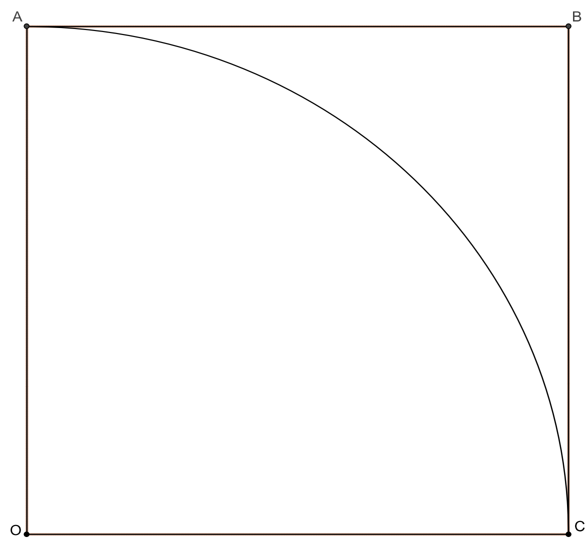
Algorithme PanaMaths
→ Estimation de π par une méthode de Monte-Carlo

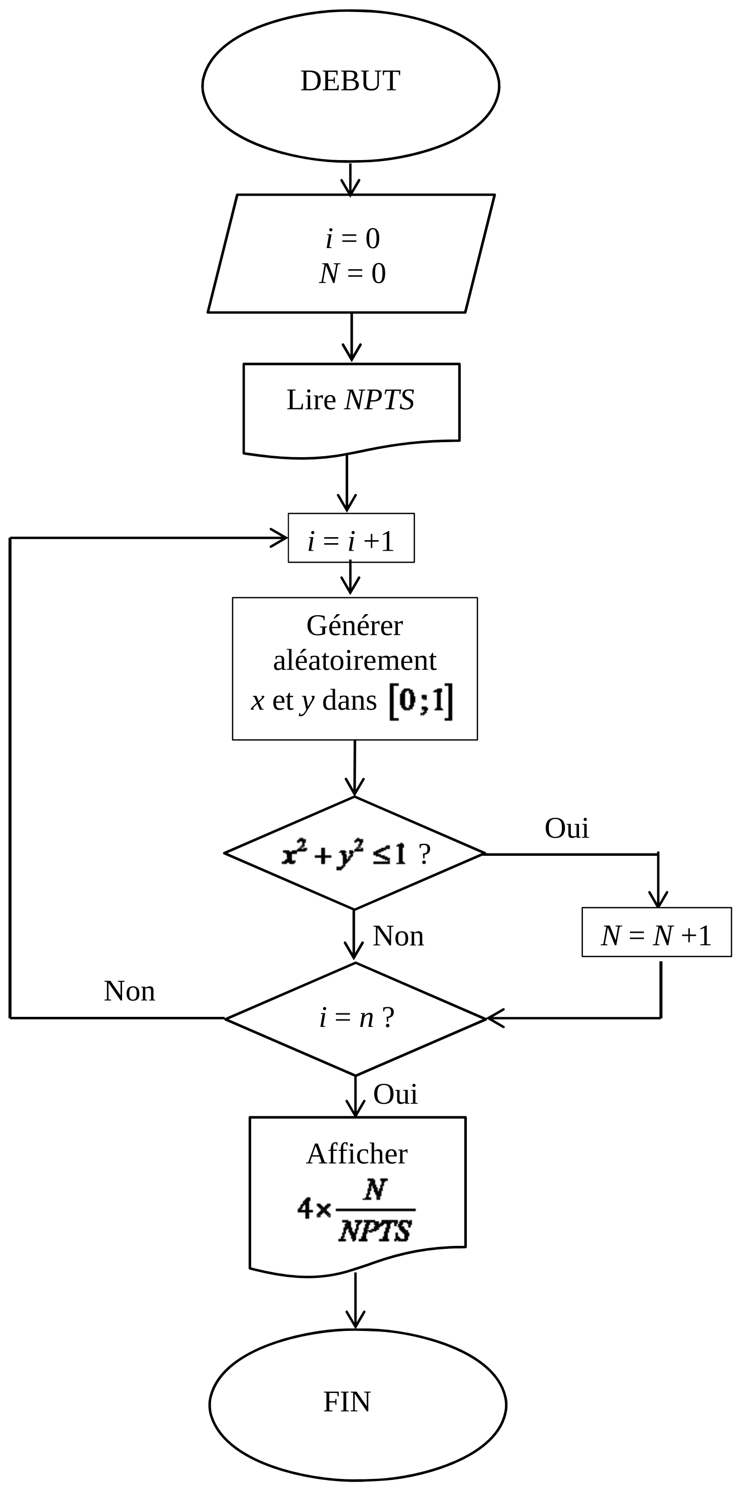
Introduction : quelques éléments mathématiques

Une méthode de Monte-Carlo permettant d'obtenir une estimation de π repose sur le résultat suivant : si l'on considère dans le plan le carré de sommets $\mathrm{O}(0\,;0)$, $\mathrm{A}(0\,;1)$, $\mathrm{B}(1\,;1)$ et $\mathrm{C}(1\,;0)$ (cf. la figure ci-dessous) et si l'on choisit un point au hasard dans ce carré (version à deux dimensions de la loi uniforme sur le segment $[0\,;1]$) alors la probabilité que ce point se trouve dans le quart de cercle de centre O passant par les points A et C est égale à $\frac{\pi}{4}$ (rapport de l'aire du demi-cercle et de celle du carré).



Le principe de la méthode numérique, et donc de l'algorithme proposé, consiste à tirer au hasard *NPTS* points dans le carré ABCD (donc à obtenir $2\times NPTS$ réalisations (*NPTS* abscisses et *NPTS* ordonnées) de la loi uniforme sur le segment $[0\,;1]$). Après chaque tirage d'un couple de coordonnées $(x\,;y)$, on effectue le test « $x^2+y^2\leq 1$? ».
Si la réponse est positive on incrémente un compteur *N*, sinon, on ne fait rien.
π est alors finalement estimé par $4\times\frac{N}{NPTS}$

Organigramme



Au niveau de la mise en œuvre de cet algorithme, tous les points sont affichés dans un graphique : en vert s'ils appartiennent au quart de cercle, en rouge sinon (cf. l'algorithme AlgoBox fourni ci-après). On notera que la couleur n'apparaît pas explicitement dans l'algorithme ci-dessous mais uniquement au moment où on ajoute dans AlgoBox la ligne correspondant à un tel affichage.

L'algorithme AlgoBox

Voici l'algorithme que vous pouvez tester en ligne :

```
EstimationPI_MC1  -  21.06.2012

******************************************
Cet algorithme vise à simuler le nombre pi par la méthode de
Monte-Carlo en tirant au hasard des points dans un carré de
sommets O(0;0), A(0;1), B(1;1) et C(1;0).
La probabilité qu'un point M(x;y) choisi au hasard dans ce
carré appartiennent au quart de cercle de centre O et de rayon
1 passant par les points A et B correspond à la probabilité de
l'événement "x²+y²<=1". Cette probabilité vaut pi/4.
Dans cet algorithme, on effectue une seule simulation.
L'algorithme EstimationPI_MC2 permet d'effectuer plusieurs
simulations et d'estimer pi à partir de la moyenne de ces
simulations.
******************************************

VARIABLES
  i EST_DU_TYPE NOMBRE
  NPTS EST_DU_TYPE NOMBRE
  N EST_DU_TYPE NOMBRE
  X EST_DU_TYPE NOMBRE
  Y EST_DU_TYPE NOMBRE
  ESTIM_PI EST_DU_TYPE NOMBRE
  ERR_REL EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  //Initialisation des variables N et NPTS.
  //La variable NPTS correspond au nombre total de points souhaités.
  //NPTS est un entier naturel non nul.
  //La variable N correspond au nombre de points situés dans le quart de disque.
  N PREND_LA_VALEUR 0
  AFFICHER "Saisir le nombre de points souhaités."
  LIRE NPTS
```

```
  TANT_QUE (NPTS<1 OU NPTS-floor(NPTS)!=0 OU NPTS>500000) FAIRE
    DEBUT_TANT_QUE
    AFFICHER "ATTENTION ! Le nombre de points doit être un entier naturel non nul inférieur ou égal à 500 000 !"
    LIRE NPTS
    FIN_TANT_QUE
  POUR i ALLANT_DE 1 A NPTS
    DEBUT_POUR
    X PREND_LA_VALEUR random()
    Y PREND_LA_VALEUR random()
    SI (pow(X,2)+pow(Y,2)<=1) ALORS
      DEBUT_SI
      N PREND_LA_VALEUR N+1
      TRACER_POINT (X,Y)
      FIN_SI
      SINON
        DEBUT_SINON
        TRACER_POINT (X,Y)
        FIN_SINON
    FIN_POUR
  ESTIM_PI PREND_LA_VALEUR 4*N/NPTS
  AFFICHER "Avec "
  AFFICHER NPTS
  AFFICHER " points, la valeur estimée de PI vaut : "
  AFFICHER ESTIM_PI
  AFFICHER ", soit une erreur relative d'environ "
  ERR_REL PREND_LA_VALEUR (ESTIM_PI/Math.PI-1)*100
  AFFICHER ERR_REL
  AFFICHER "%."
FIN_ALGORITHME
```

Remarques :

- Quelques commentaires ont été ajoutés pour rendre l’algorithme plus lisible.
- Un test triple est effectué sur la variable NPTS puisque celle-ci doit être :
    - Supérieure ou égale à 1.
    - Entière. « NPTS–floor(NPTS » correspond à la différence entre NPTS et sa partie entière et est nulle si, et seulement si, NPTS est entière.
    - Inférieure ou égale à 500 000 tout simplement parce que AlgoBox impose cette limitation au niveau des boucles (500 000 itérations au maximum et ici, la variable NPTS correspond exactement au nombre d’itérations effectuées).
- A la fin, on calcule l’erreur relative commise : variable ERR_REL prenant comme valeur le résultat du calcul (ESTIM_PI/Math.PI-1)*100.