-1 est dénommé **uE**. Il s'agit d'une courbe à résolution impaire  $m = 2\alpha + 1$ .

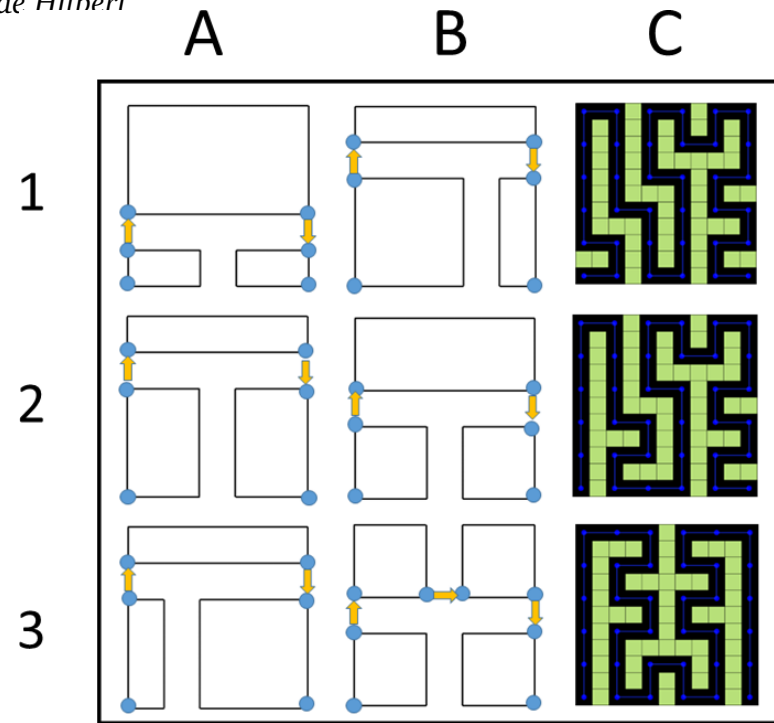


FIGURE 1 – Types de partitionnements de résolution 5

La courbe est illustrée par la figure Fig.2 pour des résolutions 3 et 5. L'occurrence de la courbe en haut à gauche de la figure est congruente avec la courbe de Hilbert classique pour  $m = 2$ . L'occurrence en haut à droite modifie le caractère symétrique de cette courbe. Les deux occurrences du bas de la figure correspondent à des résolutions 5 avec et sans symétrie. Les séquences d'appel des fonctions qui calculent les points des 4 courbes de la figure sont les suivantes ;

$[seq(\mathbf{DC0uE}(i, [3, 1]), i = 0..4 * 4 - 1)]$   
 $[seq(\mathbf{DC0uE}(i, [3, 2]), i = 0..4 * 4 - 1)]$   
 $[seq(\mathbf{DC0uE}(i, [5, 2]), i = 0..6 * 6 - 1)]$   
 $[seq(\mathbf{DC0uE}(i, [5, 3]), i = 0..6 * 6 - 1)]$

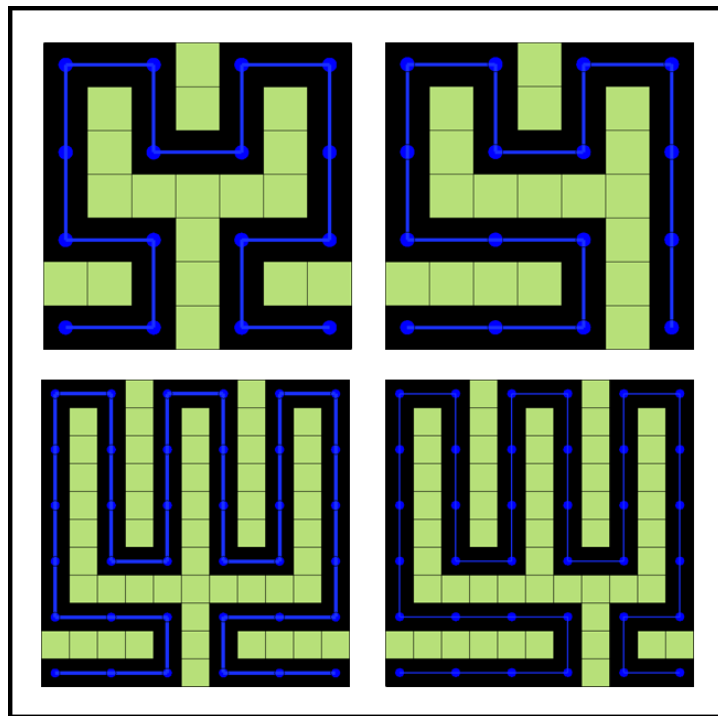


FIGURE 2 – Généralisation des courbes de Hilbert

### 2.1.1 Algorithme Alg.1 DC0uE

La fonction **Alg.1** correspond à la fonction de décodage de la courbe permettant d'associer un couple de coordonnées à un indice donné. La fonction reçoit *ind*, l'indice du point de la courbe dont on recherche les coordonnées et le couple de paramètres *v2p* correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe.

Alg. 1 – DC0uE

```

0 > DC0uE:=proc(ind, v2p)
1 > local a, b, c, d, nbrA, nbrB, nbrC, ind0, v2;
2 > a:=v2p[1];
3 > b:=v2p[2];
4 > c:=a-1-b;
5 > d:=a-2;
6 > nbrA:=2*(b+1);
7 > nbrB:=(d+1)*(a+1);
8 > nbrC:=2*(c+1);
9 > if ind >= 0 and ind < nbrA
10 > then
11 > v2:=DC0uLg(ind, b);
12 > return(v2);
13 > end if;
14 > if ind >= nbrA and ind < nbrA+nbrB
15 > then
16 > ind0:=ind-nbrA;
17 > v2:=DC0uwL(ind0, [d, 1], 2);
18 > return([v2[1], v2[2]+2]);
19 > end if;
20 > if ind >= nbrA+nbrB and ind < nbrA+nbrB+nbrC
21 > then
22 > ind0:=nbrC-1-(ind-nbrA-nbrB);
23 > v2:=DC0uLd(ind0, c);
24 > return([v2[1]+b+1, v2[2]]);
25 > end if;
26 > end proc;

```

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.1** :

- (i) **ligne 2-5**  $\Rightarrow$  Soit  $a, b, c, d$  les dimensions des rectangles
- (ii) **ligne 6**  $\Rightarrow$  Soit  $nbrA$  le nombre de points du rectangle A
- (iii) **ligne 7**  $\Rightarrow$  Soit  $nbrB$  le nombre de points du rectangle B
- (iv) **ligne 8**  $\Rightarrow$  Soit  $nbrC$  le nombre de points du rectangle C
- (v) **ligne 9**  $\Rightarrow$  Soit le test d'appartenance au rectangle A
- (vi) **ligne 11**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées associé à la courbe A
- (vii) **ligne 12,18,24**  $\Rightarrow$  Soit le retour des coordonnées finales du point après passage du repère local au repère général
- (viii) **ligne 14**  $\Rightarrow$  Soit le test d'appartenance au rectangle B
- (ix) **ligne 16**  $\Rightarrow$  Soit  $ind0$  l'indice local du point pour la courbe B
- (x) **ligne 17**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées associé à la courbe B
- (xi) **ligne 20**  $\Rightarrow$  Soit le test d'appartenance au rectangle C
- (xii) **ligne 22**  $\Rightarrow$  Soit  $ind0$  l'indice local du point pour la courbe C
- (xiii) **ligne 23**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées associé à la courbe C

**Alg. 2 – CD0uE**

```

0 > CD0uE:=proc(v2, v2p)
1 > local a, b, c, d, nbrA, nbrB, nbrC, x, y, v2a, ind, ind0;
2 > a:=v2p[1];
3 > b:=v2p[2];
4 > c:=a-1-b;
5 > d:=a-2;
6 > nbrA:=2*(b+1);
7 > nbrB:=(d+1)*(a+1);
8 > nbrC:=2*(c+1);
9 > x:=v2[1];
10 > y:=v2[2];
11 > if x < (b+1) and y < 2
12 > then
13 > ind:=CD0uLg(v2, b);
14 > return(ind);
15 > end if;
16 > if y >= 2
17 > then
18 > v2a:=[x, y-2];
19 > ind:=CD0uWl(v2a, [d,1], 2);
20 > return(ind+nbrA);
21 > end if;
22 > if x > b and y < 2
23 > then
24 > v2a:=[x-(b+1), y];
25 > ind:=CD0uLd(v2a, c);
26 > return(nbrA+nbrB+nbrC-ind-1);
27 > end if;
28 > end proc;

```

## 2.2 Codage des courbes de type uE

### 2.2.1 Algorithme ALG.2 CD0uE

La fonction **Alg.2** correspond à la fonction de codage de la courbe ayant pour occurrence la courbe de Hilbert de résolution 3 et la courbe trident de résolution 5. La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'indice et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe.

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.2** :

- (i) **ligne 2-10**  $\Rightarrow$  Soit les initialisations des paramètres (cf DC0uE)

- (ii) **ligne 11**  $\Rightarrow$  Soit le test d'appartenance au rectangle A
- (iii) **ligne 13**  $\Rightarrow$  Soit  $ind$  l'indice local du point sur la courbe A
- (iv) **ligne 14,20,26**  $\Rightarrow$  Soit l'indice global du point sur la courbe uE
- (v) **ligne 16**  $\Rightarrow$  Soit le test d'appartenance au rectangle B
- (vi) **ligne 18,24**  $\Rightarrow$  Soit  $v2a$  les coordonnées locales du point sur la courbe B et C
- (vii) **ligne 19**  $\Rightarrow$  Soit  $ind$  l'indice local du point sur la courbe B
- (viii) **ligne 22**  $\Rightarrow$  Soit le test d'appartenance au rectangle C

(ix) **ligne 25**  $\Rightarrow$  Soit *ind* l'indice local du point sur la courbe C

## 2.3 Chaînage des courbes de type uE

Le chaînage des courbes de type **uE** consiste à déterminer le type de transformation ponctuelle associé à chaque point de la courbe de manière à raccorder différentes courbes entre elles en conservant les propriétés de codage de Gray entre les coordonnées des points voisins. La figure **Fig.3** illustre la carte de chaînage de la courbe de type **uE**. Les cartes de chaînage peuvent être déterminées par une approche globale directe ou une approche locale indirecte qui utilise les cartographies locales des morceaux de courbes chaînées.

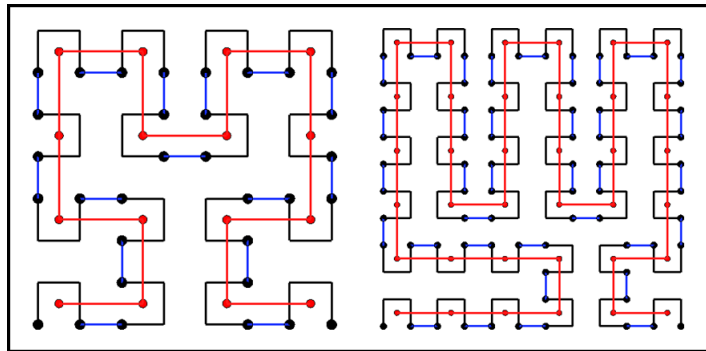


FIGURE 3 – Carte de chaînage des courbes de type E

### 2.3.1 Algorithme ALG.3 matE

La fonction **Alg.3** correspond à la fonction de chaînage de la courbe ayant pour occurrence la courbe de Hilbert de résolution 3 et la courbe trident de résolution 5. La cartographie de chaînage est réalisée dans l'algorithme présenté de manière globale et directe sans passer par

les chaînages respectifs des courbes A, B et C. La fonction reçoit *v2*, les coordonnées du point de la courbe dont on recherche l'identifiant de transformation ponctuelle et le couple de paramètres *v2p* correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe. La fonction retourne la valeur entière correspondant à la matrice ponctuelle de chaînage associée au point de coordonnées en entrée.

Alg. 3 – **matE**

```

0 > matE:=proc(v2, v2p)
1 > local a, b, c, d, x, y, cs;
2 > a:=v2p[1];
3 > b:=v2p[2];
4 > c:=a-1-b;
5 > d:=a-2;
6 > x:=v2[1];
7 > y:=v2[2];
8 > if y = 1 and x < b then return(2); end if;
9 > if y = 1 and x > b+1 then return(2); end if;
10 > if y = 1 and x = b then return(7); end if;
11 > if y = 1 and x = b+1 then return(5); end if;
12 > if y = 0 and x < b then return(0); end if;
13 > if y = 0 and x > b+1 then return(0); end if;
14 > if y = 0 and x = b then return(7); end if;
15 > if y = 0 and x = b+1 then return(5); end if;
16 > cs:=matL([x, y-2], d);
17 > return(cs);
18 > end proc;
    
```

## 2.4 Décodage des courbes de type uEV

Le second type de courbe correspondant à la partition A-2 est dénommé **uEV**. Il s'agit d'une courbe à résolution impaire  $m = 2\alpha + 1$ . La courbe est illustrée par la figure **Fig.4** pour des résolutions 3 et 5. L'occurrence de la courbe en haut à gauche de la figure est congruente

avec la courbe de Hilbert classique pour  $m = 2$ . L'occurrence en haut à droite modifie le caractère symétrique de cette courbe. Les deux occurrences du bas de la figure correspondent à des résolutions 5 et 6 avec et sans symétrie. Les séquences d'appel des fonctions qui calculent les points des 4 courbes de la figure sont les suivantes ;

$$\begin{aligned} & [seq(\mathbf{DC0uEV}(i, [3, 1]), i = 0..4 * 4 - 1)] \\ & [seq(\mathbf{DC0uEV}(i, [3, 2]), i = 0..4 * 4 - 1)] \\ & [seq(\mathbf{DC0uEV}(i, [5, 2]), i = 0..6 * 6 - 1)] \\ & [seq(\mathbf{DC0uEV}(i, [7, 0]), i = 0..8 * 8 - 1)] \end{aligned}$$

### 2.4.1 Algorithme ALG.4 DC0uEV

La fonction **Alg.4** correspond à la fonction de décodage de la courbe illustrée sur la figure **Fig.4**. La fonction reçoit *ind*, l'indice du point de la courbe dont on recherche les coordonnées et le couple de paramètres *v2p* correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe.

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.4** :

- (i) **ligne 2-5**  $\Rightarrow$  Soient  $a, b, c, d$  les dimensions des rectangles
- (ii) **ligne 6**  $\Rightarrow$  Soit  $nbrA$  le nombre de points de la courbe A
- (iii) **ligne 7**  $\Rightarrow$  Soit  $nbrB$  le nombre de points de la courbe B
- (iv) **ligne 8**  $\Rightarrow$  Soit  $nbrC$  le nombre de points de la courbe C
- (v) **ligne 9**  $\Rightarrow$  Soit le test d'appartenance à la courbe A
- (vi) **ligne 11**  $\Rightarrow$  Soit les coordonnées locales du point pour la courbe A
- (vii) **ligne 12**  $\Rightarrow$  Soit le retour des coordonnées globales pour la courbe A

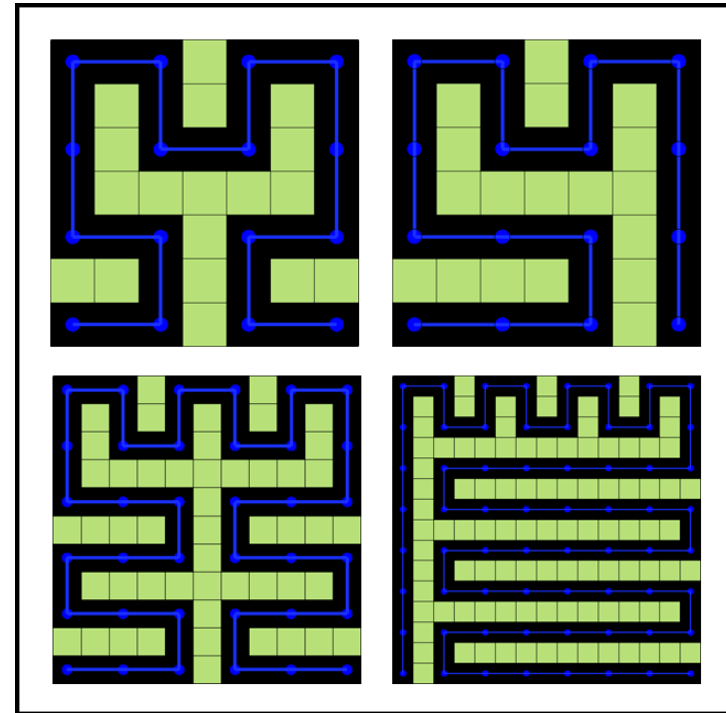


FIGURE 4 – Courbes de Hilbert de type uEV

- (viii) **ligne 14**  $\Rightarrow$  Soit le test d'appartenance à la courbe B
- (ix) **ligne 16**  $\Rightarrow$  Soit  $ind0$  l'indice local du point pour la courbe B
- (x) **ligne 17**  $\Rightarrow$  Soit les coordonnées locales du point pour la courbe B
- (xi) **ligne 18**  $\Rightarrow$  Soit le retour des coordonnées globales pour la courbe B
- (xii) **ligne 20**  $\Rightarrow$  Soit le test d'appartenance à la courbe C
- (xiii) **ligne 22**  $\Rightarrow$  Soit  $ind0$  l'indice local du point pour la courbe C
- (xiv) **ligne 23**  $\Rightarrow$  Soit les coordonnées locales du point pour la courbe C
- (xv) **ligne 24**  $\Rightarrow$  Soit le retour des coordonnées globales pour la courbe C

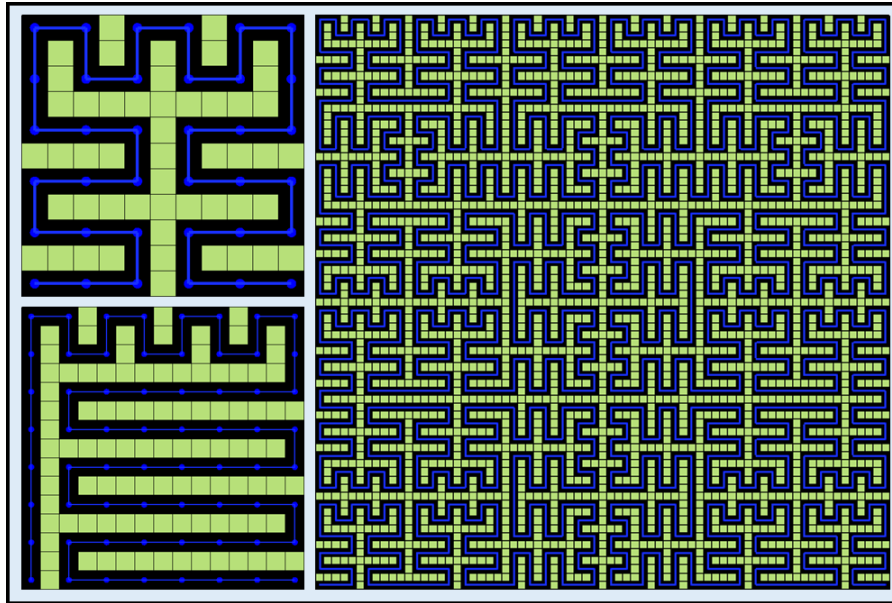


FIGURE 5 – Courbe de Hilbert de niveau 2 de type uEV

courbe C

Alg. 4 – DC0uEV

```

0 > DC0uEV:=proc(ind, v2p)
1 > local a, b, c, d, nbrA, nbrB, nbrC, ind0, v2;
2 > a:=v2p[1];
3 > b:=v2p[2];
4 > c:=a-b-1;
5 > d:=a-1;
6 > nbrA:=(b+1)*d;
7 > nbrB:=2*(a+1);
8 > nbrC:=(c+1)*d;
9 > if ind >= 0 and ind < nbrA
10 > then
11 > v2:=DC0uLg(ind, b);
12 > return(v2);
13 > end if;
14 > if ind >= nbrA and ind < nbrA+nbrB
15 > then
16 > ind0:=ind-nbrA;
17 > v2:=DC0uL(ind0, [1, 1], 2);
18 > return([v2[1], v2[2]+d]);
19 > end if;
20 > if ind >= nbrA+nbrB and ind < nbrA+nbrB+nbrC
21 > then
22 > ind0:=nbrC-1-(ind-nbrA-nbrB);
23 > v2:=DC0uLd(ind0, c);
24 > return([v2[1]+b+1, v2[2]]);
25 > end if;
26 > end proc;

```

## 2.5 Codage des courbes de type uEV

### 2.5.1 Algorithme ALG.5 CD0uEV

La fonction Alg.5 correspond à la fonction de codage de la courbe ayant pour occurrence la courbe de Hilbert de résolution 3 et la courbe de résolution 5. La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'indice et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe.

Nous avons à la suite les commentaires concernant le code de l'al-

gorithme ALG.5 :

- (i) **ligne 2-10** ⇒ Soit les initialisations des paramètres (cf DC0uE)
- (ii) **ligne 11** ⇒ Soit le test d'appartenance au rectangle A
- (iii) **ligne 12** ⇒ Soit le retour de l'indice global pour la courbe uEV
- (iv) **ligne 13** ⇒ Soit  $ind$  l'indice local du point sur la courbe A
- (v) **ligne 14** ⇒ Soit le test d'appartenance au rectangle B
- (vi) **ligne 16** ⇒ Soit  $v2a$  les coordonnées locales du point pour la courbe B
- (vii) **ligne 17** ⇒ Soit  $ind$  l'indice local du point pour la courbe B

Alg. 5 – CD0uEV

```

0 > CD0uEV:=proc(v2,v2p)
1 > local a,b,c,d,nbrA,nbrB,nbrC,x,y,v2a,ind,ind0;
2 > a:=v2p[1];
3 > b:=v2p[2];
4 > c:=a-1-b;
5 > d:=a-1;
6 > nbrA:=(b+1)*d;
7 > nbrB:=2*(a+1);
8 > nbrC:=(c+1)*d;
9 > x:=v2[1];
10 > y:=v2[2];
11 > if x < (b+1) and y < d
12 > then return(CD0uWL(v2,[b,0],2));
13 > end if;
14 > if y >= d
15 > then
16 >   v2a:=[x,y-d];
17 >   ind:=CD0uWL(v2a,[1,1],2);
18 >   return(ind+nbrA);
19 > end if;
20 > if x > b and y < d
21 > then
22 >   v2a:=[a-x,y];
23 >   ind:=CD0uWL(v2a,[c,0],2);
24 >   return(nbrA+nbrB+nbrC-ind-1);
25 > end if;
26 > end proc;

```

- (viii) **ligne 18** ⇒ Soit le retour de l'indice global pour la courbe uEV
- (ix) **ligne 20** ⇒ Soit le test d'appartenance au rectangle C
- (x) **ligne 22** ⇒ Soit  $v2a$  les coordonnées locales du point pour la courbe C
- (xi) **ligne 23** ⇒ Soit  $ind$  l'indice local du point pour la courbe C
- (xii) **ligne 24** ⇒ Soit le retour de l'indice global pour la courbe uEV

## 2.6 Chaînage des courbes de type uEV

La figure Fig.6 illustre la carte de chaînage de la courbe de type uEV. La carte de chaînage est déterminée par une approche locale indirecte qui utilise les cartographies locales des 3 morceaux de courbes chaînées.

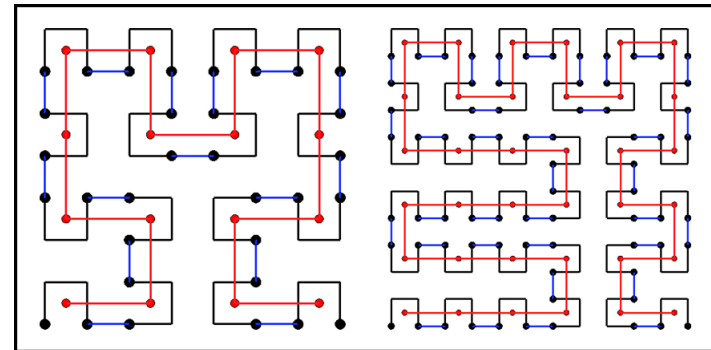


FIGURE 6 – Carte de chaînage des courbes de type EV

### 2.6.1 Algorithme ALG.6 matEV

La fonction Alg.6 correspond à la fonction de chaînage de la courbe ayant pour occurrence la courbe de Hilbert de résolution 3 et la courbe de résolution 5. La cartographie de chaînage est réalisée dans l'algorithme présenté de manière indirecte en utilisant les chaînages respectifs des courbes A, B et C. La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'identifiant de transformation ponctuelle et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe. La fonction retourne la valeur entière correspondant à la matrice ponctuelle de chaînage associée au point de coordonnées en entrée.

Alg. 6 – matEV

```

0 > matEV:=proc(v2, v2p)
1 > local a, b, c, d, x, y, cs, v2a;
2 > a:=v2p[1];
3 > b:=v2p[2];
4 > c:=a-1-b;
5 > d:=a-2;
6 > x:=v2[1];
7 > y:=v2[2];
8 > if x < (b+1) and y <= d
9 > then
10 > cs:=matLg(v2, b);
11 > return(cs);
12 > end if;
13 > if y > d
14 > then
15 > v2a:=[x, y-d-1];
16 > cs:=matL(v2a, 1);
17 > return(cs);
18 > end if;
19 > if x > b and y <= d
20 > then
21 > v2a:=[x-b-1, y];
22 > cs:=matLdv(v2a, c);
23 > return(cs);
24 > end if;
25 > end proc;

```

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.6** :

- (i) **ligne 2-5** ⇒ Soient  $a, b, c, d$  les dimensions des rectangles
- (ii) **ligne 8** ⇒ Soit le test d'appartenance du point à la courbe A
- (iii) **ligne 10** ⇒ Soit  $cs$  le calcul de matrice pour la courbe A
- (iv) **ligne 11** ⇒ Soit le retour de l'indicateur de matrice
- (v) **ligne 13** ⇒ Soit le test d'appartenance du point à la courbe B
- (vi) **ligne 15** ⇒ Soit  $v2a$  les coordonnées locales du point pour la courbe B
- (vii) **ligne 16** ⇒ Soit  $cs$  le calcul de matrice pour la courbe B

(viii) **ligne 21** ⇒ Soit  $v2a$  les coordonnées locales du point pour la courbe C

(ix) **ligne 22** ⇒ Soit  $cs$  le calcul de matrice pour la courbe C

(x) **ligne 23** ⇒ Soit le retour de l'indicateur de matrice

## 2.7 Décodage des courbes de type uEMS

Le troisième type de courbe correspondant à la partition A-3 est dénommé **uEMS**. Il s'agit d'une courbe à résolution quelconque. La courbe est illustrée par la figure **Fig.??** pour des résolutions 3, 4 et 5. L'occurrence de la courbe en haut à gauche de la figure est congruente avec la courbe de Hilbert classique pour  $m = 2$ . L'occurrence en haut à droite modifie le caractère symétrique de cette courbe. Les deux occurrences du bas de la figure correspondent à des résolutions 4 et 5. Les séquences d'appel des fonctions qui calculent les points des 4 courbes de la figure sont les suivantes ;

$[seq(DC0uEMS(i, [6, 0]), i = 0..4 * 4 - 1)]$

$[seq(DC0uEMS(i, [6, 1]), i = 0..4 * 4 - 1)]$

$[seq(DC0uEMS(i, [7, 0]), i = 0..5 * 5 - 1)]$

$[seq(DC0uEMS(i, [7, 1]), i = 0..6 * 6 - 1)]$

### 2.7.1 Algorithme ALG.7 DC0uEMS

La fonction **Alg.7** correspond à la fonction de décodage de la courbe correspondant à la partition A-3. La fonction reçoit  $ind$ , l'indice du point de la courbe dont on recherche les coordonnées et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe.

Fonctions de couplage et chaînage  
des courbes et metacourbes de Gray  
généralisant les courbes de Hilbert

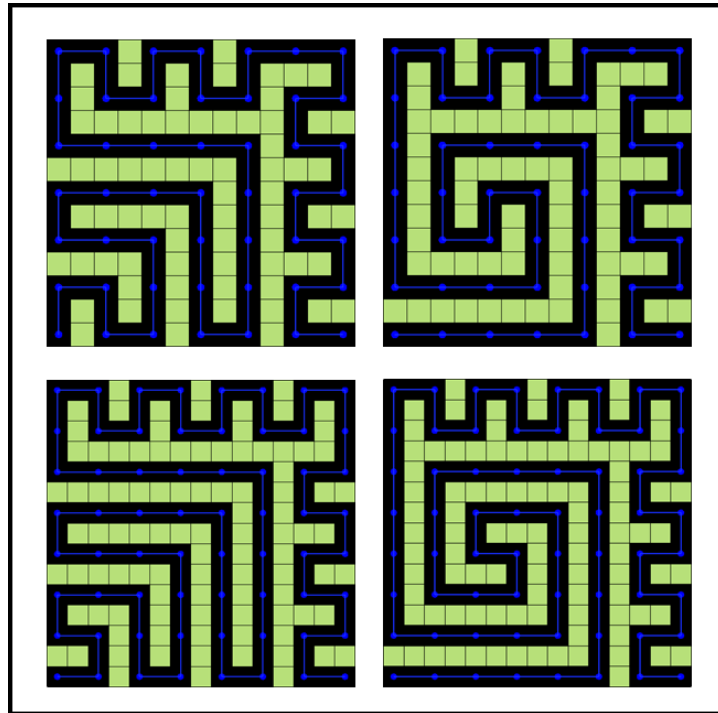


FIGURE 7 – Courbes de type uEMS

Nous avons à la suite les commentaires concernant le code de l'algorithme ALG.?? :

- (i) **ligne 3**  $\Rightarrow$  Soit  $ty$  le type de courbe dans le carré de la partition
- (ii) **ligne 6**  $\Rightarrow$  Soit  $nbrA$  le nombre de points de la courbe dans le carré
- (iii) **ligne 7**  $\Rightarrow$  Soit le test de parité de la résolution de la courbe
- (iv) **ligne 8-9**  $\Rightarrow$  Soient  $nbrB$  et  $nbrC$  les nombres de points des rectangles de la partition
- (v) **ligne 11**  $\Rightarrow$  Soit le test d'appartenance à la courbe A
- (vi) **ligne 13**  $\Rightarrow$  Soit le test de type de courbe inscrite dans le carré

Alg. 7 – DC0uEMS

```

0 > DC0uEMS:=proc(ind , v2p)
1 > local a , ty , am , ap , nbrA , nbrB , nbrC , ind0 , v2;
2 > a:=v2p[1];
3 > ty:=v2p[2];
4 > am:=a-1;
5 > ap:=a+1;
6 > nbrA:=am^2;
7 > if a mod 2 = 0
8 > then nbrB:=2*am; nbrC:=2*ap;
9 > else nbrB:=2*ap; nbrC:=2*am;
10 > end if;
11 > if ind >= 0 and ind < nbrA
12 > then
13 > if ty = 0
14 > then v2:=DC0uM(ind , am);
15 > else v2:=DC0uSi(ind , am);
16 > end if;
17 > return (v2);
18 > end if;
19 > if ind >= nbrA and ind < nbrA+nbrB
20 > then
21 > ind0:=ind-nbrA;
22 > v2:=DC0uWl(ind0 , [1 , 1] , 2);
23 > return ([v2[1] , v2[2]+am]);
24 > end if;
25 > if ind >= nbrA+nbrB and ind < nbrA+nbrB+nbrC
26 > then
27 > ind0:=nbrC-1-(ind-nbrA-nbrB);
28 > v2:=DC0uLd(ind0 , 1);
29 > return ([am+v2[1] , v2[2]]);
30 > end if;
31 > end proc;

```

- (vii) **ligne 14**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales pour le morceau méandre ( $ty=0$ )
- (viii) **ligne 15**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales pour le morceau spirale ( $ty=1$ )
- (ix) **ligne 19**  $\Rightarrow$  Soit le test d'appartenance à la courbe B
- (x) **ligne 21**  $\Rightarrow$  Soit  $ind0$  l'indice local pour la courbe B en lacet

- (xi) **ligne 22**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales pour le morceau de courbe B
- (xii) **ligne 23**  $\Rightarrow$  Soit le retour des coordonnées globales du rectangle B
- (xiii) **ligne 25**  $\Rightarrow$  Soit le test d'appartenance à la courbe C
- (xiv) **ligne 27**  $\Rightarrow$  Soit  $ind0$  l'indice local pour le morceau de courbe C
- (xv) **ligne 28**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales pour le morceau de courbe C
- (xvi) **ligne 29**  $\Rightarrow$  Soit le retour des coordonnées globales du rectangle C

## 2.8 Codage des courbes de type EMS

### 2.8.1 Algorithme ALG.8 CD0uEMS

La fonction **Alg.8** est la fonction de codage inverse de la fonction de décodage **Alg.7**. La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'indice et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du morceau de courbe inscrit dans le carré de la partition A-3.

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.8** :

- (i) **ligne 3-7**  $\Rightarrow$  Soit  $a, b, am, ap$  les dimensions de la partition
- (ii) **ligne 8**  $\Rightarrow$  Soit  $nbrA$  le nombre de points de la courbe A
- (iii) **ligne 9**  $\Rightarrow$  Soit le test de parité de la résolution
- (iv) **ligne 10-11**  $\Rightarrow$  Soient  $nbrB$  et  $nbrC$  les nombres de points des courbes B et C et les dimensions  $lg1$  et  $h0$
- (v) **ligne 15**  $\Rightarrow$  Soit le test d'appartenance à la courbe A
- (vi) **ligne 17**  $\Rightarrow$  Soit  $ty$  le type de courbe **uM** ou **uS**
- (vii) **ligne 18**  $\Rightarrow$  Soit  $ind$  l'indice local du point pour la courbe **uM**
- (viii) **ligne 19**  $\Rightarrow$  Soit  $ind$  l'indice local du point pour la courbe **uSi**

### Alg. 8 – CD0uEMS

```

0 > CD0uEMS:=proc(v2, v2p)
1 > local a, ty, b, am, ap, nbrA, nbrB, nbrC,
2 > lg1, h0, x, y, ind, v2a;
3 > a:=v2p[1];
4 > ty:=v2p[2];
5 > b:=a-2;
6 > am:=a-1;
7 > ap:=a+1;
8 > nbrA:=am^2;
9 > if a mod 2 = 0
10 > then nbrB:=2*am; nbrC:=2*ap; lg1:=am; h0:=ap;
11 > else nbrB:=2*ap; nbrC:=2*am; lg1:=ap; h0:=am;
12 > end if;
13 > x:=v2[1];
14 > y:=v2[2];
15 > if x < am and y < am
16 > then
17 > if ty = 0
18 > then ind:=CD0uM(v2, am);
19 > else ind:=CD0uSi(v2, am);
20 > end if;
21 > return(ind);
22 > end if;
23 > if y > b and x < lg1
24 > then
25 > v2a:=[x, y-b-1];
26 > ind:=CD0uWL(v2a, [1, 1], 2);
27 > return(ind+nbrA);
28 > end if;
29 > if x > b and y < h0
30 > then
31 > v2a:=[x-b-1, y];
32 > ind:=CD0uLd(v2a, 1);
33 > return(nbrA+nbrB+nbrC-ind-1);
34 > end if;
35 > end proc;

```

- (ix) **ligne 21**  $\Rightarrow$  Soit le retour de l'indice global recherché
- (x) **ligne 23**  $\Rightarrow$  Soit le test d'appartenance à la courbe B
- (xi) **ligne 31**  $\Rightarrow$  Soit  $v2a$  les coordonnées locales du point pour la courbe B

- (xii) **ligne 26**  $\Rightarrow$  Soit  $ind$  l'indice local du point pour la courbe B
- (xiii) **ligne 27**  $\Rightarrow$  Soit le retour de l'indice global recherché
- (xiv) **ligne 29**  $\Rightarrow$  Soit le test d'appartenance à la courbe C
- (xv) **ligne 31**  $\Rightarrow$  Soit  $v2a$  les coordonnées locales du point pour la courbe C
- (xvi) **ligne 32**  $\Rightarrow$  Soit le test d'appartenance à la courbe C
- (xvii) **ligne 33**  $\Rightarrow$  Soit le retour de l'indice global recherché

## 2.9 Chaînage des courbes de type uEMS

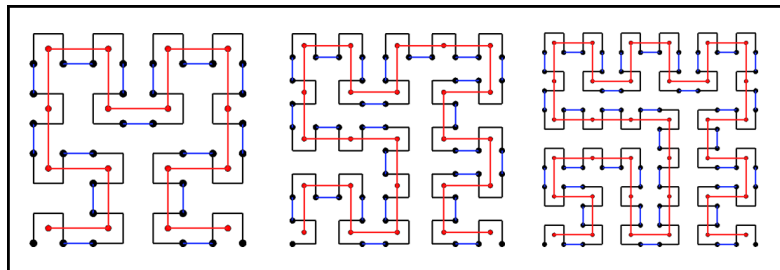


FIGURE 8 – Carte de chaînage des courbes de type uEMS de type 0

### 2.9.1 Algorithme ALG.9 matEMS

La fonction **Alg.9** est la fonction de chaînage qui permet de générer les cartes des figures **Alg.8** et **Alg.9**.

La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'identifiant de transformation ponctuelle et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe. La fonction retourne la valeur entière correspondant à la

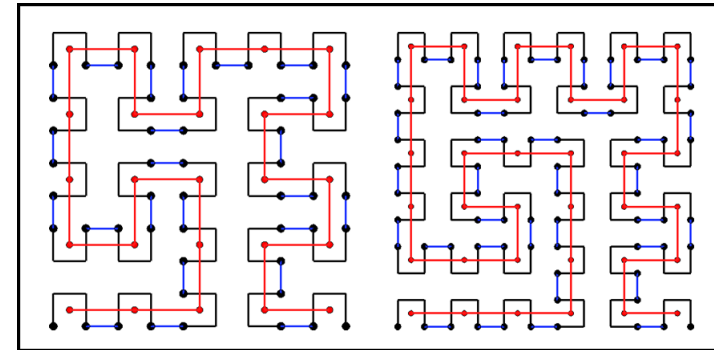


FIGURE 9 – Carte de chaînage des courbes de type uEMS de type 1

matrice ponctuelle de chaînage associée au point de coordonnées en entrée.

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.9** :

- (i) **ligne 13**  $\Rightarrow$  Soit le test d'appartenance au carré de la partition
- (ii) **ligne 15**  $\Rightarrow$  Soit le test du type de la courbe inscrit dans le carré de la partition
- (iii) **ligne 16**  $\Rightarrow$  Soit  $cs$  l'indicateur de matrice de chaînage pour la courbe méandre
- (iv) **ligne 19**  $\Rightarrow$  Soit  $cs$  l'indicateur de matrice de chaînage pour la courbe spirale inverse
- (v) **ligne 23-30**  $\Rightarrow$  Soit  $cs$  l'indicateur de matrice de chaînage pour les points n'appartenant pas au carré

## 2.10 Décodage des courbes de type uEMSi

Le second type de courbe correspondant à la partition A-2 est dénommé **uEMSi**. Il s'agit d'une courbe à résolution impaire  $m = 2\alpha + 1$ . La courbe est illustrée par la figure **Fig.10** pour des résolutions 3 et 5.

Alg. 9 – matEMS

```

0 > matEMS:=proc(v2,v2p)
1 > local a,ty,am,ap,nbrA,nbrB,nbrC,lg1,h0,x,y,cs,ind;
2 > a:=v2p[1];
3 > ty:=v2p[2];
4 > am:=a-1;
5 > ap:=a+1;
6 > nbrA:=am^2;
7 > if a mod 2 = 0
8 > then nbrB:=2*am; nbrC:=2*ap; lg1:=am; h0:=ap;
9 > else nbrB:=2*ap; nbrC:=2*am; lg1:=ap; h0:=am;
10 > end if;
11 > x:=v2[1];
12 > y:=v2[2];
13 > if x < am and y < am
14 > then
15 >   if ty = 0
16 >   then cs:=matM(v2,am);
17 >   else
18 >     ind:= CD0uSi(v2,am);
19 >     cs:=matSi(ind,am);
20 >   end if;
21 >   return(cs);
22 > end if;
23 > if y = a then return(0); end if;
24 > if y = am and a mod 2 = 0 and x = a
25 > then return(2);
26 > end if;
27 > if y = am and x mod 2 = 0 then return(7); end if;
28 > if y = am and x mod 2 <> 0 then return(5); end if;
29 > if x = am then return(5); end if;
30 > if y mod 2 = 0 then return(0); end if;
31 > return(2);
32 > end proc;

```

L'occurrence de la courbe en haut à gauche de la figure est congruente avec la courbe de Hilbert classique pour  $m = 2$ . L'occurrence en haut à droite modifie le caractère symétrique de cette courbe. Les deux occurrences du bas de la figure correspondent à des résolutions 5 et 6 avec et sans symétrie. Les séquences d'appel des fonctions qui calculent les points des 4 courbes de la figure sont les suivantes ;

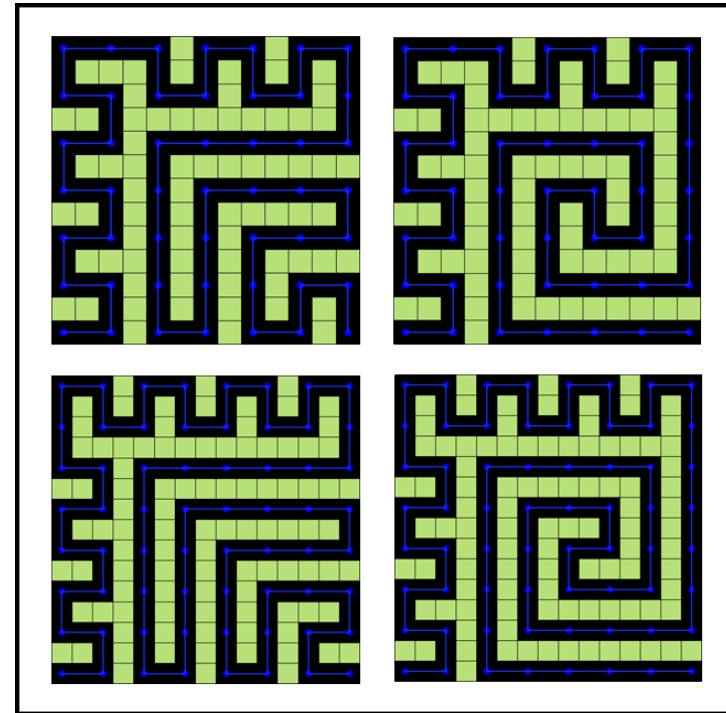


FIGURE 10 – Courbes de types DC0uEMSi 0 et 1

### 2.10.1 Algorithme ALG.10 DC0uEMSi

La fonction Alg.10 correspond à la fonction de décodage de la courbe correspondant à la partition B-1. La fonction reçoit *ind*, l'indice du point de la courbe dont on recherche les coordonnées et le couple de paramètres *v2p* correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe.

Nous avons à la suite les commentaires concernant le code de l'algorithme ALG.10 :

Alg. 10 – DC0uEMSi

```

0 > DC0uEMSi:=proc(ind , v2p)
1 > local a, ty, am, ap, inds, nbrA, nbrB, nbrC, v2, ind0;
2 > a:=v2p[1];
3 > ty:=v2p[2];
4 > am:=a-1;
5 > ap:=a+1;
6 > inds:=ap^2-(ind+1);
7 > nbrA:=am^2;
8 > if a mod 2 = 0
9 > then nbrB:=2*am; nbrC:=2*ap;
10 > else nbrB:=2*ap; nbrC:=2*am;
11 > end if;
12 > if inds >= 0 and inds < nbrA
13 > then
14 > if ty = 0
15 > then v2:=DC0uM(inds, am);
16 > else v2:=DC0uSi(inds, am);
17 > end if;
18 > return([a-v2[1], v2[2]]);
19 > end if;
20 > if inds >= nbrA and inds < nbrA+nbrB
21 > then
22 > ind0:=inds-nbrA;
23 > v2:=DC0uWL(ind0, [1, 1], 2);
24 > return([a-v2[1], v2[2]+am]);
25 > end if;
26 > if inds >= nbrA+nbrB and inds < nbrA+nbrB+nbrC
27 > then
28 > ind0:=nbrC-1-(inds-nbrA-nbrB);
29 > v2:=DC0uWL(ind0, [1, 0], 2);
30 > return([v2[1], v2[2]]);
31 > end if;
32 > end proc;

```

- (i) **ligne 2-6** ⇒ Soit l'initialisation des variables
- (ii) **ligne 7** ⇒ Soit  $nbrA$  le nombre de points de la courbe A
- (iii) **ligne 8** ⇒ Soit le test de parité de la résolution de la courbe
- (iv) **ligne 9-10** ⇒ Soient  $nbrB$  et  $nbrC$  les nombres de points des courbes B et C
- (v) **ligne 12** ⇒ Soit l'appartenance du point au rectangle A

- (vi) **ligne 15** ⇒ Soit  $v2$  les coordonnées locales par rapport à la courbe uM
- (vii) **ligne 16** ⇒ Soit  $v2$  les coordonnées locales par rapport à la courbe uSi
- (viii) **ligne 18** ⇒ Soit le retour de l'indice global
- (ix) **ligne 20** ⇒ Soit l'appartenance du point au rectangle B
- (x) **ligne 22** ⇒ Soit  $ind0$  l'indice local pour la courbe B
- (xi) **ligne 23** ⇒ Soit  $v2$  les coordonnées locales par rapport à la courbe B
- (xii) **ligne 24** ⇒ Soit le retour de l'indice global
- (xiii) **ligne 26** ⇒ Soit l'appartenance du point au rectangle C
- (xiv) **ligne 28** ⇒ Soit  $ind0$  l'indice local pour la courbe C
- (xv) **ligne 29** ⇒ Soit  $v2$  les coordonnées locales par rapport à la courbe C
- (xvi) **ligne 30** ⇒ Soit le retour de l'indice global

## 2.11 Codage des courbes de type uEMSi

### 2.11.1 Algorithme ALG.11 CD0uEMSi

La fonction **Alg.11** est la fonction de codage inverse de la fonction de décodage **Alg.10**. La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'indice et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du morceau de courbe inscrit dans le carré de la partition B-1.

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.11** :

- (i) **ligne 3-7** ⇒ Soit l'initialisation des variables
- (ii) **ligne 8** ⇒ Soit le test de parité de la résolution
- (iii) **ligne 9-10** ⇒ Soient les nombres de points de la courbe A et B et les dimensions  $lgb$  et  $hb$

Alg. 11 – CD0uEMSi

```

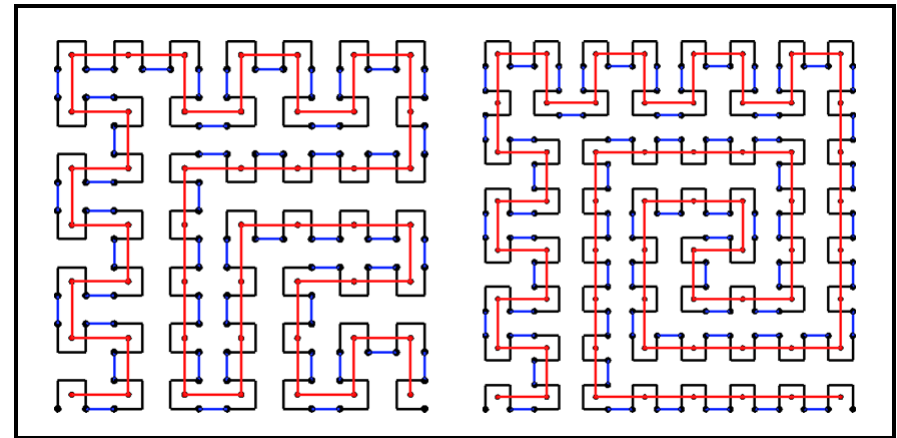
0 > CD0uEMSi:=proc(v2, v2p)
1 > local a, ty, c, am, ap, nbrA, nbrB, lgb, hb,
2 > nbrC, x, y, v2a, ind;
3 > a:=v2p[1];
4 > ty:=v2p[2];
5 > c:=a-2;
6 > am:=a-1;
7 > ap:=a+1;
8 > if a mod 2 = 0
9 > then nbrA:=2*ap; nbrB:=2*am; lgb:=2; hb:=ap;
10 > else nbrA:=2*am; nbrB:=2*ap; lgb:=0; hb:=am;
11 > end if;
12 > nbrC:=a^2;
13 > x:=v2[1];
14 > y:=v2[2];
15 > if x < 2 and y < hb
16 > then ind:=CD0uLg(v2, 1);
17 > return(ind);
18 > end if;
19 > if x >= lgb and y > c
20 > then
21 > if a mod 2 = 0
22 > then v2a:=[x-1, y-am];
23 > else v2a:=[x, y-am];
24 > end if;
25 > ind:=CD0uL(v2a, [1, 1], 2);
26 > return(ind+nbrA-lgb);
27 > end if;
28 > if x > 1 and y <= c
29 > then
30 > v2a:=[x-1, y];
31 > if ty mod 2 = 0
32 > then ind:=CD0uMi(v2a, am);
33 > else ind:=CD0uS(v2a, am);
34 > end if;
35 > return(nbrA+nbrB+ind-(2*a-1));
36 > end if;
37 > end proc;

```

- (vi) **ligne 17** ⇒ Soit le retour de l'indice global
- (vii) **ligne 19** ⇒ Soit le test d'appartenance du point à la courbe A
- (viii) **ligne 22-23** ⇒ Soit  $v2a$  les coordonnées locales du point de la courbe B
- (ix) **ligne 25** ⇒ Soit  $ind$  l'indice local du point pour la courbe B
- (x) **ligne 26** ⇒ Soit le retour de l'indice global
- (xi) **ligne 28** ⇒ Soit le test d'appartenance du point à la courbe A
- (xii) **ligne 30** ⇒ Soit  $v2a$  les coordonnées locales du point de la courbe C
- (xiii) **ligne 32** ⇒ Soit  $ind$  l'indice local du point pour la courbe uMi
- (xiv) **ligne 33** ⇒ Soit  $ind$  l'indice local du point pour la courbe uS
- (xv) **ligne 35** ⇒ Soit le retour de l'indice global

## 2.12 Chaînage des courbes de type uEMSi

La fonction Alg.12 est la fonction de chaînage qui permet de générer les cartes de la figure Alg.11.



- (iv) **ligne 12** ⇒ Soit  $nbrC$  le nombre de points de la courbe C
- (v) **ligne 15** ⇒ Soit le test d'appartenance du point à la courbe A

FIGURE 11 – Carte de chaînage des courbes de types uEMSi 0 et 1

### 2.12.1 Algorithme ALG.12 matEMSi

La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'identifiant de transformation ponctuelle et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe. La fonction retourne la valeur entière correspondant à la matrice ponctuelle de chaînage associée au point de coordonnées en entrée.

Nous avons à la suite les commentaires concernant le code de l'algorithme ALG.12 :

- (i) **ligne 3-14**  $\Rightarrow$  Soit l'initialisation des paramètres identique à la fonction **CD0uEMSi**
- (ii) **ligne 15**  $\Rightarrow$  Soit le test d'appartenance à un point de raccord singulier
- (iii) **ligne 18**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe A
- (iv) **ligne 19**  $\Rightarrow$  Soit  $cs$  l'indicateur de matrice de transformation ponctuelle pour la courbe A
- (v) **ligne 22**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe B
- (vi) **ligne 25**  $\Rightarrow$  Soit  $v2a$  les coordonnées locales de la courbe paire B
- (vii) **ligne 26**  $\Rightarrow$  Soit  $v2a$  les coordonnées locales de la courbe impaire B
- (viii) **ligne 28**  $\Rightarrow$  Soit  $cs$  l'indicateur de matrice de transformation ponctuelle pour la courbe B
- (ix) **ligne 35**  $\Rightarrow$  Soit  $cs$  l'indicateur de matrice de transformation ponctuelle pour la courbe uMi
- (x) **ligne 36**  $\Rightarrow$  Soit  $cs$  l'indicateur de matrice de transformation ponctuelle pour la courbe uS
- (xi) **ligne 20,29,38**  $\Rightarrow$  Soit le retour de  $cs$  l'indicateur de matrice de transformation ponctuelle
- (xii) **ligne 31**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe C

Alg. 12 – matEMSi

```

0 > matEMSi:=proc(v2,v2p)
1 > local a,ty,c,am,ap,nbrA,nbrB,
2 > lgb,hb,nbrC,x,y,cs,v2a,ind;
3 > a:=v2p[1];
4 > ty:=v2p[2];
5 > c:=a-2;
6 > am:=a-1;
7 > ap:=a+1;
8 > if a mod 2 = 0
9 > then nbrA:=2*ap; nbrB:=2*am; lgb:=2; hb:=ap;
10 > else nbrA:=2*am; nbrB:=2*ap; lgb:=0; hb:=am;
11 > end if;
12 > nbrC:=a^2;
13 > x:=v2[1];
14 > y:=v2[2];
15 > if a mod 2 = 0 and x = 1 and y = a
16 > then return(0);
17 > end if;
18 > if x < 2 and y < hb
19 > then cs:=matLg(v2,1);
20 > return(cs);
21 > end if;
22 > if x >= lgb and y > c
23 > then
24 > if a mod 2 = 0
25 > then v2a:=[x-1,y-am];
26 > else v2a:=[x,y-am];
27 > end if;
28 > cs:=matL(v2a,1);
29 > return(cs);
30 > end if;
31 > if x > 1 and y <= c
32 > then
33 > v2a:=[x-1,y];
34 > if ty mod 2 = 0
35 > then cs:=matMi(v2a,am);
36 > else ind:=CD0uS(v2a,am); cs:=matS(ind,am);
37 > end if;
38 > return(cs);
39 > end if;
40 > end proc;

```

(xiii) **ligne 33**  $\Rightarrow$  Soit  $v2a$  les coordonnées locales de la courbe C

## 2.13 Décodage des courbes de type uLMS

Le second type de courbe correspondant à la partition A-2 est dénommé **uLMS**. Il s'agit d'une courbe à résolution impaire  $m = 2\alpha + 1$ . La courbe est illustrée par la figure Fig.?? pour des résolutions 3 et 5.

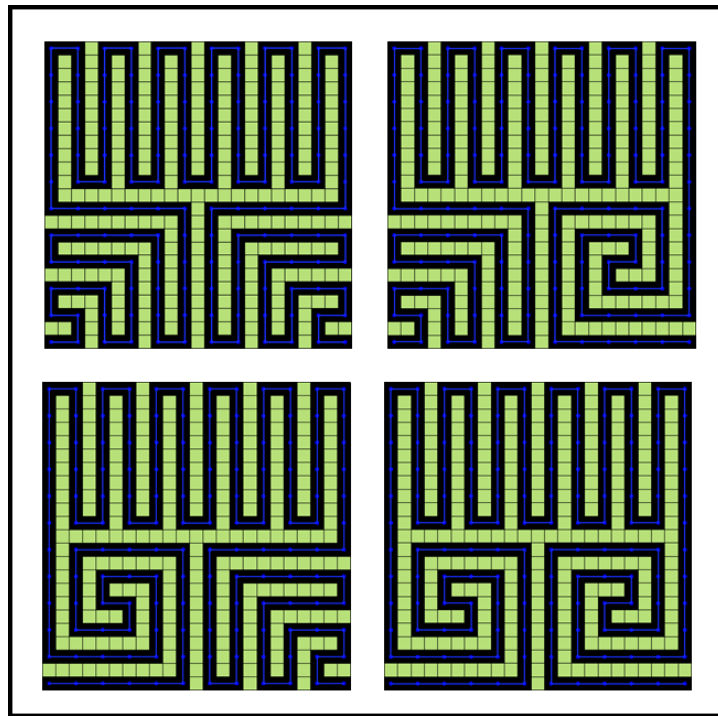


FIGURE 12 – Courbes de types DCOLMS 0,1,2,3

### 2.13.1 Algorithme ALG.13 DC0uLMS

La fonction **Alg.13** correspond à la fonction de décodage de la courbe correspondant à la partition B-1. La fonction reçoit  $ind$ , l'indice du point de la courbe dont on recherche les coordonnées et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe.

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.13** :

- (i) **ligne 2-8**  $\Rightarrow$  Soit l'initialisation des variables
- (ii) **ligne 9**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe A
- (iii) **ligne 11**  $\Rightarrow$  Soit le test du type de la courbe A
- (iv) **ligne 12**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales de la courbe uM
- (v) **ligne 13**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales de la courbe uSi
- (vi) **ligne 15**  $\Rightarrow$  Soit le retour du couple de coordonnées globales
- (vii) **ligne 17**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe B
- (viii) **ligne 19**  $\Rightarrow$  Soit  $ind0$  l'indice local sur la courbe C
- (ix) **ligne 20**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales de la courbe uwL
- (x) **ligne 21**  $\Rightarrow$  Soit le retour du couple de coordonnées globales
- (xi) **ligne 23**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe C
- (xii) **ligne 25**  $\Rightarrow$  Soit  $ind0$  l'indice local sur la courbe C
- (xiii) **ligne 26**  $\Rightarrow$  Soit le test du type de la courbe C
- (xiv) **ligne 28**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales de la courbe uMi
- (xv) **ligne 30**  $\Rightarrow$  Soit  $v2$  le couple de coordonnées locales de la courbe uS
- (xvi) **ligne 32**  $\Rightarrow$  Soit le retour du couple de coordonnées globales

### Alg. 13 – DC0uLMS

```

0 > DC0uLMS:=proc(ind , v2p)
1 > local a, ty , am, ap, nbrA, nbrB, nbrC, ind0 , v2;
2 > a:=v2p[1];
3 > ty:=v2p[2];
4 > am:=a-1;
5 > ap:=a+1;
6 > nbrA:=ap^2/4;
7 > nbrB:=nbrA*2;
8 > nbrC:=nbrA;
9 > if ind >= 0 and ind < nbrA
10 > then
11 >   if ty < 2
12 >     then v2:=DC0uM(ind , am/2+1);
13 >     else v2:=DC0uSi(ind , am/2+1);
14 >   end if;
15 >   return(v2);
16 > end if;
17 > if ind >= nbrA and ind < nbrA+nbrB
18 > then
19 >   ind0:=ind-nbrA;
20 >   v2:=DC0uwL(ind0 , [am/2 , 1] , 2);
21 >   return([v2[1] , v2[2]+am/2+1]);
22 > end if;
23 > if ind >= nbrA+nbrB and ind < nbrA+nbrB+nbrC
24 > then
25 >   ind0:=ind-3*a^2/4-a/2+5/4;
26 >   if ty mod 2 = 0
27 >     then
28 >       v2:=DC0uMi(ind0 , am/2+1);
29 >     else
30 >       v2:=DC0uS(ind0 , am/2+1);
31 >     end if;
32 >     return([v2[1]+am/2 , v2[2]]);
33 > end if;
34 > end proc;

```

## 2.14 Codage des courbes de type uLMS

### 2.14.1 Algorithme ALG.14 CD0uLMS

La fonction **Alg.14** est la fonction de codage inverse de la fonction de décodage **Alg.13**. La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'indice et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du morceau de courbe inscrit dans le carré de la partition B-1.

Nous avons à la suite les commentaires concernant le code de l'algorithme **ALG.14** :

- (i) **ligne 3-9**  $\Rightarrow$  Soit l'initialisation des variables
- (ii) **ligne 10**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe A
- (iii) **ligne 12**  $\Rightarrow$  Soit le test de type de courbe A
- (iv) **ligne 13**  $\Rightarrow$  Soit l'indice local par rapport à la courbe uM
- (v) **ligne 14**  $\Rightarrow$  Soit l'indice local par rapport à la courbe uSi
- (vi) **ligne 16**  $\Rightarrow$  Soit le retour de l'indice global par rapport à la courbe générale
- (vii) **ligne 18**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe B
- (viii) **ligne 20**  $\Rightarrow$  Soit  $v2a$  le couple de coordonnées locales du point par rapport à la courbe B
- (ix) **ligne 22**  $\Rightarrow$  Soit le retour de l'indice global par rapport à la courbe générale
- (x) **ligne 24**  $\Rightarrow$  Soit le test d'appartenance du point à la courbe C
- (xi) **ligne 26**  $\Rightarrow$  Soit  $v2a$  le couple de coordonnées locales du point par rapport à la courbe C
- (xii) **ligne 27**  $\Rightarrow$  Soit le test de type de courbe C
- (xiii) **ligne 29**  $\Rightarrow$  Soit  $ind0$  l'indice local par rapport à la courbe uMi
- (xiv) **ligne 31**  $\Rightarrow$  Soit  $ind0$  l'indice local par rapport à la courbe uS
- (xv) **ligne 33**  $\Rightarrow$  Soit le retour de l'indice global par rapport à la courbe générale

Alg. 14 – CD0uLMS

```

0 > CD0uLMS:=proc(v2, v2p)
1 > local a, ty, am, am2, ap, nbrA, x, y, ind, v2a, ind0;
2 > a:=v2p[1];
3 > ty:=v2p[2];
4 > am:=a-1;
5 > am2:=am/2;
6 > ap:=a+1;
7 > nbrA:=ap^2/4;
8 > x:=v2[1];
9 > y:=v2[2];
10 > if x <= am2 and y <= am2
11 > then
12 > if ty < 2
13 > then ind:=CD0uM(v2, am2+1);
14 > else ind:=CD0uSi(v2, am2+1);
15 > end if;
16 > return(ind);
17 > end if;
18 > if y > am2
19 > then
20 > v2a:=[v2[1], v2[2]-am2-1];
21 > ind0:=CD0uL(v2a, [am2, 1], 2);
22 > return(ind0+nbrA);
23 > end if;
24 > if x > am2 and y <= am2
25 > then
26 > v2a:=[v2[1]-am2, v2[2]];
27 > if ty mod 2 = 0
28 > then
29 > ind0:=CD0uMi(v2a, am2+1);
30 > else
31 > ind0:=CD0uS(v2a, am2+1);
32 > end if;
33 > return(ind0+3*a^2/4+a/2-5/4);
34 > end if;
35 > end proc;

```

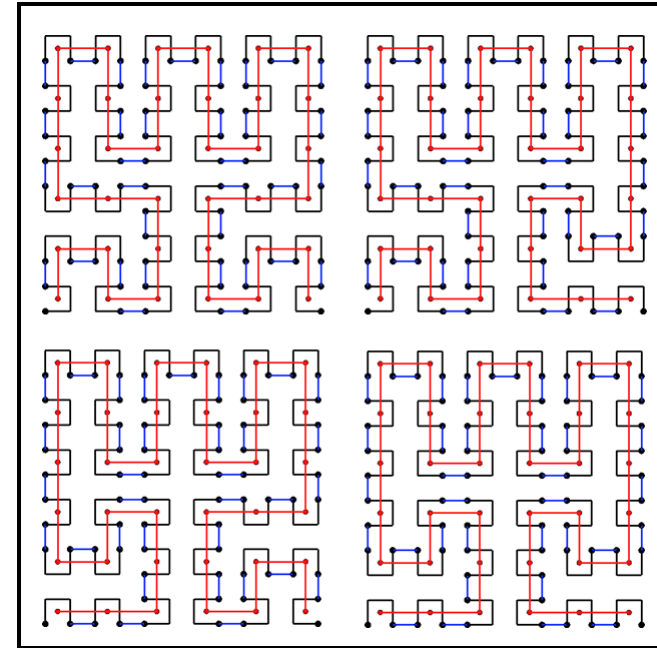


FIGURE 13 – Carte de chaînage des courbes de types uLMSi 0,1,2,3

## 2.15 Chaînage des courbes de type uLMS

La fonction Alg.15 est la fonction de chaînage qui permet de générer les cartes de la figure Alg.13.

### 2.15.1 Algorithme ALG.15 matLMS

La fonction reçoit  $v2$ , les coordonnées du point de la courbe dont on recherche l'identifiant de transformation ponctuelle et le couple de paramètres  $v2p$  correspondant à la résolution de la courbe et à la résolution du premier morceau de courbe contrôlant la symétrie de la courbe. La fonction retourne la valeur entière correspondant à la matrice ponctuelle de chaînage associée au point de coordonnées en entrée.

Nous avons à la suite les commentaires concernant le code de l'algorithme ALG.15 :

- (i) ligne 2-9  $\Rightarrow$  Soit l'initialisation des paramètres

Alg. 15 – matLMS

```

0 > matLMS:=proc(v2,v2p)
1 > local a,ty,am,am2,ap,nbrA,x,y,cs,ind,v2a;
2 > a:=v2p[1];
3 > ty:=v2p[2];
4 > am:=a-1;
5 > am2:=am/2;
6 > ap:=a+1;
7 > nbrA:=ap^2/4;
8 > x:=v2[1];
9 > y:=v2[2];
10 > if x <= am2 and y <= am2
11 > then
12 >   if ty < 2
13 >   then cs:=matM(v2,am2+1);
14 >   else
15 >     ind:=CD0uSi(v2,am2+1);
16 >     cs:=matSi(ind,am2+1);
17 >   end if;
18 >   return(cs);
19 > end if;
20 > if y > am2
21 > then
22 >   v2a:=[v2[1],v2[2]-am2-1];
23 >   cs:=matL(v2a,am2);
24 >   return(cs);
25 > end if;
26 > if x > am2 and y <= am2
27 > then
28 >   v2a:=[v2[1]-am2,v2[2]];
29 >   if ty mod 2 = 0
30 >   then
31 >     cs:=matMi(v2a,am2+1);
32 >   else
33 >     ind:=CD0uS(v2a,am2+1);
34 >     cs:=matS(ind,am2+1);
35 >   end if;
36 >   return(cs);
37 > end if;
38 > end proc;
```

- (ii) **ligne 10**  $\Rightarrow$  Soit l'appartenance du point à la courbe A  
 (iii) **ligne 12**  $\Rightarrow$  Soit le test  $\{0, 1, 2, 3\}$  concernant le type de courbe

- (iv) **ligne 15**  $\Rightarrow$  Soit  $ind$  l'indice local du point sur la courbe A  
 (v) **ligne 16**  $\Rightarrow$  Soit  $cs$  l'indicateur de transformation ponctuelle  
 (vi) **ligne 18**  $\Rightarrow$  Soit le retour de  $cs$  l'indicateur de transformation  
 (vii) **ligne 20**  $\Rightarrow$  Soit l'appartenance du point à la courbe B  
 (viii) **ligne 22**  $\Rightarrow$  Soit  $v2a$  le couple des coordonnées du point sur la courbe B  
 (ix) **ligne 23**  $\Rightarrow$  Soit  $cs$  l'indicateur de transformation ponctuelle  
 (x) **ligne 24**  $\Rightarrow$  Soit le retour de  $cs$  l'indicateur de transformation  
 (xi) **ligne 26**  $\Rightarrow$  Soit l'appartenance du point à la courbe C  
 (xii) **ligne 29**  $\Rightarrow$  Soit le test  $\{0, 1, 2, 3\}$  concernant le type de courbe  
 (xiii) **ligne 28**  $\Rightarrow$  Soit  $v2a$  le couple des coordonnées du point sur la courbe C  
 (xiv) **ligne 31**  $\Rightarrow$  Soit  $cs$  le calcul d'indicateur de transformation ponctuelle  
 (xv) **ligne 33**  $\Rightarrow$  Soit  $ind$  l'indice local du point sur la courbe C  
 (xvi) **ligne 34**  $\Rightarrow$  Soit  $cs$  le calcul d'indicateur de transformation ponctuelle  
 (xvii) **ligne 36**  $\Rightarrow$  Soit le retour de  $cs$  l'indicateur de transformation ponctuelle

### 3 Génération par rotors de niveau 1

Les courbes de Hilbert peuvent-être obtenues et généralisées par une seconde méthode en particulierisant la signature des metacourbes de type **DC1**, et en forçant les résolutions des courbes rotors à 1 tout en choisissant un type  $TP$  quelconque de courbe de classe **U**

*Fonctions de couplage et chaînage  
des courbes et metacourbes de Gray  
généralisant les courbes de Hilbert*

$$\begin{aligned} \mathbf{lsDC1} &= \{m, lsRS, lsTP\} \\ \text{avec } lsRS &= \{1, \dots, 1\}_n \\ \text{et } lsTP &= \{TP_U\} \end{aligned}$$

La metacourbe ainsi définie représente les  $n$  premières itérations ou ordres de la courbe de Hilbert classique qui est dans notre représentation constituée de courbes rotor de résolution 1. Contrairement à la construction d'Hilbert basée sur une courbe élémentaire discrète de classe **U** non typée, nous illustrons des généralisations de la courbe d'Hilbert classique en typant la metacourbe, et en modifiant les résolutions des ordres  $> 2$ . Ces courbes seront dénommées "hilbertiennes". La séquence des figures **Fig.14**, **Fig.15** et **Fig.16**, illustre des metacourbes "hilbertiennes" de types respectifs 0, 1, 2 et de signatures diophantiennes respectives  $\{1, 1, 1, 1\}$ ,  $\{1, 1, 1, 3\}$ , et  $\{1, 1, 1, 5\}$ . Cette approche revient à modifier en premier la résolution des rotors feuilles. Une autre approche consiste à modifier en premier le rotor principal racine. La figure **Fig.17** illustre une metacourbe constituée de courbes de Hilbert d'ordre 3 chaînées à partir d'une courbe rotor racine de type 1 et de résolution 4.

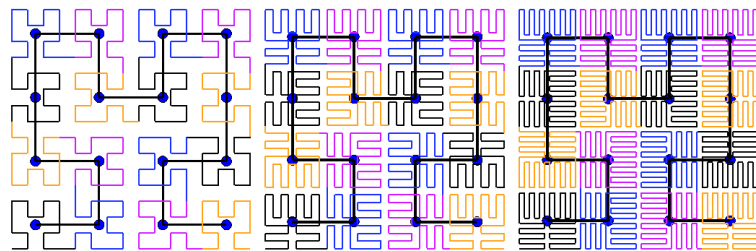


FIGURE 14 – Courbe de Hilbert à rotors de type lacet

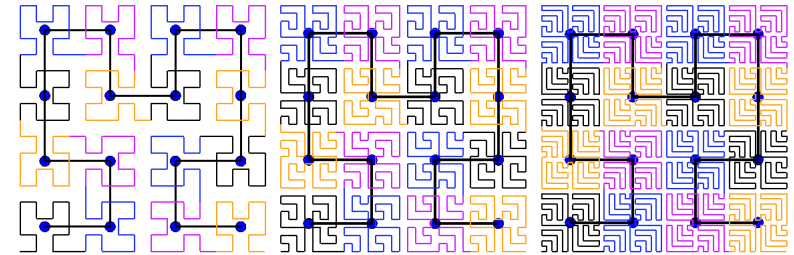


FIGURE 15 – Courbe de Hilbert à rotors de type méandre

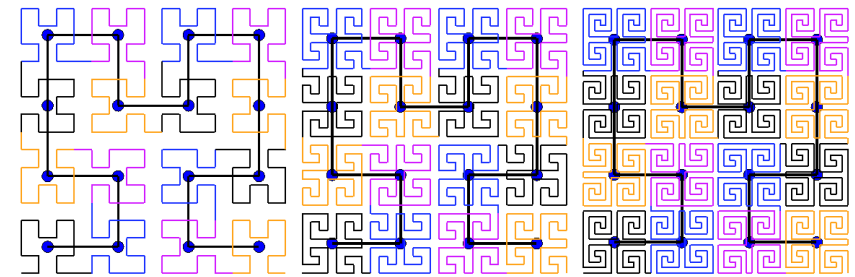


FIGURE 16 – Courbe de Hilbert à rotors de type spirale

Les coordonnées des points des deux courbes du bas de la figure de niveau 1, sont quant à elles générées par les séquences suivantes ;

$$\begin{aligned} &[seq(\mathbf{DC1}(i, 2, [3, 3]), [5, [3, 1], 5, [3, 1]]) \\ &[seq(\mathbf{DC1}(i, 2, [3, 3]), [5, [3, 2], 5, [3, 2]]) \end{aligned}$$

Le nombre  $m$  du niveau de rotors utilisé correspond au niveau de la courbe de Hilbert générée par cette première méthode.

### 3.1 Fractalisation de Hilbert généralisée

Le mécanisme récursif de Hilbert correspond à une fractalisation des courbes successives de Hilbert, chaque morceau de courbe en forme de **U** étant subdivisée en quatre nouveaux morceaux de même forme. Une fractalisation généralisée des courbes de Hilbert consiste à remplacer la courbe de Hilbert de niveaux  $m = 1$  par toute autre courbe de Gray. La figure **Fig.17** illustre une fractalisation d'une courbe de type **uSi**.

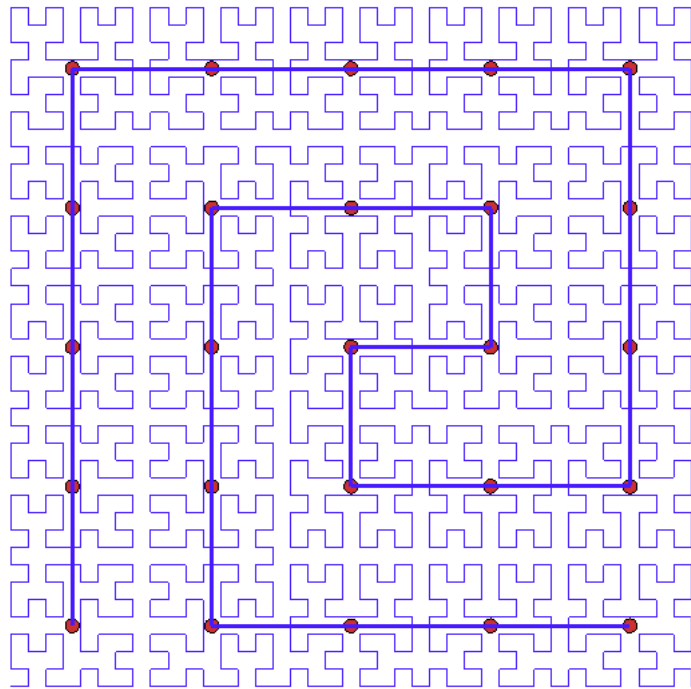


FIGURE 17 – DC1(i,4,[4,1,1,1],[1]),i=0..1599

## 4 Génération par rotors de niveau 3

Les courbes de Hilbert peuvent-êtré obtenues et généralisées par une troisième méthode en particulierisant la signature des metacourbes de type **DC3**. Cette approche correspond à la partition B-3 et permet d'obtenir des courbes de résolution impaire. Le principe des courbes à rotors de niveau 3 est décrit en **Bib.5** et s'applique à des courbes de nature hybride.La figure **Fig.18** illustre des exemples de courbes de Hilbert générées par rotors de niveau 3.

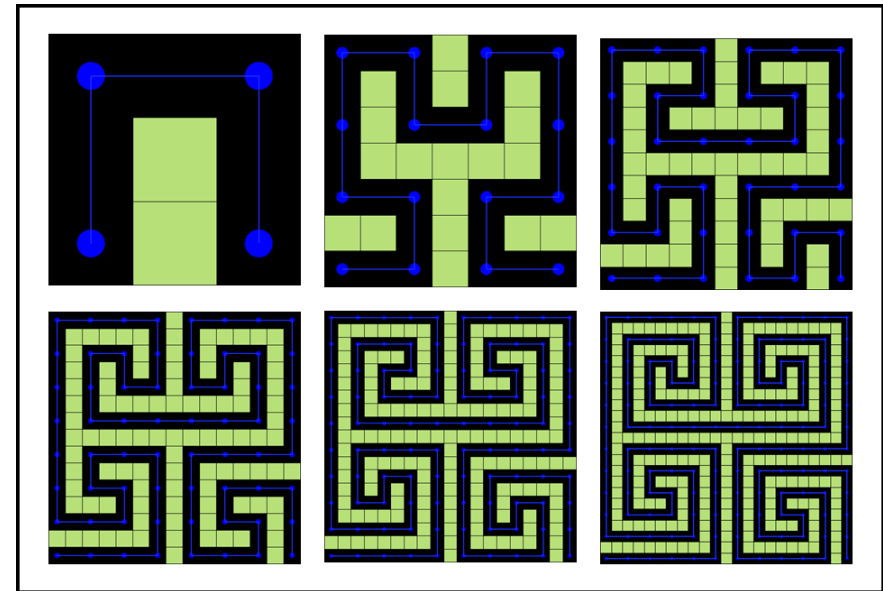


FIGURE 18 – Courbes de Hilbert par rotors de niveau 3

Ces courbes sont générées par des séquences de la forme **hilDC3(a)** où le script **Scr.16** est présenté ci-dessous.

Alg. 16 – hilDC3

```
0 > hilDC3:=proc(a)
1 > local m,lsRS,lsTP,lsSY,lsSY0,lsSY1,lsSY2,
2 > lsHE0,lsHE1,lsHE,nbr,RES;
3 > "RS=====";
4 > m:=2;
5 > lsRS:=[1,a];
6 > "TP=====";
7 > lsTP:=[0,[],0,[]];
8 > "SYS=====";
9 > lsSY0:=[1];
10 > lsSY1:=[1,3];
11 > lsSY:=[lsSY0,lsSY1];
12 > "HE=====";
13 > lsHE0:=[[1,3,[]]];
14 > lsHE1:=[[1,2,[]],[2,1,[]],[4,1,[]],[3,1,[]]];
15 > lsHE:=[lsHE0,lsHE1];
16 > nbr:=NBRSIGn(m,lsRS,2);
17 > RES:=[seq(DC3(i,m,lsRS,lsTP,lsSY,lsHE),i=0..nbr)];
18 > return(RES);
19 > end proc;
```

## 5 Notes bibliographiques

La théorie des courbes remplissantes et en particulier des courbes de Hilbert est introduite dans un grand nombre d'ouvrages. Les références Ref.[9, 10, 12] permettent de se familiariser avec les principales notions, algorithmes de construction et d'indexation de ces courbes. La théorie des courbes particulières de Hilbert prend naissance avec l'article Ref.[3] contextualisé quelques années plus tard par Moore Ref.[4]. Ces travaux s'inscrivent à la suite des travaux précurseurs de Cantor Ref.[1] et de Péano Ref.[2]. Une généralisation de la construction des courbes remplissantes est suggérée par Péano Ref.[5] par le moyen de schémas constructifs. La liaison des courbes remplissantes et des fractales est réalisée dans un livre de Sainte-Laguë Ref.[6] bien avant les travaux de Mandelbrot. Les courbes de

Hilbert sont présentées comme appartenant à la théorie des fractales dans Ref.[8]. Les différents modes de formalisation des courbes de Hilbert sont développées en particulier dans Ref.[14]. L'approche énumérative des courbes remplissantes relève de l'énumération des chemins hamiltoniens couvrant des grilles carrées. Deux sites principaux permettent d'explorer cette approche Ref.[16, 15]. Les notions de localité des courbes de Hilbert sont couvertes en Ref.[11, 14]. Finalement le principe des courbes remplissantes par des schémas constructifs est introduit par Wunderlich Ref.[7] et généralisé par Rotgé au travers de la théorie des metacourbes de Gray Ref.[24] qui s'appuie elle-même sur une série de nouvelles familles de courbes remplissantes dites de Gray Ref.[20, 21, 22, 23, 18, 17, 19, 25]. La théorie des metacourbes de Gray permet en particulier de s'affranchir des contraintes historiques des courbes de Hilbert construites récursivement dans des carrés. La référence Ref.[13] présente également des stratégies pour échapper aux contraintes initiales des courbes de Hilbert.

## Références

- [1] Georg Cantor. Ein beitrag zur mannigfaltigkeitslehre. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1878(84) :242–258, 1878.
- [2] G. Peano. Sur une courbe, qui remplit toute une aire plane. *Mathematische Annalen*, 36 :157–160, 1890.
- [3] David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. *Mathematische Annalen*, 38 :459–460, 1891.
- [4] Eliakim Hastings Moore. On certain crinkly curves. *Transactions of the American Mathematical Society*, 1(1) :72–90, 1900.
- [5] G. Peano. *Formulario mathematico*. Bocca, 1908.

- [6] André Sainte-Laguë. Le monde des formes. Librairie Arthème Fayard, 1948.
- [7] W Wunderlich. Über peano-kurven. *Elemente der Mathematik*, 28 :1–10, 1973.
- [8] Heinz-Otto Peitgen, Hartmut Jürgens, Dietmar Saupe, and Mitchell J Feigenbaum. Chaos and fractals : new frontiers of science. Springer, 1992.
- [9] Hans Sagan. Space-filling curves. Springer Science & Business Media, 1994.
- [10] Levi Valgaerts. Space-filling curves an introduction. *Technical University Munich*, 2005.
- [11] Konstantin Evgen'evich Bauman. The dilation factor of the peano-hilbert curve. *Mathematical Notes*, 80(5) :609–620, 2006.
- [12] Michael Bader. Space-filling curves : an introduction with applications in scientific computing. Springer Science & Business Media, 2012.
- [13] Christian Böhm, Martin Perdacher, and Claudia Plant. A novel hilbert curve for cache-locality preserving loops. *IEEE Transactions on Big Data*, 7(2) :241–254, 2018.
- [14] Jeffrey Shallit. Hilbert's spacefilling curve described by automatic, regular, and synchronized sequences. *arXiv preprint arXiv :2106.01062*, 2021.
- [15] Self avoiding walks. <https://datagenetics.com/blog/december22018/index.html>, 2022.
- [16] Nathan Clisby. Hamiltonian path generator. [https://clisby.net/projects/hamiltonian\\_path/](https://clisby.net/projects/hamiltonian_path/), 2022.
- [17] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques à centre. <https://hal.archives-ouvertes.fr/hal-03745896/file/Article13.pdf>, August 2022. working paper or preprint.
- [18] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques à noyaux. <https://hal.archives-ouvertes.fr/hal-03745216/file/Article14.pdf>, August 2022. working paper or preprint.
- [19] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques à réflexions. <https://hal.archives-ouvertes.fr/hal-03744206/file/Article15.pdf>, August 2022. working paper or preprint.
- [20] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme de lacets. <https://hal.archives-ouvertes.fr/hal-03737365/file/Article10.pdf>, July 2022. working paper or preprint.
- [21] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme de méandres. <https://hal.archives-ouvertes.fr/hal-03737998/file/Article11.pdf>, July 2022. working paper or preprint.
- [22] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme de spirales. <https://hal.archives-ouvertes.fr/hal-03738025/file/Article12.pdf>, July 2022. working paper or preprint.
- [23] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme d'escaliers. <https://hal.archives-ouvertes.fr/hal-03745141/file/Article16.pdf>, August 2022. working paper or preprint.
- [24] Jean-François Rotgé. Fonctions de couplage et chaînage des metacourbes de Gray surfaciques à rotors. <https://hal.archives-ouvertes.fr/hal-03768804/file/Article2.pdf>, September 2022. working paper or preprint.

*Fonctions de couplage et chaînage  
des courbes et metacourbes de Gray  
généralisant les courbes de Hilbert*

- [25] Jean-François Rotgé. Principes de chaînage des courbes de Gray. <https://hal.archives-ouvertes.fr/hal-03759184/file/Article5.pdf>, August 2022. working paper or preprint.