

Exemples de constructions de fractales par le jeu du chaos : commentaires

1—On modifie l’algorithme précédent. Il suffit de rajouter une fonction mais il faut ici veiller à considérer deux nouvelles variables notées ici x_{new} et y_{new} (si ça n’a pas été fait dans l’exemple précédent) comme coordonnées du nouveau point créé.

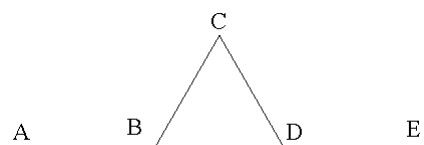
```
VARIABLES
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  N EST_DU_TYPE NOMBRE
  i EST_DU_TYPE NOMBRE
  D EST_DU_TYPE NOMBRE
  xnew EST_DU_TYPE NOMBRE
  ynew EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE N
  //N est le nombre de points à construire
  x PREND_LA_VALEUR random()
  y PREND_LA_VALEUR random()
  TRACER_POINT (x,y)
  //x et y sont donc les abscisses et ordonnées des points à construire
  POUR i ALLANT_DE 1 A N-1
    DEBUT_POUR
      D PREND_LA_VALEUR 1+floor(4*random())
      SI (D==1) ALORS
        DEBUT_SI
          xnew PREND_LA_VALEUR x/3
          //xnew correspond à x(n+1) alors que x correspond à x(n)
          ynew PREND_LA_VALEUR y/3
          //ynew correspond à y(n+1) alors que y correspond à y(n)
          FIN_SI
        SINON
          DEBUT_SINON
            SI (D==2) ALORS
              DEBUT_SI
                xnew PREND_LA_VALEUR x/6-(y*sqrt(3))/6+1/3
                ynew PREND_LA_VALEUR (x*sqrt(3))/6+y/6
                FIN_SI
              SINON
                DEBUT_SINON
                  SI (D==3) ALORS
                    DEBUT_SI
                      xnew PREND_LA_VALEUR x/6+(y*sqrt(3))/6+1/2
                      ynew PREND_LA_VALEUR -(x*sqrt(3))/6+y/6+sqrt(3)/6
                      FIN_SI
                    SINON
                      DEBUT_SINON
                        xnew PREND_LA_VALEUR x/3+2/3
                        ynew PREND_LA_VALEUR y/3
                        FIN_SINON
                      FIN_SINON
                    FIN_SINON
                  TRACER_POINT (xnew,ynew)
                  x PREND_LA_VALEUR xnew
                  y PREND_LA_VALEUR ynew
                FIN_POUR
              FIN_ALGORITHME
```

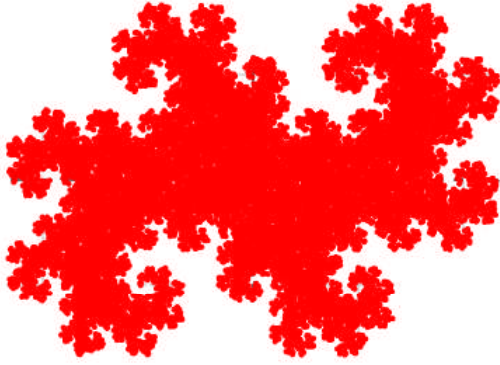


En terminale, en approfondissement lors de l’accompagnement personnalisé, on peut mettre en évidence les IFS utilisées ici en remarquant que la courbe de Koch est l’attracteur dans le plan des 4 similitudes de rapport $1/3$ transformant (A, E) successivement en (A, B), (B, C), (C, D) et (D, E) (avec $BD = AB$).

Pour cela on raisonne à l’aide des complexes en adaptant le travail effectué sur les transformations au nouveau programme de terminale.

Il est à noter que, selon le repère choisi, les IFS obtenues ne sont pas forcément les mêmes que celles qui ont été données dans l’énoncé.



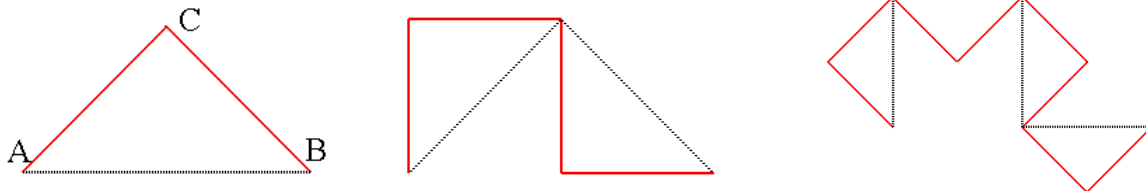


```

VARIABLES
  X EST_DU_TYPE NOMBRE
  Y EST_DU_TYPE NOMBRE
  D EST_DU_TYPE NOMBRE
  I EST_DU_TYPE NOMBRE
  N EST_DU_TYPE NOMBRE
  Xnew EST_DU_TYPE NOMBRE
  Ynew EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE N
  X PREND_LA_VALEUR random()
  Y PREND_LA_VALEUR random()
  TRACER_POINT (X, Y)
  POUR I ALLANT_DE 1 A N-1
    DEBUT_POUR
      D PREND_LA_VALEUR floor(2*random())
      SI (D==0) ALORS
        DEBUT_SI
          Xnew PREND_LA_VALEUR X/2-Y/2+1
          Ynew PREND_LA_VALEUR (X+Y)/2
        FIN_SI
      SINON
        DEBUT_SINON
          Xnew PREND_LA_VALEUR X/2-Y/2-1
          Ynew PREND_LA_VALEUR (X+Y)/2
        FIN_SINON
      TRACER_POINT (Xnew, Ynew)
      X PREND_LA_VALEUR Xnew
      Y PREND_LA_VALEUR Ynew
    FIN_POUR
  FIN_ALGORITHME

```

En terminale, en approfondissement lors de l'accompagnement personnalisé, on peut mettre en évidence les IFS utilisées ici en remarquant que la courbe du dragon est l'attracteur dans le plan des deux similitudes directes transformant, l'une (A,B) en (A,C) , l'autre (A,B) en (B,C) où ABC est un triangle rectangle isocèle en C ainsi que le montrent les figures ci-dessous.



On raisonne alors à l'aide des complexes en adaptant le travail effectué sur les transformations au nouveau programme de terminale.

Il est à noter que, selon le repère choisi, les IFS obtenues ne sont pas forcément les mêmes que celles qui ont été données dans l'énoncé.

```

VARIABLES
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  N EST_DU_TYPE NOMBRE
  i EST_DU_TYPE NOMBRE
  D EST_DU_TYPE NOMBRE
  xnew EST_DU_TYPE NOMBRE
  ynew EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  x PREND_LA_VALEUR 0
  y PREND_LA_VALEUR 0
  TRACER_POINT (x, y)
  LIRE N
  POUR i ALLANT_DE 1 A N-1
    DEBUT_POUR
      D PREND_LA_VALEUR random()
      SI (D<0.01) ALORS
        DEBUT_SI
          xnew PREND_LA_VALEUR 0
          ynew PREND_LA_VALEUR 0.16*y
          FIN_SI
        SINON
          DEBUT_SINON
            SI (D>0.01 ET D<0.08) ALORS
              DEBUT_SI
                xnew PREND_LA_VALEUR 0.2*x-0.26*y
                ynew PREND_LA_VALEUR 0.23*x+0.22*y+1.6
                FIN_SI
              SINON
                DEBUT_SINON
                  SI (D>0.08 ET D<0.15) ALORS
                    DEBUT_SI
                      xnew PREND_LA_VALEUR -0.15*x+0.28*y
                      ynew PREND_LA_VALEUR 0.26*x+0.24*y+0.44
                      FIN_SI
                    SINON
                      DEBUT_SINON
                        xnew PREND_LA_VALEUR 0.85*x+0.04*y
                        ynew PREND_LA_VALEUR -0.04*x+0.85*y+1.6
                        FIN_SINON
                      FIN_SINON
                    FIN_SINON
                TRACER_POINT (xnew, ynew)
                x PREND_LA_VALEUR xnew
                y PREND_LA_VALEUR ynew
              FIN_POUR
            FIN_ALGORITHME

```



```

VARIABLES
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  N EST_DU_TYPE NOMBRE
  i EST_DU_TYPE NOMBRE
  D EST_DU_TYPE NOMBRE
  xnew EST_DU_TYPE NOMBRE
  ynew EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  x PREND_LA_VALEUR random()
  y PREND_LA_VALEUR random()
  TRACER_POINT (x,y)
  LIRE N
  POUR i ALLANT_DE 1 A N-1
    DEBUT_POUR
      D PREND_LA_VALEUR random()
      SI (D<0.3) ALORS
        DEBUT_SI
          xnew PREND_LA_VALEUR 0.5*x
          ynew PREND_LA_VALEUR 0.5*y+0.4
          FIN_SI
        SINON
          DEBUT_SINON
            SI (D>0.3 ET D<0.6) ALORS
              DEBUT_SI
                xnew PREND_LA_VALEUR 0.7*x
                ynew PREND_LA_VALEUR 0.7*y
                FIN_SI
              SINON
                DEBUT_SINON
                  SI (D>0.6 ET D<0.8) ALORS
                    DEBUT_SI
                      xnew PREND_LA_VALEUR 0.4*x+0.3*y+0.22
                      ynew PREND_LA_VALEUR -0.3*x+0.4*y+0.15
                      FIN_SI
                    SINON
                      DEBUT_SINON
                        xnew PREND_LA_VALEUR 0.4*x-0.3*y-0.22
                        ynew PREND_LA_VALEUR 0.3*x+0.4*y+0.15
                        FIN_SINON
                      FIN_SINON
                    FIN_SINON
                  TRACER_POINT (xnew, ynew)
                x PREND_LA_VALEUR xnew
                y PREND_LA_VALEUR ynew
              FIN_POUR
            FIN_ALGORITHME

```



5—En choisissant différentes valeurs pour les nombres de a et f , on définit ainsi des transformations auxquelles on associe des probabilités. On définit ainsi des IFS produisant des images fractales pourvu que les applications considérées soient contractantes.

	a	b	c	d	e	f	p_i
f_1	0.4	0	0	0	0.3	-0.4	0.2
f_2	-0.7	0	0	0	0.7	0	0.3
f_3	-0.4	0.3	0.5	-0.3	0.4	0.15	0.4
f_4	0.3	-0.5	0.2	0.3	-0.1	0.1	0.1

