HP-28C

LIVRET D'APPLICATION

ÉQUATIONS LINÉAIRES, VECTEURS ET MATRICES

HEWLETT PACKARD

Equations linéaires, vecteurs et matrices

Livret d'application du HP-28C

Avertissement

1. Les informations contenues dans ce document peuvent être modifiées par Hewlett-Packard sans préavis.

2. **HEWLETT-PACKARD FRANCE ne garantit ni la fiabilité ni les conséquences de l'utilisation de ses progiciels lorsqu'ils sont exploités sur du matériel dont il n'a pas assuré la fourniture.**

3. Les informations contenues dans ce document sont originales. Elles ont été conçues et mises au point par Hewlett-Packard. L'acheteur s'interdit en conséquence, sauf accord préalable et écrit de HEWLETT-PACKARD FRANCE :
   - de les divulguer ou d'en faciliter la divulgation ;
   - de les copier ou de les reproduire en tout ou en partie, par n'importe quel moyen et sous n'importe quelle forme ;
   - de les traduire dans toute autre langue.

Portable Computer Division
1000 N.E. Circle Blvd
Corvallis, OR 97330
U.S.A.

Historique d'édition

Première édition : Mars 1987 Mfg No : 00028-90059

Introduction...

...aux livrets d'applications du calculateur HP-28C. Ces livrets sont conçus pour vous aider à tirer le meilleur parti votre nouveau calculateur.

Le but de ce livret *Equations linéaires, vecteurs et matrices* est de vous fournir des exemples et des techniques de résolution de problèmes sur le HP-28C. De nombreux exemples de calculs matriciels y sont présentés afin de vous familiariser avec toutes les fonctions de gestion de tableaux dont est doté ce calculateur.

Avant de vous lancer dans les exemples de ce livret, vous devez bien connaître les principaux concepts de fonctionnement du HP-28C, exposés dans son manuel d'utilisation :

* Les principes de fonctionnement de votre calculateur : comment passer d'un menu à l'autre, comment sortir du mode graphique ou du mode d'édition, et comment affecter des valeurs à des variables utilisateurs et résoudre leur équation.

* Saisie des nombres et des expressions algébriques.

Les exemples de ce livret spécifient généralement un format d'affichage quant au nombre de décimales affichées sur l'écran. Si vous êtes dans un format d'affichage différent, modifiez-le en utilisant le menu MODE et la touche de fonction FIX de ce menu (par exemple MODE 2 FIX).

Pour des explications détaillées sur les calculs présentés ici, reportez-vous à des ouvrages mathématiques que vous trouverez dans les librairies spécialisées. Les exemples donnés ici n'ont pas un but éducatif. Ils sont destinés à illustrer l'approche HP-28C à adopter dans certains problèmes mathématiques, mais ne couvre pas toutes les solutions possibles.

Les problèmes exposés dans ce livret ont été développés par Brenda C. Bowman de l'université de l'état de l'Oregon.

Table des matières

Utilisation de ce livret

Consacrez quelques minutes pour vous familiariser avec les conventions typographiques de ce livret :

Touches et sélection des menus

Les touches du calculateur sont représentées encadrées :

[ENTER]
[1/x]
[STO]

[ARRAY]
[PLOT]
[ALGEBRA]

Très souvent, les sélections de menus utilisent la touche rouge (shift) en préfixe à la touche du menu. Dans les problèmes présentés, cette touche rouge N'EST PAS représentée (ainsi, [ARRAY] nécessite la pression de la touche préfixe rouge (shift) puis de ARRAY (sur la touche "A" du clavier de gauche).

Les libellés des touches de fonction des menus sont représentées ainsi :

DRAW	(dans le menu [PLOT])
ISOL	(dans le menu [ALGEBRA])
ABCD	(nom créé par l'utilisateur dans le menu [USER])

Les menus contiennent généralement plus de fonctions que ne peuvent en afficher les six labels de l'affichage. Ces autres fonctions sont accessibles à l'aide des touches [NEXT] et [PREV], permettant de faire défiler à l'affichage toutes les options contenues dans les menus. [NEXT] et [PREV] NE SONT PAS indiquées dans nos exemples pour ne pas les alourdir. Mais pensez à les utiliser pour accéder à des fonctions non présentées à l'affichage pour un menu donné.

La résolution d'une variable utilisateur par la fonction [SOLVR] est lancée par la touche utilisateur appropriée, *précédée* de la pression de la touche préfixe (shift) :

[ ] [ABCD].

L'exemple ci-dessus illustre la pression de la touche préfixe, puis de la touche utilisateur intitulée ABCD, afin de lancer une recherche de solution pour la variable ABCD dans une équation spécifiée.

Le symbole [< >] représente la touche menu du curseur.

Tracés interactifs et curseur graphique

Les valeurs de coordonnées obtenues à partir de tracés effectués à l'aide des touches de numérisation [INS] et [DEL] peuvent être différentes des valeurs affichées du fait de petites différences au niveaux des positions du curseur graphique. Ces valeurs sont cependant suffisamment précises pour la recherche de solutions subséquente par la fonction SOLVR.

Formats d'affichage et saisie numérique*

Les nombres négatifs sont affichés et représentés ainsi :

```
-5
-12345.678
[[-1,-2,-3 [ -4,-5,-6 [ ...
```

et créés à l'aide la touche [CHS] :

```
5 [CHS]
12345.678 [CHS]
[[1 [CHS] ,2 [CHS] , ...
```

Les exemples de ce livret spécifient toujours un format d'affichage en ce qui concerne le nombre de décimales. Pour modifier votre format d'affichage, utilisez la touche [FIX] du menu [MODE] (Ex : [MODE] 2 [FIX]).

* Les exemples numériques donnés dans ce livret sont en format américain (point décimal). Pour afficher la virgule décimale, sélectionnez [RDX,] dans le menu [MODE].

Opérations élémentaires sur les matrices

Ce chapitre illustre plusieurs types d'opérations matricielles classiques : addition, multiplication, calcul du déterminant, etc. Il présente également plusieurs programmes de calcul de mineurs et de rang.

Addition de matrices

Cet exemple illustre deux méthodes de création d'une matrice.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

$$B = \begin{bmatrix} 2 & -3 & 0 & 1 \\ 0 & 4 & -1 & 2 \\ 1 & -3 & 2 & -2 \end{bmatrix}$$

Calculez $A + B$

[CLEAR] [< >]

```
4:
3:
2:
1:
```

Entrez les éléments de la matrice A dans l'ordre de ses lignes. Placez chaque élément individuellement dans la pile.

1 [ENTER]
2 [ENTER]
3 [ENTER]
4 [ENTER]
5 [ENTER]
6 [ENTER]
7 [ENTER]
8 [ENTER]
9 [ENTER]
10 [ENTER]
11 [ENTER]
12 [ENTER]

```
4:                 9
3:                10
2:                11
1:                12
```

Entrez le format { m , n } de la matrice A. N'oubliez pas de séparer les deux nombres par un espace.

{ 3 4 } [ENTER]

```
4:                10
3:                11
2:                12
1:           { 3 4 }
```

Mettez les éléments de la pile dans la matrice.

[ARRAY]
→ARRY

```
1: [[ 1 2 3 4 ]
    [ 5 6 7 8 ]
    [ 9 10 11 12 ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez cette matrice dans *A* pour le problème suivant.

'A [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Entrez la matrice *B* en séparant ses éléments par un espace. Remarquez les deux méthodes présentées pour entrer les éléments de *A* et de *B*.

```
[[2 -3 0 1[0 4 -1 2
[1 -3 2 -2 [ENTER]
```

```
1: [[ 2 -3 0 1 ]
    [ 0 4 -1 2 ]
    [ 1 -3 2 -2 ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez la somme $A + B$.

A [ENTER]

```
1: [[ 1 2 3 4 ]
    [ 5 6 7 8 ]
    [ 9 10 11 12 ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

[+]

```
1: [[ 3 -1 3 5 ]
    [ 5 10 6 10 ]
    [ 10 7 13 10 ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

Multiplication de matrices

Calculez le produit de deux matrices. La première matrice doit être de format $k \times m$, la seconde de format $m \times n$, le résultat de format $k \times n$. Dans cet exemple, $k=3$, $m=4$ et $n=2$.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

$$D = \begin{bmatrix} -1 & 1 \\ 2 & 4 \\ -2 & 3 \\ 5 & 4 \end{bmatrix}$$

Calculez $A*D$.

Rappelez la matrice 3 x 4 appelée A dans l'exemple précédent.

A [ENTER]

```
1: [[ 1 2 3 4 ]
    [ 5 6 7 8 ]
    [ 9 10 11 12 ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

Entrez la matrice D suivante de format 3 x 2.

`[[-1 1[2 4[-2 3[5 4` [ENTER]

```
1: [[ -1 1 ]
    [ 2 4 ]
    [ -2 3 ]
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez le produit $A*D$.

[×]

```
1: [[ 17 34 ]
    [ 33 82 ]
    [ 49 130 ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

Déterminant d'une matrice

Calculez le déterminant d'une matrice carrée de format n x n.

$$A = \begin{bmatrix} 2 & -3 & 1 \\ 0 & 5 & 2 \\ -1 & -2 & 3 \end{bmatrix}$$

Entrez la matrice 3 x 3.

[CLEAR]
`[[2 -3 1[0 5 2[-1 -2 3`
[ENTER] [ARRAY]

```
1: [[ 2 -3 1 ]
    [ 0 5 2 ]
    [ -1 -2 3 ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez son déterminant, det(A).

DET

```
3:
2:
1:                      49
CROSS DOT DET ABS RNRM CNRM
```

Le déterminant est 49.

Inverse d'une matrice

Calculez l'inverse d'une matrice carrée n x n.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$

Effacez la pile et sélectionnez le mode d'affichage standard à deux décimales.

[CLEAR]
[MODE] 2 [FIX]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez les éléments de la matrice 3 x 3.

[[1 2 3[2 4 5[3 5 6
[ENTER]

```
1: [[ 1.00 2.00 3.00 ]
    [ 2.00 4.00 5.00 ]
    [ 3.00 5.00 6.00 ]]
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Calculez $(A)^{-1}$.

[1/x]

```
1: [[ 1.00 -3.00 2.00 ]
    [ -3.00 3.00 -1.00...
    [ 2.00 -1.00 6.66E...
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Transposée d'une matrice

Calculez la transposée d'une matrice A de format m x n. A^T sera de format n x m.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Effacez l'affichage et positionnez le mode d'affichage standard. Entrez la matrice A de format 3 x 2.

CLEAR
MODE
STD
`[[1 2[3 4[5 6` ENTER

```
1: [[ 1 2 ]
    [ 3 4 ]
    [ 5 6 ]]
[ STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Calculez $(A)^T$.

ARRAY
TRN

```
2:
1: [[ 1 3 5 ]
    [ 2 4 6 ]]
SIZE RDM TRN CON IDN RSD
```

$(A)^T$ est une matrice 2 x 3.

Conjuguée d'une matrice complexe

Calculez la matrice conjuguée d'une matrice A d'éléments complexes.

$$A = \begin{bmatrix} 1+3i & i \\ 3 & 2-4i \end{bmatrix}$$

[CLEAR]

```
3:
2:
1:
SIZE RDM TRN CON IDN RSD
```

Entrez les éléments individuellement, ligne par ligne. Les deux chiffres de chaque doublet représentent respectivement la partie réelle et la partie imaginaire. Vous remarquerez que vous pouvez utiliser, comme ci-dessous, des virgules au lieu d'espaces.

`(1,3` [ENTER]

```
3:
2:
1:                (1,3)
SIZE RDM TRN CON IDN RSD
```

`(0,1` [ENTER]

```
3:
2:                (1,3)
1:                (0,1)
SIZE RDM TRN CON IDN RSD
```

`(3,0` [ENTER]

```
3:                (1,3)
2:                (0,1)
1:                (3,0)
SIZE RDM TRN CON IDN RSD
```

`(2,-4` [ENTER]

```
3:                (0,1)
2:                (3,0)
1:               (2,-4)
SIZE RDM TRN CON IDN RSD
```

Entrez le format $\{n \quad n\}$ de la matrice.

{ 2 2 } [ENTER]

```
3:                (3,0)
2:               (2,-4)
1:              { 2 2 }
SIZE RDM TRN CON IDN RSD
```

Placez les éléments de la pile dans un tableau.

[ARRAY]
→ARRY

```
2:
1: [[ (1,3) (0,1) ]
    [ (3,0) (2,-4) ]]
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez la matrice conjuguée de A.

CONJ

```
2:
1: [[ (1,-3) (0,-1) ]
    [ (3,0) (2,4) ]]
R→C C→R RE IM CONJ NEG
```

Mineurs d'une matrice

Le mineur M_{ij} est le déterminant de la matrice (M_{ij}) de format $(n-1) \times (n-1)$ obtenue par suppression de la ligne i et de la colonne j d'une matrice carrée A d'ordre n. Un programme doit être entré pour pouvoir calculer les mineurs d'une matrice $n \times n$.

Le sous-programme ROW ci-dessous permet de retirer une ligne et une colonne d'une matrice.

Listage du programme	Explication
SWAP ARRY→ LIST→	Fait passer la matrice au niveau 1, puis la décompose en tableau d'éléments et en liste de format.
DROP	Efface le nombre d'éléments de la liste.
→n m «	Stocke la ligne et la colonne respectivement dans *n* et *m*.
n DUP m x 2 +	Calcul le décalage du nombre de lignes (colonnes) dans la pile.
ROLL - m x →LIST →list	Place $(n - i)*m$ éléments dans la liste.
« m DROPN	Efface le rang *i* (col *j*) de la pile.
list » LIST →	Décompose temporairement la liste en éléments individuels.
DROP	Efface le nombre d'éléments de la liste.
n 1 - m 2 →LIST	Reconstruit la matrice avec une ligne (une colonne) en moins.
→ARRY	

Le programme MINOR suivant utilise le sous-programme ROW pour retirer une ligne, puis une colonne, de la matrice.

Listage du programme	Explication
3 ROLLD	Descend la matrice et le rang *i*.
ROW TRN	Retire le rang *i* et effectue une transposition pour retirer une colonne.
SWAP ROW TRN	Retire la colonne *j* et retranspose la matrice.

Entrez le sous-programme ROW.

[CLEAR]

```
1: « SWAP ARRY→ LIST→
   DROP → n m « n DUP m
   * 2 + ROLL - m *
   →LIST → list « m
```

```
«  SWAP ARRY→ LIST→
DROP → n m «  n DUP
m × 2 + ROLL - m × →LIST
→ list «  m DROPN list »
LIST→ DROP n 1 - m 2
→LIST →ARRY [ENTER] [<>]
```

Stockez le sous-programme ROW.

'ROW [STO]

```
4:
3:
2:
1:
```

Entrez le programme MINOR

```
«  3 ROLLD ROW TRN SWAP
ROW TRN [ENTER] [<>]
```

```
3:
2:
1: « 3 ROLLD ROW TRN
   SWAP ROW TRN »
```

Stockez le programme MINOR.

'MINOR [STO]

```
4:
3:
2:
1:
```

Calculez M_{23} de la matrice suivante.

$$A = \begin{bmatrix} 2 & -3 & 4 & -4 \\ 6 & 5 & 2 & -1 \\ 1 & 0 & 3 & -2 \\ 0 & -5 & 3 & -6 \end{bmatrix}$$

Entrez la matrice.

[[2 -3 4 -4[6 5 2 -1[1 0 3 -2[0 -5 3 -6 [ENTER]

```
1: [[ 2 -3 4 -4 ]
   [ 6 5 2 -1 ]
   [ 1 0 3 -2 ]
   [ 0 -5 3 -6 ]]
```

Entrez le rang et la colonne à retirer.

2 [ENTER] 3 [ENTER]

```
4:
3: [[ 2 -3 4 -4 ]  [ 6...
2:                     2
1:                     3
```

Calculez M_{23}.

[USER] MINO

```
1: [[ 2 -3 -4 ]
   [ 1 0 -2 ]
   [ 0 -5 -6 ]]
MINO ROW
```

Calculez le mineur = det(M_{ij}).

[ARRAY] DET

```
3:
2:
1:                   -18
CROSS DOT DET ABS RNRM CNRM
```

Ce mineur, det(M_{23}), est égal à -18.

Calcul du rang d'une matrice

Le rang d'une matrice est le format de la plus grande des sous-matrices carrées de cette matrice, dont le déterminant n'est pas nul. Le rang d'une matrice correspond au nombre maximal de vecteurs-lignes et de vecteurs-colonnes linéairement indépendants. Cherchez le rang de la matrice A suivante :

$$A = \begin{bmatrix} 4 & 2 & -1 \\ 0 & 5 & -1 \\ 12 & -4 & -1 \end{bmatrix}$$

Le programme MDET ci-dessous permet d'obtenir le déterminant pour le mineur d'une matrice quelconque. Ce programme utilise le programme MINOR du problème précédent.

Listage du programme	Explication
3 PICK	Duplique la matrice.
3 ROLLD MINOR	Génère le mineur de la matrice.
DET	Calcule lé déterminat du mineur.

Entrez le programme.

[CLEAR]

```
3:
2:
1: « 3 PICK 3 ROLLD
   MINOR DET »
```

« 3 PICK 3 ROLLD
MINOR DET [ENTER] [< >]

Stockez ce programme dans MDET

'MDET [STO]

```
4:
3:
2:
1:
```

Entrez la matrice.

[[4 2 -1[0 5 -1
[12 -4 -1 [ENTER]

```
2:
1: [[ 4 2 -1 ]
    [ 0 5 -1 ]
    [ 12 -4 -1 ]]
```

Faites une copie de la matrice et calculez le déterminant pour savoir si le rang n = 3.

[ENTER] [ARRAY] [DET]

```
3:
2: [[ 4 2 -1 ]  [ 0 5 …
1:          .000000000048
CROSS DOT DET ABS RNRM CNRM
```

Det(A) = 0 (approximativement), donc le rang de A est différent de 3.

Effacez det(A).

[DROP] [< >]

```
2:
1: [[ 4  2 -1 ]
    [ 0  5 -1 ]
    [ 12 -4 -1 ]]
```

Calculez les mineurs des sous-matrices 2 x 2 de A, jusqu'à obtention d'un mineur différent de 0.

Calculez le déterminant de M_{11}.

1 [ENTER] [ENTER]
[USER] [MDET]

```
3:
2: [[ 4 2 -1 ]  [ 0 5 …
1:                    -9
MDET MINO ROW
```

Det(M_{11}) est égal à -9, donc le rang de A est 2.

Vous pouvez effacer les programmes ROW, MDET et MINO avant de passer aux problèmes suivants.

Matrices d'Hermitte

Il s'agit de déterminer si une matrice est une matrice d'Hermitte. Une matrice carrée d'éléments réels ou complexes est une matrice d'Hermitte si elle est égale à sa transposée conjuguée.

Déterminez si la matrice A de format 4 x 4 est une matrice d'Hermitte.

$$A = \begin{bmatrix} 1 & 2\text{-}i & 2 & \text{-}3+i \\ 2+i & 3 & i & 3 \\ 2 & \text{-}i & 4 & 1\text{-}i \\ 3\text{-}i & 3 & 1+i & 0 \end{bmatrix}$$

Mettez les éléments de A individuellement dans la pile.

[CLEAR] [<>] 1 [ENTER] (2,-1 [ENTER] 2 [ENTER] (-3,1 [ENTER]	4: 1 3: (2,-1) 2: 2 1: (-3,1)
(2,1 [ENTER] 3 [ENTER] (0,1 [ENTER] 3 [ENTER]	4: (2,1) 3: 3 2: (0,1) 1: 3
2 [ENTER] (0,-1 [ENTER] 4 [ENTER] (1,-1 [ENTER]	4: 2 3: (0,-1) 2: 4 1: (1,-1)
(3,-1 [ENTER] 3 [ENTER] (1,1 [ENTER] 0 [ENTER]	4: (3,-1) 3: 3 2: (1,1) 1: 0

Entrez le format de A.

{ 4 4 [ENTER]

```
4:                3
3:            (1,1)
2:                0
1:          { 4 4 }
```

Placez les éléments dans la matrice.

[ARRAY]
[→ARRY]

```
1: [[ (1,0) (2,-1) (2,…
    [ (2,1) (3,0) (0,1…
    [ (2,0) (0,-1) (4,…
→ARRY ARRY→ PUT GET PUTI GETI
```

Vous pouvez visualiser la totalité de la matrice pour vérifier si elle est correcte à l'aide de [EDIT] ou de [VIEW↓] .

Faites une copie de la matrice.

[ENTER]

```
1: [[ (1,0) (2,-1) (2,…
    [ (2,1) (3,0) (0,1…
    [ (2,0) (0,-1) (4,…
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez la matrice transposée conjuguée de A. Puisque A est formée d'éléments complexes, la fonction TRN effectue à la fois le calcul de la transposée et des conjugués des éléments complexes.

[TRN]

```
1: [[ (1,0) (2,-1) (2,…
    [ (2,1) (3,0) (0,1…
    [ (2,0) (0,-1) (4,…
SIZE RDM TRN CON IDN RSD
```

Testez l'équivalence de conj(A^T) et de A. Si A est une matrice d'Hermitte, conj(A^T) et A sont égales et [SAME] renvoie un état vrai (1).

[TEST] [SAME]

```
3:
2:
1:                0
AND OR XOR NOT SAME ==
```

La matrice A n'est pas une matrice d'Hermitte.

Systèmes d'équations linéaires

L'une des applications les plus fréquentes et les plus importantes des matrices consiste en leur utilisation pour la résolution de systèmes de m équations linéaires à n inconnues.
Le HP-28C peut être utilisé aussi bien pour la résolution de systèmes d'équations avec second membre que de systèmes sans second membre (homogènes) de la forme $AX = B$.

Systèmes d'équations linéaires avec second membre

Vous allez résoudre un système d'équations de la forme $AX = B$.

$$\begin{aligned} x_1 + x_2 - 2x_3 + x_4 + 3x_5 &= 1 \\ 3x_1 + 2x_2 - 4x_3 - 3x_4 - 8x_5 &= 2 \\ 2x_1 - x_2 + 2x_3 + 2x_4 + 5x_5 &= 3 \end{aligned}$$

Effacez l'affichage et positionnez le mode d'affichage à deux décimales.

[CLEAR]
[MODE] 2 [FIX]

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez les coefficients du système d'équations.

```
[[1 1 -2 1 3[3 2 -4 -3
-8 [2 -1 2 2 5
```
[ENTER]

```
1: [[ 1.00 1.00 -2.00 ...
    [ 3.00 2.00 -4.00 ...
    [ 2.00 -1.00 2.00 ...
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Stockez la matrice A.

'A [STO]

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez les éléments de B.

`[[1[2[3` [ENTER]

```
1: [[ 1.00 ]
    [ 2.00 ]
    [ 3.00 ]]
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Stockez la matrice B.

'B [STO]

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Pour résoudre le système en X, nous utiliserons la méthode : $X = \frac{A^T B}{A^T A}$

Calculez A^T.

[ARRAY]
A [ENTER]

```
1: [[ 1.00 1.00 -2.00 ...
    [ 3.00 2.00 -4.00 ...
    [ 2.00 -1.00 2.00 ...
→ARRY ARRY→ PUT GET PUTI GETI
```

TRN

```
1: [[ 1.00 3.00 2.00 ]
    [ 1.00 2.00 -1.00 ]
    [ -2.00 -4.00 2.00...
SIZE RDM TRN CON IDN RSD
```

Multipliez par B.

B [×]

```
1: [[ 13.00 ]
    [ 2.00 ]
    [ -4.00 ]
SIZE RDM TRN CON IDN RSD
```

Calculez A^T.

A [ENTER]

```
1: [[ 1.00 3.00 2.00 ]
    [ 1.00 2.00 -1.00 ]
    [ -2.00 -4.00 2.00...
SIZE RDM TRN CON IDN RSD
```

TRN

```
1: [[ 1.00 3.00 2.00 ]
    [ 1.00 2.00 -1.00 ]
    [ -2.00 -4.00 2.00...
SIZE RDM TRN CON IDN RSD
```

Multipliez par A.

A [×]

```
1: [[ 14.00 5.00 -10.0...
    [ 5.00 6.00 -12.00...
    [ -10.00 -12.00 24...
SIZE RDM TRN CON IDN RSD
```

Divisez A^TB par A^TA.

[÷]

```
1: [[ 1.12 ]
    [ 1.24 ]
    [ 0.80 ]
SIZE RDM TRN CON IDN RSD
```

[VIEW↑] et [VIEW↓] peuvent être utilisés pour afficher chacun des éléments..Ceux-ci sont : x_1 = 1.12 ; x_2 = 1.24 ; x_3 = 0.80 ; x_4 = -0.08 et x_5 = 0.11.

Systèmes d'équations linéaires homogènes (sans second membre)

Résolvez un système d'équations linéaires homogène, de la forme $AX = 0$.

$$\begin{aligned} x_1 - 2x_2 + 3x_3 &= 0 \\ 2x_1 + 6x_2 + x_3 &= 0 \\ 3x_1 - 4x_2 + 8x_3 &= 0 \end{aligned}$$

Le programme UT ci-dessous traite une pile de vecteurs représentant des équations homogènes et les transforme en une matrice triangulaire supérieure.

Listage du programme	Explication
DUP SIZE LIST →	Stocke plusieurs éléments dans s.
DROP → s	
« s 2	Pour j = s (décroissant) à 2, transforme les vecteurs du bas j-1.
FOR j s j -	m = s - j + 1
1 + → m	
« 1 j 1 -	Boucle pour i=1 à j-1
FOR ii ROLL j PICK	
m 1 →LIST DUP2 GET 4 PICK	Transforme les vecteurs.
ROT GET SWAP ÷ x -	
i ROLLD NEXT » -1 STEP »	

Entrez le programme.

[CLEAR]

```
«  DUP SIZE LIST→
DROP → s «  s 2
FOR j s j - 1 +
→ m «  1 j 1 -
FOR i i ROLL j PICK
m 1 →LIST DUP2 GET
4 PICK ROT GET SWAP
÷ × - i ROLLD
NEXT » -1 STEP »
```

[ENTER] [<>]

```
1: « DUP SIZE LIST→
   DROP → s « s 2 FOR j
   s j - 1 + → m « 1 j
   1 - FOR i i ROLL j
```

Stockez le programme dans UT.

'UT [STO]

```
4:
3:
2:
1:
```

Positionnez le mode d'affichage à une décimale.

[MODE] 1 FIX

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez les coefficients

[[1 -2 3[2 6 1[3 -4 8
[ENTER]

```
1: [[ 1.0 -2.0 3.0 ]
    [ 2.0 6.0 1.0 ]
    [ 3.0 -4.0 8.0 ]]
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Stockez la matrice dans *ARR* à des fins de vérification en fin de problème.

'ARR [STO]

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Modifiez la matrice *ARR* pour la mettre sous forme linéaire.

[USER]
ARR

```
1: [[ 1.0 -2.0 3.0 ]
    [ 2.0 6.0 1.0 ]
    [ 3.0 -4.0 8.0 ]]
 ARR  UT
```

Utilisez le mode EDIT et la touche [DEL] pour supprimer les crochets du tableau *ARR* et placer les lignes dans trois vecteurs-lignes indépendants. Après avoir supprimé les crochets extrêmes, entrez les lignes ainsi modifiées :

```
[ 1 -2 3 ]
  [ 2 6 1 ]
  [ 3 -4 8 ] [ENTER]
```

```
3:       [ 1.0 -2.0 3.0 ]
2:        [ 2.0 6.0 1.0 ]
1:       [ 3.0 -4.0 8.0 ]
 ARR  UT
```

Maintenant, transformez la matrice en une matrice triangulaire supérieure.

UT

```
3:       [ 1.0 -2.0 3.0 ]
2:     [ 0.0 10.0 -5.0 ]
1:        [ 0.0 0.0 0.0 ]
 ARR  UT
```

La matrice est maintenant sous forme linéaire et nous pouvons résoudre le système des trois équations transformées. La matrice représente le système suivant d'équations linéaires.

$$\begin{aligned} x_1 - 2x_2 + 3x_3 &= 0 \\ 10x_2 - 5x_3 &= 0 \\ 0 &= 0 \end{aligned}$$

Supprimez l'équation 0 = 0.

[DROP]

```
3:
2:       [ 1.0 -2.0 3.0 ]
1:      [ 0.0 10.0 -5.0 ]
 ARR | UT |    |    |    |
```

Entrez l'équation de la ligne 2.

'10×X2 - 5×X3=0 [ENTER]

```
3:       [ 1.0 -2.0 3.0 ]
2:      [ 0.0 10.0 -5.0 ]
1:        '10*X2-5*X3=0'
 ARR | UT |    |    |    |
```

Résolvez l'équation pour x_3.

'X3 [ENTER]

```
3:      [ 0.0 10.0 -5.0 ]
2:        '10*X2-5*X3=0'
1:                  'X3'
 ARR | UT |    |    |    |
```

Isolez la variable x_3.

[ALGEBRA]
[ISOL]

```
3:       [ 1.0 -2.0 3.0 ]
2:      [ 0.0 10.0 -5.0 ]
1:             '10*X2/5'
TAYLR|ISOL |QUAD |SHOW |OBGET|EXGET
```

Simplifiez.

[COLCT]

```
3:       [ 1.0 -2.0 3.0 ]
2:      [ 0.0 10.0 -5.0 ]
1:                '2*X2'
COLCT|EXPAN|SIZE |FORM |OBSUB|EXSUB
```

La solution est $x_3 = 2^*x_2$. Retirez la ligne 2 pour résoudre la ligne 1.

[DROP]
[DROP]

```
3:
2:
1:       [ 1.0 -2.0 3.0 ]
COLCT|EXPAN|SIZE |FORM |OBSUB|EXSUB
```

Entrez l'équation de la ligne 1, en substituant la valeur de x_3.

'X1 − 2 × X2 + 6 × X2
[ENTER]

```
3:
2:       [ 1.0 -2.0 3.0 ]
1:        'X1-2*X2+6*X2'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Résolvez l'équation pour x_1.

'X1 [ENTER]

```
3:       [ 1.0 -2.0 3.0 ]
2:        'X1-2*X2+6*X2'
1:                  'X1'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Isolez le terme.

ISOL

```
3:
2:       [ 1.0 -2.0 3.0 ]
1:       '-(6*X2)+2*X2'
TAYLR ISOL QUAD SHOW OBGET EXGET
```

Simplifiez les termes.

COLCT

```
3:
2:       [ 1.0 -2.0 3.0 ]
1:            '-(4*X2)'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Le résultat est $x_1 = -4{*}x_2$. Une solution est $x_1 = -4$, $x_2 = 1$ et $x_3 = 2$. Vérifiez ce vecteur-solution X de format 3 x 1. Entrez le vecteur .

[[−4[1[2 [ENTER]

```
1: [[ -4.0 ]
    [ 1.0 ]
    [ 2.0 ]]
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Mettez les coefficients de la matrice ARR dans la pile.

[USER]
ARR

```
1: [[ 1.0 -2.0 3.0 ]
    [ 2.0 6.0 1.0 ]
    [ 3.0 -4.0 8.0 ]]
ARR UT
```

Echangez les positions de *ARR* et de X.

[SWAP]

```
1: [[ -4.0 ]
    [ 1.0 ]
    [ 2.0 ]]
ARR UT
```

Effectuez le produit *ARR* * *X*.

[×]

```
1: [[ 0.0 ]
    [ 0.0 ]
    [ 0.0 ]]
 ARR  UT
```

ARR * *X* = 0. Donc *X* est bien la solution du système.

Gardez le programme UT : il sera utilisé dans un problème ultérieur.

Raffinement itératif

Du fait d'erreurs d'arrondi, dans certains cas la solution Z calculée numériquement n'est pas toujours tout à fait celle du système initial $AX = B$. Dans de nombreuses applications, Z s'avèrera une solution correcte. Lorsqu'on désire une meilleure précision, il est possible d'améliorer la solution Z calculée en utilisant une méthode itérative. Cette méthode utilise l'erreur résiduelle associée à une solution pour modifier cette solution.

Résolvez le système d'équations linéaires suivant de la forme $AX = B$.

$$A = \begin{bmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Effacez l'affichage et sélectionnez le mode d'affichage standard.

CLEAR
MODE STD

```
3:
2:
1:
[ STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Résolvez $AX = B$ et améliorez la précision de la solution en utilisant les corrections résiduelles selon un processus itératif. Entrez les coefficients de la matrice.

[[33 16 72[-24 -10 -57
[-8 -4 -17 ENTER

```
1: [[ 33 16 72 ]
    [ -24 -10 -57 ]
    [ -8 -4 -17 ]]
[ STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Stockez la matrice A.

'A STO

```
3:
2:
1:
[ STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Entrez la matrice des constantes.

[[0 [0 [1 ENTER

```
1: [[ 0 ]
    [ 0 ]
    [ 1 ]]
[ STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Stockez cette matrice *B*.

' B STO

```
3:
2:
1:
[ STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Calculez $Z = B/A$.

USER
B

```
1: [[ 0 ]
    [ 0 ]
    [ 1 ]]
 B  A  UT
```

A

```
1: [[ 33 16 72 ]
    [ -24 -10 -57 ]
    [ -8 -4 -17 ]]
 B  A  UT
```

÷

```
1: [[ -31.9999999989 ]
    [ 25.4999999991 ]
    [ 8.99999999969 ]]
 B  A  UT
```

Stockez la matrice solution approchée *Z* de format 3 x 1.

' Z STO

```
3:
2:
1:
 Z  B  A  UT
```

Calculez la matrice d'erreur résiduelle *R*, où $R = B - AZ$. La fonction RSD calcule directement R en utilisant la précision étendue du calculateur.

B
A
Z

```
1: [[ -31.9999999989 ]
    [ 25.4999999991 ]
    [ 8.99999999969 ]]
 Z  B  A  UT
```

Résolvez en utilisant la fonction RSD.

ARRAY
RSD

```
1: [[ .00000000042 ]
    [ -.00000000027 ]
    [ -.00000000007 ]]
 SIZE  RDM  TRN  CON  IDN  RSD
```

Stockez la matrice R.

'R [STO]

```
3:
2:
1:
SIZE RDM TRN CON IDN RSD
```

Trouvez l'erreur vraie $E = |Z - X| = (B - AZ)/A = R/A$.

[USER]
R
A
[÷]

```
1: [[ -1.09999999997E-…
   [ 8.99999999977E-1…
   [ 3.09999999992E-1…
 R    Z    B    A    UT
```

Calculez la solution corrigée $X = Z + E$.

Z
[+]

```
1: [[ -32 ]
   [ 25.5 ]
   [ 9 ]]
 R    Z    B    A    UT
```

X = la solution corrigée.

Espaces vectoriels

Les espaces vectoriels sont largement utilisés en mathématiques, en physique et en ingénierie pour représenter des propriétés physiques comme des grandeurs ou des directions dans un système géométrique. De nombreuses opérations sur les vecteurs peuvent être réalisées facilement à l'aide des fonctions intégrées du menu [ARRAY].

Base

Une base est un ensemble de n vecteurs linéairement indépendants sur l'espace vectoriel de dimension n, noté $V_n(\mathrm{R})$.

Déterminez si les vecteurs X_1, X_2 et X_3 constituent une base sur l'espace vectoriel de dimension 3 : $V_3(\mathrm{R})$.

$$X_1 = [\ 1 \quad 1 \ 2\ ]$$
$$X_2 = [\ 3 \quad 2 \ 4\ ]$$
$$X_3 = [\ 1 \ -3 \ 1\ ]$$

Effacez l'affichage et sélectionnez le mode d'affichage standard.

CLEAR
MODE STD

```
3:
2:
1:
[STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Entrez les trois vecteurs sous la forme d'une matrice A de format 3 x 3.

[[1 1 2[3 2 4[1 -3 1
ENTER ENTER

```
1: [[ 1 1 2 ]
    [ 3 2 4 ]
    [ 1 -3 1 ]]
[STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Stockez la matrice A pour un problème suivant.

'A STO

```
3:
2:
1:
[STD ] FIX  SCI  ENG  DEG [ RAD ]
```

Calculez det(A).

ARRAY
DET

```
3:
2:
1:          -7.00000000004
CROSS DOT  DET  ABS  RNRM CNRM
```

Det(A) = -7. Donc A est une matrice régulière, et les trois vecteurs lignes sont linéairement indépendants et constituent une base.

Orthogonalité

Deux vecteurs sont orthogonaux si leur produit scalaire est égal à zéro.

Déterminez, parmi les vecteurs du problème précédent, ceux qui sont orthogonaux deux à deux.

[CLEAR]

```
3:
2:
1:
CROSS DOT DET ABS RNRM CNRM
```

Mettez la matrice A dans la pile.

A [ENTER]

```
1: [[ 1 1 2 ]
    [ 3 2 4 ]
    [ 1 -3 1 ]]
CROSS DOT DET ABS RNRM CNRM
```

Utilisez [EDIT] pour supprimer les crochets les plus extrêmes du tableau A, et formez trois vecteurs lignes. Après avoir supprimé le premier crochet et le dernier crochet à l'aide de [DEL], validez les lignes ainsi modifiées.

```
[ 1 1 2 ]
[ 3 2 4 ]
[ 1 -3 1 ] [ENTER]
```

```
3:              [ 1 1 2 ]
2:              [ 3 2 4 ]
1:             [ 1 -3 1 ]
CROSS DOT DET ABS RNRM CNRM
```

Remarque : Deux sous-programmes utilitaires de modification de tableaux à deux dimensions en vecteurs lignes et vice-versa figurent à la fin de cette section. Ces sous-programmes effectuent les modifications présentées ci-dessus.

Le troisième vecteur est X_3.

'X3 [STO]

```
3:
2:              [ 1 1 2 ]
1:              [ 3 2 4 ]
CROSS DOT DET ABS RNRM CNRM
```

Le deuxième vecteur est X_2.

'X2 [STO]

```
3:
2:
1:              [ 1 1 2 ]
CROSS DOT DET ABS RNRM CNRM
```

Le premier vecteur est X_1.

'X1 [STO]

```
3:
2:
1:
CROSS DOT DET ABS RNRM CNRM
```

Calculez les produits scalaires de ces vecteurs.

X1 [ENTER]
X2 [ENTER]

```
3:
2:                 [ 1 1 2 ]
1:                 [ 3 2 4 ]
CROSS DOT DET ABS RNRM CNRM
```

DOT

```
3:
2:
1:                        13
CROSS DOT DET ABS RNRM CNRM
```

$X_1 . X_2 = 13$. Ces deux vecteurs lignes ne sont pas orthogonaux.

[DROP]
X2 [ENTER]
X3 [ENTER]

```
3:
2:                 [ 3 2 4 ]
1:               [ 1 -3 1 ]
CROSS DOT DET ABS RNRM CNRM
```

DOT

```
3:
2:
1:                         1
CROSS DOT DET ABS RNRM CNRM
```

$X_2 . X_3 = 1$. Ces deux vecteurs lignes ne sont pas orthogonaux.

[DROP]
X1 [ENTER]
X3 [ENTER]

```
3:
2:                 [ 1 1 2 ]
1:               [ 1 -3 1 ]
CROSS DOT DET ABS RNRM CNRM
```

DOT

```
3:
2:
1:                         0
CROSS DOT DET ABS RNRM CNRM
```

$X_1 . X_3 = 0$. Ces deux vecteurs sont orthogonaux.

Sous-programmes utilitaires de transformation

De nombreux problèmes exposés jusqu'ici faisaient usage du mode EDIT pour mettre une matrice sous forme de lignes. Les sous-programmes utilitaires suivants peuvent être également utilisés pour ces transformations.

Le sous-programme ROW→ ci-dessous met une pile de *n* vecteurs et le nombre *n* au niveau 1, et affiche la matrice combinant ces *n* vecteurs lignes.

```
«  OVER SIZE LIST→ DROP
→  n m «  0 n 1 -
FOR i i m × n i - +
ROLL ARRY→ DROP NEXT
n m 2 →LIST →ARRY » »
[ENTER] [<>]
```

```
1: « OVER SIZE LIST→
   DROP → n m « 0 n 1 -
   FOR i i m * n i - +
   ROLL ARRY→ DROP NEXT
```

Après avoir entré et stocké ce sous-programme, mettez les lignes du tableau *A* sous forme matricielle.

```
'ROW→ [STO]

[1,1,2]
[3,2,4]
[1,-3,1] [ENTER]
3 [USER] [ROW→]
```

```
1: [[ 1 1 2 ]
    [ 3 2 4 ]
    [ 1 -3 1 ]]
ROW→ | UT |  |  |  |
```

Le sous-programme →ROW ci-dessous prend une matrice et la décompose en ses lignes individuelles dans la pile.

```
«  ARRY→ LIST→ DROP
→  n m «  1 n FOR i m 1
→LIST →ARRY n i -
m × i + ROLLD NEXT » »
[ENTER] [<>]
```

```
1: « ARRY→ LIST→ DROP →
   n m « 1 n FOR i m 1
   →LIST →ARRY n i - m
   * i + ROLLD NEXT » »
```

Après avoir entré et stocké ce sous-programme, reconvertissez en lignes la matrice obtenue précédemment.

```
'→ROW [STO]
[USER] [→ROW]
```

```
3:              [ 1 1 2 ]
2:              [ 3 2 4 ]
1:             [ 1 -3 1 ]
→ROW | ROW→ | UT |  |  |  |
```

Longueur d'un vecteur

Trouvez la longueur d'un vecteur X_1, formulée par :

$$||X_1|| = \sqrt{X_1 \cdot X_1}$$

Effacez l'affichage et sélectionnez le mode d'affichage à deux décimales.

[CLEAR]
[MODE] 2 FIX

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Rappelez X_1 du problème précédent. Puisque X_1 avait été stocké, vous pouvez aussi utiliser

X1 [ENTER]

```
3:
2:
1:      [ 1.00 1.00 2.00 ]
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

La fonction ABS donne directement la norme Frobenius d'un tableau, qui est équivalente à la longueur d'un vecteur.

[ARRAY]
ABS

```
3:
2:
1:                    2.45
CROSS  DOT  DET  ABS  RNRM CNRM
```

$||X_1|| = 2.45.$

Norme d'un vecteur

Pour normer un vecteur X en son unique vecteur unité U, divisez chacun des composantes de X par $||X||$. Nous allons normer X_1. Les vecteurs X_1, X_2 et X_3 proviennent du problème présenté à la section "Orthogonalité".

Entrez un programme calculant $X/||X||$.

[CLEAR]
« DUP ABS $\frac{1}{x}$ × » [ENTER]

```
3:
2:
1:     « DUP ABS INV * »
CROSS DOT DET ABS RNRM CNRM
```

Stockez le programme NORM.

'NORM [STO]

```
3:
2:
1:
CROSS DOT DET ABS RNRM CNRM
```

Entrez le vecteur à normer.

[USER]
X1

```
3:
2:
1:    [ 1.00 1.00 2.00 ]
NORM X1 X2 X3 →ROW ROW→
```

Calculez la norme du vecteur X_1.

NORM

```
3:
2:
1:    [ 0.41 0.41 0.82 ]
NORM X1 X2 X3 →ROW ROW→
```

Le résultat est U_1 = [0.41 0.41 0.82].

Calculez la norme du vecteur X_2.

X2

```
3:
2:    [ 0.41 0.41 0.82 ]
1:    [ 3.00 2.00 4.00 ]
NORM X1 X2 X3 →ROW ROW→
```

NORM

```
3:
2:    [ 0.41 0.41 0.82 ]
1:    [ 0.56 0.37 0.74 ]
NORM X1 X2 X3 →ROW ROW→
```

Le résultat est U_2 = [0.56 0.37 0.74].

Calculez, enfin, la norme du vecteur X_3 .

X3

```
3:      [ 0.41 0.41 0.82 ]
2:      [ 0.56 0.37 0.74 ]
1:    [ 1.00 -3.00 1.00 ]
NORM  X1  X2  X3  →ROW  ROW→
```

NORM

```
3:      [ 0.41 0.41 0.82 ]
2:      [ 0.56 0.37 0.74 ]
1:    [ 0.30 -0.90 0.30 ]
NORM  X1  X2  X3  →ROW  ROW→
```

Le résultat est U_3 = [0.30 -0.90 0.30].

Orthogonalisation de Gram-Schmidt

Constituez une base orthogonale sur l'espace vectoriel V_3(R), en utilisant la méthode de Gram-Schmidt. Etant donné que X_1, X_2 et X_3 forment une base, alors les vecteurs Y_1, Y_2 et Y_3 forment une base orthogonale, issue du processus :

$$Y_1 = X_1$$

$$Y_2 = X_2 - \left(\frac{Y_1 \cdot X_2}{Y_1 \cdot Y_1} * Y_1 \right)$$

$$Y_3 = X_3 - \left(\frac{Y_2 \cdot X_3}{Y_2 \cdot Y_2} * Y_2 \right) - \left(\frac{Y_1 \cdot X_3}{Y_1 \cdot Y_1} * Y_1 \right)$$

Les vecteurs X_1, X_2 et X_3 proviennent du problème présenté à la section "Orthogonalité".

Calculez Y_1.

[CLEAR]
[USER] X1

```
3:
2:
1:     [ 1.00 1.00 2.00 ]
 X1  X2  X3  VT
```

Stockez Y_1.

'Y1 [STO]

```
3:
2:
1:
 Y1  X1  X2  X3
```

Ecrivez un programme pour calculer Y_2.

« X2 Y1 X2 DOT Y1 Y1 DOT ÷ Y1 × - »
[ENTER]

```
2:
1: « X2 Y1 X2 DOT Y1 Y1
   DOT / Y1 * - »
 Y1  X1  X2  X3
```

Exécutez le programme.

[EVAL]

```
3:
2:
1: [ 0.83 -0.17 -0.33 ]
 Y1  X1  X2  X3
```

Y_2 = [0.83 -0.17 -0.33]. Stockez Y_2.

'Y2 [STO]

```
3:
2:
1:
 Y2 | Y1 | X1 | X2 | X3 |
```

Ecrivez un programme pour calculer Y_3.

« X3 Y2 X3 DOT Y2 Y2 DOT ÷ Y2 × - Y1 X3 DOT Y1 Y1 DOT ÷ Y1 × - » [ENTER]

```
1: « X3 Y2 X3 DOT Y2 Y2
   DOT / Y2 * - Y1 X3
   DOT Y1 Y1 DOT / Y1 *
 Y2 | Y1 | X1 | X2 | X3 |
```

Exécutez le programme.

[EVAL]

```
3:
2:
1: [ 4.00E-12 -2.80 1.…
 Y2 | Y1 | X1 | X2 | X3 |
```

Y_3 = [4.0E-12 -2.80 1.40]. Stockez Y_3.

'Y3 [STO]

```
3:
2:
1:
 Y3 | Y2 | Y1 | X1 | X2 | X3
```

Les vecteurs Y_1, Y_2 et Y_3 forment une base orthogonale.

Sous-programme général

Le programme GSO suivant est un sous-programme général d'orthogonalisation de Gram-Schmidt. Il permet de trouver la base orthogonale d'une liste de vecteurs quelconque.

```
«  DUP SIZE LIST→ DROP
DUP DUP 2 + ROLLD →LIST
→ M «  2 SWAP FOR n M
n GET 1 n 1 - FOR i M i
GET DUP DUP2 DOT INV ×
SWAP 3 PICK DOT × -
NEXT n M SWAP ROT PUT 'M'
STO NEXT M LIST→
DROP » » [ENTER] [<>]
```

```
1: « DUP SIZE LIST→
   DROP DUP DUP 2.00 +
   ROLLD →LIST → M «
   2.00 SWAP FOR n M n
```

Après avoir entré et stocké ce programme, trouvez la base orthogonale des trois vecteurs de l'exemple précédent.

```
'GSO [STO]

[1,1,2]
[3,2,4]
[1,-3,1] [USER] [GSO]
```

```
3:    [ 1.00 1.00 2.00 ]
2: [ 0.83 -0.17 -0.33 ]
1: [ 4.00E-12 -2.80 1....
GSO | Y3 | Y2 | Y1 | X1 | X2
```

Base orthonormée

Formez une base orthonormée G_i de vecteurs unités orthogonaux dans la variété linéaire V_3(R). Les vecteurs Y_1, Y_2 et Y_3 ainsi que le programme NORM proviennent des problèmes précédents.

$$G_i = \frac{Y_i}{||Y_i||}$$

Calculez G_1.

CLEAR
USER Y1

```
3:
2:
1:     [ 1.00 1.00 2.00 ]
NORM  Y3  Y2  Y1  X1  X2
```

Utilisez le programme NORM de la section "Norme d'un vecteur".

NORM

```
3:
2:
1:     [ 0.41 0.41 0.82 ]
NORM  Y3  Y2  Y1  X1  X2
```

Stockez le résultat dans G_1.

'G1 STO

```
3:
2:
1:
G1  NORM  Y3  Y2  Y1  X1
```

Calculez G_2'.

Y2

```
3:
2:
1: [ 0.83 -0.17 -0.33 ]
G1  NORM  Y3  Y2  Y1  X1
```

Calculez la norme.

NORM

```
3:
2:
1: [ 0.91 -0.18 -0.37 ]
G1  NORM  Y3  Y2  Y1  X1
```

Stockez le résultat dans G_2.

'G2 STO

```
3:
2:
1:
G2  G1  NORM  Y3  Y2  Y1
```

Calculez G_3.

Y3

```
3:
2:
1: [ 4.00E-12 -2.80 1....
 G2 | G1 |NORM| Y3 | Y2 | Y1
```

Calculez la norme.

NORM

```
3:
2:
1: [ 1.28E-12 -0.89 0....
 G2 | G1 |NORM| Y3 | Y2 | Y1
```

Stockez le résultat dans G_3.

'G3 STO

```
3:
2:
1:
 G3 | G2 | G1 |NORM| Y3 | Y2
```

Vérifiez que les trois vecteurs sont orthogonaux deux à deux.

G1
G2

```
3:
2:    [ 0.41 0.41 0.82 ]
1: [ 0.91 -0.18 -0.37 ]
 G3 | G2 | G1 |NORM| Y3 | Y2
```

Calculez le produit scalaire $(G_1.G_2)$.

ARRAY
DOT

```
3:
2:
1:              -8.98E-12
CROSS| DOT | DET | ABS |RNRM|CNRM
```

$G_1 \cdot G_2 \approx 0$.

Calculez le produit scalaire $(G_2.G_3)$.

DROP
USER
G2
G3
ARRAY
DOT

```
3:
2:
1:               5.18E-13
CROSS| DOT | DET | ABS |RNRM|CNRM
```

$G_2 \cdot G_3 \approx 0$.

Calculez le produit scalaire $(G_1.G_3)$.

[DROP]
[USER]
G1
G3
[ARRAY]
DOT

```
3:
2:
1:                2.97E-12
CROSS DOT  DET  ABS  RNRM CNRM
```

$G_1 \cdot G_3 \approx 0.$

Donc, puisque les trois produits scalaires sont pratiquement égaux à zéro, les trois vecteurs sont orthogonaux deux à deux.

Maintenant, vérifiez qu'ils constituent une base. Combinez ces trois vecteurs en un tableau, en plaçant leurs éléments dans la pile et en supprimant leurs listes de format respectives.

[DROP]
[USER] G1
[ARRAY] ARRY→
[DROP]
[USER] G2
[ARRAY] ARRY→
[DROP]
[USER] G3
[ARRAY] ARRY→
[DROP]

```
3:                1.28E-12
2:                   -0.89
1:                    0.45
→ARRY ARRY→ PUT  GET  PUTI GETI
```

Remarque : Nous aurions pu également utiliser le programme utilitaire →ROW, décrit dans la section "Orthogonalité", pour constituer cette liste de vecteurs.

Ensuite, entrez le format de la matrice qui sera formée des trois vecteurs.

{ 3 3 } [ENTER]

```
3:                   -0.89
2:                    0.45
1:          { 3.00 3.00 }
→ARRY ARRY→ PUT  GET  PUTI GETI
```

Enfin, mettez les trois vecteurs sous forme matricielle.

≣ →ARRY ≣

```
1: [[ 0.41 0.41 0.82 ]
    [ 0.91 -0.18 -0.37…
    [ 1.28E-12 -0.89 0…
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez le déterminant.

≣ DET ≣

```
3:
2:
1:                  -1.00
CROSS DOT DET ABS RNRM CNRM
```

Det = -1. Cette matrice est donc régulière et les vecteurs considérés constituent une base orthonormée.

Effacez les vecteurs $X_1, X_2, X_3, Y_1, Y_2, Y_3, G_1, G_2, G_3$ ainsi que le programme NORM.

Valeurs propres

Une autre utilisation majeure des matrices est le développement d'une structure permettant de représenter des transformations linéaires dans un système géométrique. Toute matrice représentant une transformation linéaire spécifique reflète les propriétés de cette transformation.

Puisque des matrices similaires ont en commun toutes les propriétés géométriques intrinsèques d'une transformation, un problème classique est la recherche d'une forme canonique simple pour chaque catégorie de similitude. Nous allons illustrer deux méthodes de calcul des valeurs propres, ainsi qu'une méthode de recherche des vecteurs propres.

L'équation caractéristique

L'équation caractéristique d'une matrice peut être écrite sous la forme :

$$
\begin{aligned}
AX &= \lambda X \\
AX - \lambda X &= 0 \\
(A - \lambda I)X &= 0 \\
X &= 0 \quad \text{Solution triviale} \\
\det(A - \lambda I) &= 0 \quad \text{Solution non triviale}
\end{aligned}
$$

Le développement du premier membre de l'équation caractéristique est un polynôme de la forme :

$$s_0\lambda^n + s_1\lambda^{n-1} + \cdots + s_{n-1}\lambda + s_n = 0.$$

Les trois programmes ci-dessous sont associés pour définir le polynôme caractéristique d'une matrice quelconque dans la pile.

Le premier programme, TRCN, crée une liste des traces des n premières puissances de la matrice.

Le second programme, SYM, utilise la liste créée par TRCN pour calculer les coefficients du polynôme caractéristique.

Le troisième et dernier programme, PSERS, utilise les coefficients obtenus par SYM et un nom de variable entré en niveau 1 pour créer une expression du polynôme caractéristique.

Entrez le premier programme.

```
[CLEAR]
«  DUP SIZE 1 GET → g
n «  g 'tmp' STO { } 1 n
START 0 1 n FOR i tmp i
DUP 2 →LIST GET + NEXT 1
→LIST + 'tmp' g STO*
NEXT 'tmp'
PURGE » » [ENTER] [<>]
```

```
1: « DUP SIZE 1.00 GET
   → g n « g 'tmp' STO
   { } 1.00 n START
   0.00 1.00 n FOR i
```

Stockez ce programme.

'TRCN [STO]

```
4:
3:
2:
1:
```

Entrez le second programme.

```
«  DUP SIZE → b n «
{1} 1 n FOR i → s
«  0 1 i FOR j b j
GET s i j - 1 + GET ×
- NEXT i ÷ 1 →LIST s
SWAP + » NEXT » »
```

[ENTER] [<>]

```
1: « DUP SIZE → b n « {
   1.00 } 1.00 n FOR i
   → s « 0.00 1.00 i
   FOR j b j GET s i j
```

Stockez ce programme.

'SYM [STO]

```
4:
3:
2:
1:
```

Entrez le troisième programme.

```
«  → x «  LIST→ 0 SWAP
1 FOR n n 1 + ROLL
x n 1 - ^ × + -1
STEP » »
```

[ENTER] [<>]

```
1: « → x « LIST→ 0.00
   SWAP 1.00 FOR n n
   1.00 + ROLL x n 1.00
   - ^ * + -1.00 STEP »
```

Stockez ce programme.

'PSERS [STO]

```
4:
3:
2:
1:
```

Recherchez le polynôme caractéristique de la matrice suivante.

$$ARR = \begin{bmatrix} -17 & -57 & -69 \\ 1 & 5 & 3 \\ 5 & 15 & 21 \end{bmatrix}$$

Entrez la matrice des coefficients.

```
[[-17 -57 -69[1 5 3
[5 15 21 ENTER
```

```
2:
1: [[ -17.00 -57.00 -6…
    [ 1.00 5.00 3.00 ]
    [ 5.00 15.00 21.00…
```

Créez une liste des traces des *n* premières puissances de la matrice.

USER TRCN

```
2:
1: { 9.00 41.00 225.00
   }
PSERS SYM TRCN UT
```

Calculez les coefficients du polynôme caractéristique.

SYM

```
2:
1: { 1.00 -9.00 20.00
   -12.00 }
PSERS SYM TRCN UT
```

Créez l'expression algébrique du polynôme caractéristique sous le nom de variable *L*.

'L' PSERS

```
3:
2:
1:   'L^3-9*L^2+20*L-12'
PSERS SYM TRCN UT
```

Le polynôme caractéristique est :

$$\lambda^3 - 9\lambda^2 + 20\lambda - 12$$

Stockez ce polynôme comme expression courante dans EQ, il vous servira dans le problème suivant.

SOLV
STEQ

```
3:
2:
1:
STEQ RCEQ SOLVR ISOL QUAD SHOW
```

Calcul des valeurs propres (Méthode 1)

Méthode 1 : Développement du polynôme caractéristique et calcul de ses racines.

Calculez les valeurs propres en utilisant le polynôme caractéristique du problème précédent.

Effacez l'affichage et sélectionnez le mode d'affichage à deux décimales.

[CLEAR]
[MODE] 2 FIX

Effacez les paramètres de tracé courant.

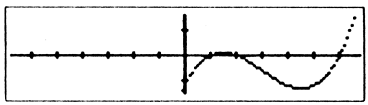
[PLOT]
'PPAR [PURGE]

Ajustez la hauteur du tracé à 10.

10 *H

Tracez un graphe du polynôme caractéristique qui avait été stocké dans EQ au problème précédent.

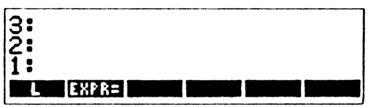
DRAW



Remarquez que les trois racines du quadrant indiquent qu'il existe trois valeurs propres distinctes pour la matrice *ARR* de format 3 x 3.

Utilisez la fonction SOLVR (du menu PLOT) pour obtenir des estimations de ces racines et calculez les trois valeurs propres.

[ATTN]
[SOLV]
SOLVR



Faites une estimation initiale de 0,5 pour la première racine.

0.5 L

```
L: 0.50
2:
1:
L EXPR=
```

Calculez la première racine.

Si vous appuyez sur la touche [ENTER] comme indiqué ci-dessous, vous pourez observer à l'affichage le processus d'itération de ce calcul.

□ L [ENTER]

```
L: 1.00
Zero
1:                  1.00
L EXPR=
```

La première valeur propre est $\lambda_1 = 1$

Faites une estimation initiale de 2,5 pour la deuxième valeur propre.

[CLEAR]
2.5 L

```
L: 2.50
2:
1:
L EXPR=
```

Calculez la deuxième racine.

□ L [ENTER]

```
L: 2.00
Zero
1:                  2.00
L EXPR=
```

La deuxième valeur propre est $\lambda_2 = 2$

Faites une estimation initiale de 5 pour la troisième valeur propre.

[CLEAR]
5 L

```
L: 5.00
2:
1:
L EXPR=
```

Calculez la troisième racine.

□ L [ENTER]

```
L: 6.00
Zero
1:                  6.00
L EXPR=
```

La troisième valeur propre est $\lambda_3 = 6$

Calcul des vecteurs propres

Nous allons calculer les vecteurs propres associés aux trois valeurs propres calculées dans le problème précédent.

$$ARR = \begin{bmatrix} -17 & -57 & -69 \\ 1 & 5 & 3 \\ 5 & 15 & 21 \end{bmatrix}$$

Effacez l'affichage et sélectionnez le mode d'affichage à une décimale.

CLEAR
MODE 1 FIX

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez la matrice *ARR*.

[[-17 -57 -69 [1 5 3
[5 15 21 ENTER

```
1: [[ -17.0 -57.0 -69....
     [ 1.0 5.0 3.0 ]
     [ 5.0 15.0 21.0 ]]
 STD [ FIX ] SCI  ENG [ DEG ] RAD
```

Créez la matrice identité *I* de format 3 x 3.

3 ENTER

```
3:
2: [[ -17.0 -57.0 -69....
1:                    3.0
 STD [ FIX ] SCI  ENG [ DEG ] RAD
```

ARRAY IDN

```
1: [[ 1.0 0.0 0.0 ]
     [ 0.0 1.0 0.0 ]
     [ 0.0 0.0 1.0 ]]
 SIZE  RDM  TRN  CON  IDN  RSD
```

Formez λ*I pour $\lambda_1 = 1$.

1 ENTER
×

```
1: [[ 1.0 0.0 0.0 ]
     [ 0.0 1.0 0.0 ]
     [ 0.0 0.0 1.0 ]]
 SIZE  RDM  TRN  CON  IDN  RSD
```

Soustrayez ce résultat de *ARR* pour obtenir la matrice $(ARR - \lambda_1 I)$.

−

```
1: [[ -18.0 -57.0 -69....
     [ 1.0 4.0 3.0 ]
     [ 5.0 15.0 20.0 ]]
 SIZE  RDM  TRN  CON  IDN  RSD
```

Stockez la matrice (ARR - $\lambda_1 I$) sous le nom de EIG.

'EIG [STO]

```
3:
2:
1:
SIZE RDM TRN CON IDN RSD
```

Rappelez la matrice EIG.

[USER] EIG

```
1: [[ -18.0 -57.0 -69....
    [ 1.0 4.0 3.0 ]
    [ 5.0 15.0 20.0 ]]
EIG L PPAR EQ UT
```

Vérifiez que det(A - λI) = 0.

[ARRAY]
DET

```
3:
2:
1:                -1.5E-10
CROSS DOT DET ABS RNRM CNRM
```

Le déterminant est très proche de zéro.

Rappeler à nouveau la matrice EIG.

[DROP]
[USER] EIG

```
1: [[ -18.0 -57.0 -69....
    [ 1.0 4.0 3.0 ]
    [ 5.0 15.0 20.0 ]]
EIG L PPAR EQ UT
```

Mettez cette matrice sous forme linéaire afin de calculer le vecteur propre X_1, où $(A - \lambda_1 I)X_1 = 0$.

Utilisez le mode EDIT et la touche [DEL] pour supprimer les crochets externes délimitant le tableau et pour former trois vecteurs lignes séparés. Chacun de ces vecteurs lignes correspond à une équation du système. Après avoir procédé à cette opération, validez les trois vecteurs lignes.

```
[ -18 -57 -69 ]
 [ 1 4 3 ]
 [ 5 15 20 ] [ENTER]
```

```
3: [ -18.0 -57.0 -69.0...
2:         [ 1.0 4.0 3.0 ]
1:      [ 5.0 15.0 20.0 ]
EIG L PPAR EQ UT
```

Les sous-programmes utilitaires de la section "Orthogonalité" peuvent également être utilisés pour procéder à ces transformations.

Utilisez le programme UT figurant dans la section "Systèmes d'équations linéaires sans second membre" pour transformer cette matrice en matrice triangulaire supérieure.

UT

```
3: [ -18.0 -57.0 -69.0...
2:         [ 0.0 0.8 -0.8 ]
1: [ 0.0 0.0 -1.0E-11 ]
EIG  L  PPAR  EQ  UT
```

Retirez le vecteur qui représente l'équation 0 = 0.

[DROP]

```
3:
2: [ -18.0 -57.0 -69.0...
1:         [ 0.0 0.8 -0.8 ]
EIG  L  PPAR  EQ  UT
```

Entrez l'équation représentée par le second vecteur.

' .8 × X2 - .8 × X3 =0
[ENTER]

```
3: [ -18.0 -57.0 -69.0...
2:         [ 0.0 0.8 -0.8 ]
1:       '0.8*X2-0.8*X3=0'
EIG  L  PPAR  EQ  UT
```

Résolvez-la pour x_2.

[ALGEBRA]
'X2 [ENTER]

```
3:         [ 0.0 0.8 -0.8 ]
2:       '0.8*X2-0.8*X3=0'
1:                    'X2'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Isolez le terme.

ISOL

```
3: [ -18.0 -57.0 -69.0...
2:         [ 0.0 0.8 -0.8 ]
1:            '0.8*X3/0.8'
TAYLR ISOL QUAD SHOW OBGET EXGET
```

Simplifiez.

COLCT

```
3: [ -18.0 -57.0 -69.0...
2:         [ 0.0 0.8 -0.8 ]
1:                    'X3'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Nous obtenons $x_2 = x_3$. Retirez cette solution de la pile ainsi que le second vecteur.

[DROP]
[DROP]

```
3:
2:
1: [ -18.0 -57.0 -69.0...
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Entrez l'équation représentée par le premier vecteur, en substituant x_2 à x_3.

`'-18 × X1 -57 × X2`
`-69 × X2 =0` [ENTER]

```
2: [ -18.0 -57.0 -69.0…
1: '-18*X1-57*X2-69*X2=
   0'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Résolvez-la pour x_1.

`'X1` [ENTER]

```
3: [ -18.0 -57.0 -69.0…
2: '-18*X1-57*X2-69*X2…
1:                 'X1'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Isolez le terme.

ISOL

```
2: [ -18.0 -57.0 -69.0…
1: '(69*X2+57*X2)/(-18)
   '
TAYLR ISOL QUAD SHOW OBGET EXGET
```

Simplifiez.

COLCT

```
3:
2: [ -18.0 -57.0 -69.0…
1:            '-(7.0*X2)'
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Nous obtenons $x_1 = -7^* x_2$.

Il existe donc un vecteur propre solution : $x_1 = -7$, $x_2 = 1$, $x_3 = 1$, soit $X_1 = [-7\ 1\ 1]$. Vérifiez que $(A - \lambda I)X = 0$.

[CLEAR]
`[-7 1 1` [ENTER]

```
3:
2:
1:       [ -7.0 1.0 1.0 ]
COLCT EXPAN SIZE FORM OBSUB EXSUB
```

Rappelez $(A - \lambda I)$.

[USER]
EIG

```
1: [[ -18.0 -57.0 -69.…
    [ 1.0 4.0 3.0 ]
    [ 5.0 15.0 20.0 ]]
EIG L PPAR EQ UT
```

Multipliez les deux matrices.

[SWAP]
[×]

```
3:
2:
1:        [ 0.0 0.0 0.0 ]
EIG L PPAR EQ UT
```

Le résultat est zéro. Ce qui vérifie que X_1 est vraiment un vecteur propre associé à λ_1 .

La même procédure peut être utilisée pour calculer les deux autres vecteurs propres. Le résultat est $\lambda_2 = 2$ et $\lambda_3 = 6$.

Effacez les variables et programmes utilisateur ayant servi aux trois derniers problèmes.

`{ 'EIG' 'L' 'PPAR' 'EQ' 'UT' }` [PURGE].

Calcul des valeurs propres (Méthode 2)

Méthode 2 : Calculez les valeurs propres directement à partir de la fonction det(λI - A), sans passer par l'équation caractéristique.

$$A = \begin{bmatrix} -7.8 & -29.7 & -39.6 \\ 0 & 2.1 & 0 \\ 3.3 & 9.9 & 15.3 \end{bmatrix}$$

Effacez l'affichage et sélectionnez le mode d'affichage à deux décimales.

[CLEAR]
[MODE] 2 [FIX]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Effacez les paramètres de tracé courant.

'PPAR [PURGE]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Introduisez la matrice de format 3 x 3.

```
[[-7.8 -29.7 -39.6
[0 2.1 0[3.3 9.9 15.3
```
[ENTER]

```
1: [[ -7.80 -29.70 -39...
    [ 0.00 2.10 0.00 ]
    [ 3.30 9.90 15.30 ...
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Stockez la matrice A.

'A [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez un programme calculant la fonction det(λI - A), dans laquelle λ est la variable indépendante.

« 3 IDN L × A - DET »
[ENTER]

```
2:
1: « 3.00 IDN L * A -
   DET »
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Stockez la fonction en tant qu'expression courante dans EQ.

[PLOT]
[STEQ]

```
3:
2:
1:
STEQ RCEQ PMIN PMAX INDEP DRAW
```

Ajustez la hauteur du graphe.

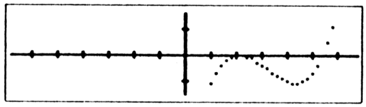
5 *H

Définissez une résolution plus élevée.

2 RES

Tracez la fonction, avec λ en abscisse. L'exécution du programme dure plusieurs minutes parce qu'il doit effectuer un calcul de déterminant pour chaque point à tracer.

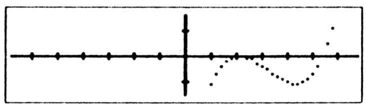
DRAW



La courbe montre qu'il n'y a que deux racines distinctes. La racine de gauche est un maximum local qui doit représenter une racine double de valeur propre.

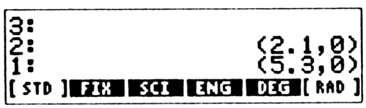
Numérisez ces racines pour vos estimations de départ à la fonction SOLVR.

[>] . . . [>] [INS]
[>] . . . [>] [INS]



Sélectionnez le mode d'affichage standard.

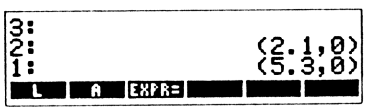
[ATTN]
[MODE] STD



Remarque : les valeurs affichées sont légèrement différentes de leur représentation graphique.

Utilisez la fonction SOLVR pour calculer les racines du graphe.

[SOLV]
SOLVR



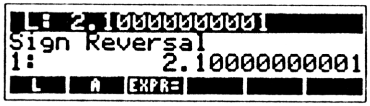
Calculez la racine de droite.

L ☐ L ENTER

Une racine est $\lambda_1 = 5.40$.

Retirez ce résultat de la pile et calculez la racine suivante.

DROP
L ☐ L ENTER



La valeur propre double est $\lambda_2 = \lambda_3 = 2.10$.

Moindres carrés

La méthode des moindres carrés est un algorithme statistique standard utilisé pour ajuster une droite à des données afin d'estimer une fonction, prévoir une tendance ou approximer des valeurs manquantes. Les résultats de cet algorithme sont facilement calculés par le HP-28C dont l'affichage graphique est particulièrement commode pour visualiser les ajustements réalisés.

Ajustement linéaire

Ajustez à une droite, par leurs moindres carrés, les quatre points : (0,1), (1,3), (2,4) et (3,4).

La solution est donnée par $Y = MV$ pour la droite d'ajustement $y = ax + b$.

Remarque : La solution proposée ci-dessous permet d'illustrer des opérations matricielles. Elle pourrait être remplacée, dans le cas de $y = ax + b$, par les fonctions statistiques intégrées (régression linéaire) du HP-28C.

$$Y = \begin{bmatrix} 1 \\ 3 \\ 4 \\ 4 \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}$$

$$V = \begin{bmatrix} a \\ b \end{bmatrix}$$

Le calcul pour V donne :

$$V = \frac{M^T Y}{M^T M}$$

CLEAR
MODE 2 FIX

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez les valeurs y des points de données.

[[1 [3 [4 [4 ENTER

```
1:  [[ 1.00 ]
     [ 3.00 ]
     [ 4.00 ]
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Stockez la matrice *Y* de format 4 x 1.

'Y [STO]

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez les coefficients *a* et *b* pour la droite $y = ax + b$.

[[0 1[1 1[2 1[3 1 [ENTER]

```
1: [[ 0.00 1.00 ]
    [ 1.00 1.00 ]
    [ 2.00 1.00 ]
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Stockez la matrice *M* de format 4 x 2.

'M [STO]

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Calculez *V* par la méthode des moindres carrés.

M [ENTER]
[ARRAY]
TRN
Y [×]

```
2:
1: [[ 23.00 ]
    [ 12.00 ]]
 SIZE  RDM  TRN  CON  IDN  RSD
```

M [ENTER]
TRN
M [×]

```
2: [[ 23.00 ]  [ 12.00...
1: [[ 14.00 6.00 ]
    [ 6.00 4.00 ]]
 SIZE  RDM  TRN  CON  IDN  RSD
```

[÷]

```
2:
1: [[ 1.00 ]
    [ 1.50 ]]
 SIZE  RDM  TRN  CON  IDN  RSD
```

Stockez les coefficients de la matrice *V* dans les variables individuelles *a* et *b*.

ARRY→

```
3:                    1.00
2:                    1.50
1:          { 2.00 1.00 }
 →ARRY ARRY→  PUT  GET  PUTI  GETI
```

Effacez la liste de format.

[DROP]

```
3:
2:                    1.00
1:                    1.50
 →ARRY ARRY→  PUT  GET  PUTI  GETI
```

Stockez les deux coefficients.

'B [STO]

```
3:
2:
1:                    1.00
→ARRY ARRY→ PUT GET PUTI GETI
```

'A [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Entrez l'équation de la droite.

'A × X + B [ENTER]

```
3:
2:
1:                 'A*X+B'
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez cette équation sous le nom LINE.

'LINE [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Rappelez l'équation LINE.

[USER] LINE

```
3:
2:
1:                 'A*X+B'
LINE A B M Y
```

Stockez l'équation LINE en tant qu'expression courante de EQ.

SOLVR EXPR=

```
EXPR='1.00*X+1.50'
2:
1:          '1.00*X+1.50'
A X B EXPR=
```

Utilisez la fonction SOLVR pour calculer la droite recherchée.

[SOLV] STEQ

```
3:
2:
1:
STEQ RCEQ SOLVR ISOL QUAD SHOW
```

La droite d'ajustement recherchée est $y = 1.5x + 1$.

Maintenant utilisez le menu PLOT pour tracer cette droite et vérifier que les données *y* sont bien ajustées. Effacez les paramètres de tracé courants.

[PLOT]
'PPAR [PURGE]

```
3:
2:
1:              '1.00*X+1.50'
STEQ RCEQ PMIN PMAX INDEP DRAW
```

Choisissez *X* comme variable indépendante.

'X INDEP

```
3:
2:
1:              '1.00*X+1.50'
STEQ RCEQ PMIN PMAX INDEP DRAW
```

Choisissez la valeur 5 comme unité de hauteur du tracé.

5 *H

```
3:
2:
1:              '1.00*X+1.50'
PPAR RES AXES CENTR *W *H
```

Recentrez les axes pour que le point (0,1) soit affiché.

(-1,-1) [ENTER]
AXES

```
3:
2:
1:              '1.00*X+1.50'
PPAR RES AXES CENTR *W *H
```

Maintenant utilisez le menu STAT pour tracer le nuage de points.

[STAT] CLΣ

```
3:
2:
1:              '1.00*X+1.50'
Σ+ Σ- NΣ CLΣ STOΣ RCLΣ
```

Entrez les coordonnées des quatre points au moyen de Σ+ .

[0,1 Σ+
[1,3 Σ+
[2,4 Σ+
[3,4 Σ+

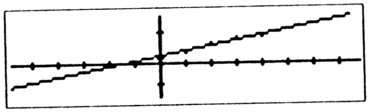
```
3:
2:
1:              '1.00*X+1.50'
Σ+ Σ- NΣ CLΣ STOΣ RCLΣ
```

Entrez un programme qui va superposer le tracé de la fonction sur le nuage de points.

[PLOT]
« CLLCD DRWΣ DRAW
[ENTER]

```
3:
2:              '1.00*X+1.50'
1:   « CLLCD DRWΣ DRAW »
STEQ RCEQ PMIN PMAX INDEP DRAW
```

Tracez le graphe.

[EVAL] 

Nous observons que la droite ajuste correctement les quatre points.

Effacez les variables utilisées dans ce problème. `{ 'ΣPAR' 'ΣDAT' 'PPAR' 'EQ' 'A' 'B' 'M' 'Y' }` [PURGE].

Equation du second degré

Selon la seconde loi du mouvement de Newton, tout corps lâché près de la surface de terre tombe verticalement selon l'équation :

$$y = y_0 + v_0 t + \frac{1}{2} g\, t^2$$

où

y = distance à l'instant t.
y_0 = distance initiale à l'instant t_0= 0.
v_0 = vitesse initiale à l'instant t_0= 0.
g = constante de Newton représentant l'accélération de la gravité près de la surface de la terre.

Une expérience est mise en oeuvre pour évaluer g. Un poids est lâché à une distance initiale et à une vitessse inconnues. A intervalles fixes, la distance parcourue est mesurée à partir d'un point de référence fixe. On obtient les résultats suivants : aux instants t = 0.1 0.2 .... 0.5 secondes, le poids a parcouru les distances, exprimées en mètres, y = -0.55 0.094 0.314 0.756 et 1.138 respectivement, à partir du point de référence. Calculez la valeur de la constante de Newton g à l'aide ces données.

Nous allons procéder à un ajustement à la courbe du second degré

$$y = a + bt + ct^2$$

pour ces cinq points. La solution des moindres carrés est donnée par :

$$Y = MV$$

où

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \\ 1 & t_4 & t_4^2 \\ 1 & t_5 & t_5^2 \end{bmatrix}$$

et

$$V = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

La résolution pour V donne : $V = \dfrac{M^T Y}{M^T M}$

Effacez la pile et sélectionnez le mode d'affichage à trois décimales.

CLEAR
MODE 3 FIX

```
3:
2:
1:
STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez la matrice des coordonnées y.

`[[-.055[.094[.314[.756[`
`1.138` ENTER

```
1: [[ -0.055 ]
    [ 0.094 ]
    [ 0.314 ]
STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Stockez cette matrice Y de format 5 x 1.

'Y STO

```
3:
2:
1:
STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez les éléments du tableau M.

Entrez $\text{ligne}_1 = 1, .1 , .1^2$.

1 ENTER
.1 ENTER
ENTER
x^2

```
3:                1.000
2:                0.100
1:                0.010
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez $\text{ligne}_2 = 1, .2 , .2^2$.

1 ENTER
.2 ENTER
ENTER
x^2

```
3:                1.000
2:                0.200
1:                0.040
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez $\text{ligne}_3 = 1, .3 , .3^2$.

1 ENTER
.3 ENTER
ENTER
x^2

```
3:                1.000
2:                0.300
1:                0.090
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez $\text{ligne}_4 = 1, .4 , .4^2$.

1 ENTER
.4 ENTER
ENTER
x^2

```
3:                1.000
2:                0.400
1:                0.160
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Enfin, entrez $\text{ligne}_5 = 1, .5 , .5^2$.

1 ENTER
.5 ENTER
ENTER
x^2

```
3:                1.000
2:                0.500
1:                0.250
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez le format de la matrice *M*.

{5 3 ENTER

```
3:                0.500
2:                0.250
1:    { 5.000 3.000 }
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Mettez ses éléments dans le tableau.

[ARRAY]
≡ →ARRY ≡

```
1: [[ 1.000 0.100 0.01…
    [ 1.000 0.200 0.04…
    [ 1.000 0.300 0.09…
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez la matrice M.

'M [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez V en utilisant la méthode des moindres carrés.

M [ENTER]
≡ TRN ≡
Y [×]

```
1: [[ 2.247 ]
    [ 0.979 ]
    [ 0.437 ]]
SIZE RDM TRN CON IDN RSD
```

M [ENTER]
≡ TRN ≡
M [×]
[÷]

```
1: [[ -0.121 ]
    [ 0.099 ]
    [ 4.914 ]]
SIZE RDM TRN CON IDN RSD
```

Stockez la matrice V.

'V [STO]

```
3:
2:
1:
SIZE RDM TRN CON IDN RSD
```

Evaluez g, la constante de gravité de Newton. Rappelez l'élément c du vecteur solution V, puis multipliez c par 2. $g = 2*c$..

V [ENTER]
{3 1}
≡ GET ≡
2 [×]

```
3:
2:
1:                9.829
→ARRY ARRY→ PUT GET PUTI GETI
```

Maintenant, utilisez la fonction SOLVR pour calculer le polynôme du second degré de l'équation recherchée.

'A + B × T + C × T^2
[ENTER]

```
3:               32.246
2:                 'ft'
1:       'A+B*T+C*T^2'
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez l'équation sous le nom de POLY.

'POLY [STO]

```
3:
2:                    32.246
1:                      'ft'
→ARRY ARRY→  PUT  GET  PUTI GETI
```

Rappelez les coefficients de la matrice *V*.

V [ENTER]

```
1: [[ -0.121 ]
    [ 0.099 ]
    [ 4.914 ]]
→ARRY ARRY→  PUT  GET  PUTI GETI
```

ARRY→

```
3:                     0.099
2:                     4.914
1:        { 3.000 1.000 }
→ARRY ARRY→  PUT  GET  PUTI GETI
```

Retirez la liste de format.

[DROP]

```
3:                    -0.121
2:                     0.099
1:                     4.914
→ARRY ARRY→  PUT  GET  PUTI GETI
```

Stockez les trois coefficients *a*, *b* et *c*.

'C [STO]

```
3:                      'ft'
2:                    -0.121
1:                     0.099
→ARRY ARRY→  PUT  GET  PUTI GETI
```

'B [STO]

```
3:                    32.246
2:                      'ft'
1:                    -0.121
→ARRY ARRY→  PUT  GET  PUTI GETI
```

'A [STO]

```
3:
2:                    32.246
1:                      'ft'
→ARRY ARRY→  PUT  GET  PUTI GETI
```

Rappelez l'équation.

[USER] POLY

```
3:                32.246
2:                  'ft'
1:       'A+B*T+C*T^2'
  A    B    C   POLY  V    M
```

Stockez cette équation en tant qu'expression courante de EQ.

[SOLV]
STEQ

```
3:
2:                32.246
1:                  'ft'
STEQ RCEQ SOLVR ISOL QUAD SHOW
```

Utilisez la fonction SOLVR pour calculer l'équation recherchée.

SOLVR
EXPR=

```
EXPR='-0.121+0.099*T+
1:  '-0.121+0.099*T+
    4.914*T^2'
  A    B    T    C  EXPR=
```

L'équation solution par les moindres carrés est -0.121 + 0.099t + 4.914 t^2 .

Nous allons maintenant superposer la courbe de cette fonction sur le nuage de points des données observées pour vérifier l'ajustement.

Tout d'abord, effacez les paramètres de tracé courants et choisissez t comme variable indépendante.

[CLEAR]
[PLOT] 'PPAR [PURGE]
'T INDEP

```
3:
2:
1:
STEQ RCEQ PMIN PMAX INDEP DRAW
```

Choisissez .1 comme unité de largeur du tracé pour obtenir en abscisse des intervalles de 0,1 seconde.

.1 *W

```
3:
2:
1:
PPAR RES AXES CENTR *W *H
```

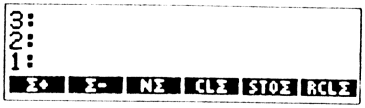
Ensuite utilisez le menu STAT pour créer le nuage de points.

[STAT]
CLΣ

```
3:
2:
1:
Σ+  Σ-  NΣ  CLΣ  STOΣ RCLΣ
```

Entrez les coordonnées des points du nuage.

```
[.1  -.055  Σ+
[.2  .094  Σ+
[.3  .314  Σ+
[.4  .756  Σ+
[.5  1.138  Σ+
```



Maintenant écrivez un programme pour superposer les deux tracés.

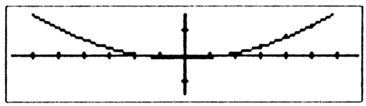
[PLOT]
« CLLCD DRWΣ DRAW
[ENTER]

Stockez ce programme sous le nom de PLT.

'PLT [STO]

Lancez le tracé.

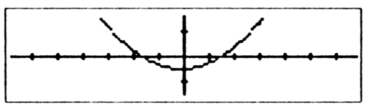
[USER] PLT



Si vous voulez modifier l'échelle du tracé pour mieux observer votre ajustement pour les deux premiers points de données, effectuez :

.25 *H

[USER]
PLT



Les tracés illustrent un bon ajustement des cinq points de données par cette équation du second degré.

Séries matricielles de Markov

Une série de Markov est un système qui change d'un état à l'autre, et dans lequel la probabilité de transition à l'état suivant ne dépend que de l'état précédent. Les états du système peuvent être probabilisés à certains instants en utilisant la technique des probabilités de transition.

La matrice de transition du processus de Markov est la matrice carrée $P = [p_{ij}]$ de format n x n, dans laquelle p_{ij} = probabilité de transition de l'état j à l'état i, avec :

$$\sum_{i=1}^{n} p_{ij} = 1.$$

Les composantes du vecteur d'état $X^{(n)}$ représentent la probabilité que le système soit à l'état i à la $n^{ième}$ observation.

$$X^{(n)} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$$

Le modèle du système est décrit par $X^{(n+1)} = PX^{(n)}$, dans lequel la matrice de transition appliquée à l'état courant détermine l'état suivant.

Un processus de Markov stable

Un chimiste procède à une expérience au cours de laquelle des films couleur sont trempés dans une solution pendant une courte période de temps, afin de provoquer un changement de couleur. Il veut calculer ces changements de couleur pour les probabilités suivantes.

Couleur initiale			**Nouvelle couleur**
Magenta	*Cyan*	*Jaune*	
.8	.3	.2	*Magenta*
.1	.2	.6	*Cyan*
.1	.5	.2	*Jaune*

Déterminez, avec deux décimales, la couleur probable d'un film cyan trempé plusieurs fois de suite dans la solution.

[CLEAR]
[MODE] 2 [FIX]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez la matrice de transition P de format 3 x 3.

[[.8 .3 .2[.1 .2 .6[.1
.5 .2 [ENTER]

```
1: [[ 0.80 0.30 0.20 ]
    [ 0.10 0.20 0.60 ]
    [ 0.10 0.50 0.20 ]]
STD [ FIX ] SCI ENG DEG [ RAD ]
```

'P [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez le vecteur d'état initial X^0 . Ce vecteur représente l'état initial du cyan.

[[0 [1 [0 [ENTER]

```
1: [[ 0.00 ]
    [ 1.00 ]
    [ 0.00 ]]
STD [ FIX ] SCI ENG DEG [ RAD ]
```

'X [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Entrez la valeur initiale pour n = état courant.

0 [ENTER]
'N [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Ecrivez un programme pour calculer l'état suivant.

« N 1 + 'N' STO P
SWAP × » [ENTER]

```
2:
1: « N 1.00 + 'N' STO P
   SWAP * »
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Stockez ce programme sous le nom de MARK.

'MARK [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Rappelez le vecteur d'état initial.

[USER]
X

```
1: [[ 0.00 ]
    [ 1.00 ]
    [ 0.00 ]]
MARK  N  X  P
```

Calculez l'état suivant.

MARK

```
1: [[ 0.30 ]
    [ 0.20 ]
    [ 0.50 ]]
MARK  N  X  P
```

A la fin de la première observation, la couleur sera probablement jaune. Calculez l'état suivant.

MARK

```
1: [[ 0.40 ]
    [ 0.37 ]
    [ 0.23 ]]
MARK  N  X  P
```

A la deuxième observation, la couleur sera probablement soit magenta soit cyan. Continuez le calcul des états successifs jusqu'à obtention d'un résultat final stable.

MARK

```
1: [[ 0.48 ]
    [ 0.25 ]
    [ 0.27 ]]
MARK  N  X  P
```

MARK

```
1: [[ 0.51 ]
    [ 0.26 ]
    [ 0.23 ]]
MARK  N  X  P
```

MARK

```
1: [[ 0.53 ]
    [ 0.24 ]
    [ 0.23 ]]
MARK  N  X  P
```

MARK

```
1: [[ 0.54 ]
    [ 0.24 ]
    [ 0.22 ]]
MARK  N  X  P
```

MARK

```
1: [[ 0.55 ]
    [ 0.23 ]
    [ 0.22 ]]
MARK  N  X  P
```

MARK

```
1: [[ 0.55 ]
    [ 0.23 ]
    [ 0.21 ]]
MARK  N  X  P
```

MARK

```
1: [[ 0.56 ]
    [ 0.23 ]
    [ 0.21 ]]
MARK  N  X  P
```

MARK

```
1: [[ 0.56 ]
    [ 0.23 ]
    [ 0.21 ]]
MARK  N  X  P
```

Le système est parvenu à un état stable. Déterminez alors combien il a fallu d'observations pour atteindre cet état.

N

```
3:
2: [[ 0.56 ]  [ 0.23 ]…
1:                 10.00
MARK  N  X  P
```

Le système parvient à un état stable au bout de n = 10 observations. La couleur future probable d'un film initialement cyan, immergé à plusieurs reprises est magenta à 56 %, cyan à 23 % et jaune à 21 %.

Effacez les variables utilisées dans ce chapitre.
`{ 'MARK'   'N'   'X'   'P' }` PURGE.

Un exemple

Les opérations matricielles servent à résoudre des problèmes multi-dimensionnels complexes. Les problèmes ci-après illustrent l'utilisation du HP-28C en traitement matriciel de problèmes économiques. Les mêmes outils analytiques peuvent être utilisés dans de nombreuses autres applications.

Gestion d'une exploitation forestière

Pour gérer une exploitation forestière selon une politique réaliste de récolte, il suffit de ré-ensemencer tous les ans le terrain des coupes de l'année, afin que la population des arbres reste constante. Un modèle matriciel peut être développer pour aider à déterminer la fréquence optimale des coupes. Ce modèle est basé sur une caractérisation des arbres par catégories hauteur/prix et sur le calcul du rendement optimal sur une longue période de temps.

Le cycle de coupe est représenté par la formule :

Forêt mûre pour la coupe - coupe + ensemencements = Forêt après coupe

soit $GX - Y + RY = X$

où

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

X = vecteur de forêt non coupée, c'est à dire les arbres restants après coupe et ensemencement.

x_i = nombre d'arbres de la catégorie i.

$i = 1 \ldots n = n$ catégories hauteur/prix.

$S = \sum_{i=1}^{n} x_i$ = nombre total d'arbres

La croissance des arbres entre deux coupes est désignées par :
g_i = partie des arbres qui passent de la catégorie i à la catégorie $i + 1$.

$1 - g_i$ = partie des arbres qui restent dans la catégorie i.

La matrice de croissance est :

$$G = \begin{bmatrix} 1-g_1 & 0 & 0 & \cdot & 0 \\ g_1 & 1-g_2 & 0 & \cdot & 0 \\ 0 & g_2 & 1-g_3 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & 1-g_{n-1} & 0 \\ 0 & 0 & 0 & g_{n-1} & 1 \end{bmatrix}$$

GX = Vecteur de forêt non coupée après la période de croissance, c'est à dire forêt mûre pour la coupe.

$$Y = \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix}$$

Y = Vecteur de coupe (arbres coupés)

$$R = \begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & 1 \\ 0 & 0 & 0 & \cdot & \cdot & 0 \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

R = matrice de remplacement.

RY = vecteur des nouvelles semences, c'est à dire les arbres plantés après la coupe.

Le modèle de la récolte

Un récoltant a une récolte de 120 sapins argentés à vendre annuellement pour Noël. A la fin de la dernière récolte, sa forêt avait la configuration suivante :

Catégorie i	**Hauteur** h_i	**Nombre** x_i
1	[0 , 4)	15
2	[4 , 8)	20
3	[8 , 12)	35
4	[12 , 16)	30
5	[16 , ∞)	20

Pendant la période de croissance, 6 arbres de la catégorie 1 sont passés dans la catégorie supérieure, ainsi que 13 arbres de la catégorie 2, 10 arbres de la catégorie 3 et 4 arbres de la catégorie 4. S'il récolte 8 arbres de la catégorie 2, 6 arbres de la catégorie 3, 13 arbres de la catégorie 4 et 6 arbres de la catégorie 5, quelle est la configuration de sa forêt après coupe et replantation ?

CLEAR
MODE 2 FIX

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Entrez le vecteur *X* de forêt non coupée, de format 5 x 1.

[[15[20[35[30[20 ENTER

```
1: [[ 15.00 ]
    [ 20.00 ]
    [ 35.00 ]
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

'X STO

```
3:
2:
1:
 STD [ FIX ] SCI  ENG  DEG [ RAD ]
```

Calculez la croissance pour chaque catégorie de hauteur.
D'abord, calculez $g_1 = 6/x_1$.

6 [ENTER]
15 [÷]

```
3:
2:
1:                    0.40
STD [ FIX ] SCI ENG DEG [ RAD ]
```

'G1 [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Calculez ensuite $g_2 = 13/x_2$.

13 [ENTER]
20 [÷]

```
3:
2:
1:                    0.65
STD [ FIX ] SCI ENG DEG [ RAD ]
```

'G2 [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Calculez ensuite $g_3 = 10/x_3$.

10 [ENTER]
35 [÷]

```
3:
2:
1:                    0.29
STD [ FIX ] SCI ENG DEG [ RAD ]
```

'G3 [STO]

```
3:
2:
1:
STD [ FIX ] SCI ENG DEG [ RAD ]
```

Calculez enfin $g_4 = 4/x_4$.

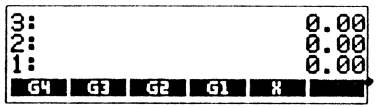
4 ENTER
30 ÷

'G4 STO

Entrez la matrice de croissance G de format 5 x 5.

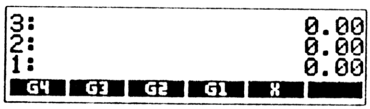
Entrez la ligne 1.

USER
1 ENTER
G1
−
0 ENTER
ENTER
ENTER
ENTER



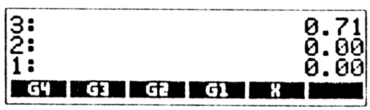
Entrez la ligne 2.

G1
1 ENTER
G2
−
0 ENTER
ENTER
ENTER



Entrez la ligne 3.

0 ENTER
G2
1 ENTER
G3
−
0 ENTER
ENTER



Entrez la ligne 4.

0 [ENTER]
[ENTER]
G3
1 [ENTER]
G4
[-]
0 [ENTER]

```
3:                  0.29
2:                  0.87
1:                  0.00
 G4   G3   G2   G1   X
```

Entrez la ligne 5.

0 [ENTER]
[ENTER]
[ENTER]
G4
1 [ENTER]

```
3:                  0.00
2:                  0.13
1:                  1.00
 G4   G3   G2   G1   X
```

Entrez le format de *G*.

{5 5} [ENTER]

```
3:                  0.13
2:                  1.00
1:         { 5.00 5.00 }
 G4   G3   G2   G1   X
```

Stockez cette matrice *G*.

[ARRAY]
→ARRY

```
1: [[ 0.60 0.00 0.00 0...
    [ 0.40 0.35 0.00 0...
    [ 0.00 0.65 0.71 0...
→ARRY ARRY→ PUT GET PUTI GETI
```

'G [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Entrez le vecteur de coupe *Y*, de format 5 x 1.

[[0[8[6[13[6 [ENTER]

```
1: [[ 0.00 ]
    [ 8.00 ]
    [ 6.00 ]
→ARRY ARRY→ PUT GET PUTI GETI
```

'Y [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Créez la matrice de remplacement *R*. D'abord, entrez le format de *R*.

{5 5} [ENTER]

```
3:
2:
1:           { 5.00 5.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

Créez une matrice de constantes dont tous les éléments sont des zéros.

0 [ENTER]
CON

```
1: [[ 0.00 0.00 0.00 0...
    [ 0.00 0.00 0.00 0...
    [ 0.00 0.00 0.00 0...
SIZE RDM TRN CON IDN RSD
```

Maintenant entrez des 1 sur la première ligne de *R*.

{1 1} [ENTER]

```
3:
2: [[ 0.00 0.00 0.00 0...
1:          { 1.00 1.00 }
SIZE RDM TRN CON IDN RSD
```

1 PUTI
1 PUTI
1 PUTI
1 PUTI
1 PUTI

```
3:
2: [[ 1.00 1.00 1.00 1...
1:          { 2.00 1.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

Retirez la liste d'index.

[DROP]

```
1: [[ 1.00 1.00 1.00 1...
    [ 0.00 0.00 0.00 0...
    [ 0.00 0.00 0.00 0...
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez la matrice *R*.

'R [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Ecrivez un programme pour calculer la configuration de la forêt après la coupe.

[USER]
« G X × Y − R Y × + »
[ENTER]

```
2:
1: « G X * Y - R Y * +
   »
 R  Y  G  G4  G3  G2
```

Stockez ce programme sous le nom de CROP.

'CROP [STO]

```
3:
2:
1:
CROP  R  Y  G  G4  G3
```

Calculez le nouveau vecteur de forêt non coupée avec le programme CROP.

CROP

```
1: [[ 42.00 ]
    [ 5.00 ]
    [ 32.00 ]
CROP  R  Y  G  G4  G3
```

Vous pouvez utiliser [EDIT] ou [VIEW↓] pour visualiser la totalité du vecteur. La touche [ATTN] vous permet de sortir du mode EDIT.

Le nouveau vecteur de forêt non coupée est :

$$X = \begin{bmatrix} 42 \\ 5 \\ 32 \\ 23 \\ 18 \end{bmatrix}$$

Le programme peut être utilisé avec le nouveau vecteur de forêt non coupée pour prévoir tous les ans les nouvelles configurations de la forêt, après les coupes.

Rendement optimal

Si ce récoltant souhaite optimiser son bénéfice, il doit déterminer le rendement optimal de ses coupes. Cela est possible en ne coupant *que* les arbres, et tous les arbres, d'une catégorie hauteur/prix particulière. Le rendement optimal est donc à la fois une fonction du prix et de la croissance des arbres, mais il est indépendant du vecteur courant de forêt non coupée. Remarquez que si la catégorie k produit le rendement optimal, la première année, toutes les catégories $\geq k$ seront coupées. Les années suivantes, seule la catégorie k sera coupée, et il n'y aura plus d'arbres de catégories supérieures.

S = nombre total d'arbres dans la forêt.

$$P = \begin{bmatrix} p_1 & 0 & \cdot & 0 \\ \cdot & p_2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & p_n \end{bmatrix} = \text{matrice de prix}$$

p_i = prix obtenu pour la catégorie i.

$$GG = \begin{bmatrix} gg_1 \\ gg_2 \\ \cdot \\ \cdot \\ gg_n \end{bmatrix} = \text{vecteur de ratio de croissance.}$$

où :

$$\begin{cases} gg_i = \dfrac{1}{\displaystyle\sum_{k=1}^{i-1} \frac{1}{g_k}} & \textit{pour } i = 2,\ldots,n \\ gg_1 = 0 \end{cases}$$

$$YL = \begin{bmatrix} yl_1 \\ yl_2 \\ \cdot \\ \cdot \\ yl_n \end{bmatrix} = \text{vecteur de rendement.}$$

yl_k = rendement (en valeur) obtenu de la coupe de tous les arbres de la catégorie i, à l'exclusion de toute autre catégorie.

La catégorie optimale de coupe peut être choisie en calculant la valeur maximale de yl_k dans le vecteur YL, où :

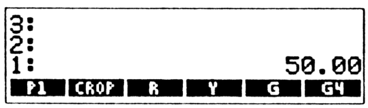
$$YL = P * S * GG$$

Supposons que les prix, pour chacune des cinq catégories, sont les suivants : $p_1 = 0$, $p_2 = 50$, $p_3 = 100$, $p_4 = 150$ et $p_5 = 200$.
Déterminez la catégorie à couper.

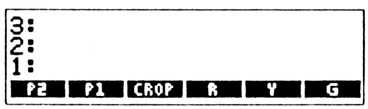
Entrez les prix des cinq catégories et stockez-les dans les variables p_1 à p_5.

$\boxed{\text{CLEAR}}$ $\boxed{\text{USER}}$
0 $\boxed{\text{ENTER}}$
'P1 $\boxed{\text{STO}}$

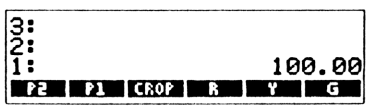
50 $\boxed{\text{ENTER}}$



'P2 $\boxed{\text{STO}}$



P2
2 $\boxed{\times}$



'P3 [STO]

```
3:
2:
1:
 P3  P2  P1  CROP  R  Y
```

[P2]
3 [×]

```
3:
2:
1:                150.00
 P3  P2  P1  CROP  R  Y
```

'P4 [STO]

```
3:
2:
1:
 P4  P3  P2  P1  CROP  R
```

[P2]
4 [×]

```
3:
2:
1:                200.00
 P4  P3  P2  P1  CROP  R
```

'P5 [STO]

```
3:
2:
1:
 P5  P4  P3  P2  P1  CROP
```

Entrez le format de la matrice de prix, *P*.

{5 5} [ENTER]

```
3:
2:
1:        { 5.00 5.00 }
 P5  P4  P3  P2  P1  CROP
```

Créez la matrice de prix *P* de format 5 x 5. Puisque *P* est une matrice clairsemée, composée principalement de zéros, créez d'abord un tableau de constantes dont tous les éléments sont des zéros.

0 [ENTER]
[ARRAY] [CON]

```
1: [[ 0.00 0.00 0.00 0...
    [ 0.00 0.00 0.00 0...
    [ 0.00 0.00 0.00 0...
 SIZE  RDM  TRN  CON  IDN  RSD
```

Maintenant, entrez les valeurs p_i sur la diagonale.

{1 1} [ENTER]

```
3:
2: [[ 0.00 0.00 0.00 0...
1:          { 1.00 1.00 }
SIZE RDM TRN CON IDN RSD
```

P1 [ENTER]

```
3: [[ 0.00 0.00 0.00 0...
2:          { 1.00 1.00 }
1:                   0.00
SIZE RDM TRN CON IDN RSD
```

PUTI

```
3:
2: [[ 0.00 0.00 0.00 0...
1:          { 1.00 2.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

Utilisez la fonction [EDIT] pour modifier l'index des positions affiché. Entrez ensuite le nouvel index. Une autre solution consiste à retirer {1.00 2.00} (par [DROP]) et à entrer l'index des positions {2 2}.

{ 2 2 } [ENTER]

```
3:
2: [[ 0.00 0.00 0.00 0...
1:          { 2.00 2.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

P2 [ENTER]

```
3: [[ 0.00 0.00 0.00 0...
2:          { 2.00 2.00 }
1:                  50.00
→ARRY ARRY→ PUT GET PUTI GETI
```

PUTI

```
3:
2: [[ 0.00 0.00 0.00 0...
1:          { 2.00 3.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

Utilisez la fonction [EDIT] pour modifier l'index des positions. Entrez-le ensuite.

{ 3 3 } [ENTER]

```
3:
2: [[ 0.00 0.00 0.00 0...
1:          { 3.00 3.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

P3 [ENTER]

```
3:  [[ 0.00 0.00 0.00 0...
2:           { 3.00 3.00 }
1:                  100.00
→ARRY ARRY→ PUT GET PUTI GETI
```

PUTI

```
3:
2:  [[ 0.00 0.00 0.00 0...
1:           { 3.00 4.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

Utilisez la fonction [EDIT] pour modifier l'index des positions. Entrez-le ensuite.

{ 4 4 } [ENTER]

```
3:
2:  [[ 0.00 0.00 0.00 0...
1:           { 4.00 4.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

P4 [ENTER]

```
3:  [[ 0.00 0.00 0.00 0...
2:           { 4.00 4.00 }
1:                  150.00
→ARRY ARRY→ PUT GET PUTI GETI
```

PUTI

```
3:
2:  [[ 0.00 0.00 0.00 0...
1:           { 4.00 5.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

Utilisez la fonction [EDIT] pour modifier l'index des positions. Entrez-le ensuite.

{ 5 5 } [ENTER]

```
3:
2:  [[ 0.00 0.00 0.00 0...
1:           { 5.00 5.00 }
→ARRY ARRY→ PUT GET PUTI GETI
```

P5 [ENTER]

```
3:  [[ 0.00 0.00 0.00 0...
2:           { 5.00 5.00 }
1:                  200.00
→ARRY ARRY→ PUT GET PUTI GETI
```

Retirez la liste d'index.

[DROP]

```
1: [[ 0.00 0.00 0.00 0...
    [ 0.00 50.00 0.00 ...
    [ 0.00 0.00 100.00...
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez la matrice *P*.

'P [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez le nombre total d'arbres dans la variable *S*.

120 [ENTER]

```
3:
2:
1:                   120.00
→ARRY ARRY→ PUT GET PUTI GETI
```

'S [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez la matrice des ratios de croissance, *GG*, de format 5 x 5.

Entrez $gg_1 = 0$.

0 [ENTER]
'GG1 [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez $gg_2 = 1/g_1$.

[USER] [G1]
[1/x]

```
3:
2:
1:                     2.50
GG1 G4 G3 G2 G1 S
```

'GG2 [STO]

```
3:
2:
1:
GG2 GG1 G4 G3 G2 G1
```

Calculez $gg_3 = 1/g_1 + 1/g_2$.

GG2
G2
[1/x]

```
3:
2:                 2.50
1:                 1.54
GG2 GG1 G4 G3 G2 G1
```

[+]

```
3:
2:
1:                 4.04
GG2 GG1 G4 G3 G2 G1
```

'GG3 [STO]

```
3:
2:
1:
GG3 GG2 GG1 G4 G3 G2
```

Calculez $gg_4 = 1/g_1 + 1/g_2 + 1/g_3$.

GG3
G3
[1/x]

```
3:
2:                 4.04
1:                 3.50
GG3 GG2 GG1 G4 G3 G2
```

[+]

```
3:
2:
1:                 7.54
GG3 GG2 GG1 G4 G3 G2
```

'GG4 [STO]

```
3:
2:
1:
GG4 GG3 GG2 GG1 G4 G3
```

Calculez $gg_5 = 1/g_1 + 1/g_2 + 1/g_3 + 1/g_4$.

GG4
G4
[1/x]

```
3:
2:                 7.54
1:                 7.50
GG4 GG3 GG2 GG1 G4 G3
```

[+]

```
3:
2:
1:                15.04
GG4 GG3 GG2 GG1 G4 G3
```

'GG5 [STO]

```
3:
2:
1:
GG5 GG4 GG3 GG2 GG1 G4
```

Maintenant inversez gg_2 , gg_3 , gg_4 et gg_5 pour former les véritables éléments de la matrice *GG*.

GG2
[1/x]

```
3:
2:
1:                  0.40
 GG5  GG4  GG3  GG2  GG1  G4
```

'GG2 [STO]

```
3:
2:
1:
 GG5  GG4  GG3  GG2  GG1  G4
```

GG3
[1/x]

```
3:
2:
1:                  0.25
 GG5  GG4  GG3  GG2  GG1  G4
```

'GG3 [STO]

```
3:
2:
1:
 GG5  GG4  GG3  GG2  GG1  G4
```

GG4
[1/x]

```
3:
2:
1:                  0.13
 GG5  GG4  GG3  GG2  GG1  G4
```

'GG4 [STO]

```
3:
2:
1:
 GG5  GG4  GG3  GG2  GG1  G4
```

GG5
[1/x]

```
3:
2:
1:                  0.07
 GG5  GG4  GG3  GG2  GG1  G4
```

'GG5 [STO]

```
3:
2:
1:
 GG5  GG4  GG3  GG2  GG1  G4
```

Créez la matrice *GG* de format 5 x 1. Mettez les éléments dans la pile.

GG1
GG2
GG3
GG4
GG5

```
3:                    0.25
2:                    0.13
1:                    0.07
GG5  GG4  GG3  GG2  GG1  G4
```

Entrez le format de la matrice.

{5 1 [ENTER]

```
3:                    0.13
2:                    0.07
1:          { 5.00 1.00 }
GG5  GG4  GG3  GG2  GG1  G4
```

Créez la matrice.

[ARRAY]
→ARRY

```
1:  [[ 0.00 ]
     [ 0.40 ]
     [ 0.25 ]
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez la matrice *GG*.

'GG [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Ecrivez un programme permettant de calculer le vecteur de rendement.

« S P × GG × » [ENTER]

```
3:
2:
1:          « S P * GG * »
→ARRY ARRY→ PUT GET PUTI GETI
```

Stockez ce programme sous le nom YLD.

'YLD [STO]

```
3:
2:
1:
→ARRY ARRY→ PUT GET PUTI GETI
```

Calculez le vecteur de rendement *YL* de format 5 x 1.

[USER]
YLD

```
1: [[ 0.00 ]
    [ 2400.00 ]
    [ 2971.43 ]
 YLD  GG  GG5  GG4  GG3  GG2
```

Vous pouvez utiliser [EDIT] ou [VIEW↓] pour visualiser la totalité du vecteur.

$$YL = \begin{bmatrix} 0 \\ 2400.00 \\ 2971.43 \\ 2387.75 \\ 1595.91 \end{bmatrix}$$

Le vecteur de rendement résultant de ce calcul montre que c'est la catégorie 3 qui maximise le rendement annuel, puisque yl_3 = 2971.43 est la valeur la plus grande.

Effacez les variables utilisateurs créées dans ce problème.

TABLE DES MATIÈRES

Omniplan Saclay (1) 60.19.04.21

Réf. : 00028-90044 FR

HEWLETT PACKARD