

- Voici un petit dessin réalisé avec AlgoBox. On n'oubliera pas d'activer la case "Utiliser un repère".

On faudra définir les bornes des axes de la manière suivante :

$X_{min} : 0$; $X_{max} : 1$; $Y_{min} : 0$; $Y_{max} : 1$

```

▼VARIABLES
├── i EST_DU_TYPE NOMBRE
└── j EST_DU_TYPE NOMBRE
▼DEBUT_ALGORITHME
  ▼POUR i ALLANT_DE 1 A 2
    └── DEBUT_POUR
          ▼POUR j ALLANT_DE 1 A 20
            └── DEBUT_POUR
                  TRACER.SEGMENT (i%2,j/20)->(1-j/20,i%2)
            └── FIN_POUR
          └── FIN_POUR
    └── FIN_POUR
  └── FIN_ALGORITHME

```

- cet algorithme illustre la méthode d'Euler pour construire la courbe représentative de la fonction exponentielle. On n'oubliera pas d'activer la case "Utiliser un repère".

On faudra définir les bornes des axes de la manière suivante :

$X_{min} : -10$; $X_{max} : 3$; $Y_{min} : -1$; $Y_{max} : 10$

```

▼VARIABLES
├── i EST_DU_TYPE NOMBRE
├── j EST_DU_TYPE NOMBRE
├── pas EST_DU_TYPE NOMBRE
├── derive EST_DU_TYPE NOMBRE
├── x1 EST_DU_TYPE NOMBRE
├── y1 EST_DU_TYPE NOMBRE
├── x2 EST_DU_TYPE NOMBRE
└── y2 EST_DU_TYPE NOMBRE
▼DEBUT_ALGORITHME
  └── pas PREND_LA_VALEUR 0.5
    ▼POUR i ALLANT_DE 0 A 1
      └── DEBUT_POUR
            x1 PREND_LA_VALEUR 0
            y1 PREND_LA_VALEUR 1
          ▼POUR j ALLANT_DE 0 A 100
            └── DEBUT_POUR
                  derive PREND_LA_VALEUR y1
                  x2 PREND_LA_VALEUR x1+pas*(i*2-1)
                  y2 PREND_LA_VALEUR y1+derive*pas*(i*2-1)
                  TRACER.SEGMENT (x1,y1)->(x2,y2)
            └── x1 PREND_LA_VALEUR x2
            └── y1 PREND_LA_VALEUR y2
          └── FIN_POUR
        └── FIN_POUR
      └── FIN_POUR
    └── FIN_ALGORITHME

```

- cet algorithme présente l'algorithme d'Euclide de deux nombres demandés lors de l'exécution de celui-ci :

```

▼VARIABLES
├── a EST_DU_TYPE NOMBRE
├── b EST_DU_TYPE NOMBRE
└── c EST_DU_TYPE NOMBRE
▼DEBUT_ALGORITHME
  └── LIRE a
    └── LIRE b
      ▼SI (b>a) ALORS
        └── DEBUT_SI
              c PREND_LA_VALEUR a
              a PREND_LA_VALEUR b
              b PREND_LA_VALEUR c
            └── FIN_SI
        └── c PREND_LA_VALEUR a%b
      ▼TANT_QUE (c!=0) FAIRE
        └── DEBUT_TANT_QUE
              a PREND_LA_VALEUR b
              b PREND_LA_VALEUR c
              c PREND_LA_VALEUR a%b
            └── FIN_TANT_QUE
        └── FIN_TANT_QUE
      └── AFFICHER b
    └── FIN_ALGORITHME

```

- cet algorithme permet d'obtenir autant de décimale que voulues dans la division de deux nombres.

```

▼VARIABLES
├── a EST_DU_TYPE NOMBRE
├── b EST_DU_TYPE NOMBRE
├── c EST_DU_TYPE NOMBRE
├── i EST_DU_TYPE NOMBRE
└── prec EST_DU_TYPE NOMBRE
▼DEBUT_ALGORITHME
  └── LIRE a
    └── LIRE b
      └── prec PREND_LA_VALEUR 200
        ▼POUR i ALLANT_DE 1 A prec
          └── DEBUT_POUR
                c PREND_LA_VALEUR floor(a/b)
              └── AFFICHER c
            ▼SI (i==1) ALORS
              └── DEBUT_SI
                    AFFICHER "."
                └── FIN_SI
              └── a PREND_LA_VALEUR (a-c*b)*10
            └── FIN_POUR
          └── FIN_POUR
        └── FIN_ALGORITHME

```

- cet algorithme présente le placement aléatoire de points dans le plan $[0; 1]^2$; on ne placera que les points dont la distance à l'origine du repère sont inférieurs à 1. La dernière valeur affichée est la fréquence des points présents dans le disque de centre O et de rayon 1.

```

▼VARIABLES
|
| - i EST_DU_TYPE NOMBRE
| - compt EST_DU_TYPE NOMBRE
| - pas EST_DU_TYPE NOMBRE
| - x EST_DU_TYPE NOMBRE
| - y EST_DU_TYPE NOMBRE
|
▼DEBUT_ALGORITHME
|
| - pas PREND_LA_VALEUR 1000
| - compt PREND_LA_VALEUR 0
|
| ▼POUR i ALLANT_DE 0 A pas
| |
| | - DEBUT_POUR
| | - x PREND_LA_VALEUR random()
| | - y PREND_LA_VALEUR random()
| |
| | ▼SI (pow(x,2)+pow(y,2)<=1) ALORS
| | |
| | | - DEBUT_SI
| | | - TRACER_POINT (x,y)
| | | - compt PREND_LA_VALEUR compt+1
| | | - FIN_SI
| | - FIN_POUR
| - compt PREND_LA_VALEUR compt/pas
| - AFFICHER compt
|
| - FIN_ALGORITHME

```

- on simule le lancé de deux dés. On considère l'évènement :

A : "La somme des dés est inférieure ou égale à 4". On représente dans un repère la fréquence de l'évènement A lorsque l'effectif des essais augmentent.

On utilisera un repère dans AlgoBox en cochant l'option "Utiliser un repère".

On définira ainsi les bornes des axes de la manière suivante :

$X_{min} : 0$; $X_{max} : 1$; $Y_{min} : 0$; $Y_{max} : 1$

```

▼VARIABLES
|
| - i EST_DU_TYPE NOMBRE
| - a EST_DU_TYPE NOMBRE
| - b EST_DU_TYPE NOMBRE
| - val EST_DU_TYPE NOMBRE
| - compt EST_DU_TYPE NOMBRE
| - mmax EST_DU_TYPE NOMBRE
|
▼DEBUT_ALGORITHME
|
| - val PREND_LA_VALEUR 4
| - mmax PREND_LA_VALEUR 3000
|
| ▼POUR i ALLANT_DE 1 A mmax
| |
| | - DEBUT_POUR
| | - b PREND_LA_VALEUR floor(random()*6)+1
| | - a PREND_LA_VALEUR floor(random()*6)+1
| |
| | ▼SI ((a+b)<=val) ALORS
| | |
| | | - DEBUT_SI
| | | - compt PREND_LA_VALEUR compt+1
| | | - FIN_SI
| | - TRACER_POINT (i/mmax,compt/i)
| | - FIN_POUR
| - compt PREND_LA_VALEUR compt/mmax
| - AFFICHER compt
|
| - FIN_ALGORITHME

```