

# Système de Cramer

## Le problème

Il s'agit de résoudre un système de deux équations du premier degré à deux inconnues, tel que :

$$\begin{aligned}ax + by &= c, \\a'x + b'y &= c'.\end{aligned}$$

On sait que, lorsque  $ab' - a'b \neq 0$ , il y a un unique couple solution :

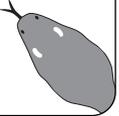
$$\left( \frac{cb' - c'b}{ab' - a'b}; \frac{ac' - a'c}{ab' - a'b} \right).$$

## Complément culturel

C'est au XVIII<sup>e</sup> siècle que le mathématicien suisse Gabriel Cramer (1704–1752) a mis au point des formules pour résoudre des systèmes d'équations linéaires.

## Les programmes

### Python



On définit la fonction « `cramer` » dont les arguments sont les six valeurs du système (`a1` désigne le paramètre  $a'$ ...).

Cette fonction donne la valeur des deux inconnues, sauf si  $D = ab' - a'b$  est nul.

le programme

```
>>> def cramer(a,b,c,a1,b1,c1):
    D=a*b1-a1*b
    if D==0:
        return('soit pas de solution, soit une infinité')
    return((c*b1-c1*b)/D , (a*c1-a1*c)/D)
```

le résultat

```
>>> cramer(1,2,-1,3,5,2)
(9.0, -5.0)
```

## SCRATCH

### Scratch



Avec Scratch, on définit les variables  $a, b, c, a', b'$  et  $c'$  (les lettres primées sont acceptées).

Ces variables sont affectées des valeurs souhaitées avant de lancer le programme. En revanche, le calcul de D est réalisé pendant l'exécution du programme.

```

quand [ ] pressé
  à a attribuer 1
  à b attribuer 2
  à c attribuer -1
  à a' attribuer 3
  à b' attribuer 5
  à c' attribuer 2
  à D attribuer a * b' - a' * b
  si D = 0
    dire Salut! pendant 2 secondes
  sinon
    dire x= pendant 1 secondes
    dire (c * b' - c' * b) / D pendant 2 secondes
    dire y= pendant 1 secondes
    dire (c' * a - c * a') / D pendant 2 secondes

```



### AlgoBox

Les coefficients du système, notés  $a, b, c$  et  $a1, b1, c1$ , seront précisés par l'utilisateur au cours de l'exécution du programme. En fin de programme, dans l'affichage des messages, penser à laisser des espaces afin que le texte renvoyé soit plus lisible.

```

VARIABLES
- a EST_DU_TYPE NOMBRE
- b EST_DU_TYPE NOMBRE
- c EST_DU_TYPE NOMBRE
- a1 EST_DU_TYPE NOMBRE
- b1 EST_DU_TYPE NOMBRE
- c1 EST_DU_TYPE NOMBRE
- D EST_DU_TYPE NOMBRE
- x EST_DU_TYPE NOMBRE
- y EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME
- LIRE a
- LIRE b
- LIRE c
- LIRE a1
- LIRE b1
- LIRE c1
- D PREND_LA_VALEUR a*b1-a1*b
- x PREND_LA_VALEUR (c*b1-c1*b)/D
- y PREND_LA_VALEUR (a*c1-a1*c)/D
- AFFICHER "x= "
- AFFICHER x
- AFFICHER " y= "
- AFFICHER y

FIN_ALGORITHME

```

le programme

### Prolongements

On peut aussi s'intéresser à la résolution d'un système de la forme :

$$a \sqrt{x} + b \sqrt{y} = c,$$

$$a' \sqrt{x} + b' \sqrt{y} = c'.$$

Par rapport au précédent, il faut tester le signe des solutions.

Le programme Python ci-contre permet de le résoudre.

```

>>> def cramer2(a,b,c,a1,b1,c1):
    D=a*b1-a1*b
    if D==0:
        return('soit pas de solution, soit')
    R1=(c*b1-c1*b)/D
    R2=(a*c1-a1*c)/D
    if (R1>=0) and (R2>=0):
        return (R1**2,R2**2)
    else:
        return ('pas de solution')

```