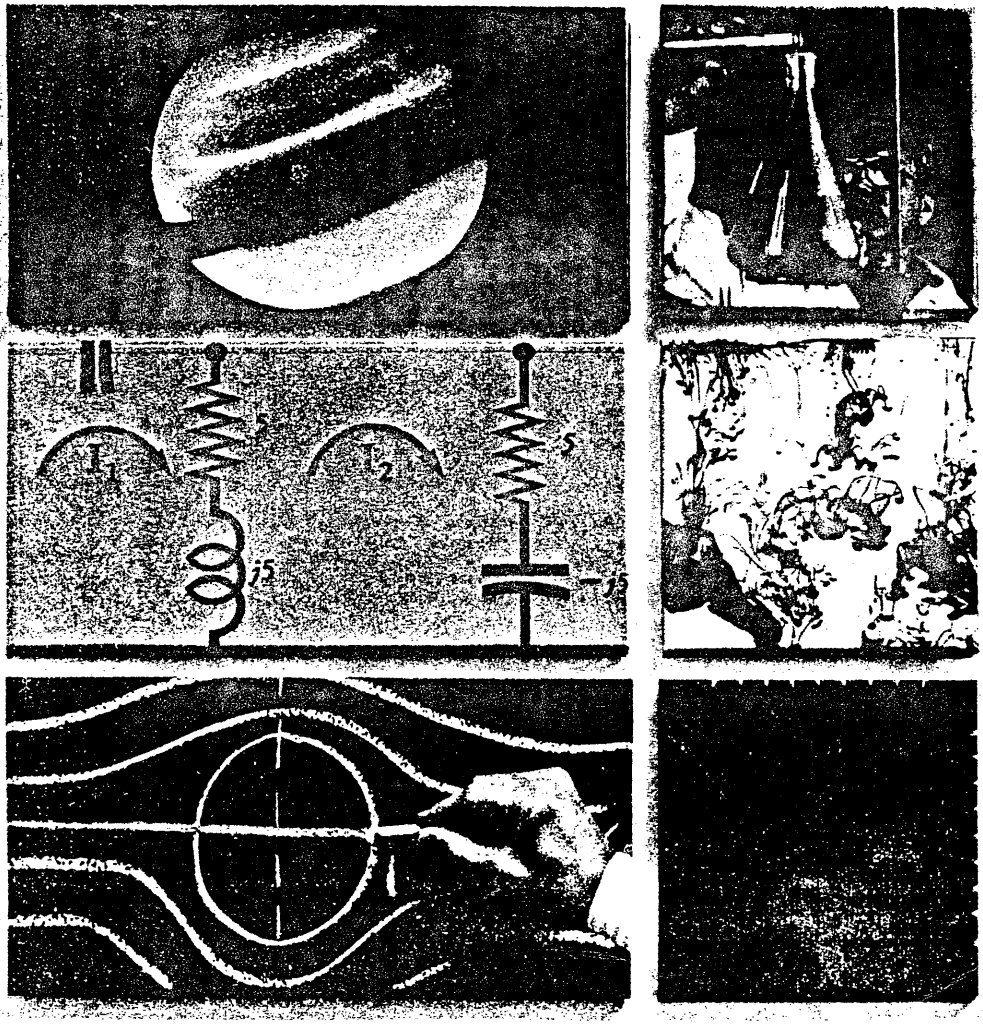
HEWLETT-PACKARD

HP-15C

MANUEL DES FONCTIONS MATHÉMATIQUES DE HAUT NIVEAU



HEWLETT PACKARD

HP-15C

Manuel des Fonctions Mathématiques de Haut Niveau

Janvier 1983

00015-90014

Imprimé à Singapour

Table des matières

Introduction

Le HP-15C est le premier calculateur programmable offrant autant de fonctions scientifiques disponibles à tout moment, où que vous soyez:

- Calcul de racines d'équations.
- Calcul d'intégrales finies.
- Calculs sur nombres complexes.
- Calcul matriciel.

Le *manuel d'utilisation du HP-15C* vous explique comment effectuer toutes ces opérations. Il contient de nombreux exemples illustrant l'utilisation de ces fonctions. Le manuel d'utilisation du HP-15C est votre manuel de référence. Le présent manuel, ***manuel des fonctions mathématiques de haut niveau du HP-15C***, complète le manuel d'utilisation du HP-15C en décrivant comment sont effectuées les fonctions étendues du calculateur HP-15C et en expliquant comment interpréter les résultats.

Ce manuel contient également de nombreux programmes (applications). Ces programmes ont une double utilité. Premièrement, ils suggèrent des méthodes d'utilisation des fonctions étendues pour que vous puissiez mettre en œuvre ces fonctions plus efficacement dans vos applications. Deuxièmement, ces programmes couvrant une vaste gamme d'applications, vous pouvez les utiliser tels quels éventuellement.

> Remarque: Les explications données ici supposent que vous connaissiez déjà les principes généraux d'utilisation des fonctions étendues et que les fonctions mathématiques décrites ici vous sont familières.

Chapitre 1

Utilisation de [SOLVE]

L'algorithme [SOLVE] offre une méthode très efficace de recherche des racines d'une équation. Ce chapitre décrit la méthode numérique utilisée par [SOLVE] et donne des conseils pratiques sur l'utilisation de [SOLVE] dans toute une variété de cas.

Recherche des racines d'une équation

En général, aucune technique numérique ne garantit dans tous les cas la résolution d'une équation même si elle a des racines. Comme on utilise un nombre fini de chiffres, la fonction calculée peut être différente de la fonction théorique dans certains intervalles de x et il peut être impossible de représenter exactement les racines ou de distinguer entre les zéros et les discontinuités de la fonction utilisée. La fonction n'étant échantillonnée que sur un nombre fini de positions, il est aussi possible de conclure à tort que l'équation n'a pas de racines.

Malgré ces limites inhérentes à toutes les méthodes numériques de recherche des racines, une méthode efficace, comme celle de [SOLVE], doit obéir aux principes suivants :

- Si une racine réelle existe et peut être représentée exactement par le calculateur, elle sera calculée. Notez que la fonction calculée peut être en dépassement de capacité inférieur (et mise à zéro) pour certaines valeurs de x autres que les vraies racines.

- Si une racine réelle existe mais ne peut être représentée exactement par le calculateur, la valeur calculée ne doit être différente de la vraie racine que sur le dernier chiffre significatif.

- Si aucune racine réelle n'existe, un message d'erreur doit être affiché.

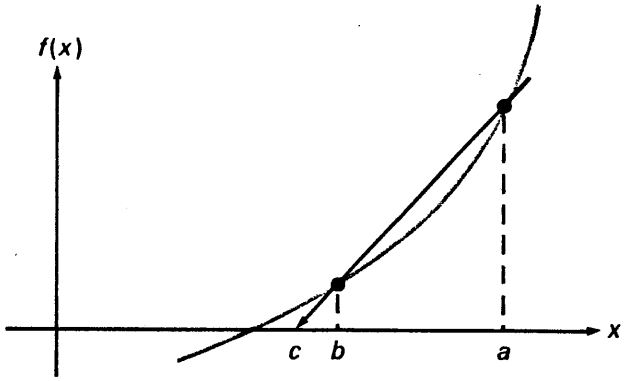
L'algorithme de [SOLVE] a été conçu pour répondre à ces principes. En outre, il est facile à utiliser et mobilise peu de mémoire. Enfin, comme [SOLVE] dans un programme peut détecter les situations de racines introuvables, vos programmes conservent leurs automatismes que [SOLVE] trouve ou non une racine.

Échantillonnage par SOLVE

Le programme SOLVE n'utilise que cinq registres de mémoire allouable sur le HP-15C. Ces cinq registres contiennent trois valeurs d'échantillonnage (a, b, et c) et deux valeurs précédentes de la fonction ($f(a)$ et $f(b)$) pendant que le sous-programme de la fonction calcule $f(c)$.

L'efficacité de SOLVE réside dans la façon dont est définie la valeur suivante c d'échantillonnage.

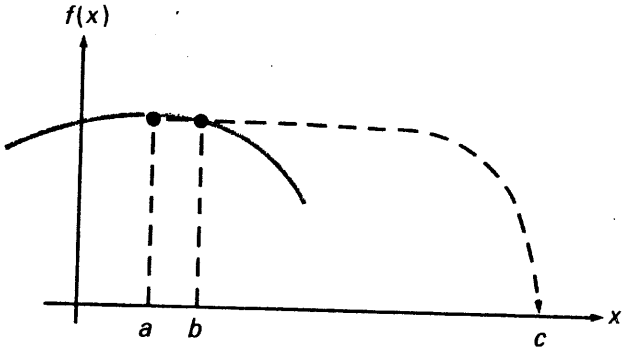
Normalement, SOLVE utilise la méthode de la sécante pour choisir la valeur suivante. Cette méthode utilise les valeurs de a, de b, de $f(a)$ et de $f(b)$ pour déterminer une valeur de c pour laquelle $f(c)$ est proche de zéro.



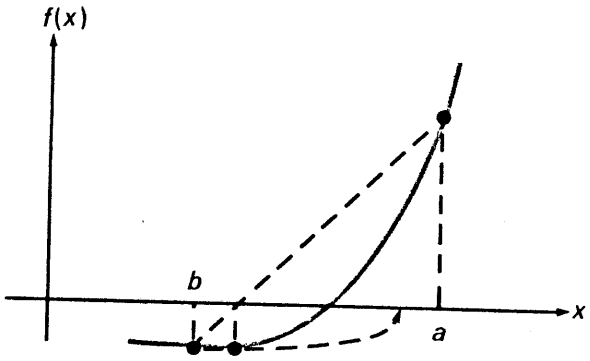
Si c n'est pas une racine mais si $f(c)$ est plus près de zéro que $f(b)$, alors b est changé en a, c est changé en b et le processus de détermination de c recommence. Lorsque la représentation graphique de $f(x)$ est régulière et si les valeurs initiales de a et de b sont proches d'une racine simple, la méthode de la sécante converge rapidement vers une racine.

Cependant dans certaines conditions, la méthode sécante ne suggère pas de valeur suivante capable d'arrêter la recherche ou de la faire aboutir à une valeur proche d'une racine : c'est le cas d'un changement de signe ou d'une amplitude plus petite. Dans ce cas, SOLVE utilise une approche différente.

Si la sécante calculée est presque horizontale, SOLVE modifie la méthode de la sécante pour s'assurer que $| c - b | \leq 100 | a - b |$. Ce procédé est très important car il réduit par ailleurs la tendance de la méthode de la sécante à s'égarer lorsque les erreurs d'arrondis deviennent significatives à proximité d'une racine.



Si SOLVE a déjà trouvé des valeurs de a et de b telles que $f(a)$ et $f(b)$ sont de signe opposé, elle modifie la méthode de la sécante pour garantir que c se trouve toujours dans l'intervalle contenant le changement de signe. Ceci garantit que l'intervalle de recherche diminue avec chaque itération, donnant toujours une racine lorsqu'elle existe.



Si SOLVE n'a pas rencontré de changement de signe et si une valeur c d'échantillonnage ne donne pas une valeur $f(c)$ d'amplitude réduite, alors SOLVE ajuste une parabole aux valeurs a, b, et c de la fonction. SOLVE recherche ensuite la valeur d à laquelle la parabole a son maximum ou son minimum, transforme d en a, et continue la recherche par la méthode de la sécante.

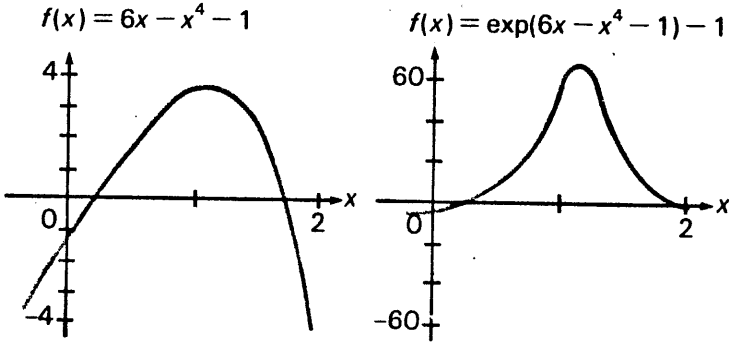
SOLVE n'abandonne la recherche d'une racine que si trois ajustements paraboliques successifs ne donnent aucune diminution de l'amplitude de la fonction ou si $d = b$. Dans ces deux cas, le calculateur affiche **Error 8**. Comme b représente le point de plus petite amplitude de la fonction échantillonnée, b et $f(b)$ sont renvoyées respectivement dans les registres X et Z. Le registre Y contient soit la valeur de a, soit la valeur de c. Avec cette information, vous pouvez décider de la suite des opérations. Ou vous recommencez la recherche là où vous l'aviez laissée, ou vous orientez différemment la recherche, ou vous décidez que $f(b)$ est si proche de 0 que $x = b$ est une racine, ou vous transformez l'équation en une autre équation, ou enfin vous concluez qu'il n'y a pas de racine.

Situations à problème

Les explications suivantes sont utiles lorsque vous travaillez sur des problèmes pouvant mener à des résultats trompeurs. Des racines imprécises sont obtenues lorsque les valeurs de la fonction calculée sont différentes des valeurs de la fonction désirée. Vous pouvez la plupart du temps éviter cette difficulté, si vous savez comment identifier l'imprécision et la réduire.

Équations faciles et équations difficiles

Les deux équations $f(x) = 0$ et $e^{f(x)} - 1 = 0$ ont les mêmes racines réelles, mais selon les cas, l'une sera toujours plus facile à résoudre numériquement que l'autre. Par exemple, lorsque $f(x) = 6x - x^4 - 1$, la première équation est la plus facile. Lorsque $f(x) = \ln(6x - x^4)$, la seconde est la plus facile. La différence dépend du comportement du graphe de la fonction, particulièrement à proximité d'une racine.



En général, toute équation est l'une d'une famille infinie d'équations équivalentes ayant les mêmes racines réelles. Et certaines de ces équations sont plus faciles à résoudre que d'autres. Alors que SOLVE peut échouer dans sa recherche des racines de l'une de ces équations, il peut très bien réussir avec une autre.

Équations imprécises

SOLVE ne calcule jamais une racine incorrecte, *sauf si la fonction est calculée incorrectement.* La précision du sous-programme de votre fonction affecte la précision de la racine que vous recherchez.

Vous devez connaître les causes éventuelles des différences entre valeur calculée de la fonction et valeur théorique de la fonction. SOLVE ne peut pas déduire de valeurs théoriques. La plupart du temps, vous devrez minimiser les erreurs de calcul en écrivant soigneusement le sous-programme de votre fonction.

Équations à plusieurs racines

Plus une équation a de racines, plus la recherche de toutes les racines d'une équation est difficile. Et lorsque ces racines ont des valeurs très proches les unes des autres, une résolution précise de l'équation est pratiquement impossible. Vous pouvez utiliser la *méthode de la déflation* pour éliminer des racines (décrite dans le *manuel d'utilisation du HP-15C*).

Une équation à plusieurs racines est caractérisée par la fonction et par ses premières dérivées d'ordre supérieur qui sont égales à zéro à la valeur des racines. Lorsque SOLVE trouve une racine double, la deuxième moitié de ses chiffres risque d'être imprécise. Dans le cas d'une racine triple, les deux tiers des chiffres de la racine tendent à perdre leur sens. Une racine quadruple tend à perdre environ les trois-quarts de ses chiffres.

Utilisation de SOLVE avec des polynômes

Les polynômes comptent parmi les fonctions les plus faciles à évaluer. C'est pourquoi ils sont traditionnellement utilisés pour approcher des fonctions de modélisation de processus physiques ou des fonctions mathématiques beaucoup plus complexes.

Un polynôme de degré n est de la forme :

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

Cette fonction est égale à zéro pour pas plus de n valeurs réelles de x, appelées les "zéros" du polynôme. Il est possible de déterminer une limite au nombre de zéros *positifs* de cette fonction en comptant le nombre de fois où changent

les signes des coefficients en lisant le polynôme de gauche à droite. De même, il est possible de déterminer une limite au nombre de zéros *négatifs* en considérant une nouvelle fonction obtenue par remplacement de x par $-x$ dans le polynôme initial. Si le nombre réel de zéros positifs ou négatifs est inférieur à sa limite, cette différence sera un nombre pair. (Ces relations sont appelées la règle des signes de Descartes).

A titre d'exemple, considérons la fonction polynomiale suivante de degré 3 :

$$f(x) = x^3 - 3x^2 - 6x + 8$$

Elle ne peut avoir plus de trois zéros réels. Elle a au plus deux zéros réels positifs (à cause des changements de signes entre le premier et le deuxième terme et entre le troisième et le quatrième terme) et elle a au plus un zéro réel négatif (car $f(-x) = -x^3 - 3x^2 + 6x + 8$).

Les fonctions polynomiales sont généralement évaluées de façon plus compact en utilisant des multiplications imbriquées. (On appelle ce procédé la méthode d'Horner). Ainsi, la fonction précédente peut être écrite sous la forme :

$$f(x) = [(x - 3)x - 6]x + 8$$

Cette représentation du polynôme est plus facile à programmer et plus rapide à exécuter que la forme de départ, puisque SOLVE, en particulier remplit la pile avec la valeur de x.

Exemple : Durant l'hiver 1978, l'explorateur de l'arctique, Jean-Claude Coulerre, isolé dans le grand Nord, s'amusa à scruter l'horizon au sud pour attendre la réapparition du soleil. Coulerre savait que le soleil ne lui apparaîtrait que début mars, lorsqu'il atteindrait une déclinaison de 5° 18' S. A quel jour et à quelle heure cet explorateur a-t-il vu le soleil réapparaître ?

La date à laquelle le soleil a atteint une déclinaison de 5° 18' S peut être calculée en résolvant pour j l'équation suivante :

$$D = a_4 j^4 + a_3 j^3 + a_2 j^2 + a_1 j + a_0$$

où D est la déclinaison exprimée en degrés, où j est le nombre de jours à partir du début du mois (jème jour) et

où

$$a_4 = 4.2725 \times 10^{-8}$$
$$a_3 = -1.9931 \times 10^{-5}$$
$$a_2 = 1.0229 \times 10^{-3}$$
$$a_1 = 3.7680 \times 10^{-1}$$
$$a_0 = -8.1806.$$

Cette équation est valide pour $1 \leq j < 32$, intervalle pour mars 1978.

Convertissez d'abord 5°18'S en degrés décimaux (**5.18** [CHS] [g] [→H]), pour obtenir — 5.3000 (en utilisant le mode d'affichage [FIX] 4). (Pour mémoire, les latitudes sud sont exprimées en nombres négatifs dans les calculs).

La solution du problème est la valeur de j satisfaisant l'égalité suivante :

$$-5.3000 = a_4 j^4 + a_3 j^3 + a_2 j^2 + a_1 j + a_0$$

Que l'on peut exprimer sous la forme :

$$0 = a_4 j^4 + a_3 j^3 + a_2 j^2 + a_1 j - 2.8806.$$

où le dernier terme (constante) tient compte de la valeur de la déclinaison.

En utilisant la méthode Horner, la fonction à résoudre est représentée par :

$$f(j) = (((a_4 j + a_3) j + a_2) j + a_1) j - 2.8806$$

Pour raccourcir le sous-programme, vous pouvez stocker et rappeler les constantes à l'aide des registres correspondant aux exposants de j.

Appuyez sur	**Affichage**	
[ON] / [−]	**Pr Error**	Efface la mémoire du calculateur*.
[←]	**0.0000**	
[g] [P/R]	**000−**	Mode programme.

* Cette étape n'est citée ici que pour s'assurer qu'il y a suffisamment de mémoire disponible pour les exemples donnés dans ce manuel.

Appuyez sur	Affichage
[f] [LBL] [A]	001-42,21,11
[RCL] 4	002- 45 4
[×]	003- 20
[RCL] 3	004- 45 3
[+]	005- 40
[×]	006- 20
[RCL] 2	007- 45 2
[+]	008- 40
[×]	009- 20
[RCL] 1	010- 45 1
[+]	011- 40
[×]	012- 20
[RCL] 0	013- 45 0
[+]	014- 40
[g] [RTN]	015- 43 32

En mode calcul, introduisez les cinq coefficients :

Appuyez sur	Affichage		
[g] [P/R]			Mode calcul
4.2725 [EEX] 8 [CHS]	4.2725	-08	
[STO] 4	4.2725	-08	Coefficient de j^4
1.9931 [CHS] [EEX]			
5 [CHS] [STO] 3	-1.9931	-05	Coefficient de j^3
1.0229 [EEX] 3 [CHS]	1.0229	-03	
[STO] 2	0.0010		Coefficient de j^2
3.7680 [EEX] 1 [CHS]	3.7680	-01	
[STO] 1	0.3768		Coefficient de j
2.8806 [CHS] [STO] 0	-2.8806		Constante

Puisque vous savez que la solution recherchée doit être comprise entre 1 et 32, introduisez ces deux valeurs comme estimations initiales. Ensuite, utilisez [SOLVE] pour rechercher les racines.

Appuyez sur	Affichage	
1 [ENTER]	1.0000	
32	32	Estimations initiales.
[f] [SOLVE] [A]	7.5137	Racine recherchée
[R↓]	7.5137	Même estimation précédente.

Appuyez sur	Affichage	
R↓	**0.0000**	Valeur de la fonction.
g R↑ g R↑	**7.5137**	Restaure la pile.

Le jour était donc le 7 mars. Convertissez maintenant la partie fractionnaire résultat en heures décimales puis en heures, minutes, secondes.

Appuyez sur	Affichage	
f FRAC	**0.5137**	Partie fractionnaire du jour.
24 ×	**12.3293**	Heures décimales.
f →H.MS	**12.1945**	Heures, minutes, secondes

L'explorateur Coulerre a donc vu le soleil le 7 mars à 12 h 19 mn 45 s (temps universel).

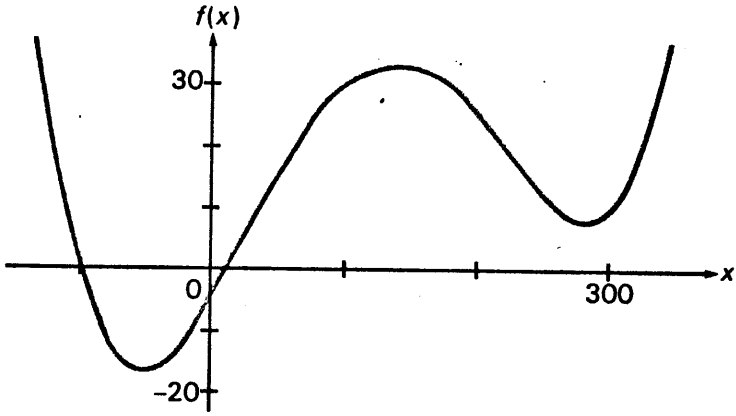
En examinant votre fonction $f(j)$, vous voyez qu'elle peut avoir jusqu'à quatre racines réelles – trois positives et une négative. Essayez de trouver d'autres racines positives en utilisant SOLVE avec des estimations positives supérieures.

Appuyez sur	Affichage	
1000 ENTER 1100	**1,100**	Deux estimations positives supérieures.
f SOLVE A	**Error 8**	Aucune racine.
←	**278.4497**	Dernière estimation.
R↓	**276.7942**	Estimation précédente.
R↓	**7.8948**	Valeur non-racine.
g R↑ g R↑	**278.4497**	Restauration de la pile.
f SOLVE A	**Error 8**	Aucune racine.
←	**278.4398**	Estimation peu différente.
R↓	**278.4497**	Estimation précédente.
R↓	**7.8948**	Même valeur de la fonction.

Vous avez trouvé un minimum local positif à la place d'une racine. Maintenant, essayez de trouver une racine négative.

Appuyez sur	Affichage	
1000 CHS ENTER	-1,000.0000	
1100 CHS	-1,100	Deux estimations négatives.
f SOLVE A	-108.9441	Racine négative.
R↓	-108.9441	Même estimation précédente.
R↓	1.6000 -08	Valeur de la fonction.

Il n'est pas nécessaire d'aller plus loin : vous avez trouvé toutes les racines possibles. La racine négative a un sens puisqu'elle est en dehors de la plage de valeurs pour lesquelles l'approximation de la déclinaison est valide. Le graphe de la fonction confirme ces résultats.



Résolution d'un système d'équations

SOLVE permet de trouver la valeur d'une seule variable satisfaisant à une seule équation. Dans le cas d'un système d'équations à plusieurs variables, vous pouvez cependant utiliser SOLVE pour rechercher une solution.

Dans le cas de certains systèmes d'équations, de la forme :

$$f_1(x_1, ..., x_n) = 0$$
$$\vdots$$
$$f_n(x_1, ..., x_n) = 0$$

il est possible d'éliminer toutes les variables sauf une par manipulation algébrique. Autrement dit, vous pouvez utiliser les équations pour dériver des

expressions pour toutes les variables sauf une en termes de variable restante. En utilisant ces expressions, vous pouvez ramener ce problème à la résolution d'une équation simple à l'aide de SOLVE. Les valeurs des autres variables à la solution peuvent être calculées à l'aide des expressions dérivées.

Ceci est souvent utile pour la résolution d'une équation complexe à racine complexe. Dans un tel problème, l'équation complexe peut être représentée sous la forme de deux équations réelles – l'une pour la partie réelle, l'autre pour la partie imaginaire – à deux variables réelles (partie réelle et partie imaginaire de la racine complexe).

Par exemple, l'équation complexe $z + 9 + 8e^{-z} = 0$ n'a pas de racines z réelles, mais a de nombreuses racines complexes de la forme $z = x + iy$. Cette équation peut être exprimée sous la forme de deux équations réelles :

$$\begin{aligned} x + 9 + 8e^{-x}\cos y &= 0 \\ y \quad - 8e^{-x}\sin y &= 0 \,. \end{aligned}$$

Les manipulations suivantes peuvent être utilisées pour éliminer y de ces équations. Comme le signe de y n'a aucune importance dans ces équations, supposons que $y > 0$ pour que toute solution (x, y) donne une autre solution $(x, -y)$. Ré-écrire la seconde équation sous la forme :

$$x = \ln(8(\sin y)/y),$$

qui nécessite $\sin y > 0$, pour que $2n\pi < y < (2n + 1)\pi$ avec n entier $= 0, 1, \ldots$

A partir de la première équation

$$\begin{aligned} y &= \cos^{-1}(-e^x(x + 9)/8) + 2n\pi \\ &= (2n + 1)\pi - \cos^{-1}(e^x(x + 9)/8) \end{aligned}$$

pour $n = 0, 1, \ldots$, substituez cette expression dans la deuxième équation :

$$x + \ln\left(\frac{(2n + 1)\pi - \cos^{-1}(e^x(x + 9)/8)}{\sqrt{64 - (e^x(x + 9))^2}}\right) = 0.$$

Vous pouvez ensuite utiliser SOLVE pour rechercher la racine x de cette équation (pour toute valeur donnée de n, le nombre de la racine). Connaissant x, vous pouvez calculer la valeur correspondante de y.

Une remarque finale sur cet exemple concerne le choix de l'estimation appropriée. Puisque l'argument du cosinus inverse doit être compris entre -1 et 1, x doit être inférieur à -0.1059 (trouvé par tentatives ou en utilisant SOLVE). Les estimations initiales pourraient être proches mais inférieures à cette valeur: -0.11 et -0.2 par exemple.

(L'équation complexe utilisée dans cet exemple est résolue à l'aide d'une procédure itérative donnée dans l'exemple de la page 81. Une autre méthode de résolution d'un système d'équations non linéaires est décrite page 122).

Recherche des extrêmes locaux d'une fonction

Utilisation de la dérivée

La méthode classique de calcul des maxima et minima locaux d'un graphe utilise la *dérivée* de la fonction. La dérivée est une fonction qui décrit la pente du graphe. Les valeurs de x pour lesquelles la dérivée est égale à zéro représentent des extrêmes locaux possibles pour la fonction. (Bien que moins connues pour les fonctions régulières, les valeurs de x où la dérivée est infinie ou indéfinie sont également des extrêmes possibles). Si vous parvenez à exprimer la dérivée d'une fonction, vous pouvez utiliser SOLVE pour calculer à quelle valeur cette dérivée est nulle pour savoir où la fonction est susceptible de présenter un maximum ou un minimum.

Exemple: Pour la conception d'une tour d'émission-radio, un ingénieur recherche l'angle par rapport à la verticale (tour), pour lequel l'intensité relative du champ est la plus négative. L'intensité relative créée par la tour est donnée par la formule suivante:

$$E = \frac{\cos(2\pi h \cos\theta) - \cos(2\pi h)}{[1 - \cos(2\pi h)]\sin\theta}$$

où E est l'intensité relative du champ, h la hauteur de l'antenne en longueurs d'onde et θ l'angle par rapport à la verticale en radians. La hauteur de l'antenne est de 0.6 longueurs d'onde dans cet exemple.

L'angle désiré est celui auquel la dérivée de l'intensité pour θ est égale à zéro.

Pour réduire l'espace mémoire de programme et le temps d'exécution, stockez les constantes suivantes dans des registres pour n'avoir qu'à les rappeler par la suite :

$r_0 = 2\pi h$	constante stockée dans R_0
$r_1 = \cos(2\pi h)$	constante stockée dans R_1
$r_2 = 1/[1 - \cos(2\pi h)]$	constante stockée dans R_2

La dérivée de l'intensité E calculée pour l'angle θ est donnée par :

$$\frac{dE}{d\theta} = r_2\left[r_0\sin(r_0\cos\theta) - \frac{\cos(r_0\cos\theta) - r_1}{\sin\theta\tan\theta}\right].$$

Enregistrez le sous-programme de calcul de la dérivée.

Appuyez sur	**Affichage**	
g P/R		Mode programme
f CLEAR PRGM	000-	
f LBL 0	001-42,21, 0	
COS	002- 24	
RCL 0	003- 45 0	
×	004- 20	
COS	005- 24	
RCL 1	006- 45 1	
−	007- 30	
x≷y	008- 34	
SIN	009- 23	
÷	010- 10	
x≷y	011- 34	
TAN	012- 25	
÷	013- 10	
CHS	014- 16	
x≷y	015- 34	
COS	016- 24	
RCL 0	017- 45 0	

Appuyez sur	Affichage
[×]	018– 20
[SIN]	019– 23
[RCL] 0	020– 45 0
[×]	021– 20
[+]	022– 40
[RCL] 2	023– 45 2
[×]	024– 20
[g] [RTN]	025– 43 32

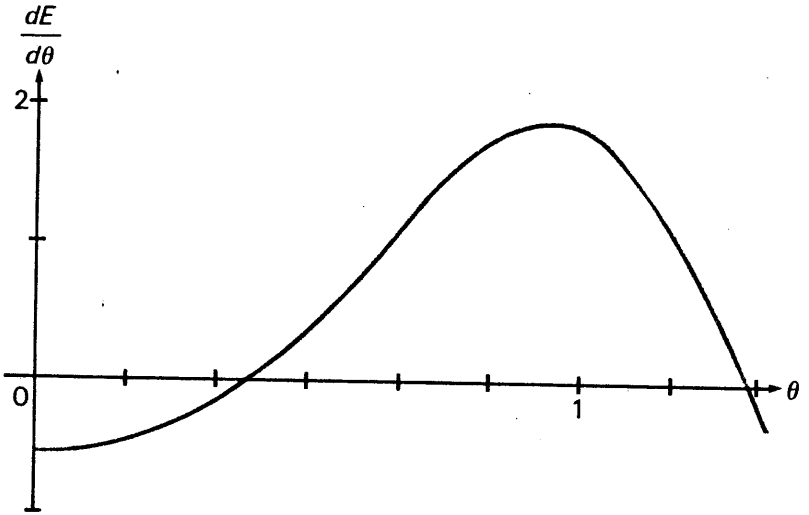
En mode Radians, calculez et stockez les trois constantes.

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
[g] [RAD]		Mode radians.
2 [g] [π] [×]	6.2832	
.6 [×] [STO] 0	3.7699	Constante de r_0.
[COS] [STO] 1	−0.8090	Constante de r_1.
[CHS] 1 [+]	1.8090	
[1/x] [STO] 2	0.5528	Constante de r_2.

L'intensité relative du champ est maximale à 90° (la perpendiculaire à la tour). Pour trouver le minimum, utilisez des angles plus proches de zéro comme estimations initiales, par exemple les équivalents en radians de 10° et 60°.

Appuyez sur	Affichage		
10 [f] [→RAD]	0.1745		
60 [f] [→RAD]	1.0472		Estimations initiales.
[f] [SOLVE] 0	0.4899		Angle donnant la pente zéro.
[R↓] [R↓]	−5.5279	−10	Pente à l'angle spécifié.
[g] [R↑] [g] [R↑]	0.4899		Restaure la pile.
[g] [→DEG]	28.0680		Angle en degrés.

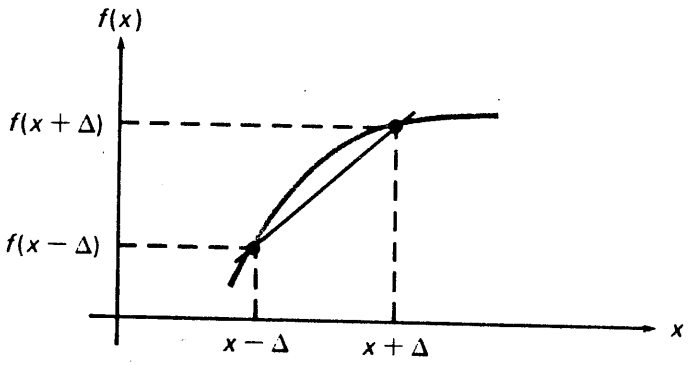
L'intensité relative du champ est la plus négative à un angle de 28.0680° par rapport à la verticale.



Utilisation d'une pente approchée

La dérivée d'une fonction peut être également calculée numériquement de façon approchée. Si vous échantillonnez une fonction sur deux points relativement proches de x (respectivement $x + \Delta$ et $x - \Delta$), vous pouvez utiliser la pente de la sécante comme approximation de la pente en x:

$$s = \frac{f(x + \Delta) - f(x - \Delta)}{2\Delta}$$



La précision de cette approximation dépend de l'écart Δ et de la nature de la fonction. De petites valeurs de Δ donnent de meilleures approximations de la dérivée, mais de trop petites valeurs risquent de provoquer une imprécision avec les arrondis. Une valeur de x pour laquelle la pente est égale à zéro est un extrême local possible de la fonction.

Exemple : Résoudre le problème précédent sans utiliser l'équation $dE/d\theta$ de la dérivée.

Calculez l'angle auquel la dérivée (calculée numériquement) de l'intensité E est égale à zéro.

En mode programme, introduisez deux sous-programmes : l'un pour estimer la dérivée de l'intensité, l'autre pour évaluer la fonction E de l'intensité. Dans le sous-programme suivant, la pente est calculée entre $\theta + 0.001$ et $\theta - 0.001$ radians (plage correspondant à environ 0.1°).

Appuyez sur	Affichage	
[g] [P/R]	000-	Mode programme.
[f] [LBL] [A]	001-42,21,11	
[EEX]	002- 26	
[CHS]	003- 16	
3	004- 3	Calcule E à $\theta + 0.001$.
[+]	005- 40	
[ENTER]	006- 36	
[GSB] [B]	007- 32 12	
[x⇄y]	008- 34	
[EEX]	009- 26	
[CHS]	010- 16	
3	011- 3	Calcule E à $\theta - 0.001$.
[−]	012- 30	
[ENTER]	013- 36	
[GSB] [B]	014- 32 12	
[−]	015- 30	
2	016- 2	
[EEX]	017- 26	
[CHS]	018- 16	
3	019- 3	

Appuyez sur	Affichage	
[÷]	020– 10	
[g] [RTN]	021– 43 32	
[f] [LBL] [B]	022–42,21,12	Sous-programme pour $E(\theta)$.
[COS]	023– 24	
[RCL] 0	024– 45 0	
[×]	025– 20	
[COS]	026– 24	
[RCL] 1	027– 45 1	
[−]	028– 30	
[x≷y]	029– 34	
[SIN]	030– 23	
[÷]	031– 10	
[RCL] 2	032– 45 2	
[×]	033– 20	
[g] [RTN]	034– 43 32	

Dans l'exemple précédent, le calculateur avait été mis en mode radians et les trois constantes stockées dans les registres R_0, R_1 et R_2. Introduisez les mêmes estimations initiales que précédemment et exécutez [SOLVE].

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
10 [f] [→RAD]	0.1745	
60 [f] [→RAD]	1.0472	Estimations initiales.
[f] [SOLVE] [A]	0.4899	Angle donné à la pente zéro.
[R↓] [R↓]	0.0000	Pente à l'angle spécifié.
[g] [R↑] [g] [R↑]	0.4899	Restaure la pile.
[ENTER] [ENTER] [f] [B]	−0.2043	Utilise le sous-programme de la fonction pour calculer l'intensité minimale.
[x≷y]	0.4899	Rappelle la valeur de θ.
[g] [→DEG]	28.0679	Angle en degrés.

Cette approximation numérique de la dérivée donne une intensité de champ minimale de − 0.2043 à un angle de 28.0679°. (Ce résultat diffère du précédent de 0.0001°).

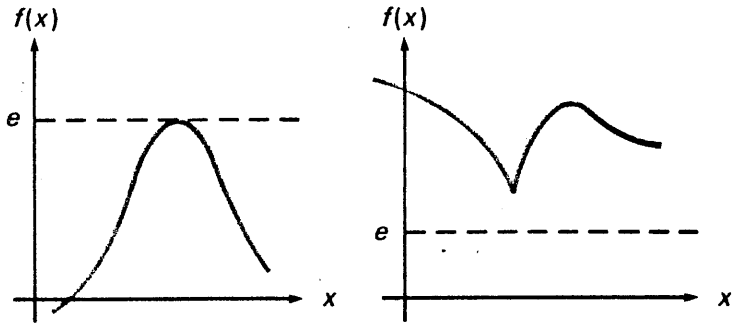
Utilisation d'une estimation répétée

Cette troisième technique est utile lorsqu'il n'est pas facile de calculer la dérivée. C'est une méthode plus lente car elle nécessite l'utilisation répétitive de [SOLVE]. Par contre, vous n'avez pas besoin de chercher une bonne valeur de Δ comme dans la méthode précédente. Pour rechercher un extrême local de la fonction $f(x)$, définissez une nouvelle fonction.

$$g(x) = f(x) - e$$

où e est un nombre légèrement supérieur à la valeur extrême estimée de la fonction $f(x)$. Si e est correctement choisi, $g(x)$ sera proche de 0 à proximité de l'extrême de $f(x)$, mais ne sera pas égale à zéro. Utilisez [SOLVE] pour analyser $g(x)$ près de l'extrême. Le résultat désiré est **Error 8.**

- Si **Error 8** est affiché, le nombre dans le registre X est une valeur de x proche de l'extrême. Le nombre contenu dans le registre Z indique grossièrement la différence entre e et la valeur extrême de $f(x)$. Modifiez e pour le rendre plus proche de la valeur extrême (mais pas égal à celle-ci). Puis utilisez [SOLVE] pour examiner la nouvelle valeur de $g(x)$ près de la valeur de x précédemment trouvée. Répétez cette procédure jusqu'à ce que les valeurs successives de x ne présentent plus d'écart significatif.
- Si une racine est trouvée pour $g(x)$, cela signifie soit que le nombre e n'est pas supérieur à la valeur extrême de $f(x)$, soit que [SOLVE] a trouvé une autre région du graphe où $f(x)$ est égale à e. Modifiez e pour qu'il soit proche – mais pas situé au-delà – de la valeur extrême de $f(x)$ et ré-exécutez [SOLVE]. Il peut être également possible de modifier $g(x)$ afin d'éliminer la racine éloignée.



Exemple : Reprenez l'exemple précédent sans calculer la dérivée de l'intensité relative du champ *E*.

Le sous-programme de calcul de *E* et les constantes nécessaires ont été introduits lors de l'exemple précédent.

En mode programme, enregistrez un sous-programme qui soustrait un nombre extrême estimé de l'intensité *E*. Le nombre extrême doit être stocké dans un registre afin de pouvoir le modifier manuellement en cas de besoin.

Appuyez sur	Affichage	
g P/R	000-	Mode programme.
f LBL 1	001-42,21, 1	Label du sous-programme.
GSB B	002- 32 12	Calcul de *E*.
RCL 9	003- 45 9	
−	004- 30	Soustraction de l'estimation de l'extrême.
g RTN	005- 43 32	

En mode calcul, estimez la valeur d'intensité minimale en échantillonnant manuellement la fonction.

Appuyez sur	Affichage	
g P/R		Mode calcul.
10 f →RAD	0.1745	Échantillonne la fonction à 10°, 30°, 50°, ...
ENTER f B	−0.1029	
30 f →RAD	0.5236	
ENTER f B	−0.2028	
50 f →RAD	0.8727	
ENTER f B	0.0405	

A partir de ces échantillons, faites un essai en utilisant une estimation de — 0.25 pour l'extrême et des estimations initiales pour [SOLVE] (en radians) proches de 10° et de 30°.

Appuyez sur	Affichage	
.25 [CHS] [STO] 9	**−0.2500**	Stocke l'estimation de l'extrême.
.2 [ENTER]	**0.2000**	
.6	**0.6**	Estimations initiales.
[f] [SOLVE] 1	**Error 8**	Aucune racine trouvée.
[←] [STO] 4	**0.4849**	Stocke l'estimation de θ.
[R↓] [STO] 5	**0.4698**	Stocke l'estimation précédente de θ.
[R↓]	**0.0457**	Distance de l'extrême.
.9 [×]	**0.0411**	Modifie l'estimation de
[STO] [+] 9	**0.0411**	l'extrême (de 90 % de la distance).
[RCL] 4	**0.4849**	Rappelle l'estimation de θ.
[ENTER] [ENTER] [f] [B]	**−0.2043**	Calcule l'intensité E.
[←]	**0.0000**	Rappelle d'autres estimations
[RCL] 5	**0.4698**	de θ en gardant la première dans le registre Y.
[f] [SOLVE] 1	**Error 8**	Aucune racine trouvée.
[←]	**0.4898**	Estimation de θ.
[x≷y]	**0.4893**	Estimation précédente de θ.
[x≷y]	**0.4898**	Rappelle l'estimation de θ.
[ENTER] [ENTER] [f] [B]	**−0.2043**	Calcule l'intensité E.
[x≷y]	**0.4898**	Rappelle la valeur de θ.
[g] [→DEG]	**28.0660**	Angle en degrés.
[g] [DEG]	**28.0660**	Restaure le mode degrés.

La seconde itération produit deux estimations de θ qui ne diffèrent qu'à la quatrième position décimale. Les intensités E pour les deux itérations, sont égales jusqu'à la quatrième position décimale. En s'arrêtant à ce niveau, on obtient une intensité de champs minimale de — 0.2043 à un angle de 28.0660°. (Avec un écart de 0.002° par rapport aux résultats des méthodes précédentes).

Applications

Les applications suivantes illustrent comment vous pouvez utiliser [SOLVE] pour simplifier un calcul habituellement difficile : la recherche d'un taux d'intérêt qui ne peut être calculé directement. D'autres applications utilisant la fonction [SOLVE] sont décrites aux chapitres 3 et 4.

Annuités et capitalisation

Ce programme permet de résoudre de nombreux problèmes financiers dans lesquels interviennent les facteurs d'argent, de temps et d'intérêt. Pour ces problèmes, vous connaissez généralement la valeur de trois ou quatre des variables suivantes et vous avez besoin de la valeur d'une autre :

n — *Nombre* de périodes de composition. (Par exemple, pour un prêt sur 30 ans avec remboursements mensuels, $n = 12 \times 300 = 360$.)

i — *Taux d'intérêt* par période, exprimé en pourcentage. (Pour calculer *i*, divisez le taux annuel par le nombre de périodes dans l'année. Autrement dit, un taux annuel d'intérêts composés de 12 % correspond à un taux périodique de 1 %).

PV — *Valeur actuelle (PRESENT VALUE)* d'une série de versements futurs ou d'un versement initial.

PMT — Montant du *remboursement (PAYMENT)* périodique.

FV — *Valeur future (FUTURE VALUE)*. C'est-à-dire le capital acquis (ou remboursé) à la fin de l'opération ou la valeur composée d'une série de versements antérieurs.

Types de problèmes d'annuités et de capitalisation

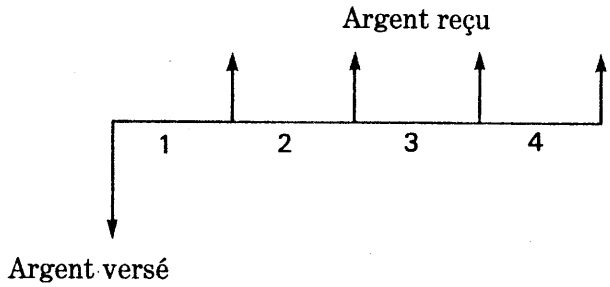
Combinaisons autorisées	Applications classiques		Procédure initiale
	Pour remboursements en fin de période	Pour remboursements en début de période	
n, i, PV, PMT (Introduire trois de ces valeurs et calculer la quatrième.)	Prêt direct. Effets escomptés. Hypothèques.	Crédit bail. Annuité à échoir.	Utiliser [f] [CLEAR] [REG] ou *FV* = 0
n, i, PV, PMT, FV (Introduire quatre de ces valeurs et calculer la cinquième.)	Prêt direct avec remboursement libératoire. Effets escomptés.	Crédit bail avec valeur résiduelle. Annuité à à échoir.	Aucune.
n, i, PMT, FV (Introduire trois de ces valeurs et calculer la quatrième.)	Fonds d'amortissement.	Épargne périodique. Assurance.	Utiliser [f] [CLEAR] [REG] ou *PV* = 0.
n, i, PV, FV (Introduire trois de ces valeurs et calculer la quatrième.)	Capitalisation. Épargne.		Utiliser [f] [CLEAR] [REG] ou *PMT* = 0.

Le programme accepte les remboursements effectués soit en fin (terme échu), soit en début (terme à échoir) de période de composition. Les remboursements effectués en fin de période (annuité ordinaire) sont courants pour les prêts directs et hypothécaires, alors que les remboursements en début de période (annuité d'avance) sont courants en crédit-bail.

Pour les remboursements effectués en fin de période, effacez l'indicateur binaire 0 (flag 0). Pour les remboursements effectués en début de période, armez l'indicateur binaire 0. Si le problème ne comporte pas de remboursements, l'état de cet indicateur est sans effet.

Ce programme utilise la convention suivante : les sommes d'argent versées sont introduites et affichées comme des quantités négatives et les sommes d'argent reçues comme des quantités positives.

Tout problème financier peut être ainsi représenté sous forme d'un diagramme de flux (positifs ou négatifs) dans le temps. Ce diagramme est constitué d'une ligne horizontale représentant le temps et divisée en intervalles égaux correspondant aux périodes de composition (années ou mois). Les flèches verticales représentent les mouvements d'argent en obéissant à la convention suivante : les flèches dirigées vers le haut (positives) représentent l'argent reçu, les flèches dirigées vers le bas (négatives) représentent l'argent versé. Exemple :



La pression de [f] CLEAR [REG] est une méthode commode de ré-initialiser le calculateur pour un nouveau problème. Cependant, il n'est pas nécessaire d'appuyer sur [f] CLEAR [REG] entre tous les problèmes. Vous ne ré-introduirez que les valeurs des variables différentes d'un problème à l'autre. Si une variable n'est pas utilisée dans un nouveau problème, donnez lui simplement la valeur 0. Par exemple, si *PMT* est utilisée dans un problème mais pas dans le suivant, introduisez simplement 0 comme valeur de *PMT* dans le second problème.

L'équation de base utilisée pour les calculs financiers est :

$$PV + \frac{PMT\,A}{i/100}[1 - (1 + i/100)^{-n}] + FV(1 + i/100)^{-n} = 0$$

où $i \neq 0$ et

$$A = \begin{cases} 1 & \text{pour les remboursements en fin de période.} \\ 1 + i/100 & \text{pour les remboursements en début de période.} \end{cases}$$

Le programme présente les caractéristiques suivantes :

- [SOLVE] est utilisée pour trouver *i*. Comme il s'agit d'une fonction itérative, le calcul de *i* est plus long que le calcul des autres variables. Certains problèmes peuvent être insolubles par cette technique. Si [SOLVE] ne trouve pas de racine, **Error 4** est affiché.
- Lors du calcul de l'une des cinq variables ci-dessous, certaines conditions provoquent l'affichage de **Error 4** :

n	$PMT = -PVi/(100\,A)$
	$(PMT\,A - FVi/100)/(PMT\,A + PVi/100) \leq 0$
	$i \leq -100$
i	[SOLVE] ne peut trouver de racine
PV	$i \leq -100$
PMT	$n = 0$
	$i = 0$
	$i \leq -100$
FV	$i \leq -100$

- Si un problème a un taux d'intérêt défini égal à 0, le programme génère un message d'erreur : **Error 0** (ou **Error 4** pour le calcul de *PMT*).
- Les problèmes ayant des valeurs de *n* ou de *i* extrêmement grandes (supérieures à 10^6) ou extrêmement petites (inférieures à 10^{-6}), risquent de donner des résultats incorrects.
- Les problèmes d'intérêts avec remboursements libératoires de signe opposé aux remboursements périodiques peuvent avoir mathématiquement plus d'une solution (ou pas de solution du tout). Ce programme peut très bien trouver l'une des solutions mais il ne donne pas les moyens de rechercher ou même d'indiquer d'autres possibilités.

Appuyez sur	**Affichage**	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	**000–**	

Appuyez sur	Affichage	
[f] [LBL] [A]	001–42,21,11	Programme pour *n*.
[STO] 1	002– 44 1	Stocke *n*.
[R/S]	003– 31	
[GSB] 1	004– 32 1	Calcule *n*.
[g] [LST*x*]	005– 43 36	
[RCL] [×] 0	006–45,20, 0	
[RCL] 5	007– 45 5	
[*x*≷*y*]	008– 34	
[−]	009– 30	Calcule $FV - 100\ PMT\ A/i$.
[g] [LST*x*]	010– 43 36	
[RCL] [+] 3	011–45,40, 3	Calcule $PV + 100\ PMT\ A/i$.
[g] [*x* = 0]	012– 43 20	Teste $PMT = -PVi/(100\ A)$.
[GTO] 0	013– 22 0	
[÷]	014– 10	
[CHS]	015– 16	
[g] [TEST] 4	016–43,30, 4	Teste $x \leq 0$.
[GTO] 0	017– 22 0	
[g] [LN]	018– 43 12	
[RCL] 6	019– 45 6	
[g] [LN]	020– 43 12	
[÷]	021– 10	
[STO] 1	022– 44 1	
[g] [RTN]	023– 43 32	
[f] [LBL] [B]	024–42,21,12	Programme pour *i*.
[STO] 2	025– 44 2	Stocke *i*.
[R/S]	026– 31	
[.]	027– 48	
2	028– 2	
[ENTER]	029– 36	
[EEX]	030– 26	

Appuyez sur	Affichage	
[CHS]	031– 16	
3	032– 3	
[g] [CF] 1	033–43, 5, 1	Arme l'indicateur 1 pour le sous-programme [SOLVE].
[f] [SOLVE] 3	034–42,10, 3	
[GTO] 4	035– 22 4	
[GTO] 0	036– 22 0	
[f] [LBL] 4	037–42,21, 4	
[EEX]	038– 26	
2	039– 2	
[×]	040– 20	Calcule *i*.
[STO] 2	041– 44 2	
[g] [RTN]	042– 43 32	
[f] [LBL] [C]	043–42,21,13	Programme pour *PV*.
[STO] 3	044– 44 3	Stocke *PV*.
[R/S]	045– 31	
[GSB] 1	046– 32 1	Calcule *PV*.
[GSB] 2	047– 32 2	
[CHS]	048– 16	
[STO] 3	049– 44 3	
[g] [RTN]	050– 43 32	
[f] [LBL] [D]	051–42,21,14	Programme pour *PMT*.
[STO] 4	052– 44 4	Stocke *PMT*.
[R/S]	053– 31	
1	054– 1	Calcule *PMT*.
[STO] 4	055– 44 4	
[GSB] 1	056– 32 1	
[RCL] 3	057– 45 3	
[GSB] 2	058– 32 2	
[x≷y]	059– 34	
[÷]	060– 10	
[CHS]	061– 16	
[STO] 4	062– 44 4	
[g] [RTN]	063– 43 32	

Appuyez sur	Affichage	
[f] [LBL] [E]	064–42,21,15	Programme pour *FV*.
[STO] 5	065– 44 5	Stocke *FV*.
[R/S]	066– 31	
[GSB] 1	067– 32 1	Calcule *FV*.
[RCL] [+] 3	068–45,40, 3	
[RCL] [÷] 7	069–45,10, 7	
[CHS]	070– 16	
[STO] 5	071– 44 5	
[g] [RTN]	072– 43 32	
[f] [LBL] 1	073–42,21, 1	
[g] [SF] 1	074–43, 4, 1	Arme l'indicateur 1 pour le sous-programme 3.
1	075– 1	
[RCL] 2	076– 45 2	
[g] [%]	077– 43 14	Calcule $i/100$.
[f] [LBL] 3	078–42,21, 3	Sous-programme [SOLVE].
[STO] 8	079– 44 8	
1	080– 1	
[STO] 0	081– 44 0	
[+]	082– 40	
[g] [TEST] 4	083–43,30, 4	Teste $i \leq 100$.
[GTO] 0	084– 22 0	
[STO] 6	085– 44 6	
[g] [F?] 0	086–43, 6, 0	Teste si les remboursements sont en fin de période.
[STO] 0	087– 44 0	
[RCL] 1	088– 45 1	
[CHS]	089– 16	
[y^x]	090– 14	Calcule $(1 + i/100)^{-n}$.
[STO] 7	091– 44 7	
1	092– 1	
[$x \gtrless y$]	093– 34	
[−]	094– 30	Calcule $1 - (1 + i/100)^{-n}$.

Appuyez sur	Affichage	
g $x=0$	095- 43 20	Teste $i = 0$ ou $n = 0$.
GTO 0	096- 22 0	
RCL × 0	097-45,20, 0	
RCL 4	098- 45 4	
RCL ÷ 8	099-45,10, 8	
×	100- 20	
g F? 1	101-43, 6, 1	Teste la position de l'indicateur 1.
g RTN	102- 43 32	
RCL + 3	103-45,40, 3	Le sous-programme SOLVE continue.
f LBL 2	104-42 21 2	
RCL 5	105- 45 5	
RCL × 7	106-45,20, 7	Calcule $FV(1 + i/100)^{-n}$.
+	107- 40	
g RTN	108- 43 32	Le sous-programme SOLVE est terminé.

Labels utilisés : A, B, C, D, E, 0, 1, 2, 3 et 4.

Registres utilisés / R_0 (A), R_1 (n), R_2 (i), R_3 (PV), R_4 (PMT), R_5 (FV), R_6, R_7 et R_8.

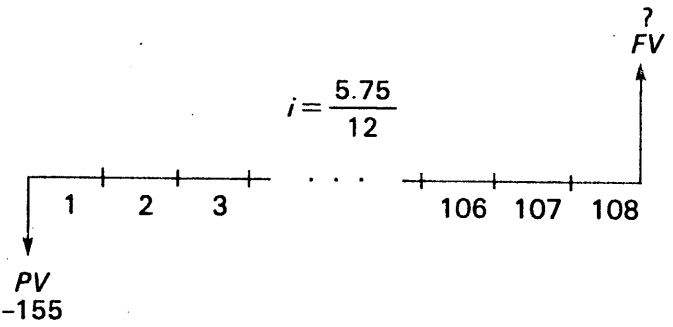
Pour utiliser le programme :

1. Appuyez sur 8 f DIM (i) pour réserver les registres R_0 à R_8.
2. Appuyez sur f USER pour valider le mode USER.
3. Si nécessaire, appuyez sur f CLEAR REG pour effacer toutes les variables. Vous n'avez pas besoin d'effacer les registres si vous avez l'intention de spécifier toutes les valeurs.
4. Armez l'indicateur 0 en fonction du type de remboursement :
   - Appuyez sur g CF 0 pour les remboursements en fin de période.
   - Appuyez sur g SF 0 pour les remboursements en début de période.
5. Introduisez les valeurs connues des variables :
   - Pour n, introduisez sa valeur et appuyez sur A.
   - Pour i, introduisez sa valeur et appuyez sur B.

- Pour *PV*, introduisez sa valeur et appuyez sur [C].
- Pour *PMT*, introduisez sa valeur et appuyez sur [D].
- Pour *FV*, introduisez sa valeur et appuyez sur [E].

6. Calculez l'inconnue :
   - Pour calculer *n*, appuyez sur [A] [R/S].
   - Pour calculer *i*, appuyez sur [B] [R/S].
   - Pour calculer *PV*, appuyez sur [C] [R/S].
   - Pour calculer *PMT*, appuyez sur [D] [R/S].
   - Pour calculer *FV*, appuyez sur [E] [R/S].
7. Pour résoudre un autre problème, répétez les étapes 3 à 6 de la procédure. Vérifiez qu'aucune variable nécessaire au calcul n'a la valeur zéro.

Exemple 1 : Vous placez 155 FF sur un compte rémunéré par composition mensuelle à 5,75 % d'intérêt annuel. Quel capital aurez-vous dans 9 ans ?



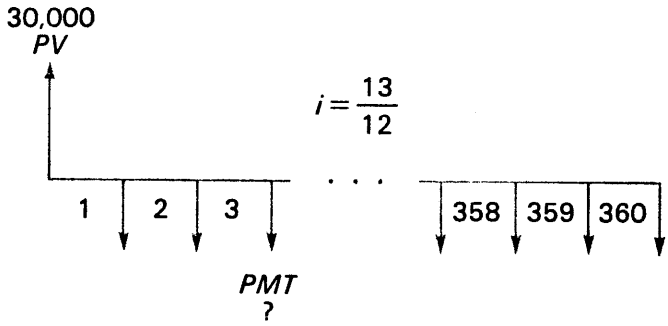
Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
[f] CLEAR [REG]		Efface les variables financières.
[f] [FIX] 2		
[f] [USER]		Valide le mode USER.
[g] [CF] 0		Annuité ordinaire.
9 [ENTER] 12 [×] [A]	**108.00**	Introduit $n = 9 \times 12$.

Appuyez sur	Affichage	
5.75 [ENTER] 12 [÷] [B]	**0.48**	Introduit $i = 5.75/12$.
155 [CHS] [C]	**−155.00**	Introduit $PV = -155$ (argent versé).
[E] [R/S]	**259.74**	Calcule *FV*.

Si vous aviez désiré un capital de 275 FF, à quel taux auriez-vous dû placer votre argent ?

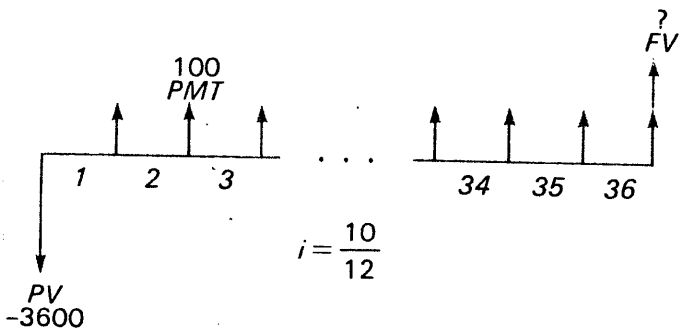
Appuyez sur	Affichage	
275 [E]	**275.00**	Introduit $FV = 275$.
[B] [R/S]	**0.53**	Calcule *i*.
12 [×]	**6.39**	Calcule le taux d'intérêt annuel.

Exemple 2 : Vous prenez une hypothèque de 30.000 FF sur 30 ans à 13 % d'intérêt. Quel sera votre remboursement mensuel ?



Appuyez sur	Affichage	
[f] CLEAR [REG]		Efface les variables.
30 [ENTER] 12 [×] [A]	**360.00**	Introduit $n = 30 \times 12$.
13 [ENTER] 12 [÷] [B]	**1.08**	Introduit $i = 13/12$.
30000 [C]	**30,000.00**	Introduit $PV = 30{,}000$.
[D] [R/S]	**−331.86**	Calcule *PMT* (argent versé).

Exemple 3 : Vous proposez de prêter 3,600 FF remboursables en 36 mensualités de 100 FF pour un taux d'intérêt annuel de 10 %. Quel sera le montant du paiement libératoire accompagnant la 36e mensualité, pour solder votre créance ?



Appuyez sur	Affichage	
f CLEAR REG		Efface les variables.
36 A	**36.00**	Introduit $n = 36$.
10 ENTER 12 ÷ B	**0.83**	Introduit $i = 10/12$.
3600 CHS C	**-3600.00**	Introduit $PV = -3600$ (argent versé).
100 D	**100.00**	Introduit $PMT = 100$ (argent reçu).
E R/S	**675.27**	Calcule FV.

Le remboursement final sera 675.27 + 100 = 775.27 FF (paiement libératoire + 36e mensualité).

Exemple 4 : Pour un emprunt de 50 000 FF remboursable en 360 mensualités au taux annuel de 14 %, trouvez le capital restant dû après le 24e versement et les intérêts payés entre les 12e et 24e versements.

Vous pouvez utiliser le programme pour calculer les intérêts payés sur un groupe d'annuités et le capital restant dû après la dernière annuité du groupe. Le montant des intérêts payés entre deux périodes est égal au montant des remboursements effectués pendant cet intervalle moins le capital amorti sur cet intervalle. Le capital amorti est égal à la différence entre le capital restant dû au début de la première période de référence et le capital restant dû à la fin de la deuxième période de référence.

Tout d'abord, calculez le montant des mensualités :

Appuyez sur	Affichage	
[f] CLEAR [REG]		Efface les variables.
360 [A]	**360.00**	Introduit $n = 360$.
14 [ENTER] 12 [÷] [B]	**1.17**	Introduit $i = 14/12$.
50000 [CHS] [C]	**−50,000.00**	Introduit $PV = -50{,}000$.
[D] [R/S]	**592.44**	Calcule *PMT*.

Maintenant, calculez le capital restant dû à la période 24 :

Appuyez sur	Affichage	
24 [A]	**24.00**	Introduit $n = 24$.
[E] [R/S]	**49,749.56**	Calcule *FV* à la période 24.

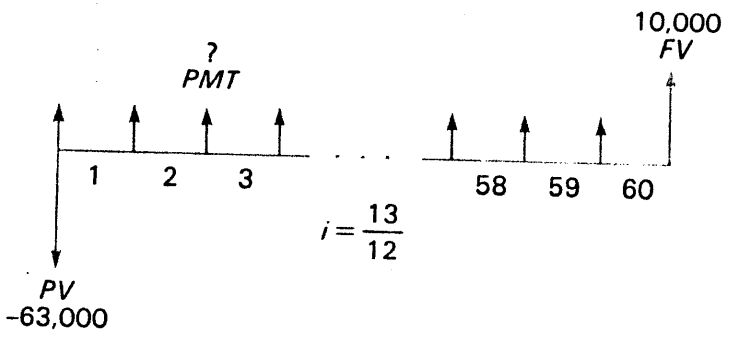
Stockez ce résultat, puis calculez le capital restant dû à la période 12 et le capital amorti entre les périodes 12 et 24 :

Appuyez sur	Affichage	
[STO] [I]	**49,749,56**	
12 [A]	**12.00**	Introduit $n = 12$.
[E] [R/S]	**49,883.48**	Calcule *FV* à la période 12.
[RCL] [I]	**49,749.56**	Rappelle *FV* à la période 24.
[−]	**133.92**	Capital amorti.

Le montant des intérêts payés est égal à la différence entre le montant de 12 mensualités et le capital amorti sur ces 12 mensualités :

Appuyez sur	Affichage	
[RCL] 4	**592.44**	Rappelle *PMT*.
12 [×]	**7,109.23**	Valeur des 12 mensualités.
[x≷y] [−]	**6,975.31**	Montant des intérêts payés.

Exemple 5 : Une société de crédit-bail envisage l'achat d'un mini-ordinateur d'une valeur de 63,000 FF et désire en tirer un profit annuel de 13 % en le louant à un client pour une période de 5 ans. Au bout de 5 ans, cette société espère revendre l'équipement au moins 10,000 FF. Quel devra-être le versement mensuel du client pour que la société de crédit-bail réalise le profit de 13 % ? (Les versements de crédit-bail étant en début de période, n'oubliez pas d'armer l'indicateur 0 comme il se doit.)



Appuyez sur	Affichage	
f CLEAR REG		Efface les variables.
g SF 0		Spécifie des versements en début de période.
5 ENTER 12 × A	60.00	Introduit $n = 5 \times 12$.
13 ENTER 12 ÷ B	1.08	Introduit $i = 13/12$.
63000 CHS C	-63,000.00	Introduit $PV = -63{,}000$.
10000 E	10,000.00	Introduit $FV = 10{,}000$.
D R/S	1,300.16	Calcule *PMT*.

Si l'ordinateur coûte 70,000 FF au montant de l'achat, quels seront les versements ?

Appuyez sur	Affichage	
70000 CHS C	-70,000.00	Introduit $PV = -70{,}000$.
D R/S	1,457.73	Calcule *PMT*.

Si les versements étaient portés à 1,500 FF, quel sera le profit réalisé ?

Appuyez sur	Affichage	
1500 [D]	**1,500.00**	Introduit $PMT = 1500$.
[B] [R/S]	**1.18**	Calcule i (mensuel).
12 [×]	**14.12**	Calcule le taux (profit) annuel.
[f] [USER]	**14.12**	Invalide le mode USER.

Analyse de flux de trésorerie escomptés

Ce programme effectue deux sortes d'analyses : la valeur actuelle nette *NPV* (Net Present Value) et le taux de rentabilité interne *IRR* (Internal Rate of Return). Il calcule soit *NPV*, soit *IRR* pour un maximum de 24 groupes de flux de trésorerie.

Les versements sont stockés dans la matrice C à deux colonnes. Chaque rang de la matrice C représente chaque groupe de versements : le premier élément est le montant du versement, le deuxième élément est le nombre de versements de ce montant (nombre de flux dans ce groupe). Le premier élément de C doit être le montant de l'investissement initial. Les versements doivent être faits à intervalles égaux ; s'il n'y a pas de versement sur plusieurs périodes, chacun de ces versements aura la valeur zéro et le nombre de 0 représentera le nombre de flux dans ce groupe.

Dès que tous ces flux sont stockés dans la matrice C, vous pouvez introduire un taux d'intérêt donné et calculer la valeur actuelle nette (*NPV*) de l'investissement. De même, vous pouvez calculer le taux de rentabilité interne (*IRR*). L'*IRR* est le taux d'intérêt pour lequel la valeur actuelle d'une série de flux de trésorerie est égale à l'investissement initial. Autrement dit, c'est le taux d'intérêt pour lequel $NPV = 0$. Ce taux de rentabilité interne est également appelé *rendement* ou *taux de rendement escompté.*

L'équation de *NPV* est :

$$NVP = \begin{cases} \displaystyle\sum_{j=1}^{k} CF_j \left(\frac{1-(1+i/100)^{-n_j}}{i/100} \right) (1+i/100)^{-\sum_{l<j} n_l} & \text{pour } i > -100,\ i \neq 0 \\ \displaystyle\sum_{j=1}^{k} CF_j n_j & \text{pour } i = 0 \end{cases}$$

où $\displaystyle\sum_{l<1} n_l$ est définie comme -1.

Le programme utilise la convention de signe suivante : toute somme d'argent reçue (introduite ou affichée) est positive, toute somme d'argent versée (introduite ou affichée) est négative.

Le programme présente les caractéristiques suivantes :

- La séquence des flux (y compris l'investissement initial) doit contenir des flux négatifs et des flux positifs. Autrement dit, il doit y avoir au moins un changement de signe.
- Le flux présentant plusieurs changements de signes peuvent avoir plus d'une solution. Ce programme n'en trouve qu'une et ne peut pas indiquer les autres possibilités.
- Le calcul de *IRR* peut durer plusieurs minutes (5 mn ou plus). Sa durée dépend du nombre de flux introduits.
- Le programme affiche **Error 4** lorsqu'il ne trouve pas de solution pour *IRR* ou lorsque le rendement *i* est inférieur ou égal à −100 % dans le calcul de *NPV*.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000-	
[f] [LBL] [A]	001-42,21,11	Programme pour *NPV*.
[EEX]	002- 26	
2	003- 2	
[÷]	004- 10	Calcule *IRR*/100.
[GSB] 2	005- 32 2	
[R/S]	006- 31	
[f] [LBL] [B]	007-42,21,12	Programme pour *IRR*.
1	008- 1	
[ENTER]	009- 36	
[EEX]	010- 26	
[CHS]	011- 16	
3	012- 3	
[f] [SOLVE] 2	013-42,10, 2	
[GTO] 1	014- 22 1	
[GTO] 0	015- 22 0	Branchement si pas de solution *IRR*.
[f] [LBL] 1	016-42,21, 1	
[EEX]	017- 26	

Appuyez sur	Affichage	
2	018– 2	
[×]	019– 20	
[R/S]	020– 31	
[f] [LBL] 2	021–42,21, 2	Calcule *NPV*.
[g] [CF] 0	022–43, 5, 0	
[STO] 2	023– 44 2	
1	024– 1	
[STO] 4	025– 44 4	
[+]	026– 40	Calcule $1 + IRR/100$.
[g] [TEST] 4	027–43,30, 4	Teste $IRR \leq -100$.
[GTO] 0	028– 22 0	Branchement si $IRR \leq -100$.
[STO] 3	029– 44 3	
0	030– 0	
[STO] 5	031– 44 5	
[f] [MATRIX] 1	032–42,16, 1	
[f] [LBL] 3	033–42,21, 3	
[g] [F?] 0	034–43, 6, 0	Teste si tous les flux sont utilisés.
[GTO] 7	035– 22 7	Branchement si tous les flux sont utilisés.
[GSB] 6	036– 32 6	
[RCL] 2	037– 45 2	
[g] [$x = 0$]	038– 43 20	Teste $IRR = 0$.
[GTO] 4	039– 22 4	Branchement si $IRR = 0$.
1	040– 1	
[+]	041– 40	
[GSB] 6	042– 32 6	
[CHS]	043– 16	
[y^x]	044– 14	
[STO] 4	045– 44 4	
1	046– 1	
[$x \gtrless y$]	047– 34	
[−]	048– 30	
[RCL] [÷] 2	049–45,10, 2	
[RCL] [×] 3	050–45,20, 3	
[GTO] 5	051– 22 5	
[f] [LBL] 4	052–42,21, 4	
[$x \gtrless y$]	053– 34	
[GSB] 6	054– 32 6	
[f] [LBL] 5	055–42,21, 5	

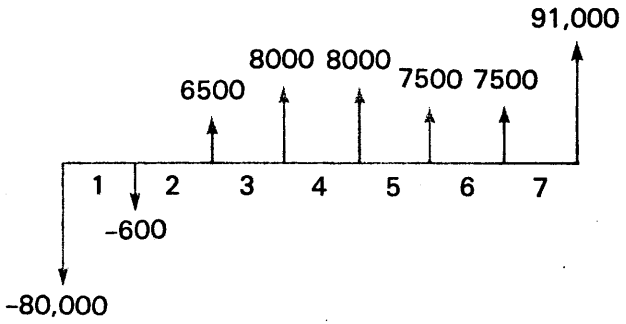
Appuyez sur	Affichage	
[X]	056- 20	
[STO] [+] 5	057-44,40, 5	
[RCL] 4	058- 45 4	
[STO] [X] 3	059-44,20, 3	
[GTO] 3	060- 22 3	
[f] [LBL] 6	061-42,21, 6	Rappelle l'élément flux.
[f] [USER] [RCL] [C]	062u 45 13	
[f] [USER]		
[g] [RTN]	063- 43 32	
[g] [SF] 0	064-43, 4, 0	Arme l'indicateur si c'est le dernier élément.
[f] [RTN]	065- 43 32	
[f] [LBL] 7	066-42,21, 7	
[RCL] 5	067- 45 5	Rappelle *NPV*.
[g] [RTN]	068- 43 32	

Labels utilisés : A, B et 0 à 7.
Registres utilisés : R_0 à R_5.
Matrice utilisée : C.

Pour utiliser le programme d'analyse des flux escomptés :

1. Appuyez sur 5 [f] [DIM] [(i)] pour allouer les registres R_0 à R_5.
2. Appuyez sur [f] [USER] pour valider le mode USER (sauf s'il est déjà validé).
3. Introduisez le nombre de groupes de flux et appuyez sur [ENTER] 2 [f] [DIM] [C] pour dimensionner la matrice C.
4. Appuyez sur [f] [MATRIX] 1 pour initialiser les numéros de rang et de colonne à 1.
5. Pour chaque groupe de flux :
   a) Introduisez la valeur des flux et appuyez sur [STO] [C].
   b) Introduisez le nombre de flux dans le groupe et appuyez sur [STO] [C].
6. Calculez le paramètre désiré :
   - Pour calculer *IRR*, appuyez sur [B].
   - Pour calculer *NPV*, introduisez le taux d'intérêt périodique *i* en pourcentage et appuyez sur [A]. Répétez cette procédure pour autant de taux d'intérêt que vous le désirez.
7. Répétez les étapes 3 à 6 de la procédure pour d'autres problèmes de flux.

Exemple 1 : Un investisseur achète 80,000 FF un duplex qu'il a l'intention de revendre au bout de 7 ans. Au cours de la première année, il doit faire des dépenses de réparations. A la fin de la septième année, le duplex est vendu 91,000 FF. Arrivera-t-il au rendement désiré de 9 % après impôts, avec l'historique ci-dessous des flux après impôts ?



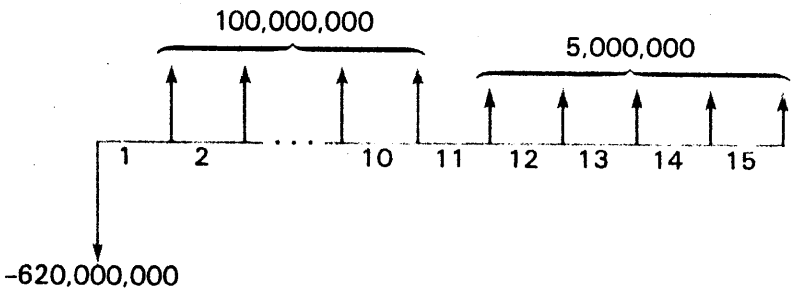
Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
[f] [FIX] 2		
5 [f] [DIM] [(i)]	5.00	Réserve les registres R_0 et R_5.
6 [ENTER] 2	2	
[f] [DIM] [C]	2.00	
[f] [MATRIX] 1	2.00	
[f] [USER]	2.00	
80000 [CHS] [STO] [C]	-80,000.00	Investissement initial.
1 [STO] [C]	1.00	
600 [CHS] [STO] [C]	-600.00	
1 [STO] [C]	1.00	
6500 [STO] [C]	6,500.00	
1 [STO] [C]	1.00	
8000 [STO] [C]	8,000.00	
2 [STO] [C]	2.00	
7500 [STO] [C]	7,500.00	
2 [STO] [C]	2.00	
91000 [STO] [C]	91,000.00	
1 [STO] [C]	1.00	
9	9	Introduit le rendement présumé.
[A]	-4,108.06	*NPV.*

Puisque *NPV* (Valeur actuelle nette) est négative, l'investissement n'assure pas la rentabilité désirée de 9 %. Calculez le taux de rentabilité interne (*IRR*).

Appuyez sur	Affichage	
B	**8.04**	*IRR* (au bout de 8 mn).

Le taux de rentabilité interne est donc inférieur à 9 %.

Exemple 2 : Il est prévu qu'un investissement de 620,000,000 FF produisent les flux de rentrées annuelles suivants au cours des 15 années à venir :



Quel taux de rentabilité peut-on espérer ?

Appuyez sur	Affichage	
3 ENTER 2	**2**	
f DIM C	**2.00**	
f MATRIX 1	**2.00**	
620000000 CHS	**-620,000,000**	
STO C	**-620,000,000.0**	
1 STO C	**1.00**	
100000000 STO C	**100,000,000.0**	
10 STO C	**10.00**	
5000000 STO C	**5,000,000.00**	
5 STO C	**5.00**	
B	**10.06**	*IRR*.
f FIX 4	**10.0649**	
f USER	**10.0649**	Invalide le mode USER.

Chapitre 2

Utilisation de $\int$

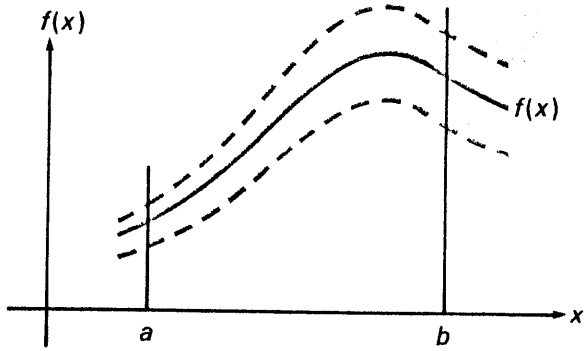
Le HP-15C vous permet les intégrations numériques à l'aide de $\int$. Ce chapitre explique comment utiliser efficacement $\int$ et décrit des techniques permettant de traiter des intégrales difficiles.

Intégration numérique avec $\int$

En général, l'intégration numérique sur calculateur n'est jamais très précise. Mais la fonction $\int$ vous demande d'une façon commode de spécifier dans quelle mesure l'erreur est tolérable. Elle vous demande de définir le format d'affichage en fonction de la précision voulue pour les chiffres de l'expression $f(x)$ à intégrer. En effet, vous spécifiez ainsi la largeur d'une bande à l'aire située sous quelque graphe non spécifié figurant entièrement à l'intérieur de la bande. Naturellement, cette estimation risque de varier en proportion avec la surface de la bande; c'est pourquoi $\int$ estime aussi cette surface. Si on appelle I l'intégrale désirée,

$$I = \left(\begin{array}{c}\text{Aire située sous un graphe}\\ \text{dessiné dans la bande}\end{array}\right) \pm \left(\begin{array}{c}\text{½ surface}\\ \text{de la bande}\end{array}\right)$$

Le HP-15C place l'estimation de la première surface (aire) dans le registre X et celle de la seconde (incertaine) dans le registre Y.



Par exemple, $f(x)$ pourrait représenter une conséquence physique dont l'amplitude ne peut être déterminée qu'à + 0.005. La valeur calculée pour $f(x)$ a donc une incertitude de 0.005. Un format d'affichage [FIX] 2 indique au calculateur que les chiffres décimaux situés au-delà de la deuxième position décimale n'ont aucune importance. Le calculateur ne doit pas perdre de temps à estimer l'intégrale avec une précision non garantie. Il va par contre vous donner une idée précise de la plage de valeurs dans laquelle doit être l'intégrale.

Le HP-15C ne vous empêche pas de déclarer que $f(x)$ est beaucoup plus précise qu'elle ne l'est. Vous pouvez soit faire une étude approfondie de l'erreur avant de spécifier le format d'affichage soit vous contenter d'une estimation. Vous pouvez laisser le format d'affichage à [SCI] 4 ou à [FIX] 4 pour simplifier. Vous obtiendrez une estimation de l'intégrale et de son imprécision, vous permettant d'interpréter le résultat plus intelligemment que si vous aviez eu la réponse sans aucune idée de sa précision ou de son imprécision.

L'algorithme de [∫] utilise la méthode de Romberg pour cumuler la valeur de l'intégrale. Plusieurs raffinements la rendent encore plus efficace.

Au lieu d'utiliser des échantillons régulièrement espacés, qui peuvent apporter une sorte de résonnance responsable de résultats trompeurs lorsque l'expression à intégrer est périodique, [∫] utilise des échantillons espacés irrégulièrement. Cet espacement peut-être démontré par substitution par exemple, de:

$$x = \frac{3}{2}u - \frac{1}{2}u^3$$

par

$$I = \int_{-1}^{1} f(x)dx = \int_{-1}^{1} f\left(\frac{3}{2}u - \frac{1}{2}u^3\right)\frac{3}{2}\ (1-u^2)\ du$$

avec un échantillonnage u uniforme. Outre la suppression de la résonnance, la substitution offre deux autres avantages. Premièrement, il n'est pas nécessaire de dessiner un échantillon à l'une ou l'autre extrémité de l'intervalle d'intégration (sauf lorsque l'intervalle est si petit qu'il n'y a pas d'autre possibilité). Il en résulte qu'une intégrale telle que:

$$\int_0^3 \frac{\sin x}{x}\,dx$$

ne sera pas interrompue par une division par zéro à un point d'extrémité. Deuxièmement, [∫] peut intégrer des fonctions se comportant comme $\sqrt{|x-a|}$, dont la pente est infinie à un point d'extrémité. De telles fonctions existent lorsqu'on calcule l'aire délimitée par une courbe régulière fermée.

Un autre raffinement est l'utilisation par [∫] de la précision étendue (13 chiffres significatifs) pour le cumul des sommes internes. Ceci permet le cumul de milliers d'échantillons sans plus de pertes d'arrondis que dans le sous-programme de la fonction.

Précision de la fonction à intégrer

La précision d'une intégrale calculée par [∫] dépend de la précision de la fonction calculée par votre sous-programme. Cette précision, que vous spécifiez à l'aide du format d'affichage, dépend principalement de trois facteurs :

- La précision de constantes empiriques dans la fonction.
- Le degré auquel la fonction peut décrire un phénomène physique avec précision.
- La portée des erreurs d'arrondis dans les calculs internes du calculateur.

Fonctions relatives à des phénomènes physiques

Des fonctions comme cos $(4\theta - \sin\theta)$ sont des *fonctions mathématiques pures*. Dans ce contexte, cela signifie que les fonctions ne contiennent aucune constante empirique et que ni les variables ni les limites de l'intégration ne représentent des quantités physiques réelles. Pour de telles fonctions, vous pouvez spécifier autant de chiffres que vous le désirez dans le format d'affichage (jusqu'à 9) pour atteindre le niveau de précision désiré dans l'intégrale*. Votre seul souci sera le compromis que vous souhaitez entre la précision désirée et la durée du calcul.

*Pourvu que $f(x)$ soit toujours calculée avec précision, en dépit des erreurs d'arrondis, au nombre de chiffres présents à l'affichage.

Cependant, d'autres facteurs jouent un rôle lorsque vous intégrez des fonctions concernant un phénomène physique réel. Avec de telles fonctions, demandez-vous simplement *si la précision que vous désirez dans l'intégrale est justifiée par la précision de la fonction.* Par exemple, si la fonction contient des constantes empiriques spécifiées par exemple sur trois chiffres significatifs seulement, cela n'aurait aucun sens de demander plus de trois chiffres dans le format d'affichage.

Une autre considération importante, sans doute plus subtile, est que toute fonction relative à un phénomène *contient une imprécision inhérente à sa nature jusqu'à un certain niveau,* parce qu'elle n'est qu'un *modèle* mathématique d'un processus ou d'un événement réel. Un modèle mathématique est lui-même une *approximation* qui ignore les effets de facteurs connus ou inconnus supposés comme insignifiants au niveau où les résultats sont utiles.

Un exemple de modèle mathématique est la *fonction de distribution normale*

$$\int_{-\infty}^{t} \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}}\,dx$$

considérée comme très utile pour dériver l'information relative à des mesures physiques sur les organismes vivants, les dimensions de produits, les températures moyennes, etc. De telles descriptions mathématiques sont soit dérivées de considérations théoriques soit issues de l'expérience. Pour être utilisables, elles ont été construites sur certaines hypothèses comme celle par exemple de l'ignorance des effets de facteurs relativement insignifiants. Par exemple, la précision des résultats obtenus en utilisant la fonction de distribution normale comme modèle de distribution de certaines quantités, dépend de la taille de la population considérée. Et la précision des résultats obtenus de l'équation $s = s_0 - \frac{1}{2}gt^2$ qui donne la hauteur d'un corps en chute libre, ignore la variation de g (accélération de la gravité) avec l'altitude.

Ainsi, les descriptions mathématiques du monde physique ne peuvent fournir des résultats que dans certaines limites de précision. Si vous avez calculé une intégrale avec une précision apparente supérieure à celle avec laquelle le modèle décrit le comportement réel du processus ou de l'événement, vous n'aurez pas nécessairement raison si vous utilisez la valeur calculée dans toute sa précision apparente.

Erreur d'arrondi dans les calculs internes

Avec le HP-15C, comme avec tout système de calcul, les résultats calculés doivent être arrondis à un nombre fini de chiffres (10 sur le HP-15C). A cause de cet *arrondi*, les résultats calculés – particulièrement les résultats d'évaluation d'une fonction contenant plusieurs opérations mathématiques – peuvent ne pas être exacts sur les 10 chiffres affichés. N'oubliez pas que l'erreur d'arrondi affecte l'évaluation de *toute* expression mathématique, et pas seulement l'évaluation d'une fonction à intégrer à l'aide de [∫f]. (Consultez l'annexe pour des explications supplémentaires.)

Si *f(x)* est une fonction décrivant un phénomène physique, son imprécision sur les arrondis est insignifiante en comparaison de l'imprécision introduite par les constantes empiriques, etc. Si *f(x)* est une fonction mathématique pure, sa précision ne dépend que de l'erreur d'arrondi. Généralement, il faut procéder à une analyse compliquée pour déterminer précisément combien de chiffres d'une fonction calculée risquent d'être affectés par l'erreur d'arrondi. En pratique, ces effets sont déterminés par l'expérience plus que par l'analyse.

Dans certains cas, l'erreur d'arrondi peut provoquer des résultats bizarres, surtout si vous comparez les résultats de calculs d'intégrales qui sont mathématiquement équivalentes mais qui diffèrent par une transformation de variables. Cependant, il est improbable que vous vous trouviez dans ces cas dans les applications classiques.

Réduction de la durée du calcul

La durée d'un calcul d'intégrale par [∫f] dépend du moment où est réalisée une certaine densité de points d'échantillonnage dans la région où la fonction est intéressante. Le calcul de l'intégrale d'une fonction sera plus long si l'intervalle d'intégration contient surtout des régions où la fonction n'est pas intéressante. Heureusement, lorsque vous devez calculer une telle intégrale, vous avez la possibilité de modifier le problème pour réduire la durée du calcul. Deux de ces techniques sont les suivantes : la subdivision de l'intervalle d'intégration et la transformation des variables.

Subdivision de l'intervalle d'intégration

Dans les régions où la pente de $f(x)$ varie beaucoup, une haute densité de points d'échantillonnage est nécessaire pour fournir une approximation qui change de façon insignifiante d'une itération à la suivante. Par contre, dans les régions où la pente de la fonction est à peu près constante, une haute densité de points d'échantillonnage n'est pas nécessaire. Ceci parce que l'évaluation de la fonction sur d'autres points d'échantillonnage ne donnerait pas beaucoup plus de renseignements sur la fonction, donc n'affecterait pas considérablement les disparités entre les approximations successives. Par conséquent, dans ce type de région, une approximation de précision comparable pourrait être réalisée avec beaucoup moins de points d'échantillonnage; donc en bien moins de temps. Lorsque vous intégrez ce genre de fonctions, vous pouvez gagner du temps en utilisant la procédure suivante:

1. Divisez l'intervalle d'intégration en sous-intervalles sur lesquels la fonction est intéressante et en sous-intervalles sur lesquels la fonction n'est pas intéressante.

2. Sur les sous-intervalles dans lesquels la fonction est intéressante, calculez l'intégrale dans le format d'affichage correspondant à la précision que vous recherchez.

3. Sur les sous-intervalles dans lesquels la fonction n'est pas intéressante ou contribue à l'intégrale de façon négligeable, calculez l'intégrale avec moins de précision, c'est-à-dire en spécifiant moins de chiffres dans le format d'affichage.

4. Pour obtenir l'intégrale sur la totalité de l'intervalle d'intégration, ajoutez les deux approximations précédentes à l'aide de la touche [Σ+].

Avant de subdiviser l'intervalle d'intégration, vérifiez si le calculateur passe en dépassement de capacité inférieur lorsqu'il évalue la fonction autour de la limite supérieure (ou inférieure) de l'intégration*. Puisqu'il n'y a aucune raison d'évaluer la fonction à des valeurs de x pour lesquelles le calculateur est en dépassement de capacité inférieur, la limite supérieure de l'intégration peut être réduite dans certains cas pour réduire la durée du calcul.

*Lorsqu'un calcul risque de résulter en un nombre inférieur à 10^{-99}, le résultat est remplacé par zéro. C'est ce qu'on appelle un dépassement de capacité inférieur.

N'oubliez pas que dès que vous avez introduit le sous-programme d'évaluation *f(x)*, vous pouvez calculer *f(x)* pour toute valeur de *x* en introduisant cette valeur dans le registre X et en appuyant sur [ENTER] [ENTER] [ENTER] [GSB] suivi du label du sous-programme.

Si le calculateur passe en dépassement de capacité inférieur à la limite supérieure de l'intégration, essayez de plus petits nombres jusqu'à ce que vous vous rapprochiez du point où le calculateur ne présente plus de dépassement de capacité inférieur.

Par exemple, pour l'approximation de

$$\int_0^{\infty} xe^{-x}dx \ .$$

Introduisez un sous-programme qui calcule la fonction $f(x) = xe^{-x}$.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	Efface la mémoire programme.
[f] [LBL] 1	001–42,21, 1	
[CHS]	002– 16	
[e^x]	003– 12	
[x]	004– 20	
[g] [RTN]	005– 43 32	

Mettez le calculateur en mode calcul et définissez le format d'affichage à [SCI] 3. Essayez ensuite plusieurs valeurs de *x* pour rechercher où le calculateur présente un dépassement de capacité inférieur pour votre fonction.

Appuyez sur	Affichage		
[g] [P/R]			Mode programme.
[f] [SCI] 3			Met le format à [SCI] 3.
[EEX] 3	1	03	Introduit 1000 dans le registre X
[ENTER] [ENTER] [ENTER]	1.000	03	Met *x* dans la pile.
[GSB] 1	0.000	00	Le calculateur donne un résultat nul pour $X = 1000$.
300 [ENTER]	3.000	02	Nouvelle valeur de *x*, plus petite.
[ENTER] [ENTER]	3.000	02	
[GSB] 1	0.000	00	Résultat nul.
200 [ENTER]	2.000	02	Nouvelle valeur de *x*, plus petite.

Appuyez sur	Affichage		
[ENTER] [ENTER]	2.000	02	
[GSB] 1	2.768	-85	Le calculateur donne un résultat non nul pour $x = 200$; essayez une nouvelle valeur comprise entre 200 et 250.
225 [ENTER]	2.250	02	
[ENTER] [ENTER]	2.250	02	
[GSB] 1	4.324	-96	Le calculateur est proche du résultat nul.

A ce niveau, vous pouvez utiliser [SOLVE] pour localiser la plus petite valeur de x à laquelle il y a dépassement de capacité inférieur.

Appuyez sur	Affichage		
[R↓]	2.250	02	Descend la pile jusqu'à ce que la dernière valeur essayée soit dans les registres X et Y.
[f] [SOLVE] 1	2.280	02	Valeur minimale de x pour laquelle il y a dépassement inférieur (= 228).

Vous avez ainsi déterminé que vous ne pouvez intégrer qu'entre 0 et 228. Puisque l'expression à intégrer n'est intéressante que pour $x < 10$, divisez à ce niveau l'intervalle d'intégration. Le problème devient le suivant :

$$\int_0^{\infty} xe^{-x}dx \approx \int_0^{228} xe^{-x}dx = \int_0^{10} xe^{-x}dx + \int_{10}^{228} xe^{-x}dx .$$

Appuyez sur	Affichage		
7 [f] [DIM] [(i)]	7.000	00	Alloue les registres statistiques.
[f] CLEAR [Σ+]	0.000	00	Efface les registres statistiques.
0 [ENTER]	0.000	00	Introduit la limite inférieure de l'intégration sur le premier sous-intervalle.
10	10		Introduit la limite supérieure de l'intégration sur le premier sous-intervalle.

Appuyez sur	Affichage		
f ∫ 1	9.995	−01	Intégrale sur (0,10) calculée en SCI 3.
Σ+	1.000	00	Ajoute l'approximation et son incertitude dans les registres R_3 et R_5.
x≷y	1.841	−04	Incertitude de l'approximation.
R↓ R↓	1.000	01	Descend la pile jusqu'à ce que la limite supérieure de la première intégrale apparaisse dans le registre X.
228	228		Introduit la limite supérieure de la seconde intégrale dans le registre X. La limite supérieure de la première intégrale monte dans le registre Y, devenant ainsi la limite inférieure de la seconde intégrale.
f SCI 0	2.	02	Spécifie SCI 0 comme format d'affichage pour un calcul rapide sur (10,228). Si l'incertitude de l'approximation devient trop imprécise, vous pouvez répéter l'approximation dans un format d'affichage plus large.
f ∫ 1	5.	−04	Intégrale sur (10,228) calculée en SCI 0.
f SCI 3	5.328	−04	Remet le format d'affichage en SCI 3.
x≷y	7.568	−05	Vérifie l'incertitude de l'approximation. Puisqu'elle est inférieure à l'incertitude de l'approximation sur le premier sous-intervalle, SCI 0 a donc fourni une approximation de précision suffisante.

Appuyez sur	Affichage		
[x⇄y]	5.328	-04	Place l'approximation et son incertitude dans les registres X et Y respectivement, avant de les ajouter dans les registres statistiques.
[Σ+]	2.000	00	Ajoute l'approximation et son incertitude.
[RCL] [Σ+]	1.000	00	Intégrale sur la totalité de l'intervalle (0,228) (rappelé de R_3).
[x⇄y]	2.598	-04	Incertitude de l'intégrale (de R_5).

Transformation de variables

Dans beaucoup de problèmes où une fonction varie peu sur la plus grande partie de l'intervalle d'intégration, une transformation de variables appropriée peut réduire la durée du calcul de l'intégrale.

Par exemple, reprenons l'intégrale

$$\int_0^{\infty} xe^{-x}dx\,.$$

Faisons $e^{-x} = u^3.$

Puis $x = -3 \ln u$

et $dx = -3\dfrac{du}{u}.$

En substituant

$$\int_0^{\infty} xe^{-x}dx = \int_{e^{-0}}^{e^{-\infty}} (-3 \ln u)(u^3)\left(-3\frac{du}{u}\right)$$

$$= \int_1^0 9u^2 \ln u\, du.$$

Introduisez le sous-programme d'évaluation de la fonction $f(u) = 9u^2 \ln u$.

Appuyez sur	Affichage	
[g] [P/R]	000–	Mode programme.
[f] [LBL] 3	001–42,21, 3	
[g] [LN]	002– 43 12	
[x≷y]	003– 34	
[g] [x²]	004– 43 11	
[x]	005– 20	
9	006– 9	
[x]	007– 20	
[g] [RTN]	008– 43 32	

Introduisez les limites de l'intégration, et appuyez sur [f] [∫] 3 pour calculer l'intégrale.

Appuyez sur	Affichage		
[g] [P/R]			Mode calcul.
1 [ENTER]	1.000	00	Introduit la limite inférieure de l'intégration.
0	0		Introduit la limite supérieure de l'intégration.
[f] [∫] 3	1.000	00	Approximation à une intégrale équivalente.
[x≷y]	3.020	–04	Incertitude de l'approximation.

L'approximation est en accord avec la valeur calculée dans le problème précédent pour la même intégrale.

Évaluation d'intégrales difficiles

Certaines conditions peuvent prolonger la durée du calcul lors de l'évaluation d'une intégrale ou provoquer des résultats imprécis. Ces conditions, décrites dans le *manuel d'utilisation du HP-15C,* sont liées à la nature de l'expression à intégrer sur l'intervalle choisi.

Une catégorie d'intégrales difficiles à évaluer est constituée par les intégrales impropres. Une intégrale impropre est une intégrale qui utilise ∞ (l'infini) de l'une des façons suivantes :

- L'une ou les deux limites de l'intégration sont $\pm \infty$, par exemple :

$$\int_{-\infty}^{\infty} e^{-u^2} du = \sqrt{\pi} .$$

- L'expression à intégrer tend vers $\pm \infty$ quelque part dans la plage d'intégration, par exemple :

$$\int_0^1 \ln(u)\, du = 1.$$

- L'expression à intégrer oscille infiniment et rapidement quelque part dans la plage d'intégration, par exemple :

$$\int_0^1 \cos(\ln u)\, du = \tfrac{1}{2}.$$

Certaines intégrales sont des intégrales presque impropres lorsque :

- L'expression à intégrer ou sa première dérivée change beaucoup dans un sous-intervalle relativement étroit de la plage d'intégration, ou oscille fréquemment à travers cette plage.

Le HP-15C tente de traiter certaines des intégrales impropres du deuxième type en n'échantillonnant pas l'expression à intégrer aux limites de l'intégration.

Comme les intégrales impropres (ou presque) ne sont pas courantes en pratique, vous pourrez les reconnaître et prendre les mesures nécessaires pour les évaluer précisément. Les exemples suivants illustrent quelques techniques utiles.

Considérons l'expression

$$f(x) = \frac{\sqrt{-2 \ln \cos(x^2)}}{x^2}.$$

Cette fonction perd sa précision lorsque x devient petit. Ceci parce que cos (x^2) est arrondi à 1, ce qui perd l'information sur la petitesse de x. Mais en utilisant $u = \cos(x^2)$, vous pouvez évaluer l'expression à intégrer comme :

$$f(x) = \begin{cases} 1 & \text{if } u = 1 \\ \dfrac{\sqrt{-2 \ln u}}{\cos^{-1} u} & \text{if } u \neq 1. \end{cases}$$

Bien que le branchement de programme pour $u = 1$ ajoute quatre étapes supplémentaires à votre sous-programme, l'intégration près de $x = 0$ devient plus précise.

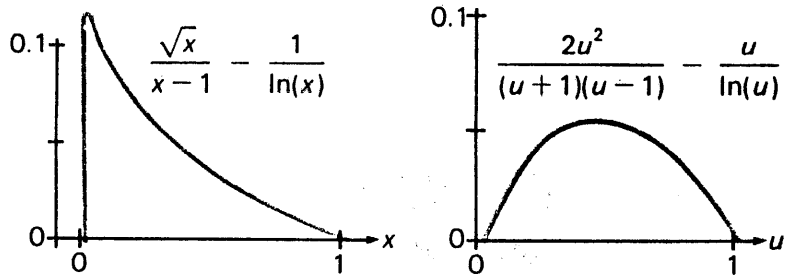
Voici un deuxième exemple d'intégrale :

$$\int_0^1 \left(\frac{\sqrt{x}}{x-1} - \frac{1}{\ln x} \right) dx.$$

La dérivée de cette expression approche l'infini lorsque x s'approche de 0, comme le montre l'illustation ci-dessous. En substituant $x = u^2$, la fonction se comporte mieux, comme le montre la seconde illustration. Cette intégrale de substitution peut être facilement évaluée :

$$\int_0^1 \left(\frac{2u^2}{(u+1)(u-1)} - \frac{u}{\ln u} \right) du.$$

Ne remplacez pas $(u+1)(u-1)$ par (u^2-1) parce que lorsque u s'approche de 1, la seconde expression perd à l'arrondi la moitié de ses chiffres significatifs et introduit un pic dans le graphe près de $u = 1$.



Comme autre exemple, considérons une fonction dont le graphe accuse une branche infinie ("queue") qui s'étale sur une région plusieurs fois plus grande que la région occupée par le "corps" principal (où le graphe est intéressant). C'est l'exemple des fonctions suivantes :

$$f(x) = e^{-x^2} \quad \text{or} \quad g(x) = \frac{1}{x^2 + 10^{-10}}.$$

Des branches infinies fines, comme celle de $f(x)$ peuvent être tronquées sans grand dommage à la précision ou à la rapidité de l'intégration. Mais $g(x)$ a une branche infinie trop large pour être ignorée lorsqu'on calcule

$$\int_{-t}^{t} g(x)\,dx$$

si t est large.

Pour ce type de fonction, une substitution comme $x = a + b \tan u$ est excellente: a est dans le "corps" principal du graphe et b est une bonne représentation de sa largeur. En faisant cela pour $f(x)$ ci-dessus avec $a = 0$ et $b = 1$, on obtient

$$\int_0^t f(x)\,dx = \int_0^{\tan^{-1}t} e^{-\tan^2 u}(1 + \tan^2 u)\,du,$$

qui est calculée directement même si t est aussi grand que 10^{10}. En adoptant la même substitution avec $g(x)$, les valeurs proches de $a = 0$ et $b = 10^5$ donnent de bons résultats.

Cet exemple implique la subdivision de l'intervalle d'intégration. Bien qu'une fonction puisse avoir des caractéristiques qui paraissent extrêmes sur la totalité de l'intervalle d'intégration, la fonction peut paraître mieux se comporter sur certaines portions de cet intervalle. La subdivision de l'intervalle d'intégration fonctionne encore mieux lorsqu'elle est combinée avec des substitutions appropriées. Considérons l'intégrale

$$\begin{aligned}
\int_0^\infty dx/(1+x^{64}) &= \int_0^1 dx/(1+x^{64}) + \int_1^\infty dx/(1+x^{64}) \\
&= \int_0^1 dx/(1+x^{64}) + \int_0^1 u^{62}du/(u^{64}+1) \\
&= \int_0^1 (1+x^{62})dx/(1+x^{64}) \\
&= 1 + \int_0^1 (x^{62}-x^{64})dx/(1+x^{64}) \\
&= 1 + \tfrac{1}{8}\int_0^1 (1-v^{1/4})v^{55/8}\,dv/(1+v^8).
\end{aligned}$$

Ces étapes opèrent les substitutions $x = 1/u$ et $x = v^{1/8}$ et font quelques manipulations algébriques. Bien que l'intégrale d'origine soit impropre, la dernière intégrale est facilement traitée par $\int$. En fait, en séparant le terme constant de l'intégrale, vous obtenez (en utilisant SCI 8) une réponse avec 13 chiffres significatifs:

$$1.000401708155 \pm 1.2 \times 10^{-12}.$$

Prenons comme dernier exemple le champ électrostatique pour une sonde ellipsoïdale dont les demi-axes principaux sont a, b et c.

$$V = \int_0^\infty \frac{dx}{(a^2+x)\sqrt{(a^2+x)(b^2+x)(c^2+x)}}$$

pour $a = 100$, $b = 2$ et $c = 1$*.

Transformez cette intégrale impropre en une intégrale correcte en substituant $x = (a^2 - c^2)/(1 - u^2) - a^2$:

$$V = p\int_r^1 \sqrt{(1-u^2)/(u^2+q)}\, du$$

où

$$p = 2/((a^2-c^2)\sqrt{a^2-b^2}) = 2.00060018 \times 10^{-6}$$

$$q = (b^2-c^2)/(a^2-b^2) = 3.001200480 \times 10^{-3}$$

$$r = c/a = 0.01\,.$$

Cependant, cette intégrale est presque impropre parce que q et r sont tous deux très proches de zéro. Mais en utilisant une intégrale de formulation proche ressemblant suffisamment à la partie gênante de V, la difficulté peut être levée. Essayez:

$$W = p\int_r^1 du/\sqrt{u^2+q} = p\ln(u+\sqrt{u^2+q})\Big|_r^1$$

$$= p\ln((1+\sqrt{1+q})/(r+\sqrt{r^2+q}))$$

$$= 8.40181880708 \times 10^{-6}.$$

Puis:

$$V = W + p\int_r^1 (\sqrt{(1-u^2)/(u^2+q)} - 1/\sqrt{u^2+q})\, du$$

$$= p\int_r^1 \left(\frac{W/p}{1-r} - \frac{u^2}{(1+\sqrt{1-u^2})\sqrt{u^2+q}}\right) du.$$

*De Strattom, J.A., *Electromagnetic Theory*, McGraw-Hill, New York, 1941, p. 201-217.

Le HP-15C traite directement cette intégrale. La valeur de $\sqrt{1-u^2}$ lorsque u tend vers 1 ne doit pas vous poser de problème puisque les chiffres perdus par les arrondis ne sont pas nécessaires.

Application

Le programme suivant calcule les valeurs de quatre fonctions spéciales pour tout argument x:

$$P(x)=\frac{1}{2\pi}\int_{-\infty}^{x} e^{-t^2/2}dt$$ (fonction de distribution normale)

$$Q(x)=1-P(x)=\frac{1}{2\pi}\int_{x}^{\infty} e^{-t^2/2}dt$$ (fonction complémentaire de distribution normale)

$$\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_{0}^{x} e^{-t^2}dt$$ (fonction d'erreur)

$$\mathrm{erfc}(x)=1-\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_{x}^{\infty} e^{-t^2}dt$$ (fonction complémentaire d'erreur)

Le programme calcule ces fonctions en utilisant la transformation $u=e^{-t^2}$ pour $|x|\geq 1.6$.

La valeur de la fonction est renvoyée dans le registre X et l'incertitude de *l'intégrale* est renvoyée dans le registre Y. (L'incertitude de la valeur de *la fonction* est à peu près du même ordre de grandeur que le nombre contenu dans le registre Y.) L'argument d'origine est dans le registre R_0.

Le programme présente les caractéristiques suivantes:

- Le format d'affichage spécifie la précision de l'expression à intégrer de la même façon qu'il le fait pour [∫]. Cependant, si vous spécifiez un nombre inutilement long de chiffres à afficher, le calcul sera prolongé.
- Des petites valeurs de fonctions, comme Q(20), P(20) et erfc(10) sont calculées très précisément aussi rapidement que des valeurs moyennes.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] [A]	001–42,21,11	Programme pour P(x).
[STO] 2	002– 44 2	Stocke x dans R_2.
[CHS]	003– 16	Calcule $-x$.
[GTO] 2	004– 22 2	Branchement pour calculer $P(x) = Q(-x)$.
[f] [LBL] [B]	005–42,21,12	Programme pour Q(x).
[STO] 2	006– 44 2	Stocke x dans R_2.
[f] [LBL] 2	007–42,21, 2	
2	008– 2	
[$\sqrt{x}$]	009– 11	
[÷]	010– 10	
[GSB] [C]	011– 32 13	Calcule erfc($x/\sqrt{2}$).
2	012– 2	
[÷]	013– 10	Calcule $Q(x) = 1/2$. erfc $(x/\sqrt{2})$.
[RCL] 2	014– 45 2	
[STO] 0	015– 44 0	Stocke x dans R_0.
[R↓]	016– 33	
[g] [RTN]	017– 43 32	Valeur de la fonction.
[f] [LBL] [C]	018–42,21,13	Programme pour erfc(x).
1	019– 1	
[GSB] 4	020– 32 4	
[g] [F?] 1	021–43, 6, 1	Teste l'indicateur 1.
[GTO] 5	022– 22 5	Branchement pour indicateur 1 armé.
1	023– 1	
[−]	024– 30	Calcule erf(x) − 1 pour indicateur 1 désarmé.
[f] [LBL] 5	025–42,21, 5	
[CHS]	026– 16	Calcule erfc (x).
[g] [RTN]	027– 43 32	Valeur de la fonction.
[f] [LBL] [E]	028–42,21,15	Programme pour erf(x).
0	029– 0	
[f] [LBL] 4	030–42,21, 4	Sous-programme pour erf(x) ou erfc(x).
[g] [CF] 1	031–43, 5, 1	Efface l'indicateur 1.

Appuyez sur	Affichage	
[STO] 1	032– 44 1	Stocke 0 pour erf(x) et 1 pour erfc(x).
[x≷y]	033– 34	
[STO] 0	034– 44 0	
[g] [ABS]	035– 43 16	Calcule $\|x\|$.
1	036– 1	
[·]	037– 48	
6	038– 6	
[g] [TEST] 8	039–43,30, 8	Teste $\|x\| > 1.6$.
[GTO] 6	040– 22 6	Branchement pour $\|x\| > 1.6$.
0	041– 0	
[RCL] 0	042– 45 0	Rappelle x.
[f] [∫] 0	043–42,20, 0	Intègre e^{-t^2} de 0 à x.
2	044– 2	
[x]	045– 20	
[f] [LBL] 3	046–42,21, 3	Sous-programme pour diviser par $\sqrt{\pi}$.
[g] [π]	047– 43 26	
[√x]	048– 11	
[÷]	049– 10	
[g] [RTN]	050– 43 32	
[f] [LBL] 6	051–42,21, 6	Sous-programme pour intégrer quand $\|x\| > 1.6$.
[g] [SF] 1	052–43, 4, 1	Arme l'indicateur 1.
0	053– 0	
[RCL] 0	054– 45 0	
[g] [x^2]	055– 43 11	
[CHS]	056– 16	
[e^x]	057– 12	Calcule e^{-x^2}.
[f] [∫] 1	058–42,20, 1	Intègre $(-\ln u)^{-1/2}$ de 0 à e^{-x^2}.
[GSB] 3	059– 32 3	Divise l'intégrale par $\sqrt{\pi}$.
[RCL] 0	060– 45 0	
[ENTER]	061– 36	
[g] [ABS]	062– 43 16	
[÷]	063– 10	Calcule le signe de x.
[x]	064– 20	

Appuyez sur	Affichage	
[RCL] 1	065– 45 1	Rappelle 1 pour erfc(x), 0 pour erf(x).
[g] [LSTx]	066– 43 36	
[--]	067– 30	
[+]	068– 40	Ajuste l'intégrale pour le signe de x et la fonction.
[CHS]	069– 16	
[g] [RTN]	070– 43 32	
[f] [LBL] 0	071–42,21, 0	Sous-programme pour calculer e^{t^2}.
[g] [x^2]	072– 43 11	
[CHS]	073– 16	
[e^x]	074– 12	
[g] [RTN]	075– 43 32	
[f] [LBL] 1	076–42,21, 1	Sous-programme pour calculer $(-\ln u)^{-1/2}$.
[g] [x = 0]	077– 43 20	
[g] [RTN]	078– 43 32	
[g] [LN]	079– 43 12	
[CHS]	080– 16	
[$\sqrt{x}$]	081– 11	
[1/x]	082– 15	
[g] [RTN]	083– 43 32	

Labels utilisés : A, B, C, E, 0, 1, 2, 3, 4, 5 et 6.

Registres utilisés : R_0 (x), R_1, R_2.

Indicateur utilisé : 1.

Pour utiliser ce programme :

1. Introduire l'argument x à l'affichage.
2. Évaluer la fonction désirée :
   - Appuyez sur [f] [A] pour évaluer P(x).
   - Appuyez sur [f] [B] pour évaluer Q(x).
   - Appuyez sur [f] [E] pour évaluer erf(x).
   - Appuyez sur [f] [C] pour évaluer erfc(x).

Exemple 1 : Calculez Q(20), P(1.234) et erf(0.5) dans le format d'affichage [SCI] 3.

Appuyez sur	Affichage		
[g] [P/R]			Mode calcul.
[f] [SCI] 3			Format d'affichage.
20 [f] [B]	**2.754**	**–89**	Q(20).
1.234 [f] [A]	**8.914**	**–01**	P(1.234).
.5 [f] [E]	**5.205**	**–01**	erf(0.5).

Exemple 2 : Pour une variable aléatoire X normalement distribuée, ayant une moyenne de 2.151 et un écart type de 1.085, calculez la probabilité $Pr[2 < X \leqslant 3]$.

$$Pr\,[2<X\leqslant 3]=Pr\left[\frac{2-2.151}{1.085}<\frac{X-\mu}{\sigma}\leqslant\frac{3-2.151}{1.085}\right]$$

$$=\mathrm{P}\left(\frac{3-2.151}{1.085}\right)-\mathrm{P}\left(\frac{2-2.151}{1.085}\right)$$

Appuyez sur	Affichage		
2 [ENTER]	**2.000**	**00**	
2.151 [–]	**–1.510**	**–01**	
1.085 [÷]	**–1.392**	**–01**	
[f] [A]	**4.447**	**–01**	Calcule $Pr[X \leqslant 2]$.
[STO] 3	**4.447**	**–01**	Stocke le résultat.
3 [ENTER]	**3.000**	**00**	
2.151 [–]	**8.490**	**–01**	
1.085 [÷]	**7.825**	**–01**	
[f] [A]	**7.830**	**–01**	Calcule $Pr[X \leqslant 3]$.
[RCL] 3	**4.447**	**–01**	Rappelle $Pr[X \leqslant 2]$.
[–]	**3.384**	**–01**	Calcule $Pr[2 < X \leqslant 3]$.
[f] [FIX] 4	**0.3384**		

Chapitre 3

Calculs en Mode Complexe

Certains problèmes importants concernant des données réelles sont très souvent résolus par des calculs simples utilisant les nombres complexes. Ce chapitre donne des explications précieuses sur les calculs en mode complexe et illustre par de nombreux exemples la résolution de problèmes sur des nombres complexes.

Utilisation du Mode Complexe

Le mode complexe dans le HP-15C vous permet d'évaluer simplement des expressions de nombres complexes. Généralement, dans le mode complexe, les expressions mathématiques sont introduites de la même façon que dans le mode "réel" normal. Par exemple, considérons un programme qui évalue le polynôme $P(x) = a_n x^n + \ldots + a_1 x + a_0$ pour la valeur x du registre X. En validant le mode complexe, ce même programme peut évaluer $P(z)$ où z est complexe. De même, d'autres expressions comme la fonction Gamma $\Gamma(x)$ dans l'exemple suivant, peuvent être évaluées pour des arguments complexes dans le mode complexe.

Exemple 1 : Écrire un programme évaluant le calcul d'approximation par fractions successives :

$$\ln(\Gamma(x)) = (x - \tfrac{1}{2})\ln x - x + a_0 + \cfrac{a_1}{x + \cfrac{a_2}{x + \cfrac{a_3}{x + \ldots}}}$$

pour les six premières valeurs de a :

$$a_0 = \tfrac{1}{2}\ln(2\pi)$$
$$a_1 = 1/12$$
$$a_2 = 1/30$$
$$a_3 = 53/210$$
$$a_4 = 195/371$$
$$a_5 = 1.011523068$$
$$a_6 = 1.517473649.$$

Puisque cette approximation est valide à la fois pour les arguments réels et pour les arguments complexes lorsque $\mathrm{Re}(z) > 0$, ce programme fait une approximation de $\ln(\Gamma(z))$ en mode complexe (pour $|z|$ suffisamment large). Quand $|z| > 4$ (et $\mathrm{Re}(z) > 0$), l'approximation comporte environ 9 ou 10 chiffres exacts.

Introduisez le programme suivant :

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] [A]	001–42,21,11	
6	002– 6	
[STO] [I]	003– 44 25	Stocke le compteur dans le registre d'Index.
[x≷y]	004– 34	
[ENTER]	005– 36	
[ENTER]	006– 36	
[ENTER]	007– 36	Remplit la pile avec z.
[RCL] 6	008– 45 6	Rappelle a_6.
[f] [LBL] 1	009–42,21, 1	Boucle pour la fraction continue.
[+]	010– 40	
[RCL] [(i)]	011– 45 24	Rappelle a_i.
[x≷y]	012– 34	Restaure z.
[÷]	013– 10	
[f] [DSE] [I]	014–42, 5,25	Diminue le compteur.
[GTO] 1	015– 22 1	
[RCL] 0	016– 45 0	Rappelle a_0.
[+]	017– 40	
[x≷y].	018– 34	Restaure z.
[−]	019– 30	
[g] [LSTx]	020– 43 36	Rappelle z.
[g] [LN]	021– 43 12	Calcule $\ln(z)$.
[g] [LSTx]	022– 43 36	Rappelle z.
[·]	023– 48	
5	024– 5	
[−]	025– 30	Calcule $z-1/2$.

Appuyez sur	Affichage	
[×]	026– 20	
[+]	027– 40	Calcule $\ln(\Gamma(z))$.
[g] [RTN]	028– 43 32	

Stocke les constantes dans les registres R_0 à R_6 en respectant l'ordre déterminé par leurs indices.

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
2 [g] [π] [×]	6.2832	
[g] [LN] 2 [÷]	0.9189	
[STO] 0	0.9189	Stocke a_0.
12 [1/x] [STO] 1	0.0833	Stocke a_1.
30 [1/x] [STO] 2	0.0333	Stocke a_2.
53 [ENTER] 210 [÷]	0.2524	
[STO] 3	0.2524	Stocke a_3.
195 [ENTER] 371 [÷]	0.5256	
[STO] 4	0.5256	Stocke a_4.
1.011523068 [STO] 5	1.0115	Stocke a_5.
1.517473649 [STO] 6	1.5175	Stocke a_6.

Utilisez ce programme pour calculer $\ln(\Gamma(4.2))$, puis comparez le résultat avec $\ln(3.2!)$ calculé avec la fonction [x!]. Calculez aussi $\ln(\Gamma(1 + 5i))$.

Appuyez sur	Affichage	
4.2 [f] [A]	2.0486	Calcule $\ln(\Gamma(4.2))$.
[f] [FIX] 9	2.048555637	Affiche 10 chiffres.
3.2 [f] [x!]	7.756689536	Calcule $(3.2)! = \Gamma(3.2 + 1)$.
[g] [LN]	2.048555637	Calcule $\ln(3.2!)$.
1 [ENTER]	1.000000000	Introduit la partie réelle de $1 + 5i$.
5 [f] [I]	1.000000000	Forme le nombre complexe $1 + 5i$.

Appuyez sur	Affichage	
f A	**-6.130324145**	Partie réelle de $\ln(\Gamma(1+5i))$.
f Re≷Im	**3.815898575**	Partie imaginaire de $\ln(\Gamma(1+5i))$.
f FIX 4	**3.8159**	

Le résultat complexe est calculé sans plus d'efforts qu'il ne faut pour introduire la partie imaginaire de l'argument z. (Le résultat $\ln(\Gamma(1+5i))$ comporte 10 chiffres exacts dans chacune de ses composantes.)

Modes trigonométriques

Bien que l'indicateur du mode trigonométrique reste affiché en mode complexe, les fonctions complexes sont *toujours* calculées en *radians*. L'indicateur ne précise le mode (Degrés, Radians ou Grades) que pour les deux conversions complexes : [→P] et [→R].

Si vous désirez évaluer $re^{i\theta}$ où θ est en degrés, [e^x] ne peut pas être utilisée directement parce que θ doit être en radians. Si vous tentez une conversion de degrés en radians, vous perdez un peu de précision surtout pour des valeurs comme 180° pour lesquelles la mesure π en radians ne peut pas être représentée exactement avec 10 chiffres.

Cependant, en mode complexe la fonction [→R] calcule $re^{i\theta}$ pour θ avec précision dans n'importe quelle unité (indiquée par l'indicateur). Introduisez simplement r et θ dans le registre X complexe sous la forme $r + i\theta$, puis exécutez [→R] pour calculer la valeur complexe :

$$re^{i\theta} = r\cos\theta + ir\sin\theta.$$

(Le programme figurant sous le titre "Calcul des $n$ièmes racines d'un nombre complexe" à la fin de ce chapitre, utilise cette fonction.)

Définitions des fonctions mathématiques

La liste suivante définit le fonctionnement du HP-15C en mode complexe. Dans ces définitions, un nombre complexe est noté sous la forme $z = x + iy$ (forme rectangulaire) ou $z = re^{i\theta}$ (forme polaire). On rencontre également la forme $|z| = \sqrt{x^2 + y^2}$.

Opérations arithmétiques

$$(a+ib)\pm(c+id)=(a\pm c)+i(b\pm d)$$

$$(a+ib)(c+id)=(ac-bd)+i(ad+bc)$$

$$z^2=z\times z$$

$$1/z=x/|z|^2-iy/|z|^2$$

$$z_1\div z_2=z_1\times 1/z_2$$

Fonctions à une valeur

$$e^z=e^x(\cos y+i\sin y)$$

$$10^z=e^{z\ln 10}$$

$$\sin z=\frac{1}{2i}(e^{iz}-e^{-iz})$$

$$\cos z=\tfrac{1}{2}(e^{iz}+e^{-iz})$$

$$\tan z=\sin z/\cos z$$

$$\sinh z=\tfrac{1}{2}(e^z-e^{-z})$$

$$\cosh z=\tfrac{1}{2}(e^z+e^{-z})$$

$$\tanh z=\sinh z/\cosh z$$

Fonctions à plusieurs valeurs

En général, l'inverse d'une fonction $f(z)$ – représenté par $f^{-1}(z)$ – comporte plus d'une valeur pour tout argument z. Par exemple, $\cos^{-1}(z)$ a un nombre infini de valeurs pour chaque argument. Mais le HP-15C calcule seulement la *valeur principale*, qui figure dans la partie de la plage de valeurs définie comme branche principale de $f^{-1}(z)$. Dans les explications ci-dessous, la fonction inverse à une valeur (réduite à sa branche principale) est représentée en lettres majuscules – par exemple, $\mathrm{COS}^{-1}(z)$ – pour la distinguer de la fonction inverse à plusieurs valeurs – $\cos^{-1}(z)$.

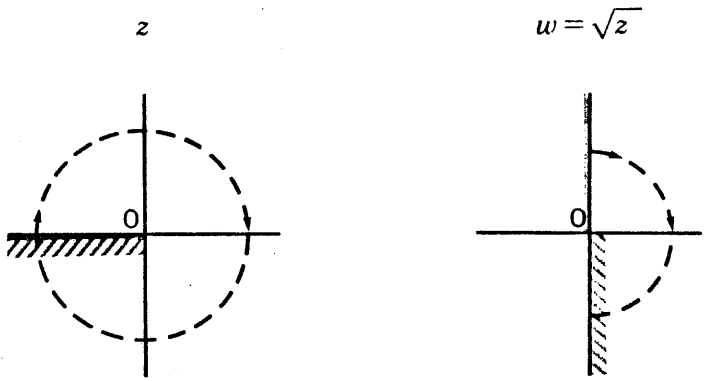
Considérons par exemple, les $n$ièmes racines d'un nombre complexe z. Représentons z sous forme polaire : $z=re^{i(\theta+2k\pi)}$ pour $-\pi<\theta\leq\pi$ et $k=0,\pm 1,\pm 2...$ Ensuite, si n est un entier positif,

$$z^{1/n}=r^{1/n}e^{i(\theta/n+2k\pi/n)}=r^{1/n}e^{i\alpha/n}e^{i2k\pi/n}$$

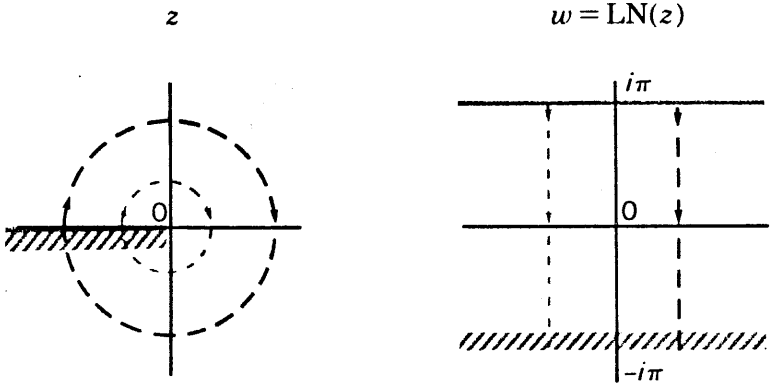
Seuls $k=0, 1, ..., n-1$ sont nécessaires puisque $e^{i2k\pi/n}$ répète ses valeurs par cycles de n. L'équation définit les $n$ièmes racines de z, et $r^{1/n}e^{i\theta/n}$ avec $-\pi<\theta\leq\pi$ est la branche principale de $z^{1/n}$. (Un programme de la page 78 calcule les $n$ièmes racines de z).

Les illustrations suivantes montrent les branches principales des relations inverses. Le graphique de gauche de chaque illustration représente le domaine tronqué de la fonction inverse. Le graphique de droite montre, dans les deux cas, la plage de la branche principale.

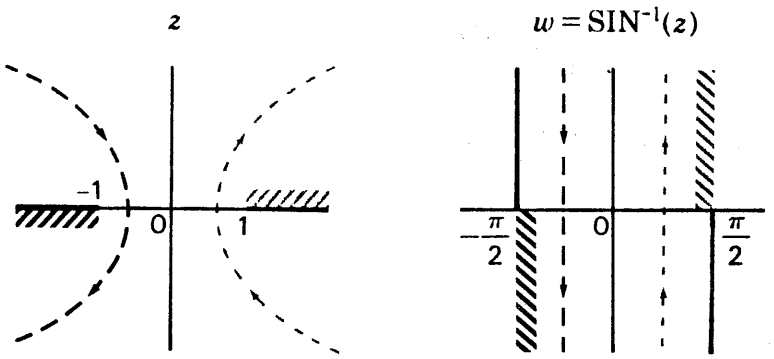
Pour certaines relations inverses, la définition de la branche principale ne fait pas consensus. Les branches principales utilisées par le HP-15C ont été soigneusement choisies. Tout d'abord, elles sont analytiques dans les régions où les arguments des fonctions inverses (évaluées en mode réel) sont définis. Autrement dit, le troncage est effectué là où la fonction inverse correspondante est indéfinie. Ensuite, la plupart des symétries importantes sont préservées. Par exemple, $\text{SIN}^{-1}(-z) = -\text{SIN}^{-1}(z)$ pour tout z.



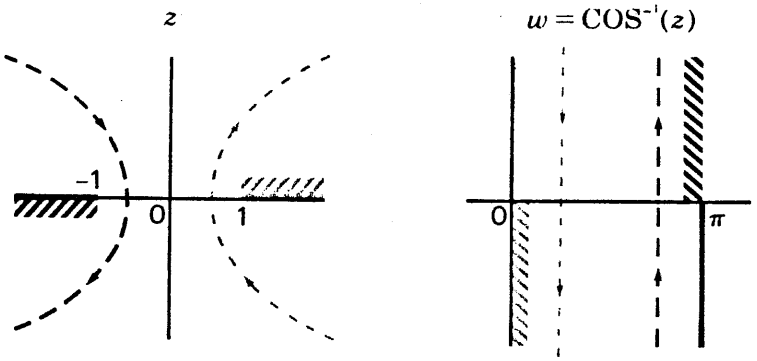
$$\sqrt{z} = \sqrt{r}\, e^{i\theta/2} \qquad \text{pour } -\pi < \theta \leqslant \pi$$



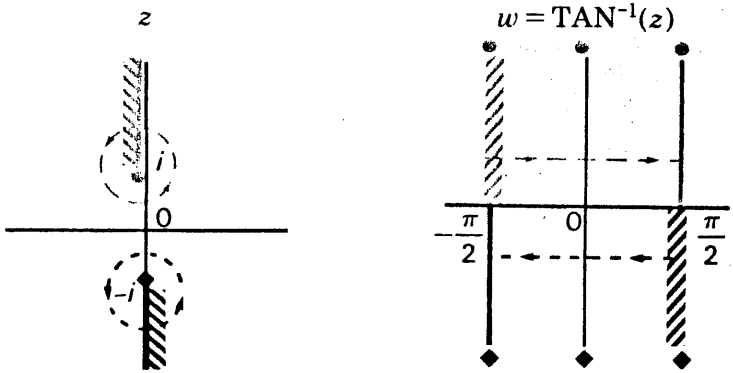
$$\text{LN}(z) = \ln r + i\theta \qquad \text{pour } -\pi < \theta \leqslant \pi$$



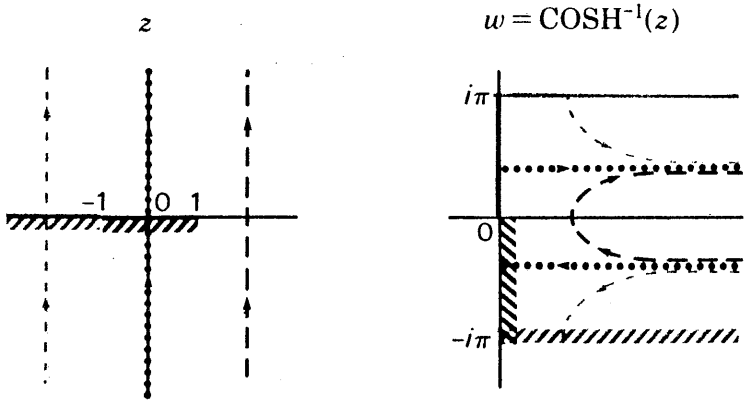
$$\sin^{-1}(z) = -i \ln[iz + (1 - z^2)^{1/2}]$$



$$\cos^{-1}(z) = -i \ln[z + (z^2 - 1)^{1/2}]$$



$$\tan^{-1}(z) = \frac{i}{2} \ln \frac{i + z}{i - z}$$



$$\cosh^{-1}(z) = \ln[z + (z^2 - 1)^{1/2}]$$

Les branches principales des quatre derniers graphes illustrés ci-dessus, sont obtenues à partir des équations correspondantes, mais n'utilisent pas nécessairement les branches principales de $\ln(z)$ et de $\sqrt{z}$.

Les fonctions inverses restantes peuvent être déterminées à partir des illustrations précédentes et des équations suivantes :

$$\mathrm{LOG}(z) = \mathrm{LN}(z)/\mathrm{LN}(10)$$

$$\mathrm{SINH}^{-1}(z) = -i\mathrm{SIN}^{-1}(iz)$$

$$\mathrm{TANH}^{-1}(z) = -i\mathrm{TAN}^{-1}(iz)$$

$$w^z = e^{z\mathrm{LN}(w)}$$

Pour déterminer *toutes* les valeurs d'une relation inverse, utilisez les expressions suivantes pour dériver ces valeurs à partir de la valeur principale calculée par le HP-15C. Dans ces expressions, $k = 0, \pm 1, \pm 2, \ldots$

$$z^{1/2} = \pm\sqrt{z}$$

$$\ln(z) = \mathrm{LN}(z) + i2k\pi$$

$$\sin^{-1}(z) = (-1)^k\mathrm{SIN}^{-1}(z) + k\pi$$

$$\cos^{-1}(z) = \pm\,\mathrm{COS}^{-1}(z) + 2k\pi$$

$$\tan^{-1}(z) = \mathrm{TAN}^{-1}(z) + k\pi$$

$$\sinh^{-1}(z) = (-1)^k\mathrm{SINH}^{-1}(z) + ik\pi$$

$$\cosh^{-1}(z) = \pm\,\mathrm{COSH}^{-1}(z) + i2k\pi$$

$$\tanh^{-1}(z) = \mathrm{TANH}^{-1}(z) + ik\pi$$

$$w^z = w^z e^{i2\pi kz}$$

Utilisation de [SOLVE] et de [∫] en mode complexe

Les fonctions [SOLVE] et [∫] utilisent des algorithmes qui échantillonnent votre fonction à des valeurs de l'axe des réels. En mode complexe, les fonctions [SOLVE] et [∫] ne fonctionnent qu'avec la pile réelle, même si le sous-programme de votre fonction est susceptible de comporter plusieurs calculs sur nombres complexes.

Par exemple, [SOLVE] ne va pas rechercher les racines d'une fonction complexe, mais va échantillonner la fonction sur l'axe des réels et rechercher le zéro de la partie réelle de la fonction. De la même façon, [∫] calcule l'intégrale de la partie réelle de la fonction sur un intervalle de l'axe des réels. Ces opérations sont utiles dans de nombreuses applications comme le calcul d'intégrales de contour et de potentiels complexes. (Reportez-vous au paragraphe "Applications" à la fin de ce chapitre.

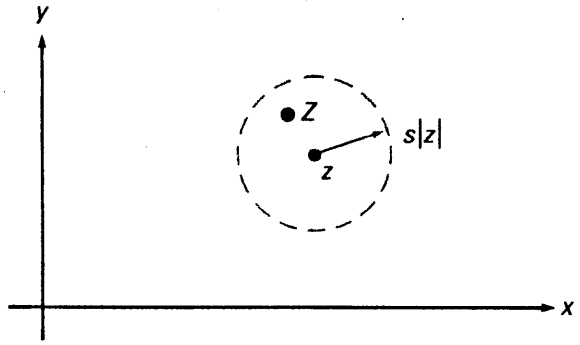
Précision en mode complexe

Les nombres complexes ayant à la fois des composantes réelles et des composantes imaginaires, la précision des calculs en mode complexe prend une autre dimension que celle des calculs en mode réel.

Avec des nombres *réels*, une approximation X est proche de x si la différence relative $E(X,x) = |(X - x)/x|$ est petite. Ceci est lié directement au nombre de chiffres significatifs exacts de l'approximation X. Autrement dit, si $E(X,x) < 5\times10^{-n}$, il y a au moins n chiffres significatifs. Pour les nombres *complexes*, définissez $E(Z,z) = |(Z - z)/z|$. Cependant ceci *n'est pas* directement lié au nombre de chiffres exacts dans chaque *composante* de Z.

Par exemple, si $E(X,x)$ et $E(Y,y)$ sont toutes deux petites, $E(Z,z)$ doit être également petite pour $z = x + iy$. Autrement dit, si $E(X,x) < s$ et $E(Y,y) < s$, alors $E(Z,z) < s$. Mais si nous considérons $z = 10^{10} + i$ et $Z = 10^{10}$, la composante imaginaire de Z est loin d'être précise et pourtant $E(Z,z) < 10^{-10}$. Même si les composantes imaginaires de z et de Z sont absolument différentes, z et Z peuvent être extrêmement proches.

Il existe une interprétation géométrique simple de l'erreur relative en mode complexe. Toute approximation Z de z satisfait $E(Z,z) < s$ (où s est un nombre réel positif) si et seulement si Z se trouve dans le cercle de rayon $s\,|z|$ centré en z dans le plan complexe.



Pour obtenir des approximations à *composantes* précises, il ne faut pas se contenter d'erreurs relatives suffisamment petites. Par exemple, dans le problème suivant, les calculs sont effectués avec quatre chiffres significatifs. Ce problème illustre les limites imposées par une précision finie dans un calcul complexe.

$$z_1 = Z_1 = 37.1 + 37.3i$$
$$z_2 = Z_2 = 37.5 + 37.3i$$

et

$$\begin{aligned} &Z_1 \times Z_2 \\ &\quad = 37.10 \times 37.50 - 37.30 \times 37.30) + i(37.10 \times 37.30 + 37.30 \times 37.50) \\ &\quad = (1391 - 1391) + i(1384. + 1399.) \\ &\quad = 0 + i(2783.) \end{aligned}$$

$z_1 z_2 = -0.04 + 2782{,}58i$ est la vraie valeur. Même si Z_1 et Z_2 n'ont pas d'erreur, la partie réelle de leur produit en quatre chiffres n'a pas de décimales significatives correctes, bien que l'erreur relative du produit complexe soit inférieure à 2×10^{-4}.

Cet exemple illustre que la multiplication en mode complexe ne propage pas ses erreurs en fonction de ses composantes. Mais même si la multiplication de nombres complexes a pour résultat des composantes exactes, les erreurs d'arrondi d'un calcul en chaîne risque de produire rapidement des composantes sans précision. D'un autre côté, l'erreur relative (correspondant à la précision du calcul), grossit très lentement.

Par exemple, avec la précision précédente de quatre chiffres:

$$z_1 = (1 + 1/3000) + i$$
$$Z_1 = 1.003 + i$$
$$z_2 = Z_2 = 1 + i$$

alors

$$\begin{aligned} Z_1 \times Z_2 &= (1.003 + i) \times (1 + i) \\ &= 0.003 + 2.003i \\ &= 3.000 \times 10^{-3} + 2.003i \end{aligned}$$

La valeur *correcte* à quatre chiffres est $3.333 \times 10^{-3} + 2.003i$. Dans cet exemple, Z_1 et Z_2 sont précis dans chacune de leurs composantes et le calcul est exact. Mais le produit est imprécis: la composante réelle n'a qu'un seul chiffre significatif. Une erreur d'arrondi résulte en une composante imprécise bien que l'erreur complexe relative du produit reste petite.

Pour le HP-15C, les résultats d'une opération complexe sont conçus pour être précis parce que l'erreur complexe relative $E(Z, z)$ reste petite. Généralement, $E(Z, z) < 6 \times 10^{-10}$.

Comme nous l'avons vu précédemment, cette erreur relative petite ne garantit pas 10 chiffres précis dans chaque composante. Parce que l'erreur est relative à la grandeur $|z|$ et que celle-ci n'est pas très différente de la valeur de la plus grande composante de z, la composante la plus petite peut avoir moins de chiffres précis. Il existe une méthode rapide pour voir quels chiffres sont généralement précis. Exprimez chaque composante en utilisant l'exposant le plus grand. Sous cette forme, les 10 premiers chiffres environ de chaque composante sont précis. Par exemple, si

$$Z = 1.234567890 \times 10^{-10} + i\,(2.222222222 \times 10^{-3}),$$

mettez Z sous la forme:

$$0.0000001234567890 \times 10^{-3} + i\,(2.222222222 \times 10^{-3}).$$

Les chiffres précis sont:

$$0.000000123 \times 10^{-3} + i\,(2.222222222 \times 10^{-3}).$$

Applications

Grâce à son mode complexe, le HP-15C vous permet de résoudre des problèmes sortant du domaine des nombres réels. Dans les pages suivantes, plusieurs programmes illustrent l'utilité des calculs sur les nombres complexes avec le HP-15C.

Stockage et rappel de nombres complexes à l'aide d'une matrice

Ce programme utilise la pile et la matrice C pour stocker et rappeler des nombres complexes. Il présente les caractéristiques suivantes :

- Si vous spécifiez un index supérieur aux dimensions de la matrice, le calculateur affiche **Erreur 3** et la pile est prête pour une nouvelle tentative.
- Si le calculateur n'est pas en mode complexe, le programme valide le mode complexe et la partie imaginaire du nombre est mise à zéro.
- Lorsque vous stockez un nombre complexe, l'index est perdu, la pile descend et le registre T est copié dans le registre Z.
- Le programme de stockage utilise la touche [D] (au-dessus de la touche [STO]). Le programme de rappel utilise la touche [E] (au-dessus de la touche [RCL]).

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	**000–**	
[f] [LBL] [D]	**001–42,21,14**	Programme de stockage
[f] [MATRIX] 1	**002–42,16, 1**	$R_0 = R_1 = 1$.
[STO] 0	**003– 44 0**	$R_0 = k$.
[R↓]	**004– 33**	
0	**005– 0**	Introduit 0 dans les registres X réels et imaginaires.
[+]	**006– 40**	Fait descendre la pile avec $a + ib$ dans le registre X.
[f] [USER] [STO] [C]	**707u 44 13**	Stocke a et incrémente les indices (mode USER).
[f] [USER]		
[f] [Re≷Im]	**008– 42 30**	

Appuyez sur	Affichage	
[STO] [C]	009– 44 13	Stocke b (pas en mode USER ici).
[f] [Re≷Im]	010– 42 30	Restaure $a + ib$ dans les registres X.
[g] [RTN]	011– 43 32	
[f] [LBL] [E]	012–42,21,15	Programme de rappel.
[STO] 0	013– 44 0	$R_0 = k$.
[g] [CLx]	014– 43 35	Invalide la pile.
2	015– 2	
[STO] 1	016– 44 1	$R_1 = 2$.
[R↓]	017– 33	
0	018– 0	
[+]	019– 40	Prépare la pile à une nouvelle tentative en cas de **Erreur 3.**
[RCL] [C]	020– 45 13	Rappelle b (partie imaginaire).
[f] [Re≷Im]	021– 42 30	
[f] [DSE] 1	022–42, 5, 1	Décrémente R_1 à 1.
[g] [CLx]	023– 43 35	Invalide la pile et efface les registres X réels.
[RCL] [C]	024– 45 13	Rappelle a (partie réelle).
[g] [RTN]	025– 43 32	

Exemple: stockez $2 + 3i$ et $7 + 4i$ dans les éléments 1 et 2 en utilisant le programme précédent. Rappelez-les puis ajoutez-les. Dimensionnez la matrice C à 5×2 pour qu'elle puisse contenir jusqu'à 5 nombres complexes.

Après avoir introduit le programme précédent,

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
5 [ENTER] 2	2	Spécifie 5 rangs et 2 colonnes.
[f] [DIM] [C]	2.0000	Dimensionne la matrice C.
2 [ENTER] 3 [f] [I]	2.0000	Introduit $2 + 3i$.
1 [f] [D]	2.0000	Stocke le nombre dans C en utilisant l'index 1.

Appuyer sur	Affichage	
7 [ENTER] 4 [f] [I]	7.0000	Introduit $7 + 4i$.
2 [f] [D]	7.0000	Stocke le nombre dans C en utilisant l'index 2.
1 [f] [E]	2.0000	Rappelle le premier nombre.
2 [f] [E]	7.0000	Rappelle le deuxième nombre.
[+]	9.0000	Partie réelle de la somme.
[f] [Re≷Im]	7.0000	Partie imaginaire de la somme.

Calcul des *n*ièmes racines d'un nombre complexe

Ce programme calcule les *n*ièmes racines d'un nombre complexe. Ces racines sont z_k pour $k = 0, 1, 2, ..., n-1$. Vous pouvez aussi utiliser le programme pour calculer $z^{1/r}$, où r n'est pas nécessairement entier. Le programme fonctionne de la même façon sauf qu'il peut y avoir un nombre infini de racines z_k pour $k = 0, \pm 1, \pm 2, ...$

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] [A]	001–42,21,11	
[x≷y]	002– 34	Place n dans le registre X, z dans les registres Y.
[1/x]	003– 15	Calcule $1/n$.
g [LST x]	004– 43 36	Extrait n.
[R↓]	005– 33	
[g] [SF] 8	006–43, 4, 8	Active le mode complexe.
[y^x]	007– 14	Calcule $z^{1/n}$.
[STO] 2	008– 44 2	Stocke la partie réelle de z_0 dans R_2.
[f] [Re≷Im]	009– 42 30	
[STO] 3	010– 44 3	Stocke la partie imaginaire de z_0 dans R_3.
3	011– 3	
6	012– 6	
0	013– 0	
[g] [R↑]	014– 43 33	
[÷]	015– 10	Calcule $360/n$.
[STO] 4	016– 44 4	Stocke $360/n$ dans R_4.
0	017– 0	
[STO] [I]	018– 44 25	Stocke 0 dans le registre d'index.

Appuyez sur	Affichage	
f LBL 0	019–42,21, 0	
RCL 4	020– 45 4	Rappelle 360/n.
RCL × I	021–45,20,25	Calcule 360 k/n en utilisant le registre d'index.
f Re≷Im	022– 42 30	
g CLx	023– 43 35	
1	024– 1	Place 1 + i (k 360/n) dans le registre X.
g DEG	025– 43 7	Mode degrés.
f →R	026– 42 1	Calcule $e^{ik360/n}$.
RCL 2	027– 45 2	Rappelle la partie réelle de z_0.
RCL 3	028– 45 3	Rappelle la partie imaginaire de z_0.
f I	029– 42 25	Reconstitue z_0.
×	030– 20	Calcule $z_0 e^{ik360/n}$, racine numéro k.
RCL I	031– 45 25	Rappelle le nombre k.
x≷y	032– 34	Place z_k dans les registres X et k dans le registre Y.
1	033– 1	
STO + I	034–44,40,25	Incrémente le nombre k dans le registre d'index.
R↓	035– 33	Restaure z_k et k dans les registres X et Y.
R/S	036– 31	Arrête l'exécution.
GTO 0	037– 22 0	Lance le calcul de la racine suivante (branchement).

Labels utilisés : A et 0.

Registres utilisés : R_2, R_3, R_4 et registre d'index.

Pour utiliser ce programme :

1. Introduire l'ordre n dans le registre Y et le nombre complexe z dans les registres X.
2. Appuyez sur f A pour calculer la racine principale, z_0, qui est placée dans les registres X (réel et imaginaire). Appuyez sur f (i) en maintenant ces touches enfoncées pour visualiser la partie imaginaire.

3. Pour calculer des racines z_k de numéro supérieur :

   - Appuyez sur [R/S] pour calculer chacune des racines successives. Chaque racine z_k est placée dans les registres X complexes et son numéro k est placé dans le registre Y. Entre ces calculs de racines, vous pouvez effectuer d'autres calculs sans affecter le déroulement du programme (à condition que R_2, R_3, R_4 et le registre d'index ne soient pas modifiés).
   - Stockez le numéro k de la racine dans le registre d'index (en utilisant [STO] [I], puis [R/S] pour calculer z_k). La racine complexe et son numéro sont placés respectivement dans les registres X et Y. (En appuyant à nouveau sur [R/S], vous pouvez continuer à calculer des racines de rang supérieur.)

Exemple : Utilisez le programme précédent pour calculer $(1)^{1/100}$. Calculez z_0, z_1 et z_{50} pour cette expression.

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
100 [ENTER] 1	1	Introduit $n = 100$ et $z = 1$ (purement réel).
[f] [A]	1.0000	Calcule z_0 (partie réelle).
[f] [(i)] (maintenu)	0.0000	Partie imaginaire de z_0.
[R/S]	0.9980	Calcule z_1 (partie réelle).
[f] [(i)] (maintenu)	0.0628	Partie imaginaire de z_1.
50 [STO] [I]	50.0000	Stocke le numéro de la racine dans le registre d'index.
[R/S]	−1.0000	Calcule z_{50} (partie réelle).
[f] [(i)] (hold)	0.0000	Partie imaginaire de z_{50}.

Résolution d'une équation pour ses racines complexes

Une méthode classique de résolution numérique de l'équation $f(z) = 0$ est l'itération de Newton. Cette méthode commence par une approximation z_0 d'une racine et calcule répétitivement :

$$z_{k+1} = z_k - f(z_k)/f'(z_k)$$

jusqu'à ce que z_k converge.

L'exemple suivant montre comment [SOLVE] peut être utilisée avec l'itération de Newton pour estimer des racines complexes.

(Une technique différente, n'utilisant pas le mode complexe, est indiquée page 16.).

Exemple : La réponse d'un système contrôlé automatiquement aux petites perturbations transitoires a été modélisée par l'équation différentielle comportant un terme de retard :

$$\frac{d}{dt} w(t) + 9\, w(t) + 8\, w(t-1) = 0.$$

Dans quelle mesure ce système est-il stable ? Autrement dit, avec quelle rapidité les solutions de cette équation décroissent-elles ?

Toute solution $w(t)$ peut être exprimée sous la forme de la somme suivante :

$$w(t) = \sum_k c(z) e^{zt}$$

où les coefficients constants $c(z)$ sont choisis pour chaque racine z de l'équation caractéristique associée à l'équation différentielle comportant un terme de retard :

$$z + 9 + 8e^{-z} = 0$$

Chaque racine $z = x + iy$ donne à $w(t)$ une composante $e^{zt} = e^{xt}(\cos(yt) + i \sin(yt))$ dont le taux de décroissance est plus rapide lorsque x (partie réelle de z) est plus négatif. La réponse à ce problème entraîne donc le calcul de *toutes* les racines z de l'équation caractéristique. Or, cette équation ayant un nombre infini de racines, dont aucune n'est réelle, le calcul de toutes ces racines risque d'être une tâche extrêmement longue.

Cependant, on sait que les racines z peuvent être approchées pour de grands entiers n par $z \approx A(n) = -1n((2n+1/2)\,\pi/8) \pm i(2n+1/2)\,\pi$ pour $n = 0, 1, 2...$ Plus n est grand, meilleure est l'approximation. C'est pourquoi, vous ne devez calculer que les quelques racines mal approchées par $A(n)$, c'est-à-dire les racines pour lesquelles $|z|$ n'est pas très grande.

En cas d'utilisation de l'itération de Newton, que doit être $f(z)$ pour ce problème ? La fonction évidente $f(z) = z + 9 + 8\, e^{-z}$ n'est pas un bon choix parce que l'exponentielle croît rapidement pour de grandes valeurs négatives de $\mathrm{Re}(z)$. Ceci ralentirait considérablement la convergence sauf si la première estimation tentée se trouvait très proche d'une racine. En outre, cette fonction $f(z)$ s'annule une infinité de fois si bien qu'il est difficile de déterminer quand toutes les racines désirées ont été calculées. Par contre, en ré-écrivant cette équation sous la forme :

$$e^z = -8/(z+9)$$

et en utilisant les logarithmes, vous obtiendrez une équation équivalente.

$$z = \ln(-8/(z+9)) \pm i2n\pi \text{ pour } n = 0, 1, 2,...$$

Cette équation n'a que deux racines complexes z conjuguées pour chaque entier n. Utilisez donc la fonction équivalente

$$f(z) = z - \ln(-8/(z+9)) \pm i2n\pi \text{ pour } n = 0, 1, 2,...$$

et appliquez l'itération de Newton

$$z_{k+1} = z_k - (z_k - \ln(-8/(z_k+9)) \pm i2n\pi)/(1 + 1/(z_k+9))$$

Comme première estimation d'essai, choisissez z_0 égale à $A(n)$, l'approximation donnée précédemment. Un peu de manipulation algébrique utilisant le fait que $\ln(\pm i) = \pm i\pi/2$, mène à la formule suivante :

$$z_{k+1} = A(n) + ((z_k - A(n)) + (z_k + 9)\ln(i\text{Im}(A(n))/(z_k+9)))/(z_k+10)$$

Dans le programme ci-dessous, $\text{Re}(A(n))$ est stocké dans R_0 et $\text{Im}(A(n))$ dans R_1. Remarquez que seule l'une des deux racines conjuguées est calculée pour chaque n.

Appuyez sur	Affichage		Commentaire
g P/R			Mode programme.
f CLEAR PRGM	000-		
f LBL A	001-	42,21,11	Programme pour $A(n)$.
g CF 8	002-	43, 5, 8	Spécifie le mode réel.
ENTER	003-	36	
+	004-	40	
.	005-	48	
5	006-	5	
+	007-	40	
g π	008-	43 26	
×	009-	20	Calcule $(2n + 1/2)\pi$.
ENTER	010-	36	
STO 1	011-	44 1	
8	012-	8	
÷	013-	10	
g LN	014-	43 12	
CHS	015-	16	Calcule $-\ln((2n + 1/2)\pi/8)$.
STO 0	016-	44 0	
x≷y	017-	34	
f I	018-	42 25	Reconstitue le nombre complexe $A(n)$.

Appuyez sur	Affichage	
g RTN	019- 43 32	
f LBL B	020-42,21,12	Programme de calcul de z_{k+1}
ENTER	021- 36	
ENTER	022- 36	
RCL 1	023- 45 1	
f Re≷Im	024- 42 30	Crée iIm($A(n)$.
x≷y	025- 34	
9	026- 9	
+	027- 40	
÷	028- 10	
g LSTx	029- 43 36	
x≷y	030- 34	
g LN	031- 43 12	
×	032- 20	
x≷y	033- 34	
RCL 1	034- 45 1	
f Re≷Im	035- 42 30	
RCL + 0	036-45,40, 0	
−	037- 30	
g LSTx	038- 43,36	
R↓	039- 33	
+	040- 40	
x≷y	041- 34	
1	042- 1	
0	043- 0	
+	044- 40	
÷	045- 10	
+	046- 40	
g RTN	047- 43 32	
f LBL C	048-42,21,13	Programme de calcul pour le résidu $\|e^z + 8/(z+9)\|$.
ENTER	049- 36	
e^x	050- 12	
9	051- 9	
g LSTx	052- 43 36	
+	053- 40	
8	054- 8	
x≷y	055- 34	
÷	056- 10	
+	057- 40	

Appuyez sur	Affichage		
g ABS	058-	43 16	Calcule $\lvert e^z + 8/(z+9)\rvert$.
g RTN	059-	43 32	

Labels utilisés : A, B et C.

Registres utilisés : R_0 et R_1.

Exécutez maintenant le programme. Pour chaque racine, appuyez sur B jusqu'à ce que la partie réelle affichée ne change plus. (Vous pourriez aussi bien vérifier que la partie imaginaire ne change plus.)

Appuyez sur	Affichage		
g P/R			Mode calcul.
f USER			Active le mode USER.
0 A	1.6279		Affiche $\mathrm{Re}(A(0)) = \mathrm{Re}\,(z_0)$.
B	-0.1487		$\mathrm{Re}(z_1)$.
B	-0.1497		$\mathrm{Re}(z_2)$.
B	-0.1497		$\mathrm{Re}(z)$.
f (i) (hold)	2.8319		$\mathrm{Im}(z)$.
C	1.0000	-10	Calcule le résidu.
x≷y	-0.1497		Restaure z dans le registre X.

En répétant la même procédure pour $n = 1$ à 5, vous obtiendrez les résultats ci-dessous (seule figure une des deux racines).

n	$A(n)$	Racine z_k	Résiduelle
0	$1.6279 + i1.5708$	$-0.1497 + i2.8319$	1×10^{-10}
1	$0.0184 + i7.8540$	$-0.4198 + i8.6361$	6×10^{-10}
2	$-0.5694 + i14.1372$	$-0.7430 + i14.6504$	2×10^{-9}
3	$-0.9371 + i20.4204$	$-1.0236 + i20.7868$	5×10^{-10}
4	$-1.2054 + i26.7035$	$-1.2553 + i26.9830$	9×10^{-10}
5	$-1.4167 + i32.9867$	$-1.4486 + i33.2103$	2×10^{-9}

Lorsque n croît, la première estimation A(n) s'approche de la racine z désirée. (Dès que vous avez terminé, appuyez sur [f] [USER] pour invalider le mode USER).

Puisque toutes les racines ont une partie réelle négative, le système est stable, mais la plage de stabilité (la plus petite en grandeur parmi les différentes parties réelles, c'est-à-dire : — 0.1497) est suffisamment petite pour être surveillée attentivement lorsque le système doit supporter beaucoup de bruits de ligne.

Intégrales de contour

Vous pouvez utiliser [∫] pour évaluer l'intégrale de contour $\int_C f(z)dz$, où C est une courbe dans le plan complexe.

Tout d'abord, paramétrez la courbe C par $z(t) = x(t) + iy(t)$ pour $t_1 \leq t \leq t_2$. Posez $G(t) = f(z(t))z'(t)$. Puis :

$$\int_C f(z)dz = \int_{t_1}^{t_2} G(t)dt$$

$$= \int_{t_1}^{t_2} \mathrm{Re}(G(t))dt + i\int_{t_1}^{t_2} \mathrm{Im}(G(t))dt.$$

Ces intégrales sont justement celles que [∫] évalue en mode complexe. Puisque $G(t)$ est une fonction complexe d'une variable réelle t, [∫] va échantillonner $G(t)$ sur l'intervalle $t_1 \leq t \leq t_2$ et intégrer Re($G(t)$) – résultat renvoyé dans le registre X réel par votre fonction. Pour la partie imaginaire, intégrez une fonction qui évalue $G(t)$ et utilise [Re≷Im] pour placer Im($G(t)$) dans le registre X réel.

Le programme général figurant ci-dessous évalue l'intégrale complexe

$$I = \int_a^b f(z)dz$$

suivant une ligne droite allant de a à b, où a et b sont des nombres complexes. Le programme suppose que le sous-programme de calcul de votre fonction complexe a le label "B", qu'il évalue la fonction complexe $f(z)$ et que les limites d'intégration a et b sont respectivement dans les registres Y et X. Les composantes complexes de l'intégrale I et l'incertitude ΔI sont renvoyées dans les registres X et Y.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	

Appuyez sur	Affichage	
f LBL A	001-42,21,11	
x≷y	002- 34	
−	003- 30	Calcule $b - a$.
STO 4	004- 44 4	Stocke Re($b - a$) dans R_4.
f Re≷Im	005- 42 30	
STO 5	006- 44 5	Stocke Im($b - a$) dans R_5.
g LSTx	007- 43 36	Rappelle a.
STO 6	008- 44 6	Stocke Re(a) dans R_6.
f Re≷Im	009- 42 30	
STO 7	010- 44 7	Stocke Im(a) dans R_7.
0	011- 0	
ENTER	012- 36	
1	013- 1	
f ∫ 0	014-42,20, 0	Calcule Im(I) et Im(ΔI).
STO 2	015- 44 2	Stocke Im(I) dans R_2.
R↓	016- 33	
STO 3	017- 44 3	Stocke Im(ΔI) dans R_3.
R↓	018- 33	
f ∫ 1	019-42,20, 1	Calcule Re(I) et Re (ΔI).
RCL 2	020- 45 2	Rappelle Im(I).
f I	021- 42 25	Reconstitue I complexe.
x≷y	022- 34	
RCL 3	023- 45 3	Rappelle Im(ΔI).
f I	024- 42 25	Reconstitue ΔI complexe.
x≷y	025- 34	Restaure I dans le registre X.
g RTN	026- 43 32	
f LBL 0	027-42,21, 0	Sous-programme de calcul de Im ($f(z)z'(t)$).
GSB 1	028- 32 1	Calcule $f(z)z'(t)$.
f Re≷Im	029- 42 30	Échange les parties réelle et imaginaire.
g RTN	030- 43 32	
f LBL 1	031-42,21, 1	Sous-programme de calcul de $f(z)z'(t)$.

Appuyez sur	Affichage	
RCL 4	032– 45 4	
RCL 5	033– 45 5	
f I	034– 42 25	Reconstitue le nombre complexe $b - a$.
×	035– 20	Calcule $(b - a)t$.
RCL 6	036– 45 6	
RCL 7	037– 45 7	
f I	038– 42 25	Reconstitue le nombre complexe a.
+	039– 40	Calcule $a + (b - a)t$.
GSB B	040– 32 12	Calcule $f(a + (b - a)t)$.
RCL 4	041– 45 4	
RCL 5	042– 45 5	
f I	043– 42 25	Reconstitue le nombre complexe $z'(t) = b - a$.
×	044– 20	Calcule $f(z)z'(t)$.
g RTN	045– 43 32	

Labels utilisés : A, 0 et 1.

Registres utilisés : R_2, R_3, R_4, R_5, R_6 et R_7.

Pour utiliser ce programme :

1. Introduisez le sous-programme de calcul de votre fonction, avec le lab "B" en mémoire programme.
2. Appuyez sur 7 f DIM (i) pour réserver les registres R_0 à R_7. (Votr sous-programme peut nécessiter des registres supplémentaires.)
3. Définissez le format d'affichage pour $\int$.
4. Introduire les deux valeurs complexes définissant les extrémités de l droite le long de laquelle votre fonction sera intégrée. La limite inférieur doit être dans les registres Y, la limite supérieure dans les registres X.
5. Appuyez sur f A pour calculer l'intégrale complexe de la droite. L valeur de l'intégrale est dans les registres X ; la valeur de l'incertitud est dans les registres Y.

Comme deux intégrales sont évaluées, le programme va mettre plus long temps que pour une intégrale réelle, bien que le programme $\int$ n'ait pas à uti liser le même nombre de points d'échantillonnage pour les deux intégrales L'intégrale la plus facile utilisera moins de calculs que la plus difficile.

Exemple: Faites une approximation des intégrales

$$I_1 = \int_1^{\infty} \frac{\cos x}{x + 1/x}\,dx \qquad \text{et} \qquad I_2 = \int_1^{\infty} \frac{\sin x}{x + 1/x}\,dx.$$

Ces expressions décroissent très lentement lorsque x tend vers l'infini. Elles nécessitent donc un large intervalle d'intégration et un temps d'exécution assez long. Vous pouvez réduire la durée de ce calcul en faisant passer le contour d'intégration de l'axe des réels au plan des complexes. Selon la théorie des variables complexes, ces intégrales peuvent être combinées sous la forme:

$$I_1 + iI_2 = \int_1^{1+i\infty} \frac{e^{iz}}{z + 1/z}\,dz.$$

Cette expression, lorsqu'elle est évaluée le long de la droite de coordonnées $x = 1$ et $y \geq 0$, décroît rapidement lorsque y augmente, comme e^{-y}.

Pour utiliser le programme précédent pour le calcul des deux intégrales en même temps, écrivez un sous-programme évaluant:

$$f(z) = \frac{e^{iz}}{z + 1/z}.$$

Appuyez sur	Affichage	
[f] [LBL] [B]	046-42,21,12	
[1/x]	047- 15	
[g] [LSTx]	048- 43 36	
[+]	049- 40	Calcule $z + 1/z$.
[g] [LSTx]	050- 43 36	
1	051- 1	
[f] [Re≷Im]	052- 42 30	Reconstitue $0 + i$.
[×]	053- 20	
[e^x]	054- 12	Calcule e^{iz}.
[x≷y]	055- 34	
[÷]	056- 10	Calcule $f(z)$.
[g] [RTN]	057- 43 32	

Faites une approximation de l'intégrale complexe en intégrant la fonction de $1 + 0i$ à $1 + 6i$, en format d'affichage [SCI] 2 pour obtenir trois chiffres significatifs. (L'intégrale n'affecte pas les trois premiers chiffres au-delà de $1 + 6i$.)

Appuyez sur	Affichage		
[g] [P/R]			Mode calcul.
[f] [SCI] 2			Spécifie le format [SCI] 2.
1 [ENTER]	1.00	00	Introduit la première limite, $1 + 0i$, de l'intégration.
1 [ENTER] 6	6		
[f] [I]	1.00	00	Introduit la seconde limite, $1 + 6i$, de l'intégration.
[f] [A]	-3.24	-01	Calcule I et affiche $\mathrm{Re}(I) = I_1$ (au bout de 9 minutes environ).
[f] [(i)] (maintenue)	3.82	-01	Affiche $\mathrm{Im}(I) = I_2$.
[x y]	7.87	-04	Affiche $\mathrm{Re}(\Delta I) = \Delta I_1$.
[f] [(i)] (maintenue)	1.23	-03	Affiche $\mathrm{Im}(\Delta I) = \Delta I_2$.
[f] [FIX] 4	0.0008		

Ce résultat I est calculé beaucoup plus rapidement que si I_1 et I_2 étaient calculées directement le long de l'axe des réels.

Potentiels complexes

La projection est utile dans des applications associées à une fonction complexe potentielle. Les explications suivantes concernent un problème d'écoulement de fluide, mais il aurait pu aussi bien s'agir de problèmes d'électricité statique ou de flux de chaleur.

Considérons la fonction potentielle $P(z)$. L'équation $\mathrm{Im}(P(z)) = c$ définit une famille de courbes appelées les *lignes de courant* du flux. C'est-à-dire, pour toute valeur de c, toute les valeurs de z qui satisfont l'équation sont dans une ligne de flux correspondant à cette valeur de c. Pour calculer des points z_k sur cette ligne de courant, spécifiez des valeurs pour x_k et utilisez ensuite [SOLVE] pour trouver les valeurs correspondantes de y_k utilisant l'équation :

$$\mathrm{Im}(P(x_k + iy_k)) = c$$

Si les valeurs x_k ne sont pas trop écartées, vous pouvez utiliser y_{k-1} comme estimation initiale de y_k. De cette façon, vous pouvez travailler sur la ligne de courant et calculer les points complexes $z_k = x_k + iy_k$. En utilisant une procédure identique, vous pouvez définir les courbes équipotentielles données par $\mathrm{Re}(P(z)) = c$.

Le programme ci-dessous permet de calculer les valeurs de y_k à partir de valeurs de x_k régulièrement espacées. Vous devez prévoir un sous-programme labellé "B" qui place Im($P(z)$) dans le registre X réel. Le programme utilise les entrées suivantes : valeur h du pas, le nombre n de points sur l'axe des réels et $z_0 = x_0 + iy_0$, point initial de la ligne de courant. Vous devez introduire n, h et z_0 dans les registres Z, Y et X avant d'exécuter le programme.

Le programme calcule les valeurs de z_k et les stocke dans une matrice A sous la forme $a_{k1} = x_{k-1}$ et $a_{k2} = y_{k-1}$ pour $k = 1, 2, ..., n$.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000-	
[f] [LBL] [A]	001-42,21,11	
[R↓]	002- 33	
[STO] 4	003- 44 4	Stocke h dans R_4.
[R↓]	004- 33	
2	005- 2	
[f] [DIM] [A]	006-42,23,11	Dimensionne la matrice A à $n \times 2$.
[g] [CLx]	007- 43 35	
[STO] [MATRIX] [A]	008-44,16,11	Met tous les éléments de A à zéro.
[STO] [I]	009- 44 25	Stocke zéro dans le registre d'index.
[f] [MATRIX] 1	010-42,16, 1	Définit $R_0 = R_1 = 1$.
[g] [R↑]	011- 43 33	Rappelle z_0 dans les registres X.
[STO] 2	012- 44 2	Stocke x_0 dans R_2.
[f] [USER] [STO] [A] [f] [USER]	013u 44 11	Définit $a_{11} = x_0$.
[f] [Re≷Im]	014- 42 30	
[STO] 3	015- 44 3	Stocke y_0 dans R_3.
[f] [USER] [STO] [A] [f] [USER]	016u 44 11	Définit $a_{12} = y_0$.
[GTO] 1	017- 22 1	Branchement si la matrice A n'est pas pleine ($n > 1$).
[f] [LBL] 0	018-42,21, 0	
[RCL] [MATRIX] [A]	019-45,16,11	Rappelle le label de la matrice A.

Appuyez sur	Affichage	
[g] [RTN]	020– 43 32	
[f] [LBL] 1	021–42,21, 1	
[f] [Re≷Im]	022– 42 30	Restaure z_0.
[GSB] [B]	023– 32 12	Calcule Im($P(z_0)$) (ou Re($P(z_0)$) pour la courbe équipotentielle).
[STO] 5	024– 44 5	Stocke c dans R_5.
[f] [LBL] 2	025–42,21, 2	Boucle de recherche de y_k.
1	026– 1	
[STO] [+] [I]	027–44,40,25	Incrémente le compteur k dans le registre d'index.
[RCL] 4	028– 45 4	Rappelle h.
[RCL] [I]	029– 45 25	Rappelle le compteur k.
[×]	030– 20	Calcule kh.
[RCL] 2	031– 45 2	Rappelle x_0.
[+]	032– 40	Calcule $x_k = x_0 + kh$.
[STO] 6	033– 44 6	Stocke x_k dans R_6.
[RCL] 3	034– 45 3	Rappelle y_{k-1} de R_3.
[ENTER]	035– 36	Duplique y_{k-1} pour une seconde estimation.
[f] [SOLVE] 3	036–42,10, 3	Recherche y_k.
[GTO] 4	037– 22 4	Branchement à une racine y_k possible.
1	038– 1	Commence à réduire la valeur du pas.
[STO] [−] [I]	039–44,30,25	Décrémente le compteur k.
4	040– 4	
[STO] [÷] 4	041–44,10, 4	Réduit h d'un facteur 4.
[STO] [×] [I]	042–44,20,25	Multiplie le compteur par 4.
[GTO] 2	043– 22 2	Boucle arrière pour chercher y_k à nouveau.
[f] [LBL] 4	044–42,21, 4	Continue à chercher y_k.
[RCL] 6	045– 45 6	
[f] [PSE]	046– 42 31	Affiche x_k.
[f] [USER] [STO] [A] [f] [USER]	047u 44 11	Définit $a_{k+1,1} = x_k$.

Appuyez sur	Affichage		
[R↓]	048–	33	
[f] [PSE]	049–	42 31	Affiche y_k.
[STO] 3	050–	44 3	Stocke y_k dans R_3.
[f] [USER] [STO] [A] [f] [USER]	051u	44 11	Définit $a_{k+1,2} - y_k$.
[GTO] 2	052–	22 2	Branchement pour $k+1 < n$ (A n'est pas pleine).
[GTO] 0	053–	22 0	Branchement pour $k+1 = n$ (A est pleine).
[f] [LBL] 3	054–	42,21, 3	Sous-programme de la fonction pour [SOLVE].
[RCL] 6	055–	45 6	Rappelle x_k.
[x≷y]	056–	34	Restaure l'estimation en cours pour y_k.
[f] [I]	057–	42 25	Crée l'estimation $z_k = x_k + iy_k$.
[GSB] [B]	058–	32 12	Calcule $\mathrm{Im}(P(z_k))$ (ou $\mathrm{Re}(P(z_k))$ pour des courbes équipotentielles).
[RCL] 5	059–	45 5	Rappelle c.
[−]	060–	30	Calcule $\mathrm{Im}(P(z_k)) - c$.
[g] [RTN]	061–	43 32	

Labels utilisés: A, B, 0, 1, 2, 3 et 4.

Registres utilisés: R_0, R_1, $R_2(x_0)$, $R_3(y_0)$, $R_4(h)$, $R_5(c)$, $R_6(x_k)$ et registre d'index (k).

Matrice utilisée: **A**.

Une caractéristique spéciale de ce programme est que si une valeur x_k se trouve au-delà du domaine de la ligne de courant (si bien qu'il n'y a pas de racine à trouver pour [SOLVE]), la valeur du pas est diminuée pour que x_k approche de la limite où la ligne de courant revient. Cette caractéristique est utile pour la détermination de la nature de la ligne de courant lorsque y_k n'est pas une fonction monadique de x_k. Si h est suffisamment petite, les valeurs de z_k se trouveront sur une branche de la ligne de courant et approcheront la limite. (Le deuxième exemple ci-dessous illustre cette particularité.)

Pour utiliser ce programme :

1. Introduisez votre sous-programme sous le label "B" dans la mémoire programme. Il doit mettre $\mathrm{Im}(P(z))$ dans le registre X réel si vous calculez des lignes de courant ou bien $\mathrm{Re}(P(z))$ si vous calculez des courbes équipotentielles.
2. Appuyez sur 6 [f] [DIM] [(i)] pour réserver les registres R_0 à R_6 (et le registre d'index). (Votre sous-programme peut nécessiter des registres supplémentaires.)
3. Introduisez les valeurs de n et de h dans les registres X et Y en appuyant sur n [ENTER] h [ENTER].
4. Introduisez la valeur complexe de $z_0 = x_0 + iy_0$ dans les registres X en appuyant sur x_0 [ENTER] y_0 [f] [I].
5. Appuyez sur [f] [A] pour afficher les valeurs successives de x_k et y_k pour $k = 1, \ldots, n$ et finalement le label de la matrice A. Les valeurs pour $k = 0, \ldots, n$ sont stockées dans la matrice A.
6. Si vous désirez, rappelez des valeurs de la matrice A.

Exemple : Calculez la ligne de courant du potentiel $P(z) = 1/z + z$ passant par le point $z = -2 + 0,1i$.

Tout d'abord, introduisez le sous-programme "B" pour calculer $\mathrm{Im}(P(z))$.

Appuyez sur	Affichage		
[f] [LBL] [B]	062-42,21,12		
[ENTER]	063-	36	Duplique z.
[1/x]	064-	15	
[+]	065-	40	Calcule $1/z + z$.
[f] [Re≷Im]	066-	42 30	Place $\mathrm{Im}(P(z))$ dans le registre X.
[g] [RTN]	067-	43 32	

Détermine la ligne de courant en utilisant $z_0 = -2 + 0.1i$, valeur du pas : $h = 0.5$ et nombre de points : $n = 9$.

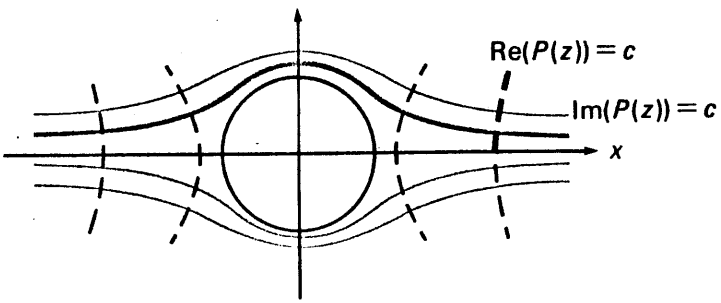
Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
9 [ENTER]	9.0000	Introduit n.
.5 [ENTER]	0.5000	Introduit h.

Appuyez sur	Affichage			
2 [CHS] [ENTER]	-2.0000			
.1 [f] [I]	-2.0000			Introduit z_0.
[f] [A]	-1.5000			x_1.
	0.1343			y_1.
	⋮			⋮
	2.0000			x_9.
	0.1000			y_9.
	A	9	2	Label de la matrice A.
[g] [CF] 8	A	9	2	Désactive le mode complexe.

La matrice A contient les valeurs suivantes de x_k et de y_k.

x_k	y_k
-2.0	0.1000
-1.5	0.1343
-1.0	0.4484
-0.5	0.9161
0.0	1.0382
0.5	0.9161
1.0	0.4484
1.5	0.1343
2.0	0.1000

Les courbes équiponentielles de courant et de vitesse sont illustrées ci-dessous. La ligne de courant dérivée est représentée par la courbe en gras.



Exemple : Pour le même potentiel que celui de l'exemple précédent, $P(z) = 1/z + z$, calculez la courbe équipotentielle de vitesse partant vers la gauche à partir du point $z = 2 + i$.

Tout d'abord, modifiez le sous-programme "B" pour qu'il donne $\mathrm{Re}(P(z))$ (en enlevant l'instruction [Re ⇄ Im] de "B"). Essayez $n = 6$ et $h = -0.5$. (Remarquez que h est négative, ce qui spécifie que x_k sera situé à gauche de x_0).

Bien que les séquences des touches ne soient pas détaillées ici, les résultats calculés et stockés dans la matrice A sont donnés ci-dessous.

x_k	y_k
2.0000	1.0000
1.8750	0.2362
1.8672	0.1342
1.8652	0.0941
1.8647	0.0811
1.8646	0.0775

Les résultats montrent la nature de la branche supérieure de la courbe (courbe en pointillés gras du graphe précédent). Notez que la valeur h du pas est automatiquement diminuée pour suivre la courbe – pour éviter un arrêt en cas d'erreur – lorsqu'aucune valeur y n'est trouvée pour $x < 1.86$.

Chapitre 4

Opérations matricielles

L'algèbre matricielle est un outil très puissant. Elle permet de formuler et de résoudre de nombreux problèmes complexes, simplifiant des calculs compliqués. Ce chapitre traite des opérations matricielles effectuées par le HP-15C ainsi que l'utilisation du calcul matriciel dans diverses applications.

Il contient aussi un résumé de certains résultats de l'algèbre linéaire mais ce n'est qu'un rappel, il existe de nombreux ouvrages de référence.

Décomposition en matrices triangulaires

Le HP-15C peut résoudre des systèmes d'équations linéaires, inverser des matrices et calculer des déterminants. Pour effectuer tous ces calculs, le HP-15C utilise une décomposition en matrices triangulaires.

Cette décomposition consiste à trouver deux matrices L et U telles que $\mathbf{A} = \mathbf{LU}$. **L** est une matrice triangulaire inférieure† dont les éléments de la diagonale sont égaux à 1 et dont les éléments situés sous la diagonale sont compris entre -1 et $+1$. **U** est une matrice triangulaire supérieure†. Par exemple :

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

† Une matrice triangulaire inférieure est une matrice dont tous les éléments situés au-dessus de la diagonale sont nuls. Une matrice triangulaire supérieure est une matrice dont les éléments situés au-dessous de la diagonale sont nuls.

Certaines matrices ne peuvent pas être décomposées ainsi. Par exemple,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \neq \mathbf{LU}$$

quelles que soient les matrices **L** et **U**. Cependant, après avoir effectué une permutation sur les rangs, on peut toujours trouver une décomposition. Il existe une matrice **P** telle que la matrice obtenue après permutation soit égale au produit **PA**. Après décomposition, on doit donc avoir **PA = LU**. Reprenons l'exemple précédent :

$$\mathbf{PA} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \mathbf{LU}.$$

La permutation des rangs peut aussi supprimer les erreurs d'arrondi qui risquent de se produire lors de la décomposition.

Pour effectuer la décomposition, le HP-15C utilise la méthode Doolittle avec une grande précision arithmétique. Le résultat de la décomposition est stocké sous la forme :

$$\begin{bmatrix} & \mathbf{U} \\ \mathbf{L} & \end{bmatrix}$$

Il est inutile de stocker les éléments de la diagonale de **L**, puisqu'ils sont tous égaux à 1. Les permutations sont aussi mises en mémoire dans cette matrice de manière codée et qui nous est invisible. La décomposition est indiquée dans le traitement et son label contient deux tirets à l'affichage.

Lors du calcul du déterminant ou de la résolution d'un système d'équations, la décomposition *LU* est automatiquement sauvegardée. Il est parfois utile de se servir de la forme décomposée de la matrice dans certains calculs ; il ne faut donc pas perdre l'information concernant la permutation : ne modifiez pas la matrice dans laquelle sont stockés les éléments de la décomposition.

Pour calculer le déterminant de la matrice **A**, le HP-15C utilise l'équation $\mathbf{A} = \mathbf{P}^{-1}\mathbf{LU}$ afin de pouvoir faire des permutations de rangs. Le déterminant est alors égal à $(-1)^r$ que multiplie le produit des éléments de la diagonale de **U** ; r représente le nombre de permutations. Le HP-15C calcule ce produit avec son signe, après décomposition de la matrice.

Il est beaucoup plus facile d'inverser une matrice triangulaire qu'une matrice quelconque. Donc pour inverser la matrice **A**, le calculateur utilise la relation :

$$\mathbf{A}^{-1} = (\mathbf{P}^{-1}\mathbf{LU})^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}$$

Il faut donc tout d'abord qu'il décompose la matrice **A**, qu'il inverse **L** et **U**, qu'il calcule le produit $\mathbf{U}^{-1}\mathbf{L}^{-1}$ puis qu'il échange les colonnes du résultat. Ces opérations s'effectuent sur la matrice résultat. Si **A** est déjà sous forme décomposée, la phase de décomposition est supprimée. Grâce à cette méthode, le HP-15C peut inverser une matrice sans utiliser de registre intermédiaire.

Résoudre un système d'équations de la forme $\mathbf{AX} = \mathbf{B}$ est beaucoup plus facile dans le cas où **A** est une matrice triangulaire que dans le cas général. En utilisant la relation $\mathbf{PA} = \mathbf{LU}$, le problème devient $\mathbf{LUX} = \mathbf{PB}$ pour **X**. Les lignes de la matrice **B** vont donc subir les mêmes permutations que celles de la matrice **A**. Le calculateur commence par résoudre l'équation $\mathbf{LY} = \mathbf{PB}$ pour **Y** (résolution en descendant), puis l'équation $\mathbf{UX} = \mathbf{Y}$ pour **X** (résolution en remontant). La décomposition est toujours sauvegardée afin de pouvoir changer **B** sans introduire à nouveau les coefficients du système.

La décomposition en matrices triangulaires est une étape très commode pour le calcul de déterminants, l'inversion de matrices ou la résolution de systèmes linéaires. Elle peut être aussi utilisée à la place de la matrice initiale dans d'autres calculs.

Matrices mal conditionnées et nombre de conditionnement

Afin de pouvoir évaluer les erreurs dans les calculs matriciels, il faut définir une distance entre deux matrices.

L'une des distances possibles entre les matrices **A** et **B** est la *norme* de leur différence notée $\|\mathbf{A}-\mathbf{B}\|$. Cette norme est aussi utilisée pour calculer le *nombre de conditionnement* d'une matrice qui indique l'erreur relative dans un calcul, comparée à l'erreur relative sur la matrice.

Le HP-15C offre 3 normes. La *norme Frobenius* d'une matrice **A** est notée $\|\mathbf{A}\|_F$; c'est la racine carrée de la somme des carrés des éléments de la matrice. Cette norme est l'analogue de la norme euclidienne pour les vecteurs.

La seconde est la *norme rang*. Pour une matrice **A** de $m \times n$, la norme rang est la plus grande somme des valeurs absolues des éléments d'une même ligne, elle est notée $\|\mathbf{A}\|_R$:

$$\|\mathbf{A}\|_R = \max_{1 \leqslant i \leqslant m} \sum_{j=1}^{n} |a_{ij}|.$$

La *norme colonne* est notée $\|\mathbf{A}\|_C$, et se calcule selon la formule $\|\mathbf{A}\|_C = \|\mathbf{A}^T\|_R$. La norme colonne est égale à la plus grande somme des valeurs absolues des éléments d'une colonne.

Prenons par exemple les matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix} \quad \text{et} \quad \mathbf{B} = \begin{bmatrix} 2 & 2 & 2 \\ 4 & 5 & 6 \end{bmatrix}.$$

Alors

$$\mathbf{A} - \mathbf{B} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

et

$\|\mathbf{A}-\mathbf{B}\|_F = \sqrt{11} \approx 3.3$ (norme Frobenius)
$\|\mathbf{A}-\mathbf{B}\|_R = 3$ (norme rang) et
$\|\mathbf{A}-\mathbf{B}\|_C = 4$ (norme colonne).

Dans toute la suite, nous utiliserons la norme rang, mais des résultats similaires sont obtenus avec les autres normes.

Le *nombre de conditionnement* d'une matrice **A** est égal à

$$K(\mathbf{A}) = \|\mathbf{A}\| \, \|\mathbf{A}^{-1}\|.$$

Donc $1 \leqslant K(\mathbf{A}) <$ quelle que soit la norme.

Ce nombre est très utile pour évaluer les erreurs dans les calculs. La matrice **A** est dite *mal conditionnée* si $K(\mathbf{A})$ est très grand.

Si des erreurs d'arrondi existent, elles risquent de se répercuter dans toute la suite des calculs. Supposons par exemple que **X** et **B** sont des vecteurs non nuls tels que $\mathbf{AX} = \mathbf{B}$. Si **A** contient une erreur $\Delta\mathbf{A}$ et si nous calculons $\mathbf{B} + \Delta\mathbf{B} = (\mathbf{A} + \Delta\mathbf{A})\mathbf{X}$, alors :

$$\frac{(\|\Delta\mathbf{B}\| / \|\mathbf{B}\|)}{(\|\Delta\mathbf{A}\| / \|\mathbf{A}\|)} \leqslant K(\mathbf{A}),$$

avec une égalité possible pour certaines valeurs de $\Delta\mathbf{A}$. Cela permet de majorer l'erreur sur **A** qui risque de se répercuter dans les calculs.

Grâce au nombre de conditionnement, il est possible d'évaluer l'erreur sur la solution d'un système par rapport à l'erreur sur les données en mémoire. Reprenons l'exemple précédent : **X** et **B** sont des vecteurs non nuls satisfaisant l'équation $\mathbf{AX} = \mathbf{B}$. S'il existe des erreurs dans la matrice **B** (erreurs d'arrondi par exemple), les erreurs étant représentées par $\Delta\mathbf{B}$, l'équation devient $\mathbf{A}(\mathbf{X} + \Delta\mathbf{X}) = \mathbf{B} + \Delta\mathbf{B}$ et alors

$$\frac{(\|\Delta\mathbf{X}\| / \|\mathbf{X}\|)}{(\|\Delta\mathbf{B}\| / \|\mathbf{B}\|)} \leqslant K(\mathbf{A}),$$

et l'égalité est possible pour certaines valeurs de $\Delta\mathbf{B}$.

Si il existe une erreur $\Delta\mathbf{A}$ sur la matrice **A**, l'équation s'écrit $(\mathbf{A} + \Delta\mathbf{A})(\mathbf{X} + \Delta\mathbf{X}) = \mathbf{B}$, si on note $d(\mathbf{A}, \Delta\mathbf{A}) = K(\mathbf{A})\ \|\Delta\mathbf{A}\| / \|\mathbf{A}\| < 1$, alors

$$\frac{(\|\Delta\mathbf{X}\| / \|\mathbf{X}\|)}{(\|\Delta\mathbf{A}\| / \|\mathbf{A}\|)} \leqslant K(\mathbf{A}) / (1 - d(\mathbf{A}, \Delta\mathbf{A})).$$

Si $\mathbf{A}^{-1} + \mathbf{Z}$ est la matrice inverse de $\mathbf{A} + \Delta\mathbf{A}$
alors

$$\frac{(\|\mathbf{Z}\| / \|\mathbf{A}^{-1}\|)}{(\|\Delta\mathbf{A}\| / \|\mathbf{A}\|)} \leqslant K(\mathbf{A}) / (1 - d(\mathbf{A}, \Delta\mathbf{A})).$$

Il existe encore des valeurs de $\Delta\mathbf{A}$ qui provoquent l'égalité.

Toutes les relations indiquées ci-dessus prouvent bien que l'erreur sur le résultat est facile à évaluer par rapport à l'erreur relative sur la matrice **A** à l'aide du nombre $K(\mathbf{A})$. Pour chaque inégalité il existe des matrices pour lesquelles l'égalité est réalisée. Plus le nombre de conditionnement est grand, plus l'erreur sur le résultat risque d'être élevée.

Des erreurs sur les données – parfois très petites en valeur relative – peuvent induire des solutions pour un système mal conditionné, très différentes de celles du système d'origine. De même, l'inverse d'une matrice mal conditionnée comportant des perturbations peut être assez différente de l'inverse de la matrice d'origine. Cette différence est majorée par $K(\mathbf{A})$, elle ne peut donc être élevée que si $K(\mathbf{A})$ est grand.

Dans le cas d'une matrice non singulière A, plus $K(\mathbf{A})$ est grand plus la matrice A est proche, au sens de la norme, d'une matrice singulière.

$$1/K(\mathbf{A}) = \min (\|\mathbf{A}\text{-}\mathbf{S}\| / \|\mathbf{A}\|)$$

et

$$1/\|\mathbf{A}^{-1}\| = \min (\|\mathbf{A}\text{-}\mathbf{S}\|)$$

où le minimum est pris sur toutes les matrices S singulières. Donc, si $K(\mathbf{A})$ est grand ou si $\|\mathbf{A}^{-1}\|$ est grand, alors la distance relative entre A et la matrice singulière S la plus proche est très petite.

Prenons par exemple :

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & .9999999999 \end{bmatrix}.$$

alors

$$\mathbf{A}^{-1} = \begin{bmatrix} -9{,}999{,}999{,}999 & 10^{10} \\ 10^{10} & -10^{10} \end{bmatrix}$$

et $\|\mathbf{A}^{-1}\| = 2 \times 10^{10}$. Si il existe une erreur sur A notée $\Delta\mathbf{A}$ telle que $\|\Delta\mathbf{A}\| = 5 \times 10^{-11}$ et telle que $\mathbf{A} + \Delta\mathbf{A}$ soit une matrice singulière. Si

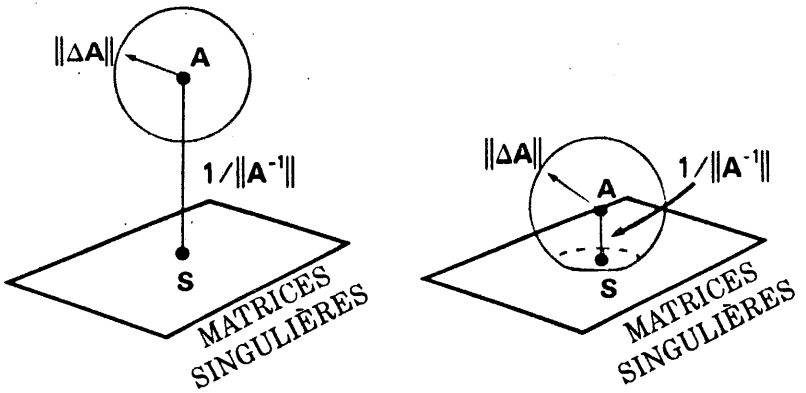
$$\Delta\mathbf{A} = \begin{bmatrix} 0 & -5 \times 10^{-11} \\ 0 & 5 \times 10^{-11} \end{bmatrix}$$

$\|\Delta\mathbf{A}\| = 5 \times 10^{-11}$ et

$$\mathbf{A} + \Delta\mathbf{A} = \begin{bmatrix} 1 & .99999999995 \\ 1 & .99999999995 \end{bmatrix}$$

$\mathbf{A} + \Delta\mathbf{A}$ est une matrice singulière.

La figure ci-dessous illustre parfaitement cette idée. La matrice **A** et la matrice **S** sont placées dans l'espace des matrices, par rapport à une surface représentant les matrices singulières. Les distances sont mesurées grâce à la norme. Autour de **A** se trouvent des matrices pratiquement semblables à **A** (par exemple, celles dont l'arrondi est le même). Cette zone a pour rayon $\|\Delta\mathbf{A}\|$. La distance entre la matrice **A** et la matrice singulière **S** la plus proche est $1/\|\mathbf{A}^{-1}\|$.



Dans le schéma de gauche, $\|\Delta\mathbf{A}\| < 1/\|\mathbf{A}^{-1}\|$. Si $\|\Delta\mathbf{A}\| << 1/\|\mathbf{A}^{-1}\|$ (ou $K(\mathbf{A})\,\|\Delta\mathbf{A}\|/\|\mathbf{A}\| << 1$), alors

$$\begin{aligned}\text{la variation relative sur } \mathbf{A}^{-1} &= \|\text{variation sur } \mathbf{A}^{-1}\|/\|\mathbf{A}^{-1}\| \\ &\approx (\|\Delta\mathbf{A}\|/\|\mathbf{A}\|)K(\mathbf{A}) \\ &= \|\Delta\mathbf{A}\|/(1/\|\mathbf{A}^{-1}\|) \\ &= (\text{rayon de la zone sphérique})/ \\ &\quad (\text{distance à la surface})\end{aligned}$$

Dans le schéma de droite, $\|\Delta\mathbf{A}\| > 1/\|\mathbf{A}^{-1}\|$, il existe ainsi une matrice singulière qui ne peut pas être distinguée de la matrice **A**, il n'est donc pas possible de calculer l'inverse de **A**.

Précision des solutions numériques des systèmes linéaires

Nous venons de voir que les imprécisions sur les données sont répercutées sur les solutions des systèmes d'équations linéaires et l'inversion des matrices. Mais même quand les données sont exactes, des imprécisions sont introduites par le calcul numérique des solutions et des inversions.

Prenons l'exemple de la résolution du système $\mathbf{AX} = \mathbf{B}$. La solution théorique est $\mathbf{X}$, mais à cause des erreurs d'arrondi, la solution calculée $\mathbf{Z}$ est plutôt la solution du système $(\mathbf{A} + \Delta\mathbf{A})\mathbf{Z} = \mathbf{B}$. $\Delta\mathbf{A}$ est telle que $\|\Delta\mathbf{A}\| \leq \varepsilon \|\mathbf{A}\|$ où ε est un nombre très petit. Dans la plupart des cas, $\Delta\mathbf{A}$ n'affecte que le 10^e chiffre des éléments de $\mathbf{A}$.

La matrice *résiduelle* $\mathbf{R} = \mathbf{B} - \mathbf{AZ}$ est telle que $\|\mathbf{R}\| \leq \varepsilon \|\mathbf{A}\| \|\mathbf{Z}\|$. Elle est donc faible en général. Cependant, si $\mathbf{A}$ est une matrice mal conditionnée, l'*erreur* $\mathbf{Z}\text{-}\mathbf{X}$ risque d'être élevée.

$$\|\mathbf{Z}\text{-}\mathbf{X}\| \leq \varepsilon \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \|\mathbf{Z}\| = \varepsilon K(\mathbf{A}) \|\mathbf{Z}\|$$

Voici une règle simple permettant d'évaluer la précision de la solution calculée :

$$\begin{pmatrix}\text{nombre de chiffres}\\ \text{décimaux corrects}\end{pmatrix} \geq \begin{pmatrix}\text{nombre de}\\ \text{chiffres traités}\end{pmatrix} - \log(\|\mathbf{A}\|\|\mathbf{A}^{-1}\|) - \log(10n)$$

n représentant la dimension de la matrice $\mathbf{A}$. Dans le cas du HP-15C, le nombre de chiffres précis traités est égal à 10.

$$(\text{nombre de chiffres corrects}) \geq 9 - \log(\|\mathbf{A}\|\|\mathbf{A}^{-1}\|) - \log(n).$$

Dans la plupart des applications, cette précision suffit. Si vous avez besoin d'une précision supplémentaire, vous pouvez améliorer la solution $\mathbf{Z}$ à l'aide de *calculs itératifs* (appelés aussi *correction des résidus*).

Par le calcul itératif, une solution est calculée puis sa précision est déterminée à l'aide de la résiduelle et cette solution est modifiée.

Pour utiliser cette méthode, commencez par calculer une solution $\mathbf{Z}$ du système $\mathbf{AX} = \mathbf{B}$. $\mathbf{Z}$ est ensuite considéré comme une valeur approchée de $\mathbf{X}$ telle que $\mathbf{E} = \mathbf{X} - \mathbf{Z}$, $\mathbf{E}$ vérifie le système $\mathbf{AE} = \mathbf{AX} - \mathbf{AZ} = \mathbf{R}$, où $\mathbf{R}$ est la résiduelle de $\mathbf{Z}$.

Il faut ensuite calculer la résiduelle et résoudre l'équation **AL R**. La solution calculée, appelée **F**, est alors considérée comme une valeur approchée de $E = X - Z$, elle s'ajoute à **Z** et une nouvelle approximation de **X** est obtenue $F + Z \approx (X - Z) + Z = X$.

Pour que la précision de $F + Z$ soit meilleure que celle de **Z**, il faut calculer $R = B - AZ$ avec une excellente précision. C'est ce que fait la fonction [MATRIX] 6 du HP-15C. La matrice **A** sert à calculer **Z** et **F**; la décomposition faite pour le calcul de **Z** est utilisée aussi pour **F** - ce qui réduit le temps de l'exécution. Le processus décrit ci-dessus peut être utilisé de nouveau, mais en pratique on constate qu'une excellente précision s'obtient dès le premier calcul.

(Vous trouverez à la fin de ce chapitre un exemple de programme pour effectuer une étape du calcul itératif.)

Simplification d'équations difficiles

Un système d'équations du type $EX = B$ est très difficile à résoudre numériquement dans le cas où **E** est une matrice mal conditionnée (presque singulière). De plus, le calcul itératif risque de ne pas donner des résultats satisfaisants dans ce cas. Cependant il existe des cas où un simple petit effort suffit à simplifier un problème difficile. La mise à l'échelle et le préconditionnement du problème sont deux méthodes de simplification du traitement.

Mise à l'échelle

Un problème mal mis à l'échelle peut conduire à des opérations erronées comme par exemple l'inversion de matrices mal conditionnées ou la résolution de systèmes d'équations à l'aide de matrice mal conditionnées. Cela peut facilement être évité.

Prenons l'exemple d'une matrice **E** obtenue à partir d'une matrice **A** telle que $E = LAR$ où **L** et **R** sont des matrices diagonales dont les éléments sont des puissances entières de 10. On dit que **E** est dérivée de **A** par *mise à l'échelle*. **L** met à l'échelle des lignes de **A** et **R** les colonnes. $E^{-1} = R^{-1}A^{-1}L^{-1}$, il est donc possible d'obtenir E^{-1} soit à l'aide de A^{-1}, soit en inversant **E**.

Prenons un exemple :

$$\mathbf{A} = \begin{bmatrix} 3\times10^{-40} & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix}.$$

Le HP-15C est capable de calculer $\mathbf{A}^{-1}$ avec une précision de 10 chiffres

$$\mathbf{A}^{-1} \approx \begin{bmatrix} -2 & 3 & -1 \\ 3 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}.$$

Si

$$\mathbf{L} = \mathbf{R} = \begin{bmatrix} 10^{20} & 0 & 0 \\ 0 & 10^{-20} & 0 \\ 0 & 0 & 10^{-20} \end{bmatrix}$$

alors

$$\mathbf{E} = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 10^{-40} & 10^{-40} \\ 2 & 10^{-40} & -10^{-40} \end{bmatrix}.$$

E est donc très proche d'une matrice singulière **S**

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

$\|\mathbf{E}\text{-}\mathbf{S}\| / \|\mathbf{E}\| = 1/3 \times 10^{-40}$. Donc $K(\mathbf{S}) \geqslant 3 \times 10^{40}$, on peut donc vérifier que l'inverse calculée $\mathbf{E}^{-1}$.

$$\mathbf{E}^{-1} \approx \begin{bmatrix} -6.67\times10^{-11} & 1 & 10^{-10} \\ 0.8569 & 8.569\times10^{9} & -4.284\times10^{9} \\ 0.07155 & -4.284\times10^{9} & 2.142\times10^{9} \end{bmatrix}$$

est très différent de la valeur vraie

$$\mathbf{E}^{-1} = \begin{bmatrix} -2\times10^{-40} & 3 & -1 \\ 3 & -4\times10^{40} & 2\times10^{40} \\ -1 & 2\times10^{40} & -10^{40} \end{bmatrix}.$$

En multipliant la matrice inverse calculée par la matrice **E** d'origine, on se rend bien compte que le calcul est inexact.

C'est parce que la mise à l'échelle de **E** n'est pas judicieuse. Une matrice bien mise à l'échelle comme **A** doit avoir des lignes et des colonnes comparables en norme et ceci doit être vrai pour la matrice inverse. C'est bien vérifié dans le cas de **E**, mais pour $\mathbf{E}^{-1}$ on voit que les normes de la première ligne et de la première colonne sont très petites comparées à celles des autres. Il faut donc mettre les lignes et colonnes à l'échelle avant d'inverser la matrice. Cela signifie qu'il faut choisir les matrices diagonales **L** et **R** de telle sorte que **LER** et $(\mathbf{LER})^{-1} = \mathbf{R}^{-1}\mathbf{E}^{-1}\mathbf{L}^{-1}$ soit assez bien mises à l'échelle.

En général on ne peut pas prévoir la valeur exacte de $\mathbf{E}^{-1}$. Il faut donc examiner la matrice **E** et la matrice calculée $\mathbf{E}^{-1}$ pour se rendre compte si le choix des matrices **L** et **R** est bon. Dans ce cas, la matrice calculée $\mathbf{E}^{-1}$ indique que le choix n'est pas très bon et nous incite à prendre

$$\mathbf{L} = \mathbf{R} = \begin{bmatrix} 10^{-5} & 0 & 0 \\ 0 & 10^{5} & 0 \\ 0 & 0 & 10^{5} \end{bmatrix}.$$

Ce qui donne :

$$\mathbf{LER} = \begin{bmatrix} 3\times10^{-10} & 1 & 2 \\ 1 & 10^{-30} & 10^{-30} \\ 2 & 10^{-30} & -10^{-30} \end{bmatrix},$$

qui n'est toujours pas excellent mais qui est néanmoins meilleur. La matrice inverse calculée est égale à

$$(\mathbf{LER})^{-1} = \begin{bmatrix} -2\times10^{-30} & 3 & -1 \\ 3 & -4\times10^{30} & 2\times10^{30} \\ -1 & 2\times10^{30} & -10^{30} \end{bmatrix}.$$

Ce résultat est juste jusqu'au 10^e chiffre, bien que vous ne puissiez pas le voir immédiatement. On peut le vérifier en utilisant la relation :

$$(\mathbf{LER})^{-1}(\mathbf{LER}) = (\mathbf{LER})(\mathbf{LER})^{-1} = \mathbf{I} \text{ (matrice identité)}$$

égalité vérifiée jusqu'au 10^e chiffre.

On peut alors calculer $\mathbf{E}^{-1}$

$$\mathbf{E}^{-1} = \mathbf{R}(\mathbf{LER})^{-1}\mathbf{L} = \begin{bmatrix} -2\times10^{-40} & 3 & -1 \\ 3 & -4\times10^{40} & 2\times10^{40} \\ -1 & 2\times10^{40} & -10^{40} \end{bmatrix},$$

avec une précision de 10 chiffres.

Si il est difficile de vérifier la précision sur $(\mathbf{LER})^{-1}$, vous pouvez utiliser la méthode de mise à l'échelle en prenant **LER** comme matrice **E** et de nouvelles matrices de mise à l'échelle.

Cette méthode de mise à l'échelle sert aussi à résoudre des équations matricielles du type $\mathbf{EX} = \mathbf{B}$. Il est possible de remplacer le système $\mathbf{EX} = \mathbf{B}$ par $(\mathbf{LER})\mathbf{Y} = \mathbf{LB}$ à résoudre pour **Y**. Les matrices **L** et **R** doivent être choisies pour que la matrice **LER** soit correctement à l'échelle. On calcule ensuite **X** grâce à la relation $\mathbf{X} = \mathbf{RY}$.

Préconditionnement

Le préconditionnement est aussi une méthode pour transformer des systèmes difficiles $\mathbf{EX} = \mathbf{B}$ en problèmes plus simples, $\mathbf{AX} = \mathbf{D}$ ayant la même solution **X**.

Prenons l'exemple d'une matrice **E** mal conditionnée (presque singulière). Vous vous en rendrez facilement compte en calculant $\mathbf{E}^{-1}$ et en remarquant que $1/\|\mathbf{E}^{-1}\|$ est beaucoup plus petit que $\|\mathbf{E}\|$ (ou en remarquant que $K(\mathbf{E})$ est très grand). On constate alors que tout vecteur ligne $\mathbf{u}^T$ possède la propriété suivante : $\|\mathbf{u}^T\| / \|\mathbf{u}^T\mathbf{E}^{-1}\|$ est très petit devant $\|\mathbf{E}\|$. En effet $\|\mathbf{u}^T\mathbf{E}^{-1}\|$ n'est pas beaucoup plus petite que $\|\mathbf{u}^T\|\,\|\mathbf{E}^{-1}\|$, et $\|\mathbf{E}^{-1}\|$ est grande. Prenons un vecteur ligne $\mathbf{u}^T$ et calculons $\mathbf{v}^T = a\mathbf{u}^T\mathbf{E}^{-1}$, le scalaire a est choisi de telle sorte que le vecteur ligne $\mathbf{r}^T$ obtenu en arrondissant chaque élément de $\mathbf{v}^T$ à un entier compris entre -100 et 100, ne soit pas très différent de $\mathbf{v}^T$.

Le vecteur-ligne $\mathbf{r}^T$ possède des éléments entiers tous inférieurs à 100. $\|\mathbf{r}^T\mathbf{E}\|$ est donc petite comparée à $\|\mathbf{r}^T\|\,\|\mathbf{E}\|$. Et c'est ce que nous recherchons.

Supposons que le $k$ième élément de $\mathbf{r}^T$ soit l'un des plus grands. Remplaçons alors le $k$ième rang de **E** par $\mathbf{r}^T\mathbf{E}$ et le $k$ième rang de **B** par $\mathbf{r}^T\mathbf{B}$. Si aucun arrondi n'a été fait dans l'évaluation des nouveaux rangs, la nouvelle matrice **A** doit être mieux conditionnée que la matrice **E**, mais le système a toujours la même solution **X**.

Ce processus est très efficace dans le cas où **E** et **A** sont à la bonne échelle, c'est-à-dire lorsque tous les rangs de **E** et de **A** ont à peu près la même norme. Cela se réalise en multipliant les rangs des systèmes d'équations $\mathbf{EX} = \mathbf{B}$ et $\mathbf{AX} = \mathbf{D}$ par les puissances convenables de 10. Si **A** ne diffère pas assez d'une matrice singulière, bien qu'elle soit bien mise à l'échelle, reprenez le processus de préconditionnement.

Afin de mieux comprendre, prenons l'exemple du système $\mathbf{EX} = \mathbf{B}$ dans lequel

$$\mathbf{E} = \begin{bmatrix} x & y & y & y & y \\ y & x & y & y & y \\ y & y & x & y & y \\ y & y & y & x & y \\ y & y & y & y & x \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

$x = 8000.00002$ et $y = -1999.99998$. Si vous essayez de résoudre directement ce système, voilà les solutions que vous donnera le HP-15C.

$$\mathbf{X} \approx \begin{bmatrix} 2014.6 \\ 2014.6 \\ 2014.6 \\ 2014.6 \\ 2014.6 \end{bmatrix} \quad \text{et} \quad \mathbf{E}^{-1} \approx 2014.6 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

puis

$$\mathbf{EX} \approx \begin{bmatrix} 1.00146 \\ 0.00146 \\ 0.00146 \\ 0.00146 \\ 0.00147 \end{bmatrix}.$$

En faisant un test (utilisant [MATRIX] 7), vous découvrez que $1/\|\mathbf{E}^{-1}\| \approx 9.9 \times 10^{-5}$, ce qui est très petit devant $\|\mathbf{E}\| \approx 1.6 \times 10^{4}$ (c'est-à-dire que le nombre calculé est très grand : $\|\mathbf{E}\|\,\|\mathbf{E}^{-1}\| \approx 1.6 \times 10^{8}$).

Choisissons un vecteur ligne quelconque $\mathbf{u}^T = (1, 1, 1, 1, 1)$ et calculons

$$\mathbf{u}^T\mathbf{E}^{-1} \approx 10{,}073\ (1, 1, 1, 1, 1).$$

Si $a = 10^{-4}$,

$$\mathbf{v}^T = a\mathbf{u}^T\mathbf{E}^{-1} \approx 1.0073\ (1, 1, 1, 1, 1)$$
$$\mathbf{r}^T = (1, 1, 1, 1, 1)$$
$$\|\mathbf{r}^T\mathbf{E}\| \approx 5 \times 10^{-4}$$
$$\|\mathbf{r}^T\|\,\|\mathbf{E}\| \approx 8 \times 10^{4}.$$

Comme nous nous y attendions, $\|\mathbf{r}^T\mathbf{E}\|$ est petite devant $\|\mathbf{r}^T\|\,\|\mathbf{E}\|$.

Remplaçons le premier rang de $\mathbf{E}$ par

$$10^7\mathbf{r}^T\mathbf{E} = (1000, 1000, 1000, 1000, 1000)$$

et le premier rang de $\mathbf{B}$ par $10^7\mathbf{r}^T\mathbf{B} = 10^7$, on obtient alors une nouvelle équation matricielle $\mathbf{AX} = \mathbf{D}$, dans laquelle

$$\mathbf{A} = \begin{bmatrix} 1000 & 1000 & 1000 & 1000 & 1000 \\ y & x & y & y & y \\ y & y & x & y & y \\ y & y & y & x & y \\ y & y & y & y & x \end{bmatrix} \quad \text{et} \quad \mathbf{D} = \begin{bmatrix} 10^7 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

$\mathbf{r}^T\mathbf{E}$ a été mis à l'échelle 10^7 et ainsi toutes les lignes de **E** et de **A** ont des normes comparables. En se servant du nouveau système, le HP-15C calcule la solution

$$\mathbf{X} = \begin{bmatrix} 2000.000080 \\ 1999.999980 \\ 1999.999980 \\ 1999.999980 \\ 1999.999980 \end{bmatrix}, \text{ avec } \mathbf{AX} = \begin{bmatrix} 10^7 \\ -10^{-5} \\ -9 \times 10^{-6} \\ 0 \\ 0 \end{bmatrix}.$$

Cette solution est différente de la solution trouvée précédemment, elle a une précision de 10 chiffres.

Il arrive parfois que les éléments d'une matrice presque singulière **E** soient calculés avec des arrondis, dans ce cas la matrice $\mathbf{E}^{-1}$ n'est pas exacte même si ses éléments sont calculés sans erreurs arithmétiques. Le préconditionnement n'est valable dans ce cas que si le rang modifié de la matrice **A** est obtenu avec une grande précision. En d'autres termes, on peut dire qu'il ne faut transformer une formule à l'aide de la méthode de préconditionnement que si l'on est sûr de pouvoir en tirer des avantages.

Méthode des moindres carrés

Les opérations matricielles sont fréquemment utilisées dans des calculs de *moindres carrés*. Dans ce type de calculs, on rencontre souvent une matrice **X** de $n \times p$ contenant des données et un vecteur **y** à n éléments pour lesquels il faut trouver un vecteur **b** à p éléments tel que l'expression suivante soit minimale :

$$\|\mathbf{r}\|_F^2 = \sum_{i=1}^{n} r_i^2$$

où $\mathbf{r} = \mathbf{y} - \mathbf{Xb}$ est appelé vecteur résiduel.

Équations normales

$$\|\mathbf{r}\|_F^2 = (\mathbf{y} - \mathbf{Xb})^T(\mathbf{y} - \mathbf{Xb}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{b}^T\mathbf{X}^T\mathbf{y} + \mathbf{b}^T\mathbf{X}^T\mathbf{Xb}.$$

La résolution de cette équation est équivalente à la recherche de la solution **b** d'*équations normales :*

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y}.$$

Mais les équations normales sont sujettes aux erreurs d'arrondi. (La factorisation orthogonale, expliquée page 113 est en effet très sensible aux erreurs d'arrondi.)

Un problème comprenant un calcul de *moindres carrés pondérés* est en fait la généralisation d'un problème de calcul de moindres carrés. Il s'agit de minimiser l'expression

$$\|\mathbf{W}\mathbf{r}\|_F^2 = \sum_{i=1}^{n} w_i^2 r_i^2$$

dans laquelle **W** est une matrice diagonale $n \times n$, dont tous les éléments diagonaux $w_1, w_2, \ldots, w_n$ sont positifs.

$$\|\mathbf{W}\mathbf{r}\|_F^2 = (\mathbf{y} - \mathbf{X}\mathbf{b})^T\mathbf{W}^T\mathbf{W}(\mathbf{y} - \mathbf{X}\mathbf{b})$$

toute solution **b** est aussi une solution des équations normales pondérées :

$$\mathbf{X}^T\mathbf{W}^T\mathbf{W}\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{W}^T\mathbf{W}\mathbf{y}.$$

Ce sont en fait les équations normales dans lesquelles **X** et **y** sont remplacées par **WX** et **Wy**. Elles sont donc très sensibles aux erreurs d'arrondi.

Dans un problème de *calcul de moindres carrés avec des contraintes linéaires*, il faut trouver **b** tel que l'équation suivante soit minimisée :

$$\|\mathbf{r}\|_F^2 = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|_F^2$$

avec :

$$\mathbf{C}\mathbf{d} = \mathbf{d} \qquad \left(\sum_{j=1}^{k} c_{ij} b_j = d_i \text{ pour } i = 1, 2, \ldots, m \right).$$

Cela revient en fait à résoudre les *équations normales augmentées*

$$\begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{y} \\ \mathbf{d} \end{bmatrix}$$

dans lesquelles **I** est un vecteur de Lagrange faisant partie de la solution mais qui n'est pas utilisé par la suite. Les équations augmentées sont, elles aussi, très sensibles aux erreurs d'arrondi. Il est possible d'inclure des pondérations en remplaçant **X** et **y** par **WX** et **Wy**.

Afin de bien prouver que les équations normales ne sont pas très fiables pour la résolution des problèmes de moindres carrés prenons un exemple numérique :

$$\mathbf{X} = \begin{bmatrix} 100{,}000. & -100{,}000. \\ 0.1 & 0.1 \\ 0.2 & 0.0 \\ 0.0 & 0.2 \end{bmatrix} \quad \text{et} \quad \mathbf{y} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}.$$

Alors

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 10{,}000{,}000{,}000.05 & -9{,}999{,}999{,}999.99 \\ -9{,}999{,}999{,}999.99 & 10{,}000{,}000{,}000.05 \end{bmatrix}$$

et

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 10{,}000.03 \\ -9{,}999.97 \end{bmatrix}.$$

Cependant en arrondissant à 10 chiffres

$$\mathbf{X}^T\mathbf{X} \approx \begin{bmatrix} 10^{10} & -10^{10} \\ -10^{10} & 10^{10} \end{bmatrix},$$

ce qui donne le même résultat que si les éléments de $\mathbf{X}$ étaient arrondis au 5^e chiffre du plus grand élément :

$$\mathbf{X} = \begin{bmatrix} 100{,}000 & -100{,}000 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Le HP-15C résout alors l'équation $\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y}$ (en perturbant légèrement la matrice singulière comme indiqué page 118) et donne

$$\mathbf{b} = \begin{bmatrix} 0.060001 \\ 0.060000 \end{bmatrix}$$

avec

$$\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\mathbf{b} = \begin{bmatrix} 0.03 \\ 0.03 \end{bmatrix}.$$

Cependant la solution correcte de la méthode des moindres carrés est

$$\mathbf{b} = \begin{bmatrix} 0.5000005 \\ 0.4999995 \end{bmatrix}$$

bien que les deux solutions satisfassent également aux équations normales.

Les équations normales ne doivent être utilisées que quand les éléments de $\mathbf{X}$ sont des entiers relativement faibles (disons entre -3000 et 3000) ou quand on sait qu'aucune perturbation sur une colonne $\mathbf{x}_j$ de $\mathbf{X}$, inférieure à $\|\mathbf{x}_j\|/10^4$, ne risque de rendre deux colonnes linéairement dépendantes.

Factorisation orthogonale

La méthode de factorisation orthogonale ci-dessous permet de résoudre les problèmes de moindres carrés ; elle est moins sensible aux erreurs d'arrondi que la méthode des équations normales. Il faut l'utiliser quand la méthode des équations normales n'est pas satisfaisante.

Toute matrice $n \times p$, $\mathbf{X}$, peut se mettre sous la forme $\mathbf{X} = \mathbf{Q}^T\mathbf{U}$ où $\mathbf{Q}$ est une matrice $n \times n$ orthogonale caractérisée par $\mathbf{Q}^T = \mathbf{Q}^{-1}$ et $\mathbf{U}$ une matrice triangulaire supérieure $n \times p$. La propriété essentielle d'une matrice orthogonale est qu'elle préserve la longueur

$$\begin{aligned} \|\mathbf{Q}\mathbf{r}\|_F^2 &= (\mathbf{Q}\mathbf{r})^T(\mathbf{Q}\mathbf{r}) \\ &= \mathbf{r}^T\mathbf{Q}^T\mathbf{Q}\mathbf{r} \\ &= \mathbf{r}^T\mathbf{r} \\ &= \|\mathbf{r}\|_F^2. \end{aligned}$$

Si $\mathbf{r} = \mathbf{y} - \mathbf{X}\mathbf{b}$, il a la même longueur que

$$\mathbf{Q}\mathbf{r} = \mathbf{Q}\mathbf{y} - \mathbf{Q}\mathbf{X}\mathbf{b} = \mathbf{Q}\mathbf{y} - \mathbf{U}\mathbf{b}.$$

La matrice triangulaire supérieure **U** et le produit **Qy** peuvent se mettre sous la forme :

$$\mathbf{U} = \begin{bmatrix} \hat{\mathbf{U}} \\ \mathbf{0} \end{bmatrix} \begin{matrix} (p \text{ rangs}) \\ (n\text{-}p \text{ rangs}) \end{matrix} \quad \text{et} \quad \mathbf{Qy} = \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} \begin{matrix} (p \text{ rangs}) \\ (n\text{-}p \text{ rangs}) \end{matrix}$$

Alors

$$\begin{aligned} \|\mathbf{r}\|_F^2 &= \|\mathbf{Qr}\|_F^2 \\ &= \|\mathbf{Qy} - \mathbf{Ub}\|_F^2 \\ &= \|\mathbf{g} - \hat{\mathbf{U}}\mathbf{b}\|_F^2 + \|\mathbf{f}\|_F^2 \\ &\geqslant \|\mathbf{f}\|_F^2 \end{aligned}$$

avec égalité quand $\mathbf{g} - \hat{\mathbf{U}}\mathbf{b} = \mathbf{0}$. En d'autres termes, la solution d'un problème de moindres carrés est aussi solution de $\hat{\mathbf{U}}\mathbf{b} = \mathbf{g}$; la valeur minimale de la somme des carrés est alors égale à $\|\mathbf{f}\|_F^2$. C'est la base de tous les programmes numériques de calcul des moindres carrés.

On peut résoudre un problème de calcul de moindres carrés en deux étapes :

1. Effectuez une factorisation orthogonale de la matrice augmentée $n \times (p+1)$:

$$\begin{bmatrix} \mathbf{X} & \mathbf{y} \end{bmatrix} = \mathbf{Q}^T\mathbf{V}$$

dans laquelle $\mathbf{Q}^T = \mathbf{Q}^{-1}$ et mettez la matrice triangulaire supérieure sous la forme

$$\mathbf{V} = \begin{bmatrix} \hat{\mathbf{U}} & \mathbf{g} \\ \mathbf{0} & q \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{matrix} (p \text{ rangs}) \\ (1 \text{ rangs}) \\ (n-p \text{ rangs}) \end{matrix} \quad \begin{matrix} (1 \text{ colonne}) \\ (p \text{ colonnes}) \end{matrix}$$

Il ne faut conserver que les $(p+1)$ rangs (et colonnes) de **V**. (**Q** est différente de l'exemple précédent puisqu'ici elle comprend aussi la factorisation de **y**).

2. Résolvez le système ci-dessous pour **b** :

$$\begin{bmatrix} \hat{\mathbf{U}} & \mathbf{g} \\ \mathbf{0} & q \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ -1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -q \end{bmatrix}.$$

(si $q = 0$, remplacez-le par un nombre très petit comme 10^{-99}). Dans la matrice solution, – 1 apparaît automatiquement sans calcul.

Si il n'y a aucune erreur d'arrondi, $q = \pm \|\mathbf{y} - \mathbf{X}\mathbf{b}\|_F$; cela peut être légèrement différent si $|q|$ est très petite, mettons inférieure à $\|\mathbf{y}\|/10^6$. Si vous désirez une meilleure approximation de $\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_F$, calculez-la à partir de **X**, de **y** et de la solution calculée **b**.

Dans le cas d'un calcul pondéré de moindres carrés, remplacez simplement **X** et **y** par **WX** et **Wy** où **W** est une matrice diagonale formée des coefficients de pondération.

Pour un calcul de moindres carrés avec contraintes linéaires il faut admettre que les contraintes sont négligeables. Il est aussi impossible d'obtenir par calcul numérique une solution parfaitement exacte à cause des erreurs d'arrondi. Il faut déterminer la tolérance t telle que les contraintes sont négligeables quand $\|\mathbf{C}\mathbf{b} - \mathbf{d}\| < t$. En général $t > \|\mathbf{d}\|/10^{10}$ permet une précision de 10 chiffres, dans certains cas une tolérance plus grande est nécessaire.

Quand t est choisie, sélectionnez le coefficient de pondération w vérifiant $w > \|\mathbf{y}\|/t$. Pour plus de simplicité, choisissez pour w une puissance de 10 supérieure à $\|\mathbf{y}\|/t$. Alors $w\|\mathbf{C}\mathbf{b} - \mathbf{d}\| > \|\mathbf{y}\|$ sauf si $\|\mathbf{C}\mathbf{b} - \mathbf{d}\| < t$.

Il se peut cependant que les contraintes ne soient pas satisfaisantes pour une des deux raisons suivantes :

- Il n'existe pas de **b** tel que $\|\mathbf{C}\mathbf{b} - \mathbf{d}\| < t$.
- Les dernières colonnes de **C** sont linéairement dépendantes.

Dans le premier cas, il faut déterminer si une solution existe pour les contraintes seules. Quand [w**C** w**d**] est factorisé sous la forme **Q**[**U g**], résolvez en **b** le système suivant :

$$\begin{matrix} (k \text{ rangs}) \\ (p+1-k \text{ rangs}) \end{matrix} \begin{bmatrix} \mathbf{U} & \mathbf{g} \\ \mathbf{0} & \text{diag}(q) \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ -1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -q \end{bmatrix} \begin{matrix} (p \text{ rangs}) \\ (1 \text{ rang}) \end{matrix}$$

en utilisant un nombre q très petit et non nul. Si la solution calculée **b** satisfait $\mathbf{C}\mathbf{b} \approx \mathbf{d}$ alors les contraintes ne sont pas négligeables.

Le second cas est beaucoup plus rare et peut être évité. Il se présente quand au moins un des éléments diagonaux de **U** est beaucoup plus petit que le plus grand des éléments qui se trouvent au-dessus dans la même colonne, où **U** provient de la factorisation orthogonale $w\mathbf{C} = \mathbf{QU}$.

Afin d'éviter cette situation, il faut réordonner les colonnes de $w\mathbf{C}$ et de **X** ainsi que les éléments (rangs) de **b**. Le nouvel ordre est facile à trouver si l'élément perturbateur de **U** est aussi beaucoup plus petit que la plupart des éléments de son rang. Il suffit alors d'échanger les colonnes correspondantes dans les données originales et de refactoriser les équations de contraintes pondérées. Répétez cette procédure si nécessaire.

Prenons l'exemple suivant dans lequel la factorisation $w\mathbf{C}$ donne :

$$\mathbf{U} = \begin{bmatrix} 1.0 & 2.0 & 0.5 & -1.5 & 0.3 \\ 0 & 0.02 & 0.5 & 3.0 & 0.1 \\ 0 & 0 & 2.5 & 1.5 & -1.2 \end{bmatrix},$$

Le second élément diagonal est très inférieur à 2.0 qui se trouve juste au-dessus. Cela indique que la première et la deuxième colonne des contraintes d'origine sont pratiquement dépendantes. Cet élément est également très inférieur à 3.0 qui se trouve dans la même ligne. La seconde et la quatrième colonne des données d'origine doivent être échangées et il faut refaire la factorisation.

Il est bon de toujours vérifier si les contraintes sont négligeables. Le test sur les éléments diagonaux de **U** peut se faire en même temps.

Pour finir, il suffit d'utiliser **U** et **g** comme k premiers rangs puis d'ajouter les rangs convenables de **X** et **y**. (Reportez-vous à la page 140.) Résolvez enfin le problème de moindres carrés sans contraintes en transformant

$$\mathbf{X} \rightarrow \begin{bmatrix} w\mathbf{C} \\ \mathbf{X} \end{bmatrix} \quad \text{et} \quad \mathbf{y} \rightarrow \begin{bmatrix} w\mathbf{d} \\ \mathbf{y} \end{bmatrix}.$$

Si la solution calculée **b** satisfait $\|\mathbf{Cb} - \mathbf{d}\| < t$ alors elle minimise aussi l'expression $\|\mathbf{y} - \mathbf{Xb}\|$ avec la contrainte $\mathbf{Cb} \approx \mathbf{d}$.

Matrices singulières et presque singulières

Une matrice est dite singulière si et seulement si son déterminant est nul. Le déterminant d'une matrice est égal à $(-1)^r$ que multiplie le produit des éléments diagonaux de **U**, dans lequel **U** est la matrice triangulaire supérieure de la décomposition LU; et r le nombre de permutations sur les rangs avant la décomposition. Donc une matrice est singulière si un des éléments diagonaux au moins de **U** est nul, sinon elle n'est pas singulière.

Cependant, puisque le HP-15C utilise un nombre fini de chiffres pour ses calculs, certaines matrices singulières et presque singulières ne peuvent pas être distinguées. Considérons par exemple la matrice

$$\mathbf{B} = \begin{bmatrix} 3 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 0 & 0 \end{bmatrix} = \mathbf{LU},$$

est singulière. Si une utilise une précision de 10 chiffres, la matrice est décomposée sous la forme

$$\mathbf{LU} = \begin{bmatrix} 1 & 0 \\ .3333333333 & 1 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 0 & 10^{-10} \end{bmatrix},$$

qui n'est pas une matrice singulière. La matrice singulière **B** ne peut pas se distinguer de la matrice non singulière

$$\mathbf{D} = \begin{bmatrix} 3 & 3 \\ .9999999999 & 1 \end{bmatrix}$$

puisque leurs décompositions LU sont identiques.

D'autre part la matrice

$$\mathbf{A} = \begin{bmatrix} 3 & 3 \\ 1 & .9999999999 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 0 & -10^{-10} \end{bmatrix} = \mathbf{LU}$$

n'est pas singulière. En utilisant une précision de 10 chiffres la matrice **A** se décompose sous la forme

$$\mathbf{LU} = \begin{bmatrix} 1 & 0 \\ .3333333333 & 1 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 0 & 0 \end{bmatrix}.$$

Ce qui indiquerait que la matrice **A** est singulière. La matrice non singulière **A** ne peut pas être distinguée de la matrice singulière

$$\mathbf{C} = \begin{bmatrix} 3 & 3 \\ .9999999999 & .9999999999 \end{bmatrix}$$

puisqu'ils ont la même décomposition *LU*.

En vous servant de votre HP-15C pour inverser une matrice ou pour résoudre un système d'équations, vous vous rendrez compte que des matrices singulières et des matrices presque singulières ont la même décomposition *LU*. C'est pour cela que le HP-15C s'assure ***toujours*** que le résultat des calculs n'a ***jamais*** de pivot nul. Si c'est nécessaire, il modifie le pivot d'une quantité inférieure à l'erreur d'arrondi. Ceci est très important dans certaines applications comme le calcul des vecteurs propres à l'aide de la méthode d'itération inverse (voir page 155).

Les erreurs d'arrondi et les modifications intentionnelles permettent le calcul d'une décomposition ne comportant aucun pivot nul et correspondant à une matrice non singulière $\mathbf{A} + \Delta\mathbf{A}$ identique ou légèrement différente de la matrice **A** de départ. En général, à moins que tous les éléments d'une même colonne de **A** ne soient inférieurs à 10^{-89} en valeur absolue, la norme colonne $\|\Delta\mathbf{A}\|_c$ est négligeable devant $\|\mathbf{A}\|_c$.

Le HP-15C calcule le déterminant d'une matrice carrée comme étant le produit des pivots calculés (éventuellement modifiés). Le déterminant calculé est celui de la matrice $\mathbf{A} + \Delta\mathbf{A}$ décomposée dans la forme *LU*. Il n'est nul que si la valeur absolue est inférieure à 10^{-99} (dépassement de capacité inférieure).

Applications

Voici quelques programmes illustrant l'utilisation du calcul matriciel pour la résolution de problèmes complexes.

Construction de la matrice identité

Ce programme est destiné à la création d'une matrice identité $\mathbf{I}_n$ dans une matrice dont le label se trouve dans le registre d'index. Ce programme suppose que la matrice est déjà dimensionnée $n \times n$. Pour exécuter le programme utilisez [GSB] 8. La matrice finale contient des 1 sur la diagonale et des 0 partout ailleurs.

Appuyer sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] 8	001–42,21, 8	
[f] [MATRIX] 1	002–42,16, 1	Initialise $i = j = 1$.
[f] [LBL] 9	003–42,21, 9	
[RCL] 0	004– 45 0	
[RCL] 1	005– 45 1	
[g] [TEST] 6	006–43,30, 6	Teste $i \neq j$.
[g] [CLx]	007– 43 35	
[g] [TEST] 5	008–43,30, 5	Teste $i = j$.
[EEX]	009– 26	Initialise l'élément à 1 si $i = j$.
[f] [USER] [STO] [(i)] [f] [USER]	010u 44 24	Saute le pas suivant pour le dernier élément.
[GTO] 9	011– 22 9	
[g] [RTN]	012– 43 32	
[g] [P/R]		Mode calcul.

Labels utilisés : 8 et 9.

Registres utilisés : R_0, R_1, et registre d'index.

Correction de la solution par une itération

Le programme ci-dessous permet de résoudre en **X** le système $\mathbf{AX} = \mathbf{B}$, puis de faire un calcul itératif à un niveau pour améliorer la précision de la solution. Ce programme utilise quatre matrices :

Matrice	A	B	C	D
Entrée	Matrice du système	Matrice de second membre		
Sortie	Matrice du système	Solution corrigée	Solution non-corrigée	Décomposition *LU* de A

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000-	
[f] [LBL] A	001-42,21,11	
[RCL] [MATRIX] [A]	002-45,16,11	
[STO] [MATRIX] [D]	003-44,16,14	Stocke la matrice du système dans D.
[RCL] [MATRIX] [B]	004-45,16,12	
[RCL] [MATRIX] [D]	005-45,16,14	
[f] [RESULT] [C]	006-42,26,13	
[÷]	007- 10	Calcule la solution C non-corrigée.
[f] [RESULT] [B]	008-42,26,12	
[f] [MATRIX] 6	009-42,16, 6	Calcule la matrice résiduelle B.
[RCL] [MATRIX] [D]	010-45,16,14	
[÷]	011- 10	Calcule la correction B.
[RCL] [MATRIX] [C]	012-45,16,13	
[+]	013- 40	Calcule la solution corrigée B.
[g] [RTN]	014- 43 32	
[g] [P/R]		Mode calcul.

Labels utilisés : A.
Matrices utilisées : A, B, C et D.

Pour utiliser ce programme :

1. Dimensionnez la matrice A selon la matrice du système puis stockez les coefficients dans A.
2. Dimensionnez la matrice B selon la matrice de second membre puis stockez ces éléments dans B.
3. Appuyez sur [GSB] [A] pour calculer la solution corrigée qui se trouve par la suite dans B.

Exemple : En utilisant le programme de correction par la résiduelle, calculez l'inverse de la matrice **A**.

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

Théoriquement :

$$\mathbf{A}^{-1} = \begin{bmatrix} -29/3 & -8/3 & -32 \\ 8 & 5/2 & 51/2 \\ 8/3 & 2/3 & 9 \end{bmatrix}.$$

Pour déterminer l'inverse par calcul, il suffit de résoudre **AX = B** où **B** est la matrice identité 3×3.

Entrez tout d'abord le programme ci-dessus et, de retour en mode calcul, entrez les coefficients de la matrice **A** (matrice du système) et de la matrice **B** (matrice identité). Appuyez sur GSB A pour exécuter le programme.

Rappelez les éléments de la solution non-corrigée **C** :

$$\mathbf{C} = \begin{bmatrix} -9.666666881 & -2.666666726 & -32.00000071 \\ 8.000000167 & 2.500000046 & 25.50000055 \\ 2.666666728 & 0.6666666836 & 9.000000203 \end{bmatrix}.$$

Cette solution est correcte jusqu'au septième chiffre. Cette précision vérifie bien l'équation indiquée page 103.

$$(\text{nombre de chiffres corrects}) \geq 9 - \log(\|\mathbf{A}\|\,\|\mathbf{C}\|) - \log(3) \approx 4.8.$$

Rappelez ensuite les éléments de la solution corrigée, matrice **B** :

$$\mathbf{B} = \begin{bmatrix} -9.666666667 & -2.666666667 & -32.00000000 \\ 8.000000000 & 2.500000000 & 25.50000000 \\ 2.666666667 & 0.6666666667 & 9.000000000 \end{bmatrix}.$$

Après une itération de correction, la précision est de 10 chiffres.

Résolution d'un système d'équations non linéaires

Considérons un système de p équations non linéaires à p inconnues de la forme :

$$f_i(x_1, x_2, ..., x_p) = 0 \quad \text{pour } i = 1, 2, ..., p$$

posons

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}, \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_p(\mathbf{x}) \end{bmatrix}, \text{ et } \mathbf{F}(\mathbf{x}) = \begin{bmatrix} F_{11}(\mathbf{x}) \dots F_{1p}(\mathbf{x}) \\ F_{21}(\mathbf{x}) \dots F_{2p}(\mathbf{x}) \\ \vdots \quad\quad \vdots \\ F_{p1}(\mathbf{x}) \dots F_{pp}(\mathbf{x}) \end{bmatrix},$$

dans lequel

$$F_{ij}(\mathbf{x}) = \frac{\partial}{\partial x_j} f_i(\mathbf{x}) \quad \text{pour } i, j = 1, 2, ..., p.$$

Le système peut alors se mettre sous la forme $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Dans la méthode de Newton, il faut déterminer une solution initiale $\mathbf{x}^{(0)}$ de l'équation $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ et calculer

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{F}(\mathbf{x}^{(k)}))^{-1}\mathbf{f}(\mathbf{x}^{(k)}) \qquad \text{for } k = 0, 1, 2, ...$$

jusqu'à ce que $\mathbf{x}^{(k+1)}$ converge.

Le programme ci-dessous effectue une itération de la méthode de Newton. Il effectue le calcul sous la forme

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{d}^{(k)},$$

dans laquelle $\mathbf{d}^{(k)}$ est la solution du système linéaire $p \times p$.

$$\mathbf{F}(\mathbf{x}^{(k)})\mathbf{d}^{(k)} = \mathbf{f}(\mathbf{x}^{(k)}).$$

Ce programme affiche pour chaque itération la longueur euclidienne de $\mathbf{f}(\mathbf{x}^{(k)})$ et la correction $\mathbf{d}^{(k)}$.

Exemple : Prenons une variable y ayant une distribution normale, d'écart-type m et de variance v^2 inconnus. Construisons un test sans biais de l'hypothèse $v^2 = v_0^2$ sachant qu'il est possible que $v^2 \neq v_0^2$ pour une valeur v_0^2 particulière.

Pour un échantillon aléatoire de y constitué de $y_1, y_2, ..., y_n$, un test sans biais rejette cette hypothèse si :

$$s_n < x_1 v_0^2 \text{ ou } s_n > x_2 v_0^2,$$

où

$$s_n = \sum_{i=1}^{n} (y_i - \bar{y})^2 \quad \text{et} \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \ ,$$

pour certaines constantes x_1 et x_2.

Si la taille du test est a ($0 < a < 1$), vous pouvez trouver x_1 et x_2 en résolvant le système d'équations $f_1(\mathbf{x}) = f_2(\mathbf{x}) = 0$, où

$$f_1(\mathbf{x}) = (n-1)\ln(x_2/x_1) + x_1 - x_2$$

$$f_2(\mathbf{x}) = \int_{x_1}^{x_2} (w/2)^m \exp(-w/2)dw \quad - \quad 2(1-a)\Gamma(m+1).$$

Ici, $x_2 > x_1 > 0$, a et n sont connus ($n > 1$), et $m = (n-1)/2 - 1$.

Une bonne valeur initiale de (x_1, x_2) est:

$$x_1^{(0)} = \chi^2_{n-1,\,a/2} \quad \text{et} \quad x_2^{(0)} = \chi^2_{n-1,\,1-a/2}$$

où $\chi^2_{d,\,p}$ est le $p$ième pourcent de la distribution du chi-carré avec d degrés de liberté.

Pour cet exemple,

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} 1-(n-1)/x_1 & (n-1)/x_2 - 1 \\ -(x_1/2)^m \exp(-x_1/2) & (x_2/2)^m \exp(-x_2/2) \end{bmatrix}.$$

Introduisez le programme suivant:

Appuyez sur	Affichage	
g P/R		Mode programme.
f CLEAR PRGM	000–	
f LBL A	001–42,21,11	
2	002– 2	
ENTER	003– 36	
f DIM C	004–42,23,13	Dimensionne à 2 × 2 la matrice F.
1	005– 1	
f DIM B	006–42,23,12	Dimensionne à 2 × 1 la matrice f.

Appuyez sur	Affichage	
GSB B	007- 32 12	Calcule **f** et **F**.
RCL MATRIX A	008-45,16,11	
RCL MATRIX B	009-45,16,12	
RCL MATRIX C	010-45,16,13	
f RESULT D	011-42,26,14	
÷	012- 10	Calcule $\mathbf{d}^{(k)}$.
f RESULT A	013-42,26,11	
−	014- 30	Calcule $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{d}^{(d)}$.
g LSTx	015- 43 36	
f MATRIX 8	016-42,16, 8	Calcule $\|\mathbf{d}^{(k)}\|_F$.
RCL MATRIX B	017-45,16,12	
f MATRIX 8	018-42,16, 8	Calcule $\|\mathbf{f}(\mathbf{x}^{(k)})\|_F$.
g RTN	019- 43 32	
f LBL B	020-42,21,12	Programme de calcul de **f** et de **F**.
f MATRIX 1	021-42,16, 1	
f USER RCL A	022u 45 11	
f USER		
STO 4	023- 44 4	Stocke $x_1^{(k)}$ dans R_4.
f USER RCL A	024u 45 11	Saute la ligne suivante pour le dernier élément.
f USER		
STO 5	025- 44 5	Stocke $x_2^{(k)}$ dans R_5.
STO 5	026- 44 5	
−	027- 30	Calcule $x_1 - x_2$.
RCL 5	028- 45 5	
RCL ÷ 4	029-45,10, 4	
g LN	030- 43 12	Calcule $\ln(x_2/x_1)$.
RCL 2	031- 45 2	
1	032- 1	
−	033- 30	
×	034- 20	Calcule $(n-1)\ln(x_2/x_1)$.
+	035- 40	Calcule f_1.
STO B	036- 44 12	Stocke f_1 dans **B**.
1	037- 1	

Appuyez sur	Affichage	
RCL 2	038– 45 2	
1	039– 1	
−	040– 30	
RCL ÷ 4	041–45,10, 4	Calcule $(n-1)/x_1$.
−	042– 30	Calcule F_{11}.
f USER STO C	043u 44 13	Stocke F_{11} dans C.
f USER		
RCL 2	044– 45 2	
1	045– 1	
−	046– 30	
RCL ÷ 5	047–45,10, 5	Calcule $(n-1)/x_2$.
1	048– 1	
−	049– 30	Calcule F_{12}.
f USER STO C	050u 44 13	Stocke F_{12} dans C.
f USER		
RCL 4	051– 45 4	
RCL 5	052– 45 5	
f ∫ C	053–42,20,13	Calcule l'intégrale.
RCL 3	054– 45 3	
1	055– 1	
−	056– 30	
2	057– 2	
×	058– 20	Calcule $2(a-1)$.
RCL 2	059– 45 2	
3	060– 3	
−	061– 30	
2	062– 2	
÷	063– 10	Calcule m.
f x!	064– 42 0	Calcule $\Gamma(m+1)$.
×	065– 20	
+	066– 40	Calcule f_2.
STO B	067– 44 12	Stocke f_2 dans B.
RCL 4	068– 45 4	
GSB C	069– 32 13	
CHS	070– 16	Calcule F_{21}.
f USER STO C	071u 44 13	Stocke F_{21} dans C.
f USER		

Appuyez sur	Affichage	
RCL 5	072- 45 5	
GSB C	073- 32 13	Calcule F_{22}.
f USER STO C	074u 44 13	Stocke F_{22} dans C.
f USER		
g RTN	075- 43 32	Saute cette ligne.
g RTN	076- 43 32	
f LBL C	077-42,21,13	Programme d'évaluation de l'expression à intégrer.
2	078- 2	
÷	079- 10	
CHS	080- 16	
e^x	081- 12	Calcule $e^{-x/2}$.
g LSTx	082- 43 36	
CHS	083- 16	
RCL 2	084- 45 2	
3	085- 3	
−	086- 30	
2	087- 2	
÷	088- 10	Calcule m.
y^x	089- 14	
×	090- 20	Calcule $(x/2)^m e^{-x/2}$.
g RTN	091- 43 32	

Labels utilisés : A, B et C.

Registres utilisés : R_0 (rang), R_1 (colonne), R_2 (n), R_3 (a), R_4 ($x_1^{(k)}$) et R_5 ($x_2^{(k)}$).

Matrices utilisées : $\mathbf{A}(\mathbf{x}^{(k+1)})$, $\mathbf{B}(\mathbf{f}(\mathbf{x}^{(k)}))$, $\mathbf{C}(\mathbf{F}(\mathbf{x}^{(k)}))$, et $\mathbf{D}(\mathbf{d}^{(k)})$.

Exécutez maintenant le programme. Par exemple, choisissez les valeurs $n = 11$ et $a = 0.05$. Les valeurs initiales suggérées sont $x_1^{(0)} = 3.25$ et $x_2^{(0)} = 20.5$. N'oubliez pas que le format d'affichage affecte l'incertitude du calcul de l'intégrale.

Appuyez sur	Affichage	
g P/R		Mode calcul.
5 f DIM (i)	5.0000	Réserve R_0 à R_5.
11 STO 2	11.0000	Stocke n dans R_2.

.05 [STO] 3	0.0500		Stocke a dans R_3.
2 [ENTER] 1	1		
[f] [DIM] [A]	1.0000		Dimensionne A à 2 × 1.
[f] [USER]	1.0000		Active le mode USER.
[f] [MATRIX] 1	1.0000		
3.25 [STO] [A]	3.2500		Stocke $x_1^{(0)}$ de la distribution du chi-carré.
20.5 [STO] [A]	20.50000		Stocke $x_2^{(0)}$ de la distribution du chi-carré.
[f] [SCI] 4	2.0500	01	Définit le format d'affichage.
[A]	1.1677	00	Affiche la norme de $\mathbf{f}(\mathbf{x}^{(0)})$.
[R↓]	1.0980	00	Affiche la norme de la correction $\mathbf{d}^{(0)}$.
[RCL] [A]	3.5519	00	Rappelle $x_1^{(1)}$.
[RCL] [A]	2.1556	01	Rappelle $x_2^{(1)}$.

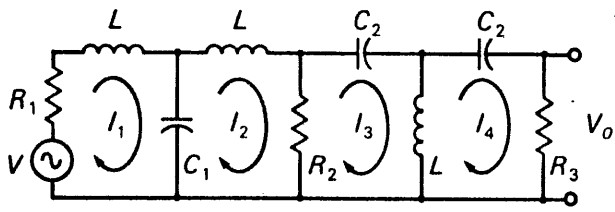
En répétant les quatre derniers pas de programmes, vous allez obtenir les résultats suivants :

k	$\|\mathbf{f}(\mathbf{x}^{(k)})\|_F$	$\|\mathbf{d}^{(k)}\|_F$	$x_1^{(k+1)}$	$x_2^{(k+1)}$
			3.2500	20.500
0	1.168	1.098	3.5519	21.556
1	1.105×10^{-1}	1.740×10^{-1}	3.5169	21.726
2	1.918×10^{-3}	2.853×10^{-3}	3.5162	21.729
3	6.021×10^{-7}	9.542×10^{-7}	3.5162	21.729

En réalité, vous n'aurez sans doute pas besoin de cette précision pour la plupart de vos problèmes. Ici, la troisième itération est suffisamment précise pour construire le test statistique. (Appuyez sur [f] [FIX] 4 pour ré-initialiser le format d'affichage et sur [f] [USER] pour désactiver le mode USER.

Résolution d'un grand système d'équations complexes

Exemple : Trouvez la tension de sortie d'une fréquence radian de $\omega = 15 \times 10^3$ rad/seconde pour le réseau de filtres illustré ci-dessous.



$V = 10$ volts
$R_1 = 100$ ohms
$R_2 = 10^6$ ohms
$R_3 = 10^5$ ohms
$L = 10^{-2}$ henry
$C_1 = 25 \times 10^{-8}$ farad
$C_2 = 25 \times 10^{-6}$ farad

Décrivez le circuit à l'aide de boucle de courant :

$$\begin{bmatrix} (R_1+i\omega L-i/\omega C_1) & (i/\omega C_1) & 0 & 0 \\ (i/\omega C_1) & (R_2+i\omega L-i/\omega C_1) & (-R_2) & 0 \\ 0 & (-R_2) & (R_2-i/\omega C_2+i\omega L) & (-i\omega L) \\ 0 & 0 & (-i\omega L) & (R_3+i\omega L-i/\omega C_2) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

résolvez ce système complexe pour I_1, I_2, I_3 et I_4. Alors, $V_0 = (R_3)(I_4)$. Comme ce système est trop grand pour une résolution par la méthode standard, la méthode suivante (décrite dans le manuel d'utilisation) est utilisée. Tout d'abord, introduisez la matrice du système dans la matrice **A** sous forme complexe et calculez son inverse. Remarquez que $\omega L = 150$, que $1/\omega C_1 = 800/3$ et que $1/\omega C_2 = 8/3$.

Appuyez sur	Affichage	
g P/R		Mode programme.
f CLEAR PRGM	000–	Efface la mémoire programme.

Appuyez sur	Affichage			
[g] [P/R]				Mode calcul.
0 [f] [DIM] [(i)]	0.0000			Dimensionne la mémoire pour une matrice maximale.
[f] [MATRIX] 0	0.0000			Dimensionne toutes les matrices à 0 × 0.
4 [ENTER] 8	8			
[f] [DIM] [A]	8.0000			Dimensionne la matrice **A** à 4 × 8.
[f] [MATRIX] 1	8.0000			
[f] [USER]	8.0000			Active le mode USER.
100 [STO] [A]	100.0000			Stocke $\mathrm{Re}(a_{11})$.
150 [ENTER]	150.0000			
800 [ENTER] 3 [÷]	266.6667			
[−] [STO] [A]	−116.6667			Stocke $\mathrm{Im}(a_{11})$.
⋮				
150 [ENTER]	150.0000			
8 [ENTER] 3 [÷]	2.6667			
[−] [STO] [A]	147.3333			Stocke $\mathrm{Im}(a_{44})$.
[RCL] [MATRIX] [A]	A	4	8	
[f] [$P_{y,x}$]	A	8	4	Transforme $\mathbf{A}^C$ en $\mathbf{A}^P$.
[f] [MATRIX] 2	A	8	8	Transforme $\mathbf{A}^P$ en $\tilde{\mathbf{A}}$.
[STO] [RESULT]	A	8	8	
[f] [1/x]	A	8	8	Calcule l'inverse de $\tilde{\mathbf{A}}$ dans **A**.

Supprimez la deuxième moitié des rangs de **A** pour avoir de la place pour stocker la matrice de second membre **B**.

Appuyez sur	Affichage	
4 [ENTER] 8	8	
[f] [DIM] [A]	8.0000	Redimensionne la matrice **A** à 4 × 8.
4 [ENTER] 2	2	
[f] [DIM] [B]	2.0000	Dimensionne la matrice **B** à 4 × 2.

Appuyez sur	Affichage			
f MATRIX 1	2.0000			
10 STO B	10.0000			Stocke Re(V). (Les autres éléments sont 0.)
RCL MATRIX A	A	4	8	
RCL MATRIX B	b	4	2	
f Py,x	b	8	1	Transforme B^C en B^P.
f MATRIX 2	b	8	2	Transforme B^P en $\tilde{B}$.
f RESULT C	b	8	2	
×	C	4	2	Calcule la solution dans C.
f MATRIX 4	C	2	4	Calcule la transposée.
f MATRIX 2	C	2	8	Transforme C en $\tilde{C}$.
1 ENTER 8	8			
f DIM C	8.0000			Redimensionne la matrice C à 1 × 8.
RCL RESULT	C	1	8	
f MATRIX 4	C	8	1	Calcule la transposée.
g Cy,x	C	4	2	Transforme C^P en C^C.

La matrice C contient les valeurs désirées de I_1, I_2, I_3 et I_4 sous forme rectangulaire. Leurs formes polaires sont faciles à calculer.

Appuyez sur	Affichage			
f MATRIX 1	C	4	2	Réinitialise R_0 et R_1.
f SCI 4	C	4	2	
RCL C	1.9950	−04		Rappelle Re(I_1).
RCL C	4.0964	−03		Rappelle Im(I_1).
$x \gtrless y$ g →P	4.1013	−03		Affiche $\|I_1\|$.
$x \gtrless y$	8.7212	01		Affiche Arg (I_1) en degrés.
RCL C	−1.4489	−03		
RCL C	−3.5633	−02		
$x \gtrless y$ g →P	3.5662	−02		Affiche $\|I_2\|$.
$x \gtrless y$	−9.2328	01		
RCL C	−1.4541	−03		
RCL C	−3.5633	−02		
$x \gtrless y$ g →P	3.5662	−02		Affiche $\|I_3\|$.

Appuyez sur	Affichage		
[x≷y]	-9.2337	01	
[RCL] [C]	5.3446	-05	
[RCL] [C]	-2.2599	-06	
[x≷y] [g] [→P]	5.3494	-05	Affiche $\|I_4\|$.
[x≷y]	-2.4212	00	
[x≷y] [EEX] 5 [×]	5.3494	00	Calcule $\|V_0\| - (R_3)\ \|I_4\|$.
[f] [FIX] 4	5.3494		
[f] [USER]	5.3494		Désactive le mode USER.

La tension de sortie est $5.3494 \angle -2.4212°$.

Moindres carrés par les équations normales

Le problème des moindres carrés sans contraintes est connu, en statistiques, sous le nom de *régression linéaire multiple.* Il utilise le modèle linéaire

$$y = \sum_{j=1}^{p} b_j x_j + r.$$

Ici, $b_1, ..., b_p$ sont les paramètres inconnus, $x_1, ..., x_p$ sont les variables indépendantes (ou "explicatives"), y est la variable dépendante (ou "de réponse") et r est l'erreur aléatoire ayant attendu la valeur $E(r) = 0$, variance σ^2.

Pour n observations de y et de $x, x_2, ..., x_p$, ce problème peut être exprimé sous la forme :

$$\mathbf{y} = \mathbf{Xb} + \mathbf{r}$$

où $\mathbf{y}$ est un vecteur de n éléments, $\mathbf{X}$ une matrice $n \times p$ et $\mathbf{r}$ un vecteur de n éléments composé des erreurs aléatoires inconnues statisfaisant à $E(\mathbf{r}) = \mathbf{0}$ et à $\text{Cov}(\mathbf{r}) = E(\mathbf{rr}^T) = \sigma^2\mathbf{I}_n$.

Si le modèle est correct et si $\mathbf{X}^T\mathbf{X}$ a une inverse, la solution calculée $\hat{\mathbf{b}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ pour les moindres carrés a les propriétés suivantes :

- $E(\hat{\mathbf{b}}) = \mathbf{b}$, si bien que $\hat{\mathbf{b}}$ est une estimation de $\mathbf{b}$.
- $\text{Cov}(\hat{\mathbf{b}}) = E((\hat{\mathbf{b}} - \mathbf{b})^T(\hat{\mathbf{b}} - \mathbf{b})) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$, matrice covariance de l'estimation $\hat{\mathbf{b}}$.

- $E(\hat{\mathbf{r}}) = 0$, où $\hat{\mathbf{r}} = \mathbf{y} - \mathbf{X}\hat{\mathbf{b}}$ est le vecteur des résiduels.
- $E(\|\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\|_F^2) = (n-p)\sigma^2$, si bien que $\hat{\sigma}^2 = \|\hat{\mathbf{r}}\|_F^2/(n-p)$ est une estimation $\hat{\mathbf{b}}$ de σ^2. Vous pouvez estimer $\text{Cov}(\hat{\mathbf{b}})$ en remplaçant σ^2 par $\hat{\sigma}^2$.

La somme totale des carrés $\|\mathbf{y}\|_F^2$ peut être découpée selon

$$\begin{aligned}\|\mathbf{y}\|_F^2 &= \mathbf{y}^T\mathbf{y}.\\ &= (\mathbf{y} - \mathbf{X}\hat{\mathbf{b}} + \mathbf{X}\hat{\mathbf{b}})^T(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}} + \mathbf{X}\hat{\mathbf{b}})\\ &= (\mathbf{y} - \mathbf{X}\hat{\mathbf{b}})^T(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) - 2\hat{\mathbf{b}}^T\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) + (\mathbf{X}\hat{\mathbf{b}})^T(\mathbf{X}\hat{\mathbf{b}})\\ &= \|\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\|_F^2 + \|\mathbf{X}\hat{\mathbf{b}}\|_F^2\\ &= \begin{pmatrix}\text{Somme des carrés}\\ \text{des résidus}\end{pmatrix} + \begin{pmatrix}\text{Somme des carrés}\\ \text{de la régression}\end{pmatrix}.\end{aligned}$$

Quand le modèle est vrai,

$$E(\|\mathbf{X}\hat{\mathbf{b}}\|_F^2/p) = \sigma^2 + \|\mathbf{X}\mathbf{b}\|_F^2/p > \sigma^2$$

et

$$E(\|\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\|_F^2/(n-p)) = \sigma^2$$

pour $\mathbf{b} \neq \mathbf{0}$. Lorsque le modèle simplifié $y = r$ est vrai, ces deux valeurs attendues sont égales à σ^2.

Vous pouvez tester l'hypothèse que le modèle simplifié est vrai (contre l'hypothèse que le modèle d'origine est vrai) en calculant le ratio F:

$$F = \frac{\|\mathbf{X}\hat{\mathbf{b}}\|_F^2/p}{\|\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\|_F^2/(n-p)}.$$

F va tendre à être plus grand lorsque le modèle d'origine est vrai ($\mathbf{b} \neq \mathbf{0}$) que lorsque le modèle simplifié est vrai ($\mathbf{b} = \mathbf{0}$). Rejetez l'hypothèse lorsque F est suffisamment grand.

Si les erreurs aléatoires ont une distribution normale, le ratio F a une distribution F centrée avec p et $(n-p)$ degrés de liberté si $\mathbf{b} = \mathbf{0}$ et une distribution non centrée si $\mathbf{b} \neq \mathbf{0}$. Un test statistique de l'hypothèse (avec une probabilité α de rejet incorrect de l'hypothèse) est de rejeter l'hypothèse si le ratio F est supérieur au 100 αième de la distribution centrée F avec p et $(n-p)$ degrés de liberté; sinon, acceptez l'hypothèse.

Le programme suivant ajuste le modèle linéaire à un ensemble de n points de données $x_{i1}, x_{i2}, ..., x_{ip}, y_i$ par la méthode des moindres carrés. Les paramètres $b_1, b_2, ..., b_p$ sont estimés par la solution $\hat{\mathbf{b}}$ aux équations normales $\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y}$. Le programme estime aussi σ^2 et la matrice covariance $\text{Cov}(\hat{\mathbf{b}})$ des paramètres. La somme des carrés de la régression et des résidus (*Reg SS* et *Res SS*) et les résidus sont également calculés.

Le programme a besoin de deux matrices :

Matrice A : $n \times p$ à rangs i $(x_{i1}, x_{i2}, ..., x_{ip})$ pour $i = 1, 2, ..., n$.

Matrice B : $n \times 1$ à éléments $i(y_i)$ pour $i = 1, 2, ..., n$.

Le programme a pour résultats :

Matrice A : inchangée.

Matrice B : $n \times 1$ contenant les résidus de l'ajustement $(y_i - \hat{b}_1 x_{i1} - ... - \hat{b}_p x_{ip})$ pour $i = 1, 2, ..., n$ où $\hat{b}_i$ est la valeur estimée de b_i.

Matrice C : matrice covariance $p \times p$ des valeurs estimées des paramètres.

Matrice D : $p \times 1$ contenant les valeurs estimées $\hat{b}_1, ..., \hat{b}_p$ des paramètres.

Registre T : contient une valeur estimée de σ^2.

Registre Y : contient la somme des carrés de la régression (*Reg SS*).

Registre X : contient la somme des carrés des résidus (*Res SS*).

Le tableau d'analyse de la variance figurant ci-dessous, découpe la somme totale des carrés (*Tot SS*) en somme de régression et somme de résidus. Vous pouvez utiliser ce tableau pour calculer le ratio F.

Tableau d'analyse de la variance

Source	Degrés de liberté	Somme des carrés	Carré moyen	Ratio F
Régression	p	*Reg SS*	$\frac{(Reg\ SS)}{p}$	$\frac{(Reg\ MS)}{(Res\ MS)}$
Résidu	$n - p$	*Res SS*	$\frac{(Res\ SS)}{(n-p)}$	
Total	n	*Tot SS*		

Le programme calcule la somme des carrés de la régression *non-ajustée* pour la moyenne parce qu'il ne doit pas y avoir de constante dans le modèle. Pour inclure une constante, introduisez dans le modèle une variable qui est identiquement égale à un. Le paramètre correspondant est alors la constante.

Pour calculer la somme des carrés de la régression *ajustée à la moyenne* pour un modèle avec constante, utilisez d'abord le programme pour ajuster le modèle et pour trouver la somme des carrés de la régression non ajustée. Ensuite, ajustez le modèle simplifié $y = b_1 + r$ en éliminant toutes les variables sauf celle qui est identiquement égale à un (b_1, par exemple) et calculez la somme des carrés de la régression pour ce modèle : $(Reg\ SS)_C$. La somme des carrés de la régression ajustée à la moyenne $(Reg\ SS)_A$ est égale à : $Reg\ SS - (Reg\ SS)_C$. Le tableau d'analyse de la variance devient donc :

Tableau d'analyse de la variance

Source	Degrés de liberté	Somme des carrés	Carré moyen	Ratio *F*
Régression| Constante	$p-1$	$(Reg\ SS)_A$	$\frac{(Reg\ SS)_A}{(p-1)}$	$\frac{(Reg\ MS)_A}{(Res\ MS)}$
Constante	1	$(Reg\ SS)_C$	$(Res\ SS)_C$	
Résidu	$n-p$	$Res\ SS$	$\frac{(Res\ SS)}{(n-p)}$	
Total	n	$Tot\ SS$		

Vous pouvez ensuite utiliser le ratio *F* pour tester si la totalité du modèle s'ajuste beaucoup mieux aux points que le modèle simplifié $y = b_1 + r$.

Vous souhaiterez peut-être effectuer une série de régressions, en éliminant les variables indépendantes entre chaque régression. Pour cela, classez les variables dans l'ordre inverse de leur élimination dans le modèle. Elles peuvent être éliminées par transposition de la matrice **A**, redimensionnement de **A** avec réduction du nombre de rangs puis seconde transposition de **A**.

Vous aurez besoin des valeurs des variables dépendantes originelles pour chaque régression. S'il n'y a pas assez de place pour stocker les données d'origine dans la matrice **E**, vous pouvez faire le calcul à partir du résultat de la régression. Un sous-programme a été ajouté dans ce but.

Ce programme a les caractéristiques suivantes :

- Si la totalité du programme est introduite dans la mémoire programme, les tailles de n et p doivent satisfaire $n \geq p$ et $(n+p)(p+1) \leq 56$, c'est-à-dire que :

si p est	1	2	3	4
alors n_{max} est :	27	16	11	7

Ceci suppose que seuls les registres de stockage R_0 et R_1 ont été alloués. Si le sous-programme "B" n'est pas introduit, alors $n \geq p$ et $(n+p)(p+1) \leq 58$, c'est-à-dire que :

si p est	1	2	3	4
alors n_{max} est :	28	17	11	7

- Même si le sous-programme "B" utilise la fonction résiduelle avec sa précision étendue, les valeurs calculées de la variable dépendante peuvent ne pas correspondre exactement aux données d'origine. La correspondante sera cependant habituellement suffisante pour une estimation et des tests statistiques. Si vous désirez une meilleure précision, vous pouvez réintroduire les données d'origine dans la matrice **B**.

Appuyez sur	**Affichage**	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] [A]	001–42,21,11	Programme d'ajustement du modèle.
[RCL] [MATRIX] [B]	002–45,16,12	
[f] [MATRIX] 8	003–42,16, 8	
[g] [x^2]	004– 43 11	Calcule *Tot SS*.
[RCL] [MATRIX] [A]	005–45,16,11	
[ENTER]	006– 36	
[f] [RESULT] [C]	007–42,26,13	
[f] [MATRIX] 5	008–42,16, 5	Calcule $\mathbf{C} = \mathbf{A}^T\mathbf{A}$.
[g] [LSTx]	009– 43 36	
[RCL] [MATRIX] [B]	010–45,16,12	
[f] [RESULT] [D]	011–42,26,14	
[f] [MATRIX] 5	012–42,16, 5	Calcule $\mathbf{D} = \mathbf{A}^T\mathbf{B}$.
[$x \gtrless y$]	013– 34	

Appuyez sur	Affichage	
[÷]	014– 10	Calcule les paramètres dans D.
[RCL] [MATRIX] [A]	015–45,16,11	
[x≷y]	016– 34	
[f] [RESULT] [B]	017–42,26,12	
[f] [MATRIX] 6	018–42,16, 6	Calcule les résidus de l'ajustement dans B.
[f] [MATRIX] 8	019–42,16, 8	
[g] [x^2]	020– 43 11	Calcule *Res SS*.
[RCL] [DIM] [A]	021–45,23,11	
[−]	022– 30	
[÷]	023– 10	Calcule la valeur estimée de σ^2.
[ENTER]	024– 36	
[ENTER]	025– 36	
[RCL] [MATRIX] [C]	026–45,16,13	
[f] [RESULT] [C]	027–42,26,13	
[÷]	028– 10	Calcule la matrice covariance dans C.
[g] [R↑]	029– 43 33	
[RCL] [MATRIX] [B]	030–45,16,12	
[f] [MATRIX] 8	031–42,16, 8	
[g] [x^2]	032– 43 11	
[−]	033– 30	Calcule *Reg SS*.
[g] [LSTx]	034– 43 36	Donne *Res SS*.
[g] [RTN]	035– 43 32	
[f] [LBL] [B]	036–42,21,12	Sous-programme de reconstitution des valeurs de la variable dépendante.
[RCL] [MATRIX] [A]	037–45,16,11	
[RCL] [MATRIX] [D]	038–45,16,14	
[CHS]	039– 16	
[f] [RESULT] [B]	040–42,26,12	
[f] [MATRIX] 6	041–42,16, 6	Calcule B = B + AD.
[RCL] [MATRIX] [D]	042–45,16,14	
[CHS]	043– 16	
[g] [RTN]	044– 43 32	

Labels utilisés : A et B.

Registres utilisés : R_0 et R_1.

Matrices utilisées : **A, B, C** et **D**.

Pour utiliser ce programme :

1. Appuyez sur 1 [f] [DIM] [(i)] pour réserver les registres R_0 et R_1.
2. Dimensionnez la matrice **A** en fonction du nombre *n* d'observations et du nombre *p* de paramètres en appuyant sur *n* [ENTER] *p* [f] [DIM] [A].
3. Dimensionnez la matrice **B** en fonction du nombre *n* d'observations (et une colonne) en appuyant sur *n* [ENTER] 1 [f] [DIM] [B].
4. Appuyez sur [f] [MATRIX] 1 pour initialiser les registres R_0 et R_1.
5. Appuyez sur [f] [USER] pour activer le mode USER.
6. Pour chaque observation, stockez les valeurs des variables *p* dans un rang de la matrice **A**. Répétez cela pour les *n* observations.
7. Stockez les valeurs de la variable dépendante dans la matrice **B**.
8. Appuyez sur [A] pour calculer et afficher *Res SS*. Le registre Y contient *Reg SS* et le registre T contient la valeur estimée de σ^2.
9. Appuyez sur [RCL] [D] pour observer chacune des valeurs estimées des paramètres *p*.
10. Si vous le désirez, appuyez sur [B] pour recalculer les données de la variable dépendante dans la matrice **B**.

Exemple : Comparez deux modèles de régression sur la variation annuelle de l'indice des prix à la consommation (IPC) en utilisant la variation annuelle de l'indice des prix à la production (IPP) et le taux de chômage (TC) :

$$y = b_1 + b_2x_2 + b_3x_3 + r \quad \text{et} \quad y = b_1 + b_2x_2 + r$$

où y, x_2 et x_3 représentent respectivement IPC, IPP et TC (tous sous forme de pourcentages). Utilisez les données suivantes :

Année	IPC	IPP	TC
1969	5.4	3.9	3.5
1970	5.9	3.7	4.9
1971	4.3	3.3	5.9
1972	3.3	4.5	5.6
1973	6.2	13.1	4.9
1974	11.0	18.9	5.6
1975	9.1	9.2	8.5
1976	5.8	4.6	7.7
1977	6.5	6.1	7.0
1978	7.6	7.8	6.0
1979	11.5	19.3	5.8

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
[f] [MATRIX] 0		
11 [ENTER] 3	**3**	
[f] [DIM] [A]	**3.0000**	Dimensionne **A** à 11 × 3.
11 [ENTER] 1	**1**	
[f] [DIM] [B]	**1.0000**	Dimensionne **B** à 11 × 1.
[f] [MATRIX] 1	**1.0000**	
[f] [USER]	**1.0000**	
1 [STO] [A]	**1.0000**	Introduit les données de la variable indépendante.
3.9 [STO] [A]	**3.9000**	
3.5 [STO] [A]	**3.5000**	
⋮	⋮	
1 [STO] [A]	**1.0000**	
19.3 [STO] [A]	**19.3000**	
5.8 [STO] [A]	**5.8000**	
5.4 [STO] [B]	**5.4000**	Introduit les données de la variable dépendante.
5.9 [STO] [B]	**5.9000**	
⋮	⋮	
11.5 [STO] [B]	**11.5000**	
[A] [f] [FIX] 9	**13.51217504**	*Res SS* pour tout le modèle.
[R↓]	**587.9878252**	*Reg SS* pour tout le modèle.

Appuyez sur	Affichage	
R↓ R↓	1.689021880	σ^2 estimée.
RCL D	1.245864326	b_1 estimée.
RCL D	0.379758235	b_2 estimée.
RCL D	0.413552218	b_3 estimée.
B	d 3 1	Recalcule les données dépendantes.
RCL MATRIX A	A 11 3	
f MATRIX 4	A 3 11	
2 ENTER 11	11	
f DIM A	11.00000000	Élimine la dernière colonne de A.
RCL MATRIX A	A 2 11	
f MATRIX 4	A 11 2	Nouvelle matrice A.
A	16.78680552	*Res SS* pour le modèle réduit.
R↓	584.7131947	*Reg SS* pour le modèle réduit.
R↓ R↓	1.865200613	σ^2 estimée.
RCL D	3.701730745	b_1 estimée.
RCL D	0.380094935	b_2 estimée.
B	d 2 1	Recalcule les données dépendantes.
RCL MATRIX A	A 11 2	
f MATRIX 4	A 2 11	
1 ENTER 11	11	
f DIM A	11.00000000	Abandonne la colonne suivante de A.
RCL MATRIX A	A 1 11	
f MATRIX 4	A 11 1	Nouvelle matrice A.
A	68.08545454	*Res SS*.
R↓	533.4145457	*Reg SS* pour la constante.
R↓ R↓	6.808545454	σ^2 estimée.
RCL D	6.963636364	b_1 estimée.
f USER	6.963636364	Désactive le mode USER.
f FIX 4	6.9636	

Reg SS pour la variable IPP ajustée pour le terme constant est:
(*Reg SS* pour le modèle réduit) − (*Reg SS* pour la constante) =
51.29864900.

Reg SS pour la variable TC ajustée pour la variable IPP et le terme constant est :

(*Reg SS* pour le modèle complet) — (*Reg SS* pour le modèle réduit)
= 3.274630500.

Établissez maintenant le tableau d'analyse de la variance suivant :

Source	Degrés de liberté	Somme des carrés	Carré moyen	Ratio *F*
TC| IPP, Constante	1	3.2746305	3.2746305	1.939
IPP|Constante	1	51.2986490	51.2986490	30.37
Constante	1	533.4145457	533.4145457	315.8
Résidu (modèle complet)	8	13.5121750	1.68902188	
Total	11	601.5000002		

Le ratio *F* pour le taux de chômage, ajusté pour la variation de l'indice des prix à la production et la constante, n'est pas très significatif statistiquement parlant au seuil significatif de 10 % ($\alpha = 0.1$). L'introduction du taux de chômage dans le modèle n'améliore pas de façon significative l'ajustement de IPC.

Cependant, le ratio *F* pour l'indice des prix à la production ajusté pour la constante, *est* significatif au seuil de 0.1 % ($\alpha = 0.001$). L'introduction de IPP dans le modèle n'améliore pas de façon significative l'ajustement de IPC.

Moindres carrés par les rangs successifs

Ce programme utilise la factorisation orthogonale pour résoudre le problème des moindres carrés. C'est-à-dire qu'il cherche les paramètres $b_1, ..., b_p$ minimisant la somme des carrés $\|\mathbf{r}\|_F^2 = (\mathbf{y} - \mathbf{Xb})^T(\mathbf{y} - \mathbf{Xb})$, les données du modèle étant données.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \quad \text{et} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Le programme traite les valeurs croissantes successives de n, bien que la solution $\mathbf{b} = \mathbf{b}^{(n)}$ n'ait un sens que pour $n \geqslant p$.

Il est possible de mettre en facteur la matrice [X y] de dimensions $n \times (p+1)$ ainsi augmentée dans $\mathbf{Q}^T\mathbf{V}$, où $\mathbf{Q}$ est une matrice orthogonale).

$$\mathbf{V} = \begin{bmatrix} \hat{\mathbf{U}} & \mathbf{g} \\ \mathbf{0} & q \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{matrix} (p \text{ rangs}) \\ (1 \text{ rang}) \\ (n-p-1 \text{ rangs}) \end{matrix}$$

et $\hat{\mathbf{U}}$ est une matrice triangulaire supérieure. Si cette factorisation résulte de l'introduction de n rangs $\mathbf{r}_m = (x_{m1}, x_{m2}, \dots, x_{mp}, y_m)$ pour $m = 1, 2, \dots, n$ dans [X y], considérez comment avancer à $n+1$ rang en ajoutant le rang $\mathbf{r}_{n+1}$ à [X y] :

$$\begin{bmatrix} \mathbf{X} & \mathbf{y} \\ \mathbf{r}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{r}_{n+1} \end{bmatrix}.$$

Les rangs zéro de V sont supprimés.

Multipliez la matrice $(p+2) \times (p+1)$

$$\mathbf{A} = \begin{bmatrix} \hat{\mathbf{U}} & \mathbf{g} \\ \mathbf{0} & q \\ \mathbf{r}_{n+1} \end{bmatrix} \begin{matrix} (p \text{ rangs}) \\ (1 \text{ rang}) \\ (1 \text{ rang}) \end{matrix}$$

par un produit de matrices orthogonales élémentaires, chacune d'elle différant de la matrice identité $\mathbf{I}_{p+2}$ dans seulement deux rangs et deux colonnes. Pour $k = 1, 2, ..., p + 1$ successivement, la $k$ième matrice orthogonale agit sur les $k$ièmes et derniers rangs pour supprimer le $k$ième élément du dernier rang pour modifier les éléments suivants de ce dernier rang. La $k$ième matrice orthogonale est de la forme

$$\begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & & & s & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & 0 & & & & \ddots & & \\ & & & & & & & 1 & \\ & & & -s & & & & & c \end{bmatrix}$$

où $c = \cos(\theta)$, $s = \sin(\theta)$, et $\theta = \tan^{-1}(a_{p+2,k}/a_{kk})$. Après que $p+1$ de ces facteurs aient été appliqués à la matrice **A**, celle-ci devient :

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{U}^* & \mathbf{g}^* \\ \mathbf{0} & q^* \\ \mathbf{0} & 0 \end{bmatrix} \begin{matrix} (p \text{ rangs}) \\ (1 \text{ rang}) \\ (1 \text{ rang}) \end{matrix}$$

où $\mathbf{U}^*$ est également une matrice triangulaire supérieure. Vous pouvez obtenir la solution $\mathbf{b}^{(n+1)}$ au système de $p + 1$ rangs incrémenté en résolvant :

$$\begin{bmatrix} \mathbf{U}^* & \mathbf{g}^* \\ \mathbf{0} & q^* \end{bmatrix} \begin{bmatrix} \mathbf{b}^{(n+1)} \\ -1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -q^* \end{bmatrix}.$$

En remplaçant le dernier rang de $\mathbf{A}^*$ par $\mathbf{r}_{n+2}$ et en répétant la factorisation, vous pouvez continuer à ajouter des rangs de données dans le système. Vous pouvez ajouter indéfiniment des rangs sans augmenter l'espace de stockage nécessaire.

Le programme suivant commence avec $n = 0$ et $\mathbf{A} = \mathbf{0}$. Vous introduisez les rangs $\mathbf{r}_m$ au fur et à mesure pour $m = 1, 2, ..., p-1$ respectivement. Vous obtenez alors la solution **b** pour chaque introduction de nouveau rang.

Vous pouvez également résoudre des problèmes de moindres carrés pondérés ainsi que des problèmes de moindres carrés avec des contraintes linéaires à l'aide de ce programme. Il vous suffit de procéder aux substitutions nécessaires décrites dans le paragraphe "Factorisation orthogonale", plus haut dans ce chapitre.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] [A]	001–42,21,11	Programme d'introduction d'un nouveau rang.
[STO] 2	002– 44 2	Stocke la pondération dans R_2.
1	003– 1	
[STO] 1	004– 44 1	Stocke $l = 1$ dans R_1.
[f] [LBL] 4	005–42,21, 4	
[RCL] [DIM] [A]	006–45,23,11	
[$x \gtrless y$]	007– 34	
[STO] 0	008– 44 0	Stocke $k = p + 2$ dans R_0.
[f] [LBL] 5	009–42,21, 5	
[RCL] 1	010– 45 1	
[R/S]	011– 31	
[RCL] 2	012– 45 2	
[×]	013– 20	
[f] [USER] [STO] [A] [f] [USER]	014u 44 11	
[GTO] 5	015– 22 5	
[GTO] 4	016– 22 4	
[f] [LBL] [B]	017–42,21,12	Programme de mise à jour de la matrice A.
[RCL] [DIM] [A]	018–45,23,11	Rappelle les dimensions $p + 2$ et $p + 1$.
[$x \gtrless y$]	019– 34	
[STO] 2	020– 44 2	Stocke $p + 2$ dans R_2.
[f] [MATRIX] 1	021–42,16, 1	Stocke $k = l = 1$.
[f] [LBL] 1	022–42,21, 1	Branchement à la mise à jour du $i$ième rang.
[g] [CF] 0	023–42, 5, 0	
[RCL] 2	024– 45 2	
[RCL] 2	025– 45 0	
[RCL] [g] [A]	026–45,43,11	Rappelle $a_{p+2,\,k}$.
[RCL] [A]	027– 45 11	Rappelle a_{kk}.

Appuyez sur	Affichage	
g TEST 2	028–43,30, 2	Teste $a_{kk} < 0$.
g SF 0	029–43, 4, 0	Arme l'indicateur 0 pour un élément diagonal négatif.
g ABS	030– 43 16	
g →P	031– 43 1	Calcule θ.
g CLx	032– 43 35	
1	033– 1	
f →R	034– 42 1	Calcule $x = \cos\theta$ et $y = \sin\theta$.
g F? 0	035–43, 6, 0	
CHS	036– 16	Définit $x = c$ et $y = s$.
f I	037– 42 25	Forme $s + ic$.
R↓	038– 33	
f LBL 2	039–42,21, 2	Sous-programme de rotation du rang k.
g R↑	040– 43 33	
RCL A	041– 45 11	Rappelle a_{kl}.
RCL 2	042– 45 2	
RCL 1	043– 45 1	
RCL g A	044–45,43,11	Rappelle $a_{p+2,l}$.
f I	045– 42 25	Forme $a_{kl} - ia_{p+2,\,l}$.
×	046– 20	
RCL 2	047– 45 2	
RCL 1	048– 45 1	
STO g A	049–44,43,11	Stocke le nouveau a_{kl}.
f Re≷Im	050– 42 30	
f USER STO A f USER	051u 44 11	Stocke le nouveau $a_{p+2,\,l}$ et incrémente R_0 et R_1.
RCL 1	052– 45 1	Rappelle l (colonne).
RCL 0	053– 45 0	Rappelle k (rang).
g x≤y	054– 43 10	Teste $k \leqslant l$.
GTO 2	055– 22 2	Boucle arrière jusqu'à ce que la colonne soit remise à 1.
g CF 8	056–43, 5, 8	Désactive le mode complexe.
STO 1	057– 44 1	Stocke k dans R_1 (l).
RCL 2	058– 45 2	

Appuyez sur	Affichage	
[g] [$x \leq y$]	059– 43 10	Teste $p + 2 \leqslant k$.
[g] [RTN]	060– 43 32	Retourne au dernier rang.
[GTO] 1	061– 22 1	Boucle arrière jusqu'au dernier rang.
[f] [LBL] [C]	062–42,21,13	Programme de calcul de la solution courante.
[RCL] [DIM] [A]	063–45,23,11	
[ENTER]	064– 36	
[f] [DIM] [A]	065–42,23,11	Élimine le dernier rang de **A**.
[STO] 0	066– 44 0	Stocke $p + 1$ dans R_0.
[STO] 1	067– 44 1	Stocke $p + 1$ dans R_1.
1	068– 1	
[f] [DIM] [C]	069–42,23,13	Dimensionne la matrice **C** à $(p + 1) \times 1$.
0	070– 0	
[STO] [MATRIX] [C]	071–44,16,13	Définit la matrice **C** à **0**.
[EEX]	072– 26	
9	073– 9	
9	074– 9	
[CHS]	075– 16	Forme 10^{-99}.
[RCL] [A]	076– 45 11	Rappelle $q = a_{p+1,p+1}$.
[g] [$x = 0$]	077– 43 20	Teste $q = 0$.
[R↓]	078– 33	Utilise 10^{-99} si $q = 0$.
[CHS]	079– 16	
[RCL] 0	080– 45 0	
1	081– 1	
[STO] [g] [C]	082–44,43,13	Définit $c_{p+1,1} = -q$.
[RCL] [MATRIX] [C]	083–45,16,13	
[RCL] [MATRIX] [A]	084–45,16,11	
[f] [RESULT] [C]	085–42,26,13	
[÷]	086– 10	Stocke $\mathbf{A}^{-1}\mathbf{C}$ dans **C**.
[RCL] 0	087– 45 0	
1	088– 1	
[+]	089– 40	
[RCL] 0	090– 45 0	
[f] [DIM] [A]	091–42,23,11	Dimensionne la matrice **A** à $(p + 2) \times (p + 1)$.
1	092– 1	

Appuyez sur	Affichage	
[−]	093− 30	
1	094− 1	
[f] [DIM] [C]	095−42,23,13	Dimensionne la matrice C à $p \times 1$.
[RCL] [A]	096− 45 11	Rappelle q.
[f] [MATRIX] 1	097−42,16, 1	Définit $k = l = 1$.
[g] [RTN]	098− 43 32	

Labels utilisés : A, B, C et 1 à 5.

Registres utilisés : R_0, R_1 et R_2 ($p + 2$ et w).

Matrices utilisées : **A** (matrices de travail) et **C** (valeurs estimées des paramètres).

Indicateurs utilisés : 0 et 8.

Après le stockage de ce programme, le HP-15C dispose de suffisamment de mémoire pour travailler avec jusqu'à $p = 4$ paramètres. Si les programmes "A" et "C" sont supprimés, vous pouvez travailler avec $p = 5$ paramètres. Dans l'un et l'autre cas, il n'y a aucune limite au nombre de rangs que vous pouvez introduire.

Pour utiliser ce programme :

1. Appuyez sur 2 [f] [DIM] [(i)] pour réserver les registres R_0 à R_2.
2. Appuyez sur [f] [USER] pour activer le mode USER.
3. Introduisez $(p + 2)$ et $(p + 1)$ dans la pile, puis appuyez sur [f] [DIM] [A] pour dimensionner la matrice **A**. Ces dimensions dépendent du nombre p de paramètres que vous utilisez.
4. Appuyez sur 0 [STO] [MATRIX] [A] pour initialiser la matrice **A**.
5. Introduisez la pondération w_k du rang courant et appuyez sur [A]. L'Affichage doit afficher 1.0000 pour indiquer que le programme est prêt pour le premier élément du rang. (Pour les problèmes classiques de moindres carrés, utilisez $w_k = 1$ pour chaque rang.)
6. Introduisez les éléments du rang m de la matrice **A** en appuyant sur x_{m1} [R/S] x_{m2} [R/S] ... x_{mp} [R/S] y_m [R/S]. Après chaque élément nouvellement introduit, l'affichage doit afficher l'indice du prochain élément à introduire. (Si vous faites une faute en introduisant les éléments, revenez en arrière et répétez les étapes 5 et 6 pour le rang considéré.)
7. Appuyez sur [B] pour mettre à jour la factorisation pour ajouter le rang introduit lors des deux étapes précédentes.

8. Éventuellement, appuyez sur [C] [g] [x^2] pour calculer et afficher la somme des carrés des résidus q^2 et pour calculer la solution **b** courante. Appuyez ensuite p fois sur [RCL] [C] pour afficher b_1, b_2, ..., b_p successivement.
9. Répétez les étapes 5 à 8 pour chaque nouveau rang.

Exemple : Utilisez ce programme et les données IPC (indice des prix à la consommation) de l'exemple précédent pour ajuster le modèle

$$y = b_1 + b_2x_2 + b_3x_3 + r,$$

où y, x_2 et x_3 représentent respectivement IPC, IPP et TC (tous en pourcentages).

Ce problème mettant en œuvre $p = 3$ paramètres, la matrice **A** doit être une matrice 5×4. Les rangs de la matrice **A** sont $(1, x_{m2}, x_{m3}, y_m)$ pour $m = 1, 2, ..., 11$. Chaque rang est pondéré à $w_m = 1$.

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
2 [f] [DIM] [(i)]	2.0000	Réserve les registres R_0 à R_2.
[f] [USER]	2.0000	Active le mode USER.
[f] [MATRIX] 0	2.0000	Efface la mémoire matrice.
5 [ENTER] 4	4	
[f] [DIM] [A]	4.0000	Dimensionne la matrice **A** à 5×4.
0 [STO] [MATRIX] [A]	0.0000	Stocke zéro dans tous les éléments.
1 [A]	1.0000	Introduit la pondération du rang 1.
1 [R/S]	2.0000	Introduit x_{11}.
3.9 [R/S]	3.0000	Introduit x_{12}.
3.5 [R/S]	4.0000	Introduit x_{13}.
5.4 [R/S]	1.0000	Introduit y_1.
[B]	5.0000	Met à jour la factorisation.
⋮	⋮	
1 [A]	1.0000	Introduit la pondération du rang 11.
1 [R/S]	2.0000	Introduit $x_{11,1}$.
19.3 [R/S]	3.0000	Introduit $x_{11,2}$.
5.8 [R/S]	4.0000	Introduit $x_{11,3}$.
11.5 [R/S]	1.0000	Introduit y_{11}.
[B]	5.0000	Met à jour la factorisation.

Appuyez sur	Affichage	
C	**3.6759**	Calcule les valeurs estimées courantes et q.
f FIX 9	**3.675891055**	
g x^2	**13.51217505**	Calcule la somme des carrés des résidus q^2.
RCL C	**1.245864306**	Affiche $b_1^{(11)}$.
RCL C	**0.379758235**	Affiche $b_2^{(11)}$.
RCL C	**0.413552221**	Affiche $b_3^{(11)}$.

Ces valeurs estimées concordent (sur 3 des neuf chiffres significatifs) avec les résultats de l'exemple précédent qui utilise l'équation normale. En outre, vous pouvez ajouter des données supplémentaires et mettre à jour les valeurs estimées des paramètres. Par exemple, ajoutez les données suivantes pour l'année 1968 : IPC = 4.2, IPP = 2.5 et TC = 3.6.

Appuyez sur	AFfichage	
1 A	**1.000000000**	Introduit la pondération de rang pour les nouveaux rangs.
1 R/S	**2.000000000**	Introduit $x_{12,1}$.
2.5 R/S	**3.000000000**	Introduit $x_{12,2}$.
3.6 R/S	**4.000000000**	Introduit $x_{12,3}$.
4.2 R/S	**1.000000000**	Introduit y_{12}.
B	**5.000000000**	Met à jour la factorisation.
C	**3.700256908**	
g x^2	**13.69190119**	Calcule la somme des carrés des résidus.
RCL C	**1.581596327**	Affiche $b_1^{(12)}$.
RCL C	**0.373826487**	Affiche $b_2^{(12)}$.
RCL C	**0.370971848**	Affiche $b_3^{(12)}$.
f FIX 4	**0.3710**	
f USER	**0.3710**	Invalide le mode USER

Valeurs propres d'une matrice réelle symétrique

Les valeurs propres d'une matrice carrée A sont les racines λ_j de son équation caractéristique

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0.$$

Quand **A** est réelle et symétrique ($\mathbf{A} = \mathbf{A}^T$), ses valeurs propres λ_j sont toutes réelles et possèdent des vecteurs propres $\mathbf{q}_j$ orthogonaux. Alors :

$$\mathbf{A}\mathbf{q}_j = \lambda_j \mathbf{q}_j$$

et

$$\mathbf{q}_j^T \mathbf{q}_k = \begin{cases} 0 & \text{if } j \neq k \\ 1 & \text{if } j = k. \end{cases}$$

Les vecteurs propres ($\mathbf{q}_1, \mathbf{q}_2, \ldots$) constituent les colonnes d'une matrice orthogonale **Q** qui satisfait :

$$\mathbf{Q}^T\mathbf{A}\mathbf{Q} = \text{diag}\,(\lambda_1, \lambda_2, \ldots)$$

et

$$\mathbf{Q}^T = \mathbf{Q}^{-1}.$$

Une variation orthogonale des variables $\mathbf{x} = \mathbf{Q}\mathbf{z}$, qui est équivalent à une rotation des axes, fait varier l'équation d'une famille d'aires quadratiques ($\mathbf{x}^T\mathbf{A}\mathbf{x} =$ constante) de la forme :

$$\mathbf{z}^T(\mathbf{Q}^T\mathbf{A}\mathbf{Q})\mathbf{z} = \sum_j^k \lambda_j z_j^2 = \text{constante}.$$

Avec l'équation sous cette forme, vous pouvez reconnaître de quelles sortes d'aires il s'agit (ellipsoïdes, hyperboloides, paraboloïdes, cones, cylindres, plans) puisque les demi-axes de l'aire se trouvent le long des nouveaux axes de coordonnées.

Le programme ci-dessous commence avec une matrice **A** donnée qui est supposés symétrique (si elle ne l'est pas, elle est remplacée par $(\mathbf{A} + \mathbf{A}^T)/2$ qui, elle, *est* symétrique).

Étant donnée une matrice symétrique **A**, le programme construit une matrice anti-symétrique (c'est-à-dire, pour laquelle $\mathbf{B} = -\mathbf{B}^T$) en utilisant la formule :

$$b_{ij} = \begin{cases} \tan(¼ \tan^{-1}(2a_{ij}/(a_{ii} - a_{jj}))) & \text{si } i \neq j \text{ et } a_{ij} \neq 0 \\ 0 & \text{si } i = j \text{ ou } a_{ij} = 0. \end{cases}$$

Ensuite, $\mathbf{Q} = 2(\mathbf{I} + \mathbf{B})^{-1} - \mathbf{I}$ doit être une matrice orthogonale dont les colonnes sont une bonne approximation des valeurs propres de **A** ; plus sont petits tous les éléments de **B**, meilleure est l'approximation. $\mathbf{Q}^T\mathbf{A}\mathbf{Q}$ doit donc être proche d'une matrice diagonale que **A** mais avec les mêmes valeurs propres.

Si $\mathbf{Q}^T\mathbf{AQ}$ n'est pas suffisamment proche de la diagonale, elle est utilisée à la place de **A** précédente pour répéter le processus.

De cette façon, des transformations orthogonales successives $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3, \ldots$ sont effectuées sur **A** pour produire une suite $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \ldots$, où :

$$\mathbf{A}_j = (\mathbf{Q}_1\,\mathbf{Q}_2 \ldots\, \mathbf{Q}_j)^T \mathbf{A}\mathbf{Q}_1\mathbf{Q}_2 \ldots\, \mathbf{Q}_j$$

avec chaque $\mathbf{A}_j$ successive plus diagonale que celle qui la précède.

Ce processus aboutit normalement à des matrices anti-symétriques dont les éléments sont tous petits, $\mathbf{A}_j$ convergeant rapidement vers une matrice diagonale **A**. Cependant, si certaines valeurs propres d'une matrice **A** sont très proches les unes des autres mais très écartées des autres valeurs, la convergeance est lente ; heureusement cette situation est rare.

Le programme s'arrête après chaque itération pour afficher

$$\tfrac{1}{2}\sum_j |\text{éléments hors diagonale de } \mathbf{A}_j| \,/\, \|\mathbf{A}_j\|_F$$

qui mesure la façon dont $\mathbf{A}_j$ est diagonale. Si cette mesure n'est pas négligeable, vous pouvez appuyer sur [R/S] pour calculer $\mathbf{A}_{j+1}$; si elle est négligeable, alors les éléments diagonaux de $\mathbf{A}_j$ sont proches des valeurs propres de **A**. Le programme n'a besoin que d'une itération pour des matrices 1×1 ou 2×2 et rarement plus de six pour des matrices 3×3. Pour les matrices 4×4, le programme prend légèrement plus de temps et utilise toute la mémoire disponible ; 6 ou 7 itérations sont généralement suffisantes, mais si certaines valeurs propres sont très proches les unes des autres mais relativement loin des autres valeurs, il faudra vraisemblablement entre 10 et 16 itérations.

Appuyez sur	**Affichage**	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] [A]	001–42,21,11	
[RCL] [MATRIX] [A]	002–45,16,11	
[STO] [MATRIX] [B]	003–44,16,12	Dimensionne B.
[STO] [MATRIX] [C]	004–44,16,13	Dimensionne C.
[f] [MATRIX] 4	005–42,16, 4	Transpose A.
[RCL] [MATRIX] [B]	006–45,16,12	
[STO] [RESULT]	007– 44 26	
[+]	008– 40	

Appuyez sur	Affichage	
2	009– 2	
÷	010– 10	
STO MATRIX A	011–44,16,11	Calcule $A = (A + A^T)/2$.
f MATRIX 8	012–42,16, 8	Calcule $\|A\|_F$.
STO 2	013– 44 2	Stocke $\|A\|_F$ dans R_2.
g CLx	014– 43 35	
STO 3	015– 44 3	Initialise la somme des éléments hors-diagonale.
STO MATRIX C	016–44,16,13	Définit $C = 0$.
f MATRIX 1	017–42,16, 1	Définit $R_0 = R_1 = 1$.
f LBL 0	018–42,21, 0	Programme de construction de **Q**.
RCL 0	019– 45 0	
RCL 1	020– 45 1	
g TEST 5	021–43,30, 5	Teste rang = colonne.
GTO 3	022– 22 3	
g TEST 7	023–43,30, 7	Teste colonne > rang.
GTO 1	024– 22 1	
x≷y	025– 34	
RCL g B	026–45,43,12	
CHS	027– 16	
f USER STO B	028u 44 12	Définit $b_{ij} = -b_{ji}$.
f USER		
GTO 0	029– 22 0	
f LBL 1	030–42,21, 1	Programme pour colonne > rang.
RCL g A	031–45,43,11	
g ABS	032– 43 16	Calcule $\|a_{ij}\|$.
STO + 3	033–44,40, 3	Cumule la somme hors diagonale.
g LSTx	034– 43 36	
ENTER	035– 36	
+	036– 40	Calcule $2a_{ij}$.
RCL 0	037– 45 0	
ENTER	038– 36	
RCL g A	039–45,43,11	Rappelle a_{ii}.
RCL 1	040– 45 1	
ENTER	041– 36	

Appuyez sur	Affichage	
RCL g A	042–45,43,11	Rappelle a_{jj}.
−	043– 30	Calcule $a_{ii} - a_{jj}$.
g TEST 3	044–43,30, 3	Teste $x \geqslant 0$.
GTO 2	045– 22 2	
CHS	046– 16	Garde l'angle de rotation compris entre $-90°$ et $90°$.
x≷y	047– 34	
CHS	048– 16	
x≷y	049– 34	
f LBL 2	050–42,21, 2	
g →P	051– 43 1	Calcule l'angle de rotation.
g CLx	052– 43 35	
4	053– 4	
÷	054– 10	
TAN	055– 25	Calcule b_{ij}.
f USER STO B	056u 44 12	
f USER		
GTO 0	057– 22 0	
f LBL 3	058–42,21, 3	Programme pour rang = colonne.
1	059– 1	
STO C	060– 44 13	Définit $c_{ii} = 1$.
f USER STO B	061u 44 12	Définit $b_{ii} = 1$.
f USER		
GTO 0	062– 22 0	
RCL 3	063– 45 3	
RCL ÷ 2	064–45,10, 2	Calcule le ratio des éléments hors diagonale.
R/S	065– 31	Affiche ce ratio.
2	066– 2	
RCL MATRIX B	067–45,16,12	
÷	068– 10	
RCL MATRIX C	069–45,16,13	
−	070– 30	Calcule $\mathbf{B} = 2(\mathbf{I} + \text{antisymétrique})^{-1} - \mathbf{I}$.
RCL MATRIX A	071–45,16,11	
f RESULT C	072–42,26,13	
f MATRIX 5	073–42,16, 5	Calcule $\mathbf{C} = \mathbf{B}^T\mathbf{A}$.

Appuyez sur	Affichage	
[RCL] [MATRIX] [B]	074-45,16,12	
[f] [RESULT] [A]	075-42,26,11	
[×]	076- 20	Calcule $\mathbf{A} = \mathbf{B}^T\mathbf{AB}$.
[GTO] [A]	077- 22 11	

Labels utilisés : A, 0, 1, 2 et 3.
Registres utilisés : R_0, R_1, R_2 (somme hors diagonale) et R_3 ($\|\mathbf{A}_j\|_F$).
Matrices utilisées : $\mathbf{A}(\mathbf{A}_j)$, $\mathbf{B}(\mathbf{Q}_j)$ et **C**.

Pour utiliser ce programme :

1. Appuyez sur 4 [f] [DIM] [(i)] pour réserver les registres R_0 à R_4.
2. Appuyez sur [f] [USER] pour activer le mode USER.
3. Dimensionne et introduit les éléments de la matrice **A** en utilisant [f] [DIM] [A] et [STO] [A]. Les dimensions peuvent aller jusqu'à 4 × 4, du moment qu'il y a suffisamment de mémoire disponible pour les matrices **B** et **C** ayant également les mêmes dimensions.
4. Appuyez sur [A] pour calculer et afficher le ratio hors diagonale.
5. Appuyez plusieurs fois sur [R/S] jusqu'à ce que le ratio affiché soit négligeable c'est-à-dire inférieur à 10^{-8}.
6. Appuyez plusieurs fois sur [RCL] [A] pour obtenir les éléments de la matrice **A**. Les éléments diagonaux sont des valeurs propres.

Exemple : Quelle aire quadratique est décrite par l'équation ci-dessous ?

$$\mathbf{x}^T\mathbf{Ax} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= 2x_1x_2 + 4x_1x_3 + 2x_2^2 + 6x_2x_3 + 4x_3^3$$

$$= 7$$

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
4 [f] [DIM] [(i)]	4.0000	Alloue la mémoire.
[f] [USER]	4.0000	Active le mode USER.

Appuyez sur	Affichage		
3 [ENTER] [f] [DIM] [A]	3.0000		Dimensionne la matrice A à 3 × 3.
[f] [MATRIX] 1	3.0000		Définit R_0 et R_1 à 1.
0 [STO] [A]	0.0000		Introduit a_{11}.
1 [STO] [A]	1.0000		Introduit a_{12}.
⋮			
3 [STO] [A]	3.0000		Introduit a_{32}.
4 [STO] [A]	4.0000		Introduit a_{33}.
[A]	0.8660		Calcule le ratio — il est trop grand.
[R/S]	0.2304		2e tentative : trop grand.
[R/S]	0.1039		3e tentative : trop grand.
[R/S]	0.0060		4e tentative : trop grand.
[R/S]	3.0463	−05	5e tentative : trop grand.
[R/S]	5.8257	−10	Ratio négligeable.
[RCL] [A]	−0.8730		Rappelle $a_{11} = \lambda_1$.
[RCL] [A]	−9.0006	−10	Rappelle a_{12}.
[RCL] [A]	−2.0637	−09	Rappelle a_{13}.
[RCL] [A]	−9.0006	−10	Rappelle a_{21}.
[RCL] [A]	9.3429	−11	Rappelle $a_{22} = \lambda_2$.
[RCL] [A]	1.0725	−09	Rappelle a_{23}.
[RCL] [A]	−2.0637	−09	Rappelle a_{31}.
[RCL] [A]	1.0725	−09	Rappelle a_{32}.
[RCL] [A]	6.8730		Rappelle $a_{33} = \lambda_3$.
[f] [USER]	6.8730		Désactive le mode USER.

Dans le nouveau système d'axes, l'équation de l'aire quadratique est approximativement

$$-0.8730z_1^2 + 0z_2^2 + 6.8730z_3^2 = 7.$$

Il s'agit de l'équation d'un cyclindre hyperbolique.

Vecteurs propres d'une matrice réelle symétrique

Comme nous l'avons vu dans l'application précédente, une matrice réelle symétrique A a des valeurs propres réelles $\lambda_1, \lambda_2, \ldots$ et des vecteurs orthogonaux correspondants $\mathbf{q}_1, \mathbf{q}_2, \ldots$

Ce programme utilise l'itération inverse pour calculer un vecteur propre $\mathbf{q}_k$ correspondant à la valeur propre X_k et tel que $\|\mathbf{q}_k\|_R = 1$. Cette technique utilise un vecteur initial $\mathbf{z}^{(0)}$ pour calculer par itération les vecteurs $\mathbf{w}^{(n)}$ et $\mathbf{z}^{(n)}$ suivants à partir des équations

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{w}^{(n+1)} = \mathbf{z}^{(n)}$$

$$\mathbf{z}^{(n+1)} = s\mathbf{w}^{(n+1)} / \|\mathbf{w}^{(n+1)}\|_R$$

où s indique le signe de la première composante de $\mathbf{w}^{(n+1)}$ ayant la valeur absolue la plus grande. Les itérations continuent jusqu'à ce que $\mathbf{z}^{(n)}$ converge. Ce vecteur est un vecteur propre $\mathbf{q}_k$ correspondant à la valeur propre λ_k.

Il n'est pas nécessaire que la valeur utilisée pour λ_k soit exacte; le vecteur propre calculé est déterminé précisément en dépit de petites imprécisions dans λ_k. Par ailleurs, vous n'êtes pas obligé d'avoir une approximation de λ_k trop précise ; le HP-15C peut calculer le vecteur propre même lorsque $\mathbf{A} - \lambda_k\mathbf{I}$ est mal conditionnée.

Cette technique exige que le vecteur $\mathbf{z}^{(0)}$ ait une composante non nulle le long du vecteur (propre) $\mathbf{q}_k$ inconnu. Puisqu'il n'y a pas d'autres restrictions sur $\mathbf{z}^{(0)}$, le programme utilise des composantes aléatoires pour $\mathbf{z}^{(0)}$. A la fin de chaque itération, le programme affiche $\|\mathbf{z}^{(n+1)} - \mathbf{z}^{(n)}\|_R$ pour montrer la rapidité de la convergence.

Ce programme accepte une matrice A non symétrique à condition qu'elle ait une forme canonique de Jordan en diagonale, c'est-à-dire qu'il existe une matrice P non singulière telle que $\mathbf{P}^{-1}\mathbf{AP} = \text{diag}(\lambda_1, \lambda_2, \ldots)$.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000–	
[f] [LBL] [C]	001–42,21,13	
[STO] 2	002– 44 2	Stocke les valeurs propres dans R_2.
[RCL] [MATRIX] [A]	003–45,16,11	
[STO] [MATRIX] [B]	004–44,16,12	Stocke A dans B.
[RCL] [DIM] [A]	005–45,23,11	
[STO] 0	006– 44 0	
[f] [LBL] 4	007–42,21, 4	
[RCL] 0	008– 45 0	
[STO] 1	009– 44 1	
[RCL] [B]	010– 45 12	

Appuyez sur	Affichage	
RCL − 2	011-45,30, 2	
STO B	012– 44 12	Modifie les éléments diagonaux de B.
f DSE 0	013–42, 5, 0	
GTO 4	014– 22 4	
RCL DIM A	015–45,23,11	
1	016– 1	
f DIM C	017–42,23,13	Dimensionne C à $n \times 1$.
f MATRIX 1	018–42,16, 1	
f LBL 5	019–42,21, 5	
f RAN #	020– 42 36	
f USER STO C	021u 44 13	Stocke les composantes aléatoires dans C.
f USER		
GTO 5	022– 22 5	
f LBL 6	023–42,21, 6	Programme d'itération pour $\mathbf{z}^{(n)}$ et $\mathbf{w}^{(n)}$.
RCL MATRIX C	024–45,16,13	
STO MATRIX D	025–44,16,14	Stocke $\mathbf{z}^{(n)}$ dans D.
STO RESULT	026– 44 26	
RCL MATRIX B	027–45,16,12	
÷	028– 10	Calcule $\mathbf{w}^{(n+1)}$ dans C.
ENTER	029– 36	
f MATRIX 7	030–42,16, 7	
÷	031– 10	Calcule $\pm\, \mathbf{z}^{(n+1)}$ dans C.
f MATRIX 1	032–42,16, 1	
f LBL 7	033–42,21, 7	Programme de recherche du signe du plus grand élément.
f USER RCL C	034u 45 13	
f USER		
ENTER	035– 36	(Cette ligne est sautée pour le dernier élément.)
g ABS	036– 43 16	
1	037– 1	
g TEST 6	038–43,30, 6	Teste $\|a_j\| \neq 1$.
GTO 7	039– 22 7	
RCL MATRIX C	040–45,16,13	
g LSTx	041– 43 36	Rappelle a_j extrême.
÷	042– 10	Calcule $\mathbf{z}^{(n+1)}$ dans C.

Appuyez sur	Affichage	
RCL MATRIX D	043-45,16,14	
STO RESULT	044- 44 26	
−	045- 30	Calcule $\mathbf{z}^{(n+1)} - \mathbf{z}^{(n)}$ dans **D**.
f MATRIX 7	046-42,16, 7	Calcule $\|\mathbf{z}^{(n+1)} - \mathbf{z}^{(n)}\|_R$.
f MATRIX 1	047-42,16, 1	Définit $R_0 = R_1 = 1$ pour visualiser **C**.
R/S	048- 31	Affiche le paramètre de convergence.
GTO 6	049- 22 6	

Labels utilisés : C, 4, 5, 6 et 7.

Registres utilisés : R_0, R_1 et R_2 (valeur propre).

Matrices utilisées : **A** (matrice d'origine), $\mathbf{B}(\mathbf{A} - \lambda\mathbf{I})$, $\mathbf{C}(\mathbf{z}^{(n+1)})$ et $\mathbf{D}(\mathbf{z}^{(n+1)} - \mathbf{z}^{(n)})$.

Pour utiliser ce programme :

1. Appuyez sur 2 f DIM (i) pour réserver les registres R_0, R_1 et R_2.
2. Appuyez sur f USER pour activer le mode USER.
3. Dimensionne et introduit les éléments dans la matrice **A** en utilisant f DIM A, f MATRIX 1 et STO A.
4. Appuyez sur la valeur propre et appuyez sur C. L'affichage montre le paramètre de correction $\|\mathbf{z}^{(1)} - \mathbf{z}^{(0)}\|_R$.
5. Appuyez plusieurs fois sur R/S jusqu'à ce que le paramètre de correction soit très petit (négligeable).
6. Appuyez plusieurs fois sur RCL C pour afficher les différentes composantes de $\mathbf{q}_k$, le vecteur propre.

Exemple : Pour la matrice **A** de l'exemple précédent

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

calculez les vecteurs propres $\mathbf{q}_1$, $\mathbf{q}_2$ et $\mathbf{q}_3$.

Appuyez sur	Affichage		
g P/R			Mode calcul.
2 f DIM (i)	2.0000		Réserve les registres R_0 à R_2.
f USER	2.0000		Active le mode USER.
3 ENTER f DIM A	3.0000		Dimensionne la matrice **A** à 3 × 3.
f MATRIX 1	3.0000		
0 STO A	0.0000		Introduit les éléments de **A**.
1 STO A	1.0000		
⋮			
4 STO A	4.0000		
.8730 CHS	−0.8730		Introduit $\lambda_1 = -0.8730$ (approximation).
C	0.8982		$\|\|\mathbf{z}^{(1)} - \mathbf{z}^{(0)}\|\|$.*
R/S	0.0001		$\|\|\mathbf{z}^{(2)} - \mathbf{z}^{(1)}\|\|$.*
R/S	2.4000	−09	$\|\|\mathbf{z}^{(3)} - \mathbf{z}^{(2)}\|\|$.*
R/S	1.0000	−10	$\|\|\mathbf{z}^{(4)} - \mathbf{z}^{(3)}\|\|$.*
R/S	0.0000		$\|\|\mathbf{z}^{(5)} - \mathbf{z}^{(4)}\|\|$.*
RCL C	1.0000		Vecteur propre pour λ_1.
RCL C	0.2254		
RCL C	−0.5492		
0 C	0.8485		Utilise $\lambda_2 = 0$ (approximation).
R/S	0.0000		
RCL C	−0.5000		Vecteur propre pour λ_2.
RCL C	1.0000		
RCL C	−0.5000		
6.8730 C	0.7371		Utilise $\lambda_3 = 6.8730$ (approximation).
R/S	1.9372	−06	
R/S	1.0000	−10	
R/S	0.0000		

*Les normes de correction vont varier en fonction de la racine courante des nombres aléatoires.

Appuyez sur	Affichage	
RCL C	0.3923	} Vecteur propre pour λ_3.
RCL C	0.6961	
RCL C	1.0000	
f USER	1.0000	Désactive le mode USER.

Si la matrice **A** n'est pas plus grande que 3×3, ce programme peut être inclus dans le programme précédent de calcul des valeurs propres. Puisque le programme de calcul des valeurs propres modifie la matrice **A**, les valeurs propres d'origine doivent être sauvegardées et la matrice d'origine réintroduite dans la matrice **A** avant l'exécution du programme des vecteurs propres. Le programme suivant peut être ajouté pour stocker les valeurs propres calculées dans la matrice **E**.

Appuyez sur	Affichage	
f LBL E	127-42,21,15	
RCL DIM A	128-45,23,11	
STO 0	129- 44 0	
1	130- 1	
f DIM E	131-42,23,15	Dimensionne **E** à $n \times 1$.
f LBL 8	132-42,21, 8	
RCL 0	133- 45 0	
ENTER	134- 36	
RCL g A	135-45,43,11	Rappelle l'élément diagonal.
RCL 0	136- 45 0	
1	137- 1	
STO g E	138-44,43,15	Stocke a_{ii} dans e_i.
f DSE 0	139-42, 5, 0	
GTO 8	140- 22 8	
f MATRIX 1	141-42,16, 1	Redéfinit $R_0 = R_1 = 1$.
g RTN	142- 43 32	
g P/R		Mode calcul.

Labels utilisés : E et 8.

Registres utilisés : pas de registres supplémentaires.

Matrices utilisées : **A** (du programme précédent) et **E** (valeurs propres).

Pour utiliser les programmes valeurs propres, stockage des valeurs propres et vecteurs propres en combinaison sur une matrice 3×3 maximum :

1. Exécutez le programme des valeurs propres comme indiqué précédemment.

2. Appuyez sur [E] pour stocker les valeurs propres.
3. Réintroduisez les éléments de la matrice d'origine dans **A**.
4. Rappelez la valeur propre désirée de la matrice **E** en utilisant [RCL] [E].
5. Exécutez le programme de calcul des vecteurs propres comme indiqué précédemment.
6. Répétez les étapes 4 et 5 pour chaque valeur propre.

Optimisation

Nous allons décrire ici une catégorie de problèmes dans lesquels le but est de trouver la valeur minimale ou maximale d'une fonction considérée. Le plus souvent, il s'agit d'éliminer le comportement d'une fonction dans une région particulière.

Le programme suivant utilise la méthode du gradient la plus abrupte pour calculer les minimums ou maximums locaux d'une fonction réelle à deux ou plusieurs variables. Cette méthode est une procédure itérative qui utilise le gradient de la fonction pour déterminer des points d'échantillonnage successifs. Quatre paramètres d'entrée contrôlent le plan d'échantillonnage.

Pour la fonction

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$$

le gradient ∇f de f est défini par

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix}.$$

Les points critiques de $f(\mathbf{x})$ sont les solutions de $\nabla f(\mathbf{x}) = 0$. Un point critique peut être un minimum local, un maximum local ou ni l'un ni l'autre.

Le gradient de $f(\mathbf{x})$ évalué à un point $\mathbf{x}$ donne la direction de la croissance la plus rapide, c'est-à-dire la façon dont il faudrait modifier $\mathbf{x}$ pour provoquer l'accroissement le plus rapide de $f(\mathbf{x})$. Le gradient négatif donne la direction de la décroissance la plus rapide. Le vecteur de direction est :

$$\mathbf{s} = \begin{cases} -\nabla f(\mathbf{x}) & \text{pour la recherche d'un minimum} \\ \nabla f(\mathbf{x}) & \text{pour la recherche d'un maximum.} \end{cases}$$

Dès que la direction est déterminée à partir du gradient, le programme recherche la distance optimale d'éloignement de $\mathbf{x}_j$ dans la direction indiquée par $\mathbf{s}_j$, c'est-à-dire la distance donnant la meilleure amélioration dans $f(\mathbf{x})$ vers un minimum ou un maximum.

Pour cela, le programme recherche la valeur optimale t_j en calculant la pente de la fonction

$$g_j(t) = f(\mathbf{x}_j + t\mathbf{s}_j)$$

pour des valeurs croissantes de t jusqu'à ce que la pente change de signe. Cette procédure est appelée "recherche de limites" puisque le programme tente de délimiter la valeur désirée t_j dans un intervalle. Lorsque le programme trouve un changement de signe, il réduit alors l'intervalle en le divisant par $j+1$ fois pour avoir la meilleure valeur t, près de $t = 0$. Cette procédure est appelée "réduction d'intervalle". Elle donne des valeurs pour t_j d'autant plus précises que $\mathbf{x}_j$ converge vers la solution désirée. (Ces deux processus font partie de la "recherche le long d'une ligne". La nouvelle valeur de $\mathbf{x}$ est alors :

$$\mathbf{x}_{j+1} = \mathbf{x}_j + t_j\mathbf{s}_j.$$

Le programme utilise quatre paramètres qui définissent comment il progresse vers la solution désirée. Bien qu'aucune méthode de recherche de ligne ne puisse garantir un résultat pour la valeur optimale de t, les deux premiers paramètres vous apportent une souplesse considérable dans la façon dont le programme échantillonne t.

d Détermine la phase initiale u_1 de la recherche de limites. La première valeur de t essayée est :

$$u_1 = \frac{d}{(j+1)\|\mathbf{s}_j\|_F}$$

Ceci correspond à une distance de

$$\|(\mathbf{x}_j + u_1\mathbf{s}_j) - \mathbf{x}_j\|_F = \frac{d}{j+1}$$

qui montre que d et le nombre d'itérations définissent à quelle distance de la dernière valeur $\mathbf{x}$ le programme commence sa recherche de limites.

a Détermine les valeurs $u_2, u_3 \ldots$ des phases suivantes de la recherche de limites. Ces valeurs de t sont définies par

$$u_{i+1} = au_i.$$

En fait, a est un facteur d'expansion, normalement supérieur à 1, générant une suite croissante de valeurs de t.

e Détermine la tolérance acceptable sur la taille du gradient. Le processus itératif s'arrête lorsque

$$\|\nabla f(\mathbf{x}_j)\|_F \leq e.$$

N Détermine le nombre maximum d'itérations que le programme va tenter dans chacune des deux procédures : la recherche de limites et la procédure générale d'optimisation. Autrement dit, le programme s'arrête si la recherche de limites ne trouve aucun changement de signe sur les N itérations. Le programme s'arrête également si la norme du gradient est encore trop grande à $\mathbf{x}_N$. Chacune de ces situations résulte en l'affichage de **Error 1**. (Elles peuvent être identifiées en appuyant sur [←]). Vous pouvez continuer l'exécution du programme si vous le désirez.

Le programme a besoin d'un sous-programme d'évaluation de $f(\mathbf{x})$ et de $\nabla f(\mathbf{x})$. Ce sous-programme doit être appelé "E", doit utiliser le vecteur $\mathbf{x}$ stocké dans la matrice **A**, doit retourner le gradient dans la matrice **E** et doit placer $f(\mathbf{x})$ dans le registre X.

En outre, le programme demande une estimation initiale $\mathbf{x}_0$ du point critique désiré. Ce vecteur doit être stocké dans la matrice **A**.

Le programme a les caractéristiques suivantes :

- Le programme recherche tout point $\mathbf{x}$ pour lequel $\nabla f(\mathbf{x}) = \mathbf{0}$. Rien n'empêche la convergence vers un point-selle par exemple. En général, vous devez utiliser d'autres moyens pour déterminer la nature du point critique trouvé. (En plus, ce programme ne traite pas le problème de localisation d'un maximum ou d'un minimum sur la limite du domaine de $f(\mathbf{x})$).
- Vous pouvez ajuster les paramètres de convergence après avoir lancé le programme. Dans la plupart des cas, ceci réduit beaucoup le temps nécessaire à la convergence. Voici quelques suggestions :
  - Si le programme introduit la phase de réduction de l'intervalle après l'échantillonnage d'un seul point u_j, la taille du pas initial risque d'être trop grande. Essayez de réduire d pour avoir une recherche plus efficace.
  - Si les résultats de la recherche des limites semble prometteurs (c'est-à-dire si les pentes décroissent), mais commencent ensuite à croître, la recherche a peut-être manqué un point critique. Essayez de réduire a pour générer un échantillonnage plus serré. Vous pouvez aussi avoir à augmenter N.

- Vous pouvez remplacer [R/S] à la ligne 102 par [PSE] ou même le supprimer si les résultats intermédiaires ne vous intéressent pas.
- Pour une fonction à n variables, le programme a besoin de $4n+1$ registres réservés aux matrices.

Appuyez sur	Affichage	
[g] [P/R]		Mode programme.
[f] CLEAR [PRGM]	000-	
[f] [LBL] 8	001-42,21, 8	Programme d'échange de A et de C à l'aide de E.
[RCL] [MATRIX] [C]	002-45,16,13	
[STO] [MATRIX] [E]	003-44,16,15	
[RCL] [MATRIX] [A]	004-45,16,11	
[STO] [MATRIX] [C]	005-44,16,13	
[RCL] [MATRIX] [E]	006-45,16,15	
[STO] [MATRIX] [A]	007-44,16,11	
[g] [RTN]	008- 43 32	
[f] [LBL] 7	009-42,21, 7	Programme de recherche le long d'une ligne.
[RCL] 4	010- 45 4	
[RCL] [÷] 6	011-45,10, 6	
[STO] 8	012- 44 8	Stocke $d/(j+1)$ dans R_8.
[GSB] [E]	013- 32 15	
[RCL] [MATRIX] [E]	014-45,16,15	
[STO] [MATRIX] [D]	015-44,16,14	
[RCL] [MATRIX] [D]	016-45,16,14	
[g] [F?] 0	017-43, 6, 0	
[CHS]	018- 16	Dans le cas d'un minimum, change le signe du gradient.
[f] [MATRIX] 8	019-42,16, 8	Calcule $\|\nabla f(\mathbf{x})\|$.
[g] [$x=0$]	020- 43 20	
[g] [RTN]	021- 43 32	Sortie si $\|\nabla f(\mathbf{x})\| = 0$.
[1/x]	022- 15	
[RCL] [×] 8	023-45,20, 8	Calcule u_1.
[STO] .1	024- 44 .1	Stocke u_1 dans R_1.
0	025- 0	
[STO] .0	026- 44 .0	
[RCL] 5	027- 45 5	
[STO] 7	028- 44 7	Stocke le compteur dans R_7.

Appuyez sur	Affichage	
f LBL 6	029–42,21, 6	Début de la recherche des limites
RCL .1	030– 45 .1	
GSB 3	031– 32 3	
f PSE	032– 42 31	Affiche la pente.
g F? 0	033–43, 6, 0	
CHS	034– 16	
g TEST 4	035–43,30, 4	Teste si changement de pente.
GTO 5	036– 22 5	Branchement à la réduction de l'intervalle.
GSB 8	037– 32 8	Restaure la matrice d'origine à A.
RCL .1	038– 45 .1	
STO .0	039– 44 .0	Stocke u_i dans R_0.
RCL 2	040– 45 2	
STO × .1	041–44,20,. 1	Stocke u_{i+1} dans R_1.
f DSE 7	042–42, 5, 7	Décrémente le compteur.
GTO 6	043– 22 6	Branchement pour continuer.
RCL MATRIX A	044–45,16,11	
g ABS	045– 43 16	Affiche **Error 1** avec A dans le registre X.
GTO 6	046– 22 6	Branchement pour continuer.
f LBL 5	047–42,21, 5	Programme de réduction de l'intervalle.
RCL 6	048– 45 6	
STO 7	049– 44 7	Stocke $j + 1$ dans R_7.
f LBL 4	050–42,21, 4	
GSB 8	051– 32 8	Restaure la matrice d'origine à A.
RCL .0	052– 45 .0	
RCL + .1	053–45,40, .1	
2	054– 2	
÷	055– 10	
STO 8	056– 44 8	Calcule le milieu de l'intervalle.
GSB 3	057– 32 3	Calcule la pente.
g F? 0	058–43, 6, 0	
CHS	059– 16	Change le signe dans le cas d'un minimum.

Appuyez sur	Affichage	
1	060– 1	
1	061– 1	
STO I	062– 44 25	Stocke le numéro du registre de l'intervalle.
R↓	063– 33	
g TEST 1	064–43,30, 1	
f DSE I	065–42, 5,25	
RCL 8	066– 45 8	
STO (i)	067– 44 24	Stocke le milieu de l'intervalle dans R_0 ou R_1.
f DSE 7	068–42, 5, 7	Décrémente le compteur.
GTO 4	069– 22 4	
g RTN	070– 43 32	Sortie quand le compteur est à zéro.
f LBL 3	071–42,21, 3	Programme de calcul de la pente.
RCL MATRIX D	072–45,16,14	
f RESULT C	073–42,26,13	
×	074– 20	
RCL MATRIX A	075–45,16,11	
+	076– 40	Calcule le point $\mathbf{x}_j + t\mathbf{s}_j$.
GSB 8	077– 32 8	Échange la matrice d'origine et le nouveau point.
GSB E	078– 32 15	Calcule $\nabla f(\mathbf{x})$ dans **E**.
STO 9	079– 44 9	Stocke $f(\mathbf{x})$ dans R_9.
RCL MATRIX E	080–45,16,15	
RCL MATRIX D	081–45,16,14	
f RESULT B	082–42,26,12	
f MATRIX 5	083–42,16, 5	Calcule la pente comme $(\nabla f)^T\mathbf{s}$.
1	084– 1	
ENTER	085– 36	
RCL g B	086–45,43,12	
g RTN	087– 43 32	Sortie avec la pente dans le registre X.
f LBL A	088–42,21,11	Programme principal.
0	089– 0	
STO 6	090– 44 6	
f LBL 2	091–42,21, 2	
1	092– 1	

Appuyez sur	Affichage	
STO + 6	093–44,40, 6	Stocke $j + 1$ dans R_6.
f SCI 3	094–42, 8, 3	
GSB 7	095– 32 7	Branchement à la recherche de courbe.
RCL 6	096– 45 6	
f FIX 0	097–42, 7, 0	
f PSE	098– 42 31	Pause avec $j + 1$ à l'affichage.
f MATRIX 1	099–42,16, 1	Définit $R_0 = R_1 = 1$ pour visualisation.
f SCI 3	100–42, 8, 3	
RCL 9	101– 45 9	Rappelle $f(x)$.
R/S	102– 31	Arrête le programme.
RCL 3	103– 45 3	Rappelle e.
RCL MATRIX E	104–45,16,15	
f MATRIX 8	105–42,16, 8	Calcule $\|\nabla f(x)\|$.
g $x \leq y$	106– 43 10	Teste $\|\nabla f(x)\| \leq e$.
GTO B	107– 22 12	Branchement pour affichage de la solution.
f PSE	108– 42 31	Affiche $\|\nabla f(x)\|$.
RCL 5	109– 45 5	
RCL 6	110– 45 6	
g TEST 8	111–43,30, 8	Teste $(j + 1) < N$.
GTO 2	112– 22 2	Branchement pour continuer l'itération.
RCL MATRIX C	113–45,16,13	
g ABS	114– 43 16	Affiche **Error 1** avec C dans le registre X.
GTO 2	115– 22 2	Branchement pour continuer.
f LBL B	116–42,21,12	Programme d'affichage de la solution.
g SF 9	117–43, 4, 9	Arme l'indicateur 9.
R/S	118– 31	Arrêt du programme avec $\|\nabla f(x_{j+1})\|$ à l'affichage.
GTO B	119– 22 12	Branchement de boucle.

Labels utilisés : A, B, et 2 à 8.

Registres utilisés : R_2 à R_9, R_0, R_1, et registre Index.

Matrices utilisées : **A, B, C, D** et **E**.

Votre sous-programme "E" peut utiliser tous labels et registres non indiqués ci-dessus, plus le registre d'index, la matrice **B** et la matrice **E** (qui doit contenir votre gradient calculé).

Pour utiliser le programme :

1. Introduisez votre sous-programme dans la mémoire programme.
2. Appuyez sur **1 1** [f] [DIM] [(i)] pour réserver les registres R_0 à R_1. (Votre sous-programme peut nécessiter des registres supplémentaires).
3. Armez l'indicateur 0 si vous recherchez un minimum local ; désarmez l'indicateur 0 si vous recherchez un maximum local.
4. Dimensionnez la matrice **A** à $n \times 1$, où n est le nombre de variables.
5. Stockez les données nécessaires en mémoire :
   - Stockez la valeur estimée initiale $\mathbf{x}_0$ dans la matrice **A**.
   - Stockez a dans R_2.
   - Stockez e dans R_3.
   - Stockez d dans R_4.
   - Stockez N dans R_5.
6. Appuyez sur [GSB] [A] pour visualiser les pentes au cours de la procédure d'itération.
   - Regardez le numéro de l'itération et la valeur de $f(\mathbf{x})$.
   - Si **Error 1** apparaît, appuyez sur [←] pour effacer le message. Allez alors à l'étape 5 en modifiant les paramètres à votre gré ou appuyez sur [←] [R/S] pour obtenir une nouvelle itération de recherche de limites ou une nouvelle itération d'optimisation. (Si le label de la matrice **A** était à l'affichage lorsque l'erreur s'est produite, c'est que le nombre d'itérations en recherche de limites était supérieur à N ; si le label de la matrice **C** était à l'affichage, c'est que le nombre d'itérations en optimisation était supérieur à N.)
7. Appuyez sur [R/S] pour afficher la norme du gradient et pour lancer l'itération suivante.
   - Si la norme du gradient clignote à l'affichage, appuyez sur [←] puis rappelez les valeurs de **x** dans la matrice **A**.

- Si le numéro de l'itération et la valeur de $f(\mathbf{x})$ sont affichés, répétez cette étape autant qu'il le faut pour obtenir la solution ou retournez à l'étape 5 et modifiez les paramètres à votre gré.

Exemple : Utilisez le programme d'optimisation pour trouver les dimensions de la boîte offrant le plus grand volume pour une somme de sa longueur et de sa périphérie (périmètre de sa section) égale à 100 cm.

Pour ce problème

$$l + (2h + 2w) = 100$$

$$v = whl$$

$$v(w,h) = wh(100 - 2h - 2w)$$

$$= 100wh - 2wh^2 - 2hw^2$$

$$\nabla v(w,h) = \begin{bmatrix} 2h(50 - h - 2w) \\ 2w(50 - w - 2h) \end{bmatrix}.$$

La solution doit satisfaire $w + h < 50$, $w > 0$ et $h > 0$.

Tout d'abord introduisez un sous-programme pour calculer le gradient et le volume.

Appuyez sur	Affichage	
f LBL E	120-42,21,15	Sous-programme de la fonction.
RCL DIM A	121-45,23,11	
f DIM E	122-42,23,15	
f MATRIX 1	123-42,16, 1	
f USER RCL A	124u 45 11	
f USER		
STO .2	125- 44 .2	Stocke w dans R_2.
STO E	126- 44 15	Stocke w dans e_2.
RCL A	127- 45 11	
STO .3	128- 44 .3	Stocke h dans R_3.
f MATRIX 1	129-42,16, 1	
STO E	130- 44 15	Stocke h dans e_1.
+	131- 40	
5	132- 5	
0	133- 0	
−	134- 30	

Appuyez sur	Affichage	
[CHS]	135- 16	
2	136- 2	
[×]	137- 20	Calcule $l = 2\,(50 - h - w)$.
[f] [$x \gtrless$] .2	138-42, 4, .2	Stocke l dans $R_{.2}$.
[STO] [×] .3	139-44,20, .3	Stocke wh dans $R_{.3}$.
[RCL] .2	140- 45 .2	
[RCL] [MATRIX] [E]	141-45,16,15	
[f] [RESULT] [E]	142-42,26,15	
[×]	143- 20	
[RCL] .3	144- 45 .3	
[RCL] [+] .3	145-45,40, .3	
[−]	146- 30	Remplace e_i par $le_i - 2wh$, les éléments du gradient.
[RCL] .2	147- 45 .2	
[RCL] [×] .3	148-45,20, .3	Calcule lwh.
[g] [RTN]	149- 43 32	

Introduisez maintenant l'information nécessaire et exécutez le programme.

Appuyez sur	Affichage	
[g] [P/R]		Mode calcul.
13 [f] [DIM] [(i)]	13.0000	Réserve R_0 à $R_{.3}$.
[g] [CF] 0	13.0000	Trouve un maximum local.
[f] [USER]	13.0000	Active le mode USER.
[f] [MATRIX] 1	13.0000	
2 [ENTER] 1	1	Introduit les dimensions pour la matrice **A**.
[f] [DIM] [A]	1.0000	Dimensionne la matrice **A** à 2×1.
15 [STO] [A]	15.0000	
[STO] [A]	15.0000	Stocke l'estimation initiale $l = w = 15$.
3 [STO] 2	3.0000	Stocke $a = 3$.
0.1 [STO] 3	0.1000	Stocke $e = 0.1$.
0.05 [STO] 4	0.0500	Stocke $d = 0.05$.

Appuyez sur	Affichage		
4 STO 5	4.0000		Stocke $N - 4$.
A	4.415	04	Pente à u_1.
	4.243	04	Pente à u_2.
	3.718	04	Pente à u_3.
	2.045	04	Pente à u_4.
	Error 1		
←	A	2 1	Recherche de limites sans succès.

Puisque les résultats semblent prometteurs (les dérivées décroissent), ajoutez cinq autres échantillons à cette recherche et définissez $N - 8$ comme nombre d'itérations restantes.

Appuyez sur	Affichage		
5 STO 7	5.000	00	Met le compteur à 5.
8 STO 5	8.000	00	Définit $N = 8$.
R/S	−3.849	04	Pente à u_5 (changement de signe).
	1.		$j + 1$.
	9.253	03	Volume à cette itération.
R/S	3.480	01	Gradient.
	1.121	03	Pente à u_1.
	9.431	02	Pente à u_2.
	4.126	02	Pente à u_3.
	−1.139	03	Pente à u_4 (changement de signe).
	2.		$j + 1$.
	9.259	03	Volume à cette itération.
R/S	5.479	−01	Gradient.
	−6.127	−01	Pente à u_1 (changement de signe).
	3.		$j + 1$.
	9.259	03	Volume à cette itération.
R/S	7.726	−02	Gradient inférieur à e.
←	7.726	−02	Arrêt du clignotement.
f FIX 4	0.0773		
RCL A	16.6661		Rappelle h de a_1.
RCL A	16.6661		Rappelle w de a_2.

Appuyez sur	Affichage	
f USER	**16.6661**	
f MATRIX 0	**16.6661**	Désalloue la mémoire matricée.

La taille optimale de la boîte est 16.6661 × 16.6661 × 33.3355 cm. (Une autre méthode consiste à résoudre ce problème en résolvant le système linéaire représenté par $\nabla v(w,h) = 0$.)

Précision des calculs numériques

Interprétation des erreurs

Une erreur est toujours possible. Ce n'est d'ailleurs pas toujours une faute. L'erreur numérique représente simplement la différence entre ce que vous souhaitiez calculer et ce que vous avez calculé. Cette différence n'est préoccupante que si elle est vraiment trop importante. Elle est généralement négligeable ; mais il arrive que l'erreur soit désespérément grande, difficile à expliquer et encore plus difficile à corriger. Cette annexe est consacrée aux erreurs, et surtout à celles qui risquent d'être importantes — un cas assez rare. En voici quelques exemples.

Exemple 1 : Un Calculateur cassé. Puisque $(\sqrt{x})^2 = x$ pour tout $x \geq 0$, on est en droit d'attendre que

$$f(x) = ((\ldots((\underbrace{\sqrt{\sqrt{\ldots\sqrt{\sqrt{x}}}}}_{\substack{50\\ \text{racines}}})^2)^2\ldots)^2)^2 \quad \underbrace{\qquad}_{\substack{50\\ \text{carrés}}}$$

soit aussi égale à x.

Un programme de 100 pas peut évaluer l'expression $f(x)$ pour tout x positif. Lorsque $x = 10$, le HP-15C calcule le résultat 1. L'erreur $10 - 1 = 9$ semble énorme si l'on considère que seulement 100 opérations arithmétiques ont été effectuées, chacune d'elle étant présumée correcte sur 10 chiffres. Or le programme, au lieu de donner $f(x) = x$, renvoie :

$$f(x) = \begin{cases} 1 & \text{pour } x \geq 1 \\ 0 & \text{pour } 0 \leq x < 1, \end{cases}$$

Ce qui est faux. Ce calculateur doit-il être envoyé en réparation ?

Exemple 2 : Beaucoup d'argent. Une société s'attache les services d'une secrétaire au tarif de 1 centime par seconde. Cette société vire les honoraires de cette secrétaire sur un compte rémunéré à 11.25 % par an, les intérêts étant composés par seconde. A la fin de l'année, tous ces francs accumulés vont présenter le total suivant :

$$\text{Total} = (\text{versement}) \times \frac{(1 + i/n)^n - 1}{i/n}$$

où versement = 0.01 F = 1 centime par seconde,
$i = 0.1125 = 11.25\%$ d'intérêt annuel,
$n = 60 \times 60 \times 24 \times 365$ = nombre de secondes (périodes de composition) dans l'année.

Utilisant son HP-15C, cette secrétaire trouve un total de 376,877.67 FF. Mais à la fin de l'année son compte présente un crédit de 333,783.35 FF. Le consultant peut-elle disposer de ce supplément (différence) de 43,094.32 FF.

Dans ces deux exemples, les différences sont dues à des erreurs d'arrondi qui auraient pu être évitées. Nous montrerons comment.

La guerre contre les erreurs commence avec une réserve à l'encontre des bonnes intentions qui risquent de nous faire confondre ce que nous voulons et ce que nous obtenons. Pour éviter toute confusion, les résultats vrais et les résultats calculés doivent être affectés de noms différents même si leur différence est si petite que cela semble exagéré.

Exemple 3 : Pi. La constante $\pi = 3.14159265358979323846264 33...$ En appuyant sur la touche $\boxed{\pi}$ du HP-15C vous obtenez une valeur différente :

$$\boxed{\pi} = 3.141592654$$

qui correspond à π sur 10 chiffres significatifs. Mais $\boxed{\pi} \neq \pi$, aussi ne soyez pas surpris si, en mode Radians, le calculateur ne donne pas $\sin \boxed{\pi} = 0$.

Supposons que nous calculons x, mais obtenons X. (Convention utilisée systématiquement dans cette annexe.) L'*erreur* est $x - X$. L'***erreur absolue*** est $| x - X |$. L'*erreur relative* est $(x - X)/x$ pour $x \neq 0$.

Exemple 4 : Un pont trop court. Les longueurs (en mètres) des trois sections d'un pont en encorbellement (pont cantilever) doivent être:

$$x = 333.76 \qquad y = 195.07 \qquad z = 333.76 .$$

Les longueurs mesurées sont en fait:

$$X = 333.69 \qquad Y = 195.00 \qquad Z = 333.72 .$$

La différence totale est:

$$d = (x + y + z) - (X + Y + Z) = 862.59 - 862.41 = 0.18 .$$

L'ingénieur responsable du pont compare la différence à la longueur totale $(x + y + z)$ et considère que cette différence relative:

$$d/(x + y + z) = 0.0002 = 2 \text{ dix millièmes}$$

est négligeable. Mais le riveur, lui, trouve la différence absolue $|d| = 0.18$ mètres beaucoup trop grande à son goût. Il faudra "allonger" la structure du pont pour pouvoir poser les rivets. Tous deux considèrent la même différence d, mais celle-ci est négligeable pour l'un alors qu'elle est inacceptable pour l'autre.

Qu'elles soient grandes ou petites les erreurs sont de deux origines qui, si elles sont comprises, permettent en général de les compenser ou de les tourner. Pour comprendre les distorsions dans l'ossature d'un pont, il faut connaître la mécanique des structures et la théorie de l'élasticité. Pour comprendre les erreurs introduites par le calcul, il suffit de connaître son outil de calcul et ses limitations. Ce sont des détails que la plupart d'entre vous désirent connaître, spécialement si les erreurs d'arrondi d'un calculateur bien conçu sont toujours minimales et apparaissent ainsi comme insignifiantes lorsqu'elles sont introduites. Mais lorsque, à de rares occasions, ces erreurs s'accumulent au niveau des calculs, elles doivent être considérées malgré tout comme "importantes".

Exemple 1: Explication. Ici $f(x) = s(r(x))$, où

$$r(x) = \underbrace{\sqrt{\sqrt{\ldots\sqrt{\sqrt{x}}}}}_{\substack{50 \\ \text{racines}}} = x^{(1/2^{50})}$$

et

$$s(r) = \underbrace{((\ldots((r)^2)^2\ldots)^2)^2}_{\substack{50 \\ \text{carrés}}} = r^{(2^{50})}.$$

Les exposants sont $1/2^{50} = 8.8818 \times 10^{-16}$ et $2^{50} = 1.1259 \times 10^{15}$. Maintenant, x doit se trouver entre 10^{-99} et $9.999\ldots \times 10^{99}$ puisqu'aucun nombre positif en dehors de cette plage de valeur ne peut être introduit dans le calculateur. Puisque r est une fonction croissante, $r(x)$ se trouve entre:

$$r(10^{-99}) = 0.9999999999997975\ldots$$

et

$$r(10^{100}) = 1.0000000000002045\ldots.$$

Ceci suggère que $R(x)$, la valeur calculée de $r(x)$, sera 1 pour tous les arguments x valides du calculateur. En fait, à cause de l'arrondi:

$$R(x) = \begin{cases} 0.9999999999 & \text{pour } 0 < x < 1 \\ 1.000000000 & \text{pour } 1 \leqslant x \leqslant 9.999999999 \times 10^{99}. \end{cases}$$

Si $0 < x < 1$, alors $x \leqslant 0.9999999999$ dans un calculateur 10 chiffres. Nous serions en droit d'attendre $\sqrt{x} \leqslant \sqrt{0.9999999999}$, qui est $0.99999999994999999998\ldots$, arrondi à nouveau à 0.9999999999. Par conséquent, si vous appuyez sur $\boxed{\sqrt{x}}$ en commençant arbitrairement par $x < 1$, le résultat ne peut pas dépasser 0.9999999999. Ceci explique pourquoi nous obtenons $R(x) = 0.9999999999$ pour $0 < x < 1$ ci-dessus. Quand $R(x)$ est mis au carré 50 fois pour donner $F(x) = S(R(x))$, le résultat est 1 pour $x \geqslant 1$, mais pourquoi $F(x) = 0$ pour $0 \leqslant x < 1$? Quand $x < 1$,

$$s(R(x)) \leqslant s(0.9999999999) = (1 - 10^{-10})^{2^{50}} \approx 6.14 \times 10^{-48898}.$$

Cette valeur est si petite que la valeur calculée $F(x) = S(R(x))$ est en dépassement inférieur de capacité à 0. Aussi le HP-15C n'est-il pas cassé ; il fait de son mieux avec 10 chiffres significatifs de précision et 2 chiffres d'exposants.

Nous avons expliqué l'exemple 1 en ne sachant rien de plus sur le HP-15C que le fait qu'il effectue chaque opération arithmétique $\boxed{\sqrt{x}}$ et $\boxed{x^2}$ aussi précisément que possible dans les limites de 10 chiffres significatifs et de 2 chiffres d'exposant. Ce dont nous avons besoin est la connaissance mathématique des fonctions *f*, *r* et *s*. Ainsi, la valeur $r(10^{100})$ ci-dessus a été évaluée comme :

$$\begin{aligned} r(10^{100}) &= (10^{100})^{(1/2^{50})} \\ &= \exp(\ln(10^{100})/2^{50}) \\ &= \exp(100(\ln 10)/2^{50}) \\ &= \exp(2.045 \times 10^{-13}) \\ &= 1 + (2.045 \times 10^{-13}) + \tfrac{1}{2}(2.045 \times 10^{-13})^2 + \ldots \end{aligned}$$

en utilisant la série $\exp(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \ldots$

De façon identique, le théorème binominal a été utilisé pour :

$$\begin{aligned} \sqrt{0.9999999999} &= (1 - 10^{-10})^{1/2} \\ &= 1 - \tfrac{1}{2}(10^{-10}) - \tfrac{1}{8}(10^{-10})^2 - \ldots . \end{aligned}$$

Ces faits mathématiques se situent bien au-delà du type de connaissances ayant pu être considérées comme suffisantes pour traiter un calcul ne mettant en œuvre qu'une poignée de multiplications et de racines carrées. L'exemple 1 nous a montré comment les erreurs pouvaient rendre les calculs difficiles à analyser. C'est pourquoi un bon calculateur comme le HP-15C introduira pour sa part aussi peu d'erreurs que possible. Des erreurs plus importantes risqueraient de transformer une tâche déjà difficile en un problème sans issue.

L'exemple 1 met en valeur deux *conditions d'erreurs* assez fréquentes :

- Les erreurs d'arrondi ne faussent un calcul que si un grand nombre d'entre elles s'accumulent.
- Un petit nombre d'erreurs d'arrondi ne faussent un calcul que si elles sont accompagnées par un effet de "compensation" quasi-totale.

En ce qui concerne la première de ces conditions, l'exemple 1 risque de mal évoluer si il est victime d'une seule erreur d'arrondi, celle qui donne $R(x) = 1$ ou 0.9999999999, en erreur sur moins d'une unité au niveau de son dernier (10ième) chiffre significatif.

En ce qui concerne la seconde condition, la "compensation" est ce qui se produit lorsque deux nombres très proches font l'objet d'une soustration. Par exemple, le calcul de :

$$c(x) = (1 - \cos x)/x^2$$

en mode radians pour de petites valeurs de x est risqué. Si nous avons $x = 1.2 \times 10^{-5}$ et des résultats arrondis à 10 chiffres,

$$\cos x = 0.9999999999$$

et

$$1 - \cos x = 0.0000000001$$

la "compensation" laissant peut-être un chiffre significatif au numérateur. Ensuite :

$$x^2 = 1.44 \times 10^{-10}.$$

Donc

$$C(x) = 0.6944\,.$$

Ce qui est faux puisque $0 \leqslant c(x) < \frac{1}{2}$ pour tout $x \neq 0$. Pour éviter la "compensation", exploitez l'égalité trigonométrique : $\cos x = 1 - 2\sin^2(x/2)$ pour supprimer ***exactement*** le 1 et obtenir une meilleure formule de calculer

$$c(x) = \frac{1}{2}\left(\frac{\sin(x/2)}{x/2}\right)^2.$$

Lorsque cette dernière expression est évaluée (en mode radians) pour $x = 1.2 \times 10^{-5}$, le résultat calculé $C(x) = 0.5$ est correct sur 10 chiffres significatifs. Cet exemple, tout en expliquant la notion de "compensation", sous-entend qu'il s'agit toujours d'une mauvaise chose. C'est une interprétation que nous étudierons un peu plus loin. Pour le moment, souvenez-vous que l'exemple 1 ne contient pas de soustraction, donc pas de "compensation", et que pourtant les résultats de ce problème sont complètement faussés par des erreurs d'arrondi.

Cet exemple 1 est quelque peu déconcertant : il ne contient nulle part des opérations arithmétiques auxquelles imputer le résultat catastrophique ; et aucune manipulation des formules, comme pour $c(x)$, ne peut redresser les choses. L'exemple 1 n'est pas, hélas, un cas unique. Plus les calculateurs et les ordinateurs sont puissants, plus ces erreurs insidieuses se glissent dans les calculs.

Pour vous aider à identifier l'ampleur des erreurs, nous allons, dans cette annexe, en examiner plusieurs types en commençant par les plus simples puis en étudiant celles qui affectent les calculs les plus sophistiqués du HP-15C.

Hiérarchie des erreurs

Certaines erreurs sont plus faciles à expliquer et à tolérer que d'autres. Par conséquent, nous avons classé les fonctions offertes par les touches du HP-15C par niveaux de difficulté à estimer leurs erreurs. Ces estimations sont plus des objectifs définis pour le calculateur à sa conception que des spécifications vous garantissant un degré assuré de précision. D'autre part ces objectifs de conception ont été testés de façon approfondis et peuvent être considérés comme tout à fait justes.

Niveau 0 : pas d'erreur

C'est le cas des fonctions qui, même sur des entiers petits (inférieurs à 10^{10}), ne provoquent pas d'erreurs.

Exemples :

$$\sqrt{4} = 2 \qquad -2^3 = -8 \qquad 3^{20} = 3{,}486{,}784{,}401$$

$$\log(10^9) = 9 \qquad 6! = 720$$

$$\cos^{-1}(0) = 90 \text{ (en mode degrés)}$$

$$\text{ABS}(4{,}684{,}660 + 4{,}684{,}659i) = 6{,}625{,}109 \text{ (en mode complexe)}$$

Également exactes sont les fonctions : [ABS], [FRAC], [INT], [RND] ainsi que les comparaisons (comme [$x \leq y$]. Par contre les fonctions matricielles [×], [÷], [1/x], [MATRIX] 6 et [MATRIX] 9 (déterminant) sont des exceptions (voir page 192).

Niveau ∞ : dépassements de capacité.

Les résultats plus proches de zéro que de 10^{-99} sont considérés comme nuls. Les résultats dépassant le seuil de $\pm 9.999999999 \times 10^{99}$ sont remplacés par ce seuil avec armement de l'indicateur 9 et clignotement de l'affichage. (Appuyez sur [ON] [ON] ou [CF] 9 ou [←] pour effacer l'indicateur 9 et arrêter le clignotement.) De nombreuses fonctions dont les résultats ont plusieurs composantes, tolèrent les dépassements de capacité inférieurs ou supérieurs sur l'une de leurs composantes, sans répercussion sur les autres. Des exemples de ces fonctions sont: [→R], [→P], les calculs sur nombres complexes et la plupart des opérations matricielles. Les exceptions sont l'inversion de matrice ([1/x] et [÷]), [MATRIX] 9 (déterminant) et [L.R.].

Niveau 1 : arrondis corrects ou presque

Les opérations donnant des résultats "arrondis correctement" dont les erreurs sont inférieures à ½ unité de leur dernier (10ième) chiffre significatif, sont les suivantes : les opérations algébriques [+], [−], [×], [÷], [√x], [1/x] et [%], les opérations [+] et [−] complexes et matricielles (sauf la division par une matrice) et la fonction [→H.MS]. Ces résultats sont les meilleurs sur 10 chiffres significatifs à l'instar des constantes [π], 1 [e^x], 2 [LN], 10 [LN] et 1 [→RAD]. D'autres opérations admettent une erreur légèrement supérieure, bien que toujours inférieure à une unité sur le 10ième chiffre significatif du résultat: [Δ%], [→H], [→RAD], [→DEG], [Py,x], et [Cy,x]; [LN], [LOG], [10^x] et [TANH] pour les arguments réels; [→P], [SIN^{-1}], [COS^{-1}], [TAN^{-1}], [SINH^{-1}], [COSH^{-1}] et [TANH^{-1}] pour les arguments réels ou complexes; [ABS], [√x] et [1/x] pour les arguments complexes; les normes matricielles [MATRIX] 7 et [MATRIX] 8; et enfin [SIN], [COS] et [TAN] pour les arguments réels en mode degrés ou en mode grades (mais pas en mode radians – voir Niveau 2, par 184).

Une fonction qui tend vers l'infini ou qui tend vers 0 de façon exponentielle lorsque son argument approche $\pm \infty$, peut supporter une erreur supérieure à une unité sur le 10ième chiffre significatif de son résultat, mais seulement si sa valeur est inférieure à 10^{-20} ou supérieure à 10^{20} ; et bien que l'erreur relative devienne de plus en plus importante lorsque les résultats deviennent extrêmes (petits ou grands), l'erreur demeure inférieure à trois unités sur le dernier (10ième) chiffre significatif. Cette erreur sera expliquée plus loin. Les fonctions ainsi affectées sont [e^x], [y^x], [x!] (pour x non entier), [SINH] et [COSH] pour des arguments réels. Le plus mauvais cas rencontré est 3^{201} qui est calculé égal à $7.968419664 \times 10^{95}$. Le dernier chiffre devrait être 6 au lieu de 4, comme dans le cas de $7.29^{33.5}$ calculé comme égal à $7.968419666 \times 10^{28}$.

La conclusion précédente sur les erreurs peut être résumée ainsi pour toutes les fonctions citées au niveau 1:

> Toute tentative de calcul d'une fonction f du niveau 1, donne comme résultat une valeur $F = (1+\varepsilon)f$ donc l'erreur relative ε, bien que non connue, est très petite:

$$|\varepsilon| < \begin{cases} 5 \times 10^{-10} & \text{si } F \text{ est arrondie correctement} \\ 1 \times 10^{-9} & \text{pour toutes les autres fonctions } F \text{ du niveau 1.} \end{cases}$$

Cette classification simple de toutes les fonctions du niveau 1 ne peut conserver d'autres propriétés importantes de ces fonctions, des propriétés telles que:

- Valeurs entières exactes: mentionnées au niveau 0.
- Symétrie du signe: $\sinh(-x) = -\sinh(x)$, $\cosh(-x) = \cosh(x)$, $\ln(1/x) = -\ln(x)$ (si $1/x$ calculé exactement).
- Monotonie: si $f(x) \geqslant f(y)$ alors $F(x)$ calculé $\geqslant F(y)$.

Ces propriétés supplémentaires ont des implications importantes; par exemple TAN(20°) = TAN(200°) = TAN(2,000°) = ... = TAN $(2 \times 10^{99}°)$ = 0.3639702343 (correct). Mais, la caractérisation simple conserve l'essentiel de ce qui est bon à savoir.

Exemple 2: Explication.

La secrétaire a fait le calcul suivant:

$$\text{total} = (\text{versement}) \times \frac{(1 + i/n)^n - 1}{i/n}$$

où

versement = 10 centimes

$i = 0.1125$

$n = 60 \times 60 \times 24 \times 365 = 31{,}526{,}000.$

Elle a calculé 376,877.67 FF sur son HP-15C mais le total donné par la banque est: 333,783.35 FF et ce dernier total est tout a fait compatible avec les résultats obtenus sur de bons calculateurs financiers tels que le HP-12C, le HP-37E, les HP-38E/38C et le HP-92. A quel niveau s'est produite la distorsion? Pas de "compensation" grave, pas de gros cumul d'erreurs. Juste une erreur d'arrondi qui a grossi insidieusement.

$$i/n = 0.000000003567351598$$
$$1 + i/n = 1.000000004$$

après arrondi à 10 chiffres significatifs. C'est l'erreur d'arrondi la grande responsable. Ensuite, lorsqu'elle calcule $(1 + i/n)^n$, la secrétaire va obtenir $(1.000000004)^{31,536,00} = 1.134445516$, résultat faux sur sa seconde position décimale.

Comment calculer la valeur correcte ? Uniquement en ne perdant pas tant de chiffres de i/n. Observez que :

$$(1 + i/n)^n = e^{n \ln(1 + i/n)},$$

aussi pourrions-nous essayer de calculer ce logarithme de façon à ne pas perdre autant de chiffres. C'est possible sur le HP-15C.

Pour calculer $\lambda(x) = \ln(1 + x)$ précisément pour tout $x > -1$, même si $|x|$ est très petit :

1. Calculez $u = 1 + x$ arrondi.
2. Ensuite

$$\lambda(x) = \begin{cases} x & \text{si } u = 1 \\ \ln(u)\ x/(u-1) & \text{si } u \neq 1. \end{cases}$$

Le programme suivant calcule $\lambda(x) = \ln(1 + x)$.

Appuyez sur	Affichage		
g P/R			
f CLEAR PRGM	000–		
f LBL A	001–	42,21,11	Suppose x dans le registre X.
ENTER	002–	36	
ENTER	003–	36	
EEX	004–	26	Place 1 dans le registre X.
+	005–	40	Calcule $u = 1 + x$ arrondi.
g LN	006–	43 12	Calcule $\ln(u)$ (zéro pour $u = 1$).
x≷y	007–	34	Restaure x dans le registre X.
g LSTx	008–	43 36	Rappelle u.

Appuyez sur	Affichage	
EEX	009– 26	Place 1 dans le registre X.
g TEST 6	010–43,30, 6	Teste $u \neq 1$.
−	011– 30	Calcule $u - 1$ quand $u \neq 1$.
÷	012– 10	Calcule $x/(u-1)$ ou $1/1$.
×	013– 20	Calcule $\lambda(x)$.
g RTN	014– 43 32	
g P/R		

La valeur calculée de u, arrondie correctement par le HP-15C est : $u = (1 + \varepsilon)(1 + x)$ où $|\varepsilon| < 5 \times 10^{-10}$. Si $u = 1$, alors :

$$|x| = |1/(1+\varepsilon) - 1| \leq 5 \times 10^{-10}$$

aussi, dans lequel cas la série de Taylor $\lambda(x) = x(1 - \frac{1}{2}x + \frac{1}{3}x^2 - \ldots)$ nous indique que la valeur correctement arrondie de $\lambda(x)$ doit être juste x. Sinon, nous allons calculer $x\,\lambda(u-1)/(u-1)$ beaucoup plus précisément, au lieu de $\lambda(x)$. Mais $\lambda(x)/x = 1 - \frac{1}{2}x + \frac{1}{3}x^2 - \ldots$ varie très lentement, si lentement que l'erreur absolue $\lambda(x)/x - \lambda(u-1)/(u-1)$ n'est pas pire que l'erreur absolue $x - (u-1) = \varepsilon(1 + x)$, et si $x \leq 1$, cette erreur est négligeable par rapport à $\lambda(x)/x$. Quand $x > 1$, alors $u - 1$ est si proche de x que l'erreur est là aussi négligeable ; $\lambda(x)$ est correcte sur 9 chiffres significatifs.

Comme fréquemment dans les analyses des erreurs, l'explication est beaucoup plus longue que la procédure simple expliquée, et cache une considération importante : les erreurs dans $\ln(u)$ et $u - 1$ ont été ignorées lors de l'explication parce que nous savions qu'elle serait négligeable. Cette information et la procédure simple décrite ici, sont *non applicables* à d'autres calculateurs ou gros ordinateurs ! Il existe des machines qui calculent $\ln(u)$ et/ou $1 - u$ avec une erreur *absolue* minime, mais une erreur *relative* assez importante lorsque u est proche de 1 ; sur ces machines, les calculs précédents seront faux ou beaucoup plus compliquées, souvent les deux. (Reportez-vous à l'explication figurant à Niveau 2).

Revenons aux honoraires de notre secrétaire. En utilisant la procédure simple déjà citée pour calculer $\lambda(i/n) = \ln(1 + i/n) = 3.567351591 \times 10^{-9}$, elle obtiendra un résultat intermédiaire meilleur

$$(1 + i/n)^n = e^{n\,\lambda(i/n)} = 1.119072257$$

lequel génère un total correct.

Pour comprendre l'erreur pour 3^{201}, remarquez que ceci est calculé comme $e^{201\ln(3)} = e^{220.821\ldots}$. Pour maintenir l'erreur relative finale à moins de une unité sur le 10ème chiffre significatif, 201 ln(3) devrait être calculé avec une erreur absolue plutôt inférieure à 10^{-10}, ce qui entraînerait de garder au moins 14 chiffres significatifs pour ce résultat intermédiaire. Le calculateur garde 13 chiffres significatifs pour certains calculs intermédiaires internes, mais un 14ième chiffre serait vraiment un luxe pour les quelques cas où sa présence serait souhaitable.

Niveau 1C : Niveau 1 des complexes

La plupart des fonctions arithmétiques sur nombres complexes ne peuvent pas garantir 9 ou 10 chiffres significatifs corrects dans chacune des parties imaginaire ou réelle, bien que le résultat soit conforme à notre conclusion sur les fonctions du niveau 1, pourvu que f, F, et ε soient interprétés comme des nombres complexes. En d'autres termes, toute fonction complexe f du niveau 1C va générer un résultat complexe calculé $F = (1 + \varepsilon)f$ dont la petite erreur relative complexe ε doit satisfaire $|\varepsilon| < 10^{-9}$. Les fonctions complexes du niveau 1C sont [×], [÷], [x^2], [LN], [LOG], [SIN^{-1}], [COS^{-1}], [TAN^{-1}], [SINH^{-1}], [COSH^{-1}] et [TANH^{-1}]. Par conséquent, une fonction telle que $\lambda(z) = \ln(1 + z)$ peut être calculée précisément pour tout z par le même programme que celui donné précédemment (et avec les mêmes explications).

Pour comprendre pourquoi les parties réelle et imaginaire d'un résultat complexe risquent de ne pas être correctes individuellement sur 9 ou 10 chiffres significatifs, considérez [×], par exemple : $(a + ib) \times (c + id) = (ac - bd) + i(ad + bc)$ idéalement. Essayez ce calcul avec $a = c = 9.999999998$, $b = 9.999999999$ et $d = 9.999999997$; la valeur exacte de la partie réelle du produit $(ac - bd)$ devait donc être :

$$\begin{aligned}(9.999999998)^2 - (9.999999999)(9.999999997) &= 99.999999980000000004 - 99.999999980000000003 \\ &= 10^{-18}\end{aligned}$$

qui nécessite au moins 20 chiffres significatifs pour le calcul intermédiaire. Comme le HP-15C ne garde que 13 chiffres significatifs pour ses résultats intermédiaires internes, il donne donc 0 au lieu de 10^{-18} pour la partie réelle ; mais cette erreur est négligeable comparée à la partie imaginaire 199.9999999.

Niveau 2 : Arrondis corrects pour introduction éventuellement faussée

Fonctions trigonométriques d'angles réels en radians

Reprenez l'exemple 3 qui indique que la touche [π] du calculateur donne une approximation correcte de π avec 10 chiffres significatifs, mais cependant légèrement différente de π, si bien que $0 = \sin(\pi) \neq \sin([\pi])$ pour lequel le calculateur donne :

$$[\text{SIN}]([\pi]) = -4.100000000 \times 10^{-10}.$$

Cette valeur calculée n'est pas tout à fait la même que la vraie valeur :

$$\sin([\pi]) = -4.10206761537356... \times 10^{-10}.$$

Que l'écart semble petit (erreur absolue inférieure à 2.1×10^{-13}) ou relativement grand (résultat faux au quatrième chiffre significatif) pour un calculateur à 10 chiffres significatifs, il mérite cependant d'être bien compris car il laisse présager d'autres erreurs qui, à première vue, sont beaucoup plus sérieuses.

Considérons :

$$10^{14}\,\pi = 314159265358979.3238462643...$$

avec $\sin(10^{14}\pi) = 0$ et

$$10^{14} \times [\pi] = 314159265400000$$

avec [SIN] $(10^{14}[\pi]) = 0.7990550814$, bien que le vrai

$$\sin(10^{14}[\pi]) = -0.78387...$$

Le signe (faux) est une erreur trop sérieuse à ignorer ; elle semble suggérer un défaut du calculateur. Pour comprendre cette erreur dans les fonctions trigonométriques, il faut faire attention aux petites différences entre π et deux approximations de π :

vrai	$\pi = 3.14159265358979323846264 33...$	
touche [π]	$= 3.141592654$	(ajuste π à 10 chiffres)
p interne	$= 3.141592653590$	(ajuste π à 13 chiffres).

Ensuite tout est dit dans la formule suivante pour la valeur calculée : [SIN] $(x) = \sin(x\pi/p)$ avec $\pm$ 0.6 unités sur son dernier (10ième) chiffre significatif.

Plus généralement, si trig(x) est l'une des fonctions $\sin(x)$, $\cos(x)$ ou $\tan(x)$, évaluée en mode radians réel, le HP-15C donne :

$$\boxed{\text{TRIG}}(x) = \text{trig}(x\pi/p)$$

à $\pm$ 0.6 unités près sur son 10ième chiffre significatif.

Cette formule a des conséquences pratiques importantes:

- Puisque $\pi/p = 1 - 2.0676... \times 10^{-13}/p = 0.9999999999999342...$, la valeur produite par $\boxed{\text{TRIG}}(x)$ ne diffère de trig(x) que de ce qui peut être attribué à deux perturbations: l'une sur le 10^e chiffre significatif du trig(x) sorti, l'autre sur le 13^e chiffre significatif du x introduit.

  Si x a été calculé et arrondi à 10 chiffres significatifs, l'erreur héritée de son 10^e chiffre est probablement, en ce qui concerne sa valeur, plus grande que la seconde perturbation de $\boxed{\text{TRIG}}$ sur le 13^e chiffre significatif de x, si bien que cette seconde perturbation peut être ignorée, à moins que x soit considéré comme connu ou calculé exactement.

- Toute égalité trigonométrique qui n'utilise pas π explicitement, est satisfaite dans la limite de l'arrondi sur le 10^e chiffre significatif des valeurs calculées dans l'égalité. Par exemple:

$$\sin^2(x) + \cos^2(x) = 1, \text{ donc } (\boxed{\text{SIN}}(x))^2 + (\boxed{\text{COS}}(x))^2 = 1$$

$$\sin(x)/\cos(x) = \tan(x), \text{ donc } \boxed{\text{SIN}}(x)/\boxed{\text{COS}}(x) = \boxed{\text{TAN}}(x)$$

  avec chaque résultat calculé correct sur neuf chiffres pour tout x. Remarquez que $\boxed{\text{COS}}(x)$ se perd s'il n'y a pas de valeur de x représentable exactement avec juste 10 chiffres significatifs. Et si $2x$ peut être calculé exactement avec x donné:

$$\sin(2x) = 2\sin(x)\cos(x), \text{ si bien que } \boxed{\text{SIN}}(2x) = 2\boxed{\text{SIN}}(x)\boxed{\text{COS}}(x)$$

  sur neuf chiffres significatifs. Essayez la dernière égalité pour $x = 52174$ radians:

$$\boxed{\text{SIN}}(2x) = -\ 0.00001100815000,$$

$$2\boxed{\text{SIN}}(x)\boxed{\text{COS}}(x) = -\ 0.00001100815000.$$

  Remarquez la similarité même si, pour cet x, $\sin(2x) = 2\sin(x)\cos(x) = -\ 0.0000110150176...$ est en désaccord avec $\boxed{\text{SIN}}(2x)$ à son quatrième chiffre significatif. Les mêmes égalités sont satisfaites par les valeurs $\boxed{\text{TRIG}}(x)$ comme par les valeurs trig(x) même si $\boxed{\text{TRIG}}(x)$ et trig(x) sont différentes.

- Malgré les deux sortes d'erreurs dans $\boxed{\text{TRIG}}$, ses valeurs calculées conservent la relation familière suivante, chaque fois que cela est possible:

  - Symétrie du signe:

$$\boxed{\text{COS}}(-x) = \boxed{\text{COS}}(x)$$

$$\boxed{\text{SIN}}(-x) = -\boxed{\text{SIN}}(x)$$

- Monotonie : si $\text{trig}(x) \geq \text{trig}(y)$, alors $\boxed{\text{TRIG}}(x) \geq \boxed{\text{TRIG}}(y)$ (pourvu que $|x - y| < 3$)

- Inégalités limitatives : $\boxed{\text{SIN}}(x)/x \leq 1$ pour tout $x \neq 0$
  $\boxed{\text{TAN}}(x)/x \geq 1$ pour $0 < |x| < \pi/2$
  $-1 \leq \boxed{\text{SIN}}(x)$ et $\boxed{\text{COS}}(x) \leq 1$
  pour tout x.

Quel est la répercussion de ces propriétés pour les calculs d'ingénierie ? *Vous n'avez pas besoin de vous en préoccuper !*

En général, les calculs d'ingénierie ne seront pas affectés par la différence entre p et π, parce que les conséquences de cette différence dans la formule définissant $\boxed{\text{TRIG}}(x)$ ci-dessus sont noyées par la différence entre $\boxed{\pi}$ et π et par l'arrondi habituel inévitable de x ou de $\text{trig}(x)$. Dans ces calculs, le ratio $\pi/p = 0.9999999999999342...$ pourrait être remplacé par 1 sans effets visibles sur le comportement de $\boxed{\text{TRIG}}$.

Exemple 5 : Phases lunaires. Si la distance entre la terre et la lune était connue avec précision, nous pourrions calculer la différence de phase entre les signaux de radars transmis à puis reflétés par la lune. Dans ce calcul le décalage de phase introduit par $p \neq \pi$ a moins d'effet que la modification de la distance terre-lune d'une valeur de l'ordre de l'épaisseur de cette page. De plus, le calcul de la force, de la direction et du taux de variation des signaux émis à proximité de la lune ou des signaux réfléchis à proximité de la terre, des calculs qui dépendent de la validité permanente des égalités trigonométriques, ne sont pas affectés par le fait que $p \neq \pi$; par contre, ils reposent sur le fait que p est une constante (indépendante de x dans la formule pour $\boxed{\text{TRIG}}(x)$), et que cette constante est très proche de π.

Les fonctions disponibles sur le clavier du HP-15C utilisant p, sont les fonctions trigonométriques $\boxed{\text{SIN}}$, $\boxed{\text{COS}}$ et $\boxed{\text{TAN}}$ pour les arguments réels et complexes ; les fonctions hyperboliques $\boxed{\text{SINH}}$, $\boxed{\text{COSH}}$ et $\boxed{\text{TANH}}$ pour les arguments complexes ; les opérations complexes $\boxed{e^x}$, $\boxed{10^x}$ et $\boxed{y^x}$; et enfin la fonction $\boxed{\rightarrow \text{R}}$ réelle et complexe.

Il vous semble peut-être que nous avons fait beaucoup de bruit pour rien. Après une avalanche de formules et d'exemples, nous concluons que l'erreur causée par $p \neq \pi$ est négligeable dans les calculs d'ingénierie et que vous n'avez pas à vous en préoccuper. Il s'agit de notre part d'une forme d'honnêteté intellectuelle : nous nous sommes posé les questions que se pose un analyste des erreurs ; si ce dernier prend pour hypothèse que les petites erreurs sont négligeables, il prend un grand risque.

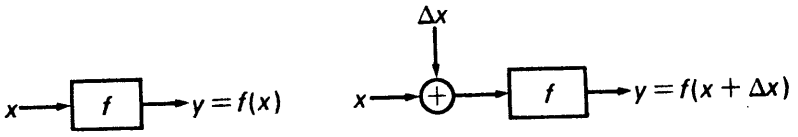
Analyse récurrente de l'erreur

Jusqu'à la fin des années 50, la plupart des experts en informatique dramatisaient les conséquences des erreurs d'arrondis. Pour justifier leur attitude, ils citaient des analyses d'erreur du type de celle faite par un chercheur réputé qui concluait que les matrices de dimensions 40×40 étaient pratiquement impossibles à inverser numériquement du fait des arrondis. Cependant, cinq ans plus tard environ, on pouvait inverser sans problèmes des matrices 100×100 et de nos jours, on est capable de résoudre des équations ayant des centaines de milliers d'inconnues. Comment réconcilier notre époque et la conclusion tout à fait correcte de ce fameux chercheur ?

Nous comprenons mieux maintenant qu'autrefois pourquoi des formules différentes servant à calculer le même résultat peuvent différer terriblement au niveau de la dégradation imposée par les erreurs d'arrondis. Par exemple, nous comprenons pourquoi les équations normales de certains problèmes de moindres carrés ne peuvent être qu'arithmétiquement résolus et avec une précision exceptionnelle ; c'est ceci que le fameux chercheur a, en fait, prouvé. Nous connaissons également des nouvelles procédures (l'une d'elles figure page 140) pouvant résoudre les mêmes problèmes de moindres carrés sans plus de précision qu'il n'en faut pour représenter les données. Les nouvelles (et meilleures) procédures numériques ne sont pas évidentes et auraient pu ne jamais être découvertes sans les nouvelles (et meilleures) techniques d'analyse des erreurs par lesquelles nous avons appris à distinguer les formules hypersensibles aux erreurs d'arrondis de celles qui ne le sont pas. L'une de ces nouvelles (en 1957) techniques est appelée "Analyse récurrente de l'erreur" et vous l'avez déjà vue en œuvre à deux reprises : tout d'abord, elle a expliqué pourquoi la procédure de calcul de $\lambda(x)$ est suffisamment précise pour chasser l'imprécision de l'exemple 2 ; ensuite, elle a expliqué pourquoi les fonctions [TRIG] du calculateur satisfont de façon très proche les mêmes égalités qui sont satisfaites par des fonctions trig même dans le cas d'arguments x très grands (en radians) pour lesquels [TRIG](x) et trig(x) peuvent être très différents. Les paragraphes suivants expliquent l'analyse récurrente de l'erreur.

Considérons un système F destiné à transformer une entrée x en une sortie $y = f(x)$. Par exemple, F peut être un amplificateur de signal, un filtre, un transducteur, un système de contrôle, une raffinerie, le système économique d'un pays, un programme informatique ou un calculateur. L'entrée et la sortie ne sont pas nécessairement des nombres ; elles peuvent être des ensembles de nombres ou des matrices ou n'importe quel élément quantitatif. Si l'entrée x devait être

contaminée par le bruit Δx, la sortie $y + \Delta y = f(x + \Delta x)$ serait contaminée par le bruit $\Delta y = f(x + \Delta x) - f(x)$.



Pas de bruit

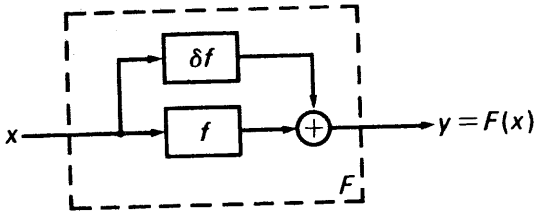
Entrée avec bruit

Certaines transformations f sont stables en présence du bruit d'entrée ; elles gardent Δy relativement petit tant que Δx est relativement petit. D'autres transformations f peuvent être instables en présence du bruit parce que certains bruits d'entrée Δx relativement petits provoquent des perturbations Δy relativement importantes sur la sortie. En général, le bruit d'entrée Δx sera modifié d'une certaine façon par la transformation considérée f, pour devenir à la sortie un bruit Δy, et aucune réduction de Δy n'est possible sans une diminution de Δx ou une modification de f. Ayant accepté f comme une spécification de performance ou comme un objectif de conception, nous devons être d'accord sur la façon dont f influence le bruit à son entrée.

Le système réel F est différent de f désirée à cause du bruit et d'autres écarts internes à F. Avant de discuter des conséquences de ce bruit interne nous devons trouver une façon de le représenter, une notation particulière. La façon la plus simple est d'écrire :

$$F(x) = (f + \delta f)(x)$$

où la perturbation δf représente le bruit interne de F.



Une petite perturbation de sortie (Niveau 1)

Nous espérons que le terme δf est négligeable comparé à f. Si cet espoir est satisfait, nous classons F au niveau 1 pour les fins de notre exposé ; ceci

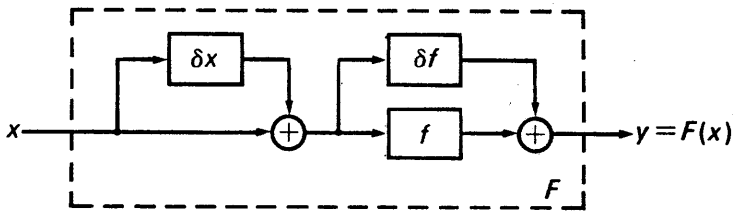
signifie que le bruit interne de F peut être expliqué comme *une* petite addition δf à la sortie f désirée.

Par exemple, $F(x) =$ [LN] (x) est classée au niveau 1 parce que les dizaines de petites erreurs commises par HP-15C pendant son calcul de $F(x) = (f+\delta f)(x)$ se chiffrent à une perturbation $\delta f(x)$ inférieure à 0.6 sur le dernier (10^e) chiffre significatif de la sortie désirée $f(x) = \ln(x)$. Mais $F(x) =$ [SIN] (x) n'est pas du niveau 1 pour x radians parce que $F(x)$ peut être trop différent de $f(x) = \sin(x)$; par exemple $F(10^{14}$ [π] $) = 0.799...$ est de signe opposé à $f(10^{14}$ [π] $) = 0.784...$, si bien que l'équation $F(x) = (f+\delta f)(x)$ ne peut être vrai que si δf est de temps en temps plutôt supérieur à f, ce qui n'est pas bon.

Les systèmes réels ressemblent plus souvent à [SIN] qu'à [LN]. Le bruit dans la plupart des systèmes réels peut se cumuler occasionnellement pour engloutir la sortie désirée, au moins pour certaines entrées, et pourtant de tels systèmes ne méritent pas nécessairement d'être condamnés. Généralement un système réel F fonctionne de façon fiable parce que son bruit interne, bien que quelquefois important, n'a jamais de conséquences plus préjudiciables que celles qui pourraient être provoquées par quelque petite perturbation δx sur le signal d'entrée x. De tels systèmes peuvent être représentés par:

$$F(x) = (f + \delta f)(x + \delta f)$$

où δf est toujours petit comparé à f et où δx est toujours inférieur (ou comparable) au bruit Δx attendu pour contaminer x. Les deux termes δf et δx du bruit sont des bruits hypothétiques introduits pour expliquer diverses sources de bruits réellement attachées à F. Certains de ces bruits apparaissent comme des petites perturbations δx tolérables pour l'entrée – d'où le terme "analyse arrière des erreurs". Un tel système F, dont le bruit peut être comptabilisé par deux petites perturbations tolérables, est donc classé au niveau 2 pour les fins de notre exposé.



Petites perturbations d'entrée et de sortie (Niveau 2)

Aucune différence n'apparaîtra à première vue entre le niveau 1 et le niveau 2 pour les lecteurs habitués aux systèmes linéaires et aux petits signaux parce que les erreurs de ces systèmes peuvent se situer indifféremment au niveau de la sortie ou de l'entrée. Cependant, d'autres systèmes plus classiques, numériques ou non linéaires, n'admettent pas une réattribution arbitraire du bruit de sortie au bruit d'entrée (ni vice-versa).

Par exemple, la totalité de l'erreur dans [COS] peut-elle être attribuée, en écrivant simplement [COS](x) = cos(x + δx), à une perturbation d'entrée δx petite par rapport à l'entrée x ? Non, quand x est très petit. Par exemple, quand x s'approche de 10^{-5} radians, cos(x) arrive très près de 0.99999999995 et doit être alors arrondi soit à 1 = cos(0) soit à 0.9999999999 = cos(1.414 × 10^{-5}). Par conséquent, [COS](x) = cos(x + δx) est vraie seulement si δx est autorisée à de relativement grandes valeurs, presque aussi grandes que x quand x est très petit. Si nous souhaitons expliquer l'erreur dans [COS] en n'utilisant que des perturbations relativement petites, il nous en faut au moins deux : l'une, une perturbation $\delta x = (-6.58\ldots \times 10^{-14})\,x$, inférieure à l'arrondi de l'entrée ; l'autre, dans la sortie, comparable à l'arrondi à ce niveau et telle que [COS](x) = (cos + δcos)(x + δx) pour une certaine inconnue $|\delta\cos| \leq (6 \times 10^{-10})|\cos|$.

Comme [COS], tout système F du niveau 2 est caractérisé par deux petites tolérances seulement – appelons-les ε et η – qui résument ce qu'il vous suffit de connaître sur ce bruit interne du système. La tolérance ε impose une contrainte sur un bruit hypothétique à la sortie, $|\delta f| \leq \varepsilon |f|$, et η contient un bruit d'entrée, $|\delta| \leq \eta |x|$, qui peuvent apparaître dans une formule simple du type :

$$F(x) = (f + \delta f)(x + \delta x) \qquad \text{pour } |\delta f| \leq \varepsilon |f| \text{ et } |\delta x| \leq \eta\ |x|.$$

L'objectif de l'analyse récurrente de l'erreur est de s'assurer que la totalité du bruit interne de F peut réellement être ramenée à une formule aussi simple avec des petites tolérances ε et η satisfaisantes. Au mieux, l'analyse récurrente de l'erreur confirme que *la valeur réalisée F(x) est à peine différente de la valeur idéale f(x + δx) qui aurait été produite par une entrée x + δx à peine différente de l'entrée x réelle*, en donnant à l'expression "à peine" un sens quantitatif (ε et η). Mais l'analyse récurrente de l'erreur n'est valable que pour les systèmes F conçus avec soin pour assurer que toute source de bruit interne est équivalente au pire à une perturbation d'entrée ou de sortie petite de façon tolérable. Les premiers essais à la conception du système, particulièrement les programmes de calcul numérique, souffrent souvent de bruit interne d'une manière plus compliquée et plus désagréable, comme le montre l'exemple suivant.

Exemple 6 : La plus petite racine d'une équation quadratique. Les deux racines x et y de l'équation quadratique $c - 2bz + az^2 = 0$ sont réelles quand $d = b^2 - ac$ n'est pas négative. Alors, la plus petite racine y peut être considérée comme une fonction $y = f(a, b, c)$ des coefficients de l'équation quadratique :

$$f(a,b,c) \begin{cases} (b - \sqrt{d}\ \text{signe}\ (b))/a & \text{si } a \neq 0 \\ (c \quad b) \ 2 & \text{dans les autres cas.} \end{cases}$$

Si cette formule était traduite dans un programme $F(a, b, c)$ destiné à calculer $f(a, b, c)$, chaque fois que ac serait si petit par rapport à b^2 que la valeur calculée de d s'arrondirait à b^2, ce programme pourrait donner $F = 0$ même pour $f \neq 0$. Une telle erreur ne peut pas être expliquée par l'analyse récurrente de l'erreur parce qu'aucune perturbation relativement petite sur chaque coefficient a, b et c ne pourrait mener c vers zéro comme il le faudrait pour mettre à zéro la plus petite racine y. D'autre part, la formule algébrique équivalente :

$$f(a,b,c) \begin{cases} c \ (b - \sqrt{d}\ \text{signe}\ (b)) & \text{si diviseur} \neq 0 \\ 0 & \text{dans les autres cas.} \end{cases}$$

se traduit dans un programme F beaucoup plus précis, dont les erreurs ne sont pas plus gênantes qu'une perturbation sur le dernier (10^0) chiffre significatif de c. L'un de ces programmes est listé page 205 et doit être utilisé dans les cas courants en ingénierie, où la plus petite racine y est demandée avec une grande précision malgré le fait que l'autre racine, non désirée, de l'équation quadratique soit relativement grande.

Presque toutes les fonctions du HP-15C ont été connues pour que l'analyse récurrente de l'erreur tienne compte de façon satisfaisante de leurs erreurs. Les exceptions sont [SOLVE], [∫], et les touches statistiques [s], [L.R.] et [ŷ,r] qui risquent des errances dans des cas difficiles. Sinon, toute fonction F du calculateur destiné à produire $f(x)$, produit à la place une valeur $F(x)$ pas plus éloignée de $f(x)$ que si le premier x avait été perturbé à $x + \delta x$ avec $|\delta x| \leq \eta\,|x|$, et $f(x + \delta x)$ avait été perturbée à $(f + \delta f)(x + \delta x)$ avec $|\delta f| \leq \varepsilon\,|f|$. Les tolérances η et ε varient un petit peu d'une fonction à une autre ; en gras, nous pouvons dire que :

$\eta = 0$ et $\varepsilon < 10^{-9}$	pour toutes les fonctions de niveau 1,
$\eta < 10^{-2}$ et $\varepsilon < 6 \times 10^{-10}$	pour les autres fonctions, réelles et complexes.

Dans le cas des opérations matricielles, les valeurs absolues $|\delta x|$, $|x|$, $|\delta f|$ et $|f|$ doivent être remplacées par des normes matricielles $\|\delta \mathbf{x}\|$, $\|\mathbf{x}\|$, $\|\delta \mathbf{f}\|$ et $\|\mathbf{f}\|$ respectivement, qui sont décrites dans le chapitre 4 et évaluées à l'aide de [MATRIX] 7 ou [MATRIX] 8. Toutes les fonctions matricielles ne figurant pas au niveau 1, passent dans le niveau 2, avec approximativement:

$\eta \leq 10^{-12}\, n\, \eta$ et $\varepsilon < 10^{-9}$ pour les opérations matricielles autres que le déterminant [MATRIX] 9, [÷] et [1/x].

$\eta \leq 10^{-9}\, n\, \eta$ et $\varepsilon < 10^{-9}$ pour le déterminant [MATRIX] 9, [1/x] et [÷] avec un diviseur matriciel.

où n est la plus grande dimension de toute matrice impliquée dans l'opération.

Les implications d'une analyse récurrente de l'erreur ne semblent simples que lorsque la donnée x d'entrée arrive contaminée par un bruit Δx inévitable et sans corrélation, comme cela est souvent le cas. Lorsque nous désirons donc calculer $f(x)$, ce que nous pouvons espérer de mieux est d'obtenir $f(x+\Delta x)$, mais en fait nous obtenons $F(x+\Delta x) = (f+\delta f)(x+\Delta x+\delta x)$, où $|\delta f| \leq \varepsilon |f|$ et $|\delta x| \leq \eta(x)$.

Ce que nous obtenons est à peine pire que le meilleur à espérer, pourvu que les tolérances ε et η soient suffisamment petites, surtout si $|\Delta x|$ est susceptible d'être au moins aussi grande que $\eta |x|$. Naturellement, le meilleur à espérer peut être très mauvais, particulièrement si f possède une singularité plus proche de x que les tolérances sur les perturbations Δx et δx de x.

Analyse récurrente de l'erreur et singularités

Le mot "singularité" se réfère à la fois à une valeur spéciale de l'argument x et à la façon dont $f(x)$ s'égare lorsque x s'approche de cette valeur spéciale. Dans la plupart des cas, $f(x)$ ou sa première dérivée $f'(x)$ peuvent devenir infinies ou osciller violemment lorsque x s'approche de la singularité. Quelquefois, les singularités de $\ln|f|$ sont appelées singularités de f, incluant par là les zéros de f parmi ses singularités; ceci est valable lorsque la précision relative d'un calcul de f est en litige, comme nous le verrons. En ce qui nous concerne, la signification de "singularité" peut rester un petit peu vague.

Ce que nous voulons habituellement faire avec les singularités est de les éviter ou de les neutraliser. Par exemple, la fonction:

$$c(x) = \begin{cases} (1-\cos x)/x^2 & \text{si } x \neq 0 \\ 1/2 & \text{dans les autres cas} \end{cases}$$

n'a pas de singularité pour $x = 0$ même si ses composantes $1 - \cos x$ et x^2 (en fait, leurs logarithmes) se comportent singulièrement lorsque x s'approche de 0. Les singularités des composantes ont des effets indésirables sur le programme calculant $c(x)$. La plupart de ces effets sont neutralisés par le choix d'une meilleure formule:

$$c(x) = \begin{cases} \dfrac{1}{2}\left(\dfrac{\sin(x/2)}{x/2}\right)^2 & \text{si } x/2 \neq 0 \\ 1/2 & \text{dans les autres cas.} \end{cases}$$

Maintenant, la singularité peut être évitée en totalité en testant si $x/2 = 0$ dans le programme de calcul de $c(x)$.

L'analyse récurrente de l'erreur complique les singularités d'une façon plus facile à illustrer avec la fonction $\lambda(x) = \ln(1 + x)$ qui a servi à résoudre le problème de l'exemple 2. La procédure utilisée dans ce cas calculait $u = 1 + x$ (arrondi) $= 1 + x + \Delta x$. Alors:

$$\lambda(x) = \begin{cases} x & \text{si } u = 1 \\ \ln(u)\, x/(u-1) & \text{dans les autres cas.} \end{cases}$$

Cette procédure exploite le fait que $\lambda(x)/x$ a une singularité susceptible d'être enlevée pour $x = 0$, ce qui signifie que $\lambda(x)/x$ varie de façon continue et s'approche de 1 lorsque x s'approche de 0. Par conséquent, $\lambda(x)/x$ est relativement bien représenté par $\lambda(x + \Delta x)/(x + \Delta x)$ lorsque $|\Delta x| < 10^{-9}$, d'où:

$$\lambda(x) = x(\lambda(x)/x) \approx x(\lambda(x + \Delta x)/(x + \Delta x)) = x(\ln(u)/(u-1)),$$

tous calculés précisément parce que [LN] est dans le niveau 1. Que pourrait-il se passer si [LN] était dans le niveau 2?

Si [LN] était dans le niveau 2, une analyse récurrente de l'erreur "réussie" montrerait que, pour des arguments u proches de 1, [LN] $(u) = \ln(u + \delta u)$ avec $|\delta u| < 10^{-9}$. Alors, la procédure ci-dessus produirait, non pas $x(\ln(u)/(u-1))$, mais:

$$\begin{aligned} x(\ln(u + \delta u)/(u-1)) &= x\lambda(x + \Delta x + \delta u)/(x + \Delta x) \\ &= x(\lambda(x + \Delta x + \delta u)/(x + \Delta x + \delta u))\frac{x + \Delta x + \delta u}{x + \Delta x} \\ &\approx x(\lambda(x)/x)(1 + \delta u/(x + \Delta x)) \\ &= \lambda(x)(1 + \delta u/(x + \Delta x)). \end{aligned}$$

Quand $|x + \Delta x|$ n'est pas beaucoup plus grand que 10^{-9}, la dernière expression peut être terriblement différente de $\lambda(x)$. Par conséquent, la procédure qui a servi à résoudre l'exemple 2 ne marchera pas sur des machines pour lesquelles [LN] n'est pas de niveau 1. De telles machines *existent*, et avec elles, la procédure échoue pour certaines entrées inoffensives par ailleurs. Des échecs similaires se produisent sur des machines qui produisent $(u + \delta' u) - 1$ au lieu de $u - 1$ lorsque leur fonction [−] est de niveau 2 et non pas de niveau 1. Et ces machines qui produisent $\ln(u + \delta u)/(u + \delta' u - 1)$ au lieu de $\ln(u)/(u - 1)$, parce que [LN] et [−] sont toutes deux de niveau 2, seraient doublement vulnérables si ce n'est pour un accident mal compris qui lie habituellement les deux erreurs récurrentes δu et $\delta' u$ de telle façon que seulement la moitié des chiffres significatifs de λ calculé, et non pas tous, sont faux.

En résumé

Maintenant que la complexité introduite par l'analyse récurrente de l'erreur dans les singularités a été exposée, il est temps de résumer, de simplifier et de consolider ce qui a été dit jusqu'ici.

- De nombreuses procédures numériques produisent des résultats trop faux pour être justifiés par n'importe quelle analyse des erreurs, récurrente ou pas.
- Quelques procédures numériques produisent des résultats seulement légèrement plus mauvais que ceux qui auraient été obtenus par résolution exacte d'un problème ne différant que légèrement du problème considéré. Ces procédures, classées au niveau 2 en ce qui nous concerne, sont largement acceptées comme satisfaisante du point de vue de l'analyse récurrente de l'erreur.
- Les procédures du niveau 2 peuvent produire des résultats relativement écartés de ceux qui auraient été obtenus si aucune erreur n'avait été commise, mais des erreurs importantes peuvent survenir uniquement pour des données relativement proches d'une singularité de la fonction en cours de calcul.
- Les procédures du niveau 1 produisent des résultats relativement précis quelle que soit la proximité d'une singularité. De telles procédures sont rares mais préférables, parce que leurs résultats sont plus faciles à interpréter, particulièrement lorsque plusieurs variables sont impliquées.

Un exemple simple illustre ces quatre points.

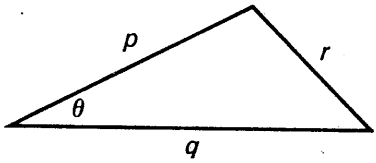
Exemple 7 : L'angle d'un triangle. La loi cosinus du triangle dit que :

$$r^2 = p^2 + q^2 - 2pq \cos\theta$$

pour la figure ci-dessous. Les calculs scientifiques nécessitent souvent que l'angle θ soit calculé à partir de valeurs, p, q et r des longueurs des côtés du triangle. Ce calcul est faisable pour que $0 < p \leqslant q + r$, $0 < q \leqslant p + r$ et $0 \leqslant r \leqslant p + q$, et ensuite :

$$0 \leqslant \theta = \cos^{-1}(((p^2 + q^2) - r^2)/(2pq)) \leqslant 180°;$$

sinon, aucun triangle n'existe avec ces longueurs de côtés, ou bien $\theta = 0/0$ est indéterminé.



La formule précédente pour θ définit une fonction $\theta = f(p, q, r)$ et aussi d'une façon naturelle, un programme $F(p, q, r)$ destiné à calculer cette fonction. Ce programme est appelé "A" ci-dessous, avec des résultats $F_A(p, q, r)$ tabulés pour certaines entrées p, q et r correspondant aux triangles très aplatis pour lesquels la formule souffre énormément de l'arrondi. L'absence de fiabilité de cette formule est bien connue de même que celle de la formule algébrique équivalente, mais plus fiable : $\theta = f(p, q, r) = 2\tan^{-1}\sqrt{ab/(cs)}$ où $s = (p+q+r)/2$, $a = s-p$, $b = s-q$ et $c = s-r$. Un autre programme $F(p, q, r)$ basé sur cette meilleure formle sera appelé "B" ci-dessous, avec des résultats $F_B(p, q, r)$ pour les entrées sélectionnées. Apparemment, F_B n'est pas beaucoup plus fiable que F_A. La plupart des résultats décevants pourraient être expliqués par l'analyse récurrente de l'erreur si nous supposons que les calculs donnent $F(p, q, r) = f(p + \delta p, q + \delta q, r + \delta r)$ pour des perturbations inconnues mais petites satisfaisant $|\delta p| < 10^{-9}\,|p|$, etc. Même si cette explication était vraie, elle aurait des conséquences troublantes et désagréables parce que les angles des triangles très aplatis peuvent varier relativement beaucoup quand les côtés sont relativement peu perturbés ; $f(p, q, r)$ est relativement instable pour les entrées marginales.

En réalité l'explication précédente est fausse. Aucune analyse récurrente de l'erreur ne pourrait tenir compte des résultats tabulés pour F_A et F_B dans le cas 1 ci-dessous à moins que des perturbations δp, δq et δr n'aient été autorisées pour corrompre le cinquième chiffre significatif de l'entrée, changeant 1 en 1.0001 ou en 0.9999. Ceci fait trop de bruit à tolérer dans un calcul sur 10 chiffres. Un meilleur programme, et de loin, est F_C ; il a le label "C" et est expliqué un peu plus loin.

trois dernières lignes de chaque compartiment du tableau ci-dessous, indi-nt les résultats de trois programmes "A", "B" et "C" basés sur trois formules érentes $F(p, q, r)$, toutes algébriquement équivalentes à :

$$\theta = f(p, q, r) = \cos^{-1}((p^2 + q^2 - r^2)/(2pq)$$

Résultats différents de trois programmes : F_A, F_B et F_C.

	Cas 1	Cas 2	Cas 3
p	1.	9.999999996	10
q	1.	9.999999994	5.000000001
r	1.00005×10^{-5}	3×10^{-9}	15.
F_A	0.	0.	180.
F_B	5.73072×10^{-4}	**Error 0**	180.
F_C	5.72986×10^{-4}	1.28117×10^{-8}	179.9985965

	Cas 4	Cas 5	Cas 6
p	0.527864055	9.999999996	9.999999999
q	9.472135941	3×10^{-9}	9.999999999
r	9.999999996	9.999999994	20.
F_A	**Error 0**	48.18968509	180.
F_B	**Error 0**	**Error 0**	180.
F_C	180.	48.18968510	**Error 0**

	Cas 7	Cas 8	Cas 9
p	1.00002	3.162277662	3.162277662
q	1.00002	2.3×10^{-9}	1.5555×10^{-6}
r	2.00004	3.162277661	3.162277661
F_A	**Error 0**	90.	90.
F_B	180.	70.52877936	89.96318706
F_C	180.	64.22853822	89.96315156

our utiliser un programme, introduisez p [ENTER] q [ENTER] r, exécutez le pro-ramme "A", "B" ou "C" et attendez l'approximation F du programme à $\theta = f$. eul le programme "C" est fiable.

Appuyez sur	Affichage
g DEG	
g P/R	
f CLEAR PRGM	000-
f LBL A	001-42,21,11
g x^2	002- 43 11
$x \gtrless y$	003- 34
g x^2	004- 43 11
g LSTx	005- 43 36
g R↑	006- 43 33
×	007- 20
$x \gtrless y$	008- 34
g LSTx	009- 43 36
g x^2	010- 43 11
+	011- 40
g R↑	012- 43 33
−	013- 30
$x \gtrless y$	014- 34
ENTER	015- 36
+	016- 40
÷	017- 10
g $\cos^{-1}$	018- 43 24
g RTN	019- 43 32
f LBL B	020-42,21,12
STO 1	021- 44 1
ENTER	022- 36
g R↑	023- 43 33
STO + 1	024-44,40, 1
g R↑	025- 43 33
STO + 1	026-44,40, 1
2	027- 2
STO ÷ 1	028-44,10, 1
R↓	029- 33
RCL − 1	030-45,30, 1
$x \gtrless y$	031- 34
RCL − 1	032-45,30, 1
×	033- 20
$\sqrt{x}$	034- 11
$x \gtrless y$	035- 34
RCL − 1	036-45,30, 1
RCL × 1	037-45,20, 1

Appuyez sur	Affichage
CHS	038- 16
√x	039- 11
g →P	040- 43 1
R↓	041- 33
×	042- 20
g RTN	043- 43 32
f LBL C	044-42,21,13
STO 0	045- 44 0
R↓	046- 33
g x≤y	047- 43 10
x≷y	048- 34
STO 1	049- 44 1
STO + 0	050-44,40, 0
x≷y	051- 34
STO + 0	052-44,40, 0
−	053- 30
g R↑	054- 43 33
STO − 1	055-44,30, 1
g LSTx	056- 43 36
ENTER	057- 36
RCL + 1	058-45,40, 1
√x	059- 11
f x≷ 0	060-42, 4, 0
√x	061- 11
STO × 0	062-44,20, 0
g CLx	063- 43 35
+	064- 40
R↓	065- 33
+	066- 40
f x≷ 1	067-42, 4, 1
g R↑	068- 43 33
g LSTx	069- 43 36
g x≤y	070- 43 10
GTO .9	071- 22 .9
R↓	072- 33
g TEST 2	073-43,30, 2
√x	074- 11
x≷y	075- 34
GTO .8	076- 22 .8
f LBL .9	077-42,21, .9

Appuyez sur	Affichage
g TEST 2	078-43,30, 2
√x	079- 11
g R↑	080- 43 33
f LBL .8	081-42,21, .8
−	082- 30
√x	083- 11
RCL 1	084- 45 1
√x	085- 11
×	086- 20
RCL 0	087- 45 0
g →P	088- 43 1
g x=0	089- 43 20
÷	090- 10
x≷y	091- 34
ENTER	092- 36
+	093- 40
g RTN	094- 43 32
g P/R	

Les résultats $F_C(p, q, r)$ sont corrects jusqu'à au moins neuf chiffres significatifs. Ils sont obtenus à partir d'un programme "C" très fiable bien que plutôt plus long que les programmes "A" et "B" non fiables. La méthode pour le programme "C" est la suivante.

1. Si $p < q$, échange de registre pour que $p \geq q$.
2. Calcul de $b = (p - q) + r$, $c = (p - r) + q$ et $s = (p + r) + q$.
3. Calcul de :

$$a = \begin{cases} r - (p - q) & \text{si } q \geq r \geq 0 \\ q - (p - r) & \text{si } r > q \geq 0 \\ \textbf{Error 0} & \text{dans les autres cas (pas de triangle).} \end{cases}$$

4. Calcul de $F_C(p, q, r) = 2 \tan^{-1}(\sqrt{ab}/\sqrt{cs})$

Cette procédure fournit $F_C(p, q, r) = \theta$ correct sur à peu près neuf chiffres significatifs, un résultat certainement plus facile à utiliser et à interpréter que les résultats donnés par les autres formules mieux connues. Mais le travail interne de cette procédure est difficile à expliquer ; en effet, cette procédure peut mal fonctionner sur certains calculateurs ou ordinateurs.

Cette procédure ne marche impeccablement que sur certaines machines comme le HP-15C, dont l'opération de soustraction est libre d'erreurs évitables et bénéficie ainsi de la propriété suivante : chaque fois que y est compris entre $x/2$ et $2x$, la soustraction n'introduit pas d'erreur d'arrondi dans la valeur calculée de $x - y$. Par conséquent, chaque fois que la compensation a pu laisser des erreurs relativement grandes, contaminant a, b ou c, la différence pertinente $(p - q)$ ou $(p - r)$ en vient à être libre d'erreur et la compensation devient avantageuse !

La compensation reste un problème sur les machines qui calculent $(x + \delta r) - (y + \delta y)$ au lieu de $x - y$ même si ni δx ni δy n'atteint la valeur 1 sur le dernier chiffre significatif de x et de y respectivement. Ces machines donnent $F_C(p, q, r) = f(p + \delta p, q + \delta q, r + \delta r)$ avec des perturbations δp, δq et δr sur les chiffres de terminaison, qui semblent toujours négligeables du point de vue de l'analyse récurrente de l'erreur mais qui peuvent avoir des conséquences déconcertantes. Par exemple, seul l'un des triplets (p, q, r) ou $(p + \delta p, q + \delta q, r + \delta r)$, pas les deux, peut constituer les longueurs des côtés d'un triangle faisable, si bien que F_C pourrait générer un message d'erreur alors qu'il ne le devrait pas, ou vice-versa, sur ces machines.

Analyse récurrente de l'erreur d'une inversion de matrice

La mesure habituelle de la grandeur d'une matrice $\mathbf{X}$ est une norme $\|\mathbf{X}\|$, telle qu'elle est calculée par [MATRIX] 7 ou par [MATRIX] 8 ; nous utiliserons la norme antérieure, la norme des rangs :

$$\|\mathbf{X}\| = \max_i \sum_j |x_{ij}|$$

dans les explications suivantes. Cette norme a des propriétés similaires à celles de la longueur d'un vecteur, ainsi que la propriété de multiplication :

$$\|\mathbf{XY}\| \leq \|\mathbf{X}\|\|\mathbf{Y}\|$$

Quand l'équation $\mathbf{Ax} = \mathbf{b}$ est résolue numériquement avec une matrice $\mathbf{A}$ donnée $n \times n$ et un vecteur-colonne $\mathbf{b}$, la solution calculée est un vecteur-colonne $\mathbf{c}$ qui satisfait à peu près la même équation que $\mathbf{x}$, c'est-à-dire :

$$(\mathbf{A} + \delta\mathbf{A})\mathbf{c} = \mathbf{b}$$

avec $\|\delta\mathbf{A}\| < 10^{-9} n \|\mathbf{A}\|$.

Par conséquent, le résidu $\mathbf{b} - \mathbf{Ac} = (\delta\mathbf{A})\mathbf{c}$ est toujours relativement petit ; très souvent, la norme résiduelle $\|\mathbf{b} - \mathbf{Ac}\|$ est inférieure à $\|\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}\|$ où $\bar{\mathbf{x}}$ est obtenu à partir de la vraie solution $\mathbf{x}$ par arrondi de chacun de ses éléments à dix chiffres significatifs. Donc, $\mathbf{c}$ ne peut différer de $\mathbf{x}$ de façon significative que si $\|\mathbf{A}^{-1}\|$ est relativement grand par rapport à $1/\|\mathbf{A}\|$;

$$\begin{aligned}\|\mathbf{x} - \mathbf{c}\| &= \|\mathbf{A}^{-1}(\mathbf{b} - \mathbf{Ac})\| \\ &\leqslant \|\mathbf{A}^{-1}\| \, \|\delta\mathbf{A}\| \, \|\mathbf{c}\| \\ &\leqslant 10^{-9} n \, \|\mathbf{c}\| / \sigma(\mathbf{A})\end{aligned}$$

où $\sigma(\mathbf{A}) = 1/(\|\mathbf{A}\|\|\mathbf{A}^{-1}\|)$ est l'inverse du nombre de condition et mesure à quelle proximité relative de $\mathbf{A}$ se situe la matrice singulière $\mathbf{S}$ la plus proche, puisque

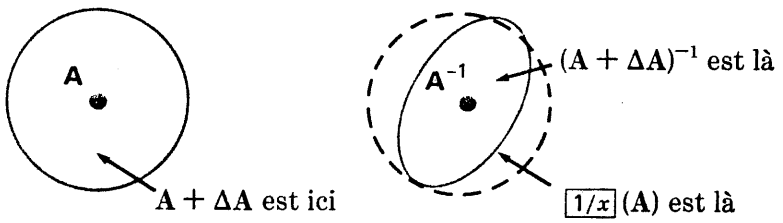
$$\min_{\det(\mathbf{S})=0} \|\mathbf{A} - \mathbf{S}\| = \sigma(\mathbf{A}) \, \|\mathbf{A}\|.$$

Ces relations et quelques-unes de leurs conséquences sont expliquées de façon approfondie au chapitre 4.

Le calcul de $\mathbf{A}^{-1}$ est plus compliquée. Chaque colonne de la matrice inverse calculée $\boxed{1/x}(\mathbf{A})$ est la colonne correspondante d'une certaine matrice $(\mathbf{A} + \delta\mathbf{A})^{-1}$, mais chaque colonne a son propre petit $\delta\mathbf{A}$. Par conséquent, aucun petit $\delta\mathbf{A}$, avec $\|\delta\mathbf{A}\| \leqslant 10^{-9} n \|\mathbf{A}\|$ n'a besoin d'exister en satisfaisant à peu près :

$$\|(\mathbf{A} + \delta\mathbf{A})^{-1} - \boxed{1/x}(\mathbf{A})\| \leqslant 10^{-9} \, \|\boxed{1/x}(\mathbf{A})\|$$

Un tel $\delta\mathbf{A}$ existe habituellement, mais pas toujours. Ceci ne contrarie pas la précédente affirmation que les opérations $\boxed{1/x}$ et $\boxed{\div}$ matricielles sont de niveau 2 ; elles sont couvertes par la seconde affirmation du résumé de la page 194. La précision de $\boxed{1/x}(\mathbf{A})$ peut être décrite dans les termes d'inverses de toutes les matrices $\mathbf{A} + \Delta\mathbf{A}$ si proches de $\mathbf{A}$ que $\|\Delta\mathbf{A}\| \leqslant 10^{-9} n \|\mathbf{A}\|$; la pire de ces matrices $(\mathbf{A} + \Delta\mathbf{A})^{-1}$ est au moins aussi loin de $\mathbf{A}^{-1}$ en norme que la matrice $\boxed{1/x}(\mathbf{A})$ calculée. La figure ci-dessous illustre la situation.



Quand $\mathbf{A} + \Delta\mathbf{A}$ se promène à travers les matrices avec $\|\Delta\mathbf{A}\|$ au moins aussi grande que l'arrondi dans $\|\mathbf{A}\|$, son inverse $(\mathbf{A} + \Delta\mathbf{A})^{-1}$ doit errer de façon au moins aussi éloignée de $\mathbf{A}^{-1}$ que la distance entre $\mathbf{A}^{-1}$ et la matrice $\boxed{1/x}(\mathbf{A})$ calculée. Tous ces mouvements sont très petits sauf si $\mathbf{A}$ est trop proche d'une matrice singulière, dans quel cas la matrice doit être pré-conditionnée loin de la proximité d'une singularité (voir chapitre 4).

Si parmi ces matrices $\mathbf{A} + \Delta\mathbf{A}$ voisines se dissimulent des matrices singulières, plusieurs $(\mathbf{A} + \Delta\mathbf{A})^{-1}$ et $\boxed{1/x}(\mathbf{A})$ risque d'être très différentes de $\mathbf{A}^{-1}$. Cependant, la norme résiduelle sera toujours relativement petite :

$$\frac{\|\mathbf{A}(\mathbf{A} + \Delta\mathbf{A})^{-1} - \mathbf{I}\|}{\|\mathbf{A}\|\,\|(\mathbf{A} + \Delta\mathbf{A})^{-1}\|} \leqslant \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} \leqslant 10^{-9}n.$$

Cette dernière inégalité reste vraie quand $\boxed{1/x}(\mathbf{A})$ remplace $(\mathbf{A} + \Delta\mathbf{A})^{-1}$.

Si $\mathbf{A}$ est suffisamment loin d'une singularité, de façon que :

$$1/\|(\mathbf{A} + \Delta\mathbf{A})^{-1}\| > 10^{-9}n\|\mathbf{A}\| \geqslant \|\Delta\mathbf{A}\|,$$

alors :

$$\frac{\|\mathbf{A}^{-1} - (\mathbf{A} + \Delta\mathbf{A})^{-1}\|}{\|(\mathbf{A} + \Delta\mathbf{A})^{-1}\|} \leqslant \frac{\|\Delta\mathbf{A}\|\,\|(\mathbf{A} + \Delta\mathbf{A})^{-1}\|}{1 - \|\Delta\mathbf{A}\|\,\|(\mathbf{A} + \Delta\mathbf{A})^{-1}\|}$$
$$\leqslant \frac{10^{-9}n\|\mathbf{A}\|\,\|(\mathbf{A} + \Delta\mathbf{A})^{-1}\|}{1 - 10^{-9}n\|\mathbf{A}\|\,\|(\mathbf{A} + \Delta\mathbf{A})^{-1}\|}.$$

Cette inégalité reste également vraie quant $\boxed{1/x}(\mathbf{A})$ remplace $(\mathbf{A} + \Delta\mathbf{A})^{-1}$, et alors tout ce qui est à droite peut être calculé, si bien que l'erreur dans $\boxed{1/x}(\mathbf{A})$ ne peut excéder une quantité évaluable. En d'autres termes, le rayon du cercle en pointillés sur la figure précédente peut être calculé.

Les estimations ci-dessus peuvent sembler pessimistes. Cependant, pour montrer pourquoi il n'existe généralement rien de mieux en plus vrai, considérons la matrice :

$$\mathbf{X} = \begin{bmatrix} 0.00002 & -50{,}000 & 50{,}000.03 & -45 \\ 0 & 50{,}000 & -50{,}000.03 & 45 \\ 0 & 0 & 0.00002 & -50{,}000.03 \\ 0 & 0 & 0 & 52{,}000 \end{bmatrix}$$

et

$$\mathbf{X}^{-1} = \begin{bmatrix} 50{,}000 & 50{,}000 & p & q \\ 0 & 0.00002 & 50{,}000.03 & 48{,}076.98077\ldots \\ 0 & 0 & 50{,}000 & 48{,}076.95192\ldots \\ 0 & 0 & 0 & 0.00001923076923\ldots \end{bmatrix}.$$

Idéalement, $p = q = 0$, mais l'approximation de $\mathbf{X}^{-1}$ par le HP-15C, c'est-à-dire $\boxed{1/x}(\mathbf{X})$, a $q = 9{,}643.269231$, soit une erreur relative

$$\frac{\|\mathbf{X}^{-1} - \boxed{1/x}(\mathbf{X})\|}{\|\mathbf{X}^{-1}\|} = 0.0964\ldots,$$

de près de 10 %. D'autre part, si $\mathbf{X} + \Delta\mathbf{X}$ ne diffère de $\mathbf{X}$ que dans sa seconde colonne où $-50{,}000$ et $50{,}000$ sont remplacés respectivement par $-50{,}000.000002$ et $49{,}999.999998$ (altérés sur le 11e chiffre significatif), alors $(\mathbf{X} + \Delta\mathbf{X})^{-1}$ ne diffère beaucoup de $\mathbf{X}^{-1}$ que dans la mesure où $p = 0$ et $q = 0$ doivent être remplacés par $p = 10{,}000.00600\ldots$ et $q = 9{,}615.396154\ldots$ d'où :

$$\frac{\|\mathbf{X}^{-1} - (\mathbf{X} + \Delta\mathbf{X})^{-1}\|}{\|\mathbf{X}^{-1}\|} = 0.196\ldots;$$

L'erreur relative dans $(\mathbf{X} + \Delta\mathbf{X})^{-1}$ est pratiquement le double de l'erreur relative dans $\boxed{1/x}(\mathbf{X})$. N'essayez pas de calculer $(\mathbf{X} + \Delta\mathbf{X})^{-1}$ directement, mais utilisez plutôt la formule :

$$(\mathbf{X} - \mathbf{c}\mathbf{b}^T)^{-1} = \mathbf{X}^{-1} + \mathbf{X}^{-1}\mathbf{c}\mathbf{b}^T\mathbf{X}^{-1} / (1 - \mathbf{b}^T\mathbf{X}^{-1}\mathbf{c}),$$

qui est valide pour tout vecteur-colonne $\mathbf{c}$ et tout vecteur-rang $\mathbf{b}^T$, et particulièrement pour

$$\mathbf{c} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbf{b}^T = [0 \quad 0.000002 \quad 0 \quad 0].$$

Malgré que :

$$\|\mathbf{X}^{-1} - \boxed{1/x}(\mathbf{X})\| < \|\mathbf{X}^{-1} - (\mathbf{X} + \Delta\mathbf{X})^{-1}\|,$$

on peut montrer qu'aucune perturbation très petite $\delta\mathbf{X}$ n'existe sur le dernier chiffre pour laquelle $(\mathbf{X} + \delta\mathbf{X})^{-1}$ est identique à $\boxed{1/x}(\mathbf{X})$ sur plus de cinq chiffres significatifs dans la norme.

Naturellement, aucune de ces horreurs ne se produirait si **X** n'était pas si singulière. Puisque $\|\mathbf{X}\|\|\mathbf{X}^{-1}\| > 10^{10}$, une modification dans **X** s'élevant à moins d'une unité sur le 10^{e} chiffre significatif de $\|\mathbf{X}\|$, pourrait rendre **X** singulière ; une telle modification pourrait remplacer l'un des éléments 0.00002 de la diagonale de **X** par zéro. Puisque **X** est si singulière, la précision de $\boxed{1/x}$ (**X**) dans ce cas est plutôt plus importante que ce que l'on attendait. Ce qui fait de cet exemple un cas particulier est une mauvaise échelle ; **X** a été obtenue à partir d'une matrice tout à fait convenable :

$$\tilde{\mathbf{X}} = \begin{bmatrix} 2. & -5. & 5.000003 & -4.5\times10^{-12} \\ 0 & 5. & -5.000003 & 4.5\times10^{-12} \\ 0 & 0 & 2. & -5.000003 \\ 0 & 0 & 0 & 5.2 \end{bmatrix}$$

en multipliant chaque rang et chaque colonne par une puissance de 10 soigneusement choisie. La division compensatrice des colonnes et des rangs de la matrice non moins convenable :

$$\tilde{\mathbf{X}}^{-1} = \begin{bmatrix} 0.5 & 0.5 & p & q \\ 0 & 0.2 & 0.5000003 & 0.4807698077... \\ 0 & 0 & 0.5 & 0.4807695192... \\ 0 & 0 & 0 & 0.1923076923... \end{bmatrix}$$

a donné $\mathbf{X}^{-1}$, avec $p = q = 0$. Le HP-15C calcule $\boxed{1/x}$ $(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}}^{-1}$ sauf que $q = 0$ est remplacé par $q = 9.6 \times 10^{-11}$, une modification mineure. Ceci illustre la façon dramatique dont l'échelle peut affecter la qualité perçue des résultats calculés. (Reportez-vous au chapitre 4 pour des explications détaillées sur l'échelle).

L'analyse récurrente de l'erreur est-elle une bonne chose ?

La seule bonne chose à dire sur l'analyse récurrente de l'erreur est qu'elle explique les erreurs internes d'une façon qui libère l'utilisateur d'un système de la nécessité d'une connaissance totale du fonctionnement interne du système. Étant données deux tolérances, l'une sur le bruit d'entrée δx et l'autre sur le bruit de sortie δf, l'utilisateur peut analyser les conséquences du bruit interne dans :

$$F(x) = (f + \delta f)(x + \delta x)$$

en étudiant les propriétés de propagation du bruit du système idéal f sans référence plus approfondie à la structure interne peut-être complexe de F.

Mais l'analyse récurrente de l'erreur n'est pas une panacée ; elle peut expliquer les erreurs mais pas les excuser. Parce qu'elle complique les calculs en cas de singularités, nous avons essayé d'éviter d'y recourir chaque fois que nous le pouvions. Si nous avions su comment éliminer le besoin de recourir à l'analyse récurrente de l'erreur pour chaque fonction intégrée du calculateur, à un coût raisonnable naturellement, nous l'aurions fait pour simplifier la vie de tout le monde. Mais cette simplicité fait appel à trop de rapidité et de mémoire pour la technologie actuelle. L'exemple suivant illustre les compromis à réaliser.

Exemple 6 (suite). Le programme figurant ci-dessous résoud l'équation quadratique réelle $c - 2bz + az^2 = 0$ pour des racines réelles ou complexes.

Pour utiliser ce programme, introduisez les constantes réelles dans la pile (c [ENTER] b [ENTER] a) et exécutez le programme "A".

Les racines x et y vont apparaître dans les registres X et Y. Si ces racines sont complexes, l'indicateur C s'allume pour signaler que le mode complexe a été activé. Le programme utilise les labels "A" et ".9" et le registre d'index (mais aucun des registres 0 à .9) ; le programme peut donc être appelé immédiatement par d'autres programmes en tant que sous-programme. Les programmes appelant (après désarmement de l'indicateur 8 si nécessaire) peuvent découvrir si les racines sont réelles ou complexes par simple test de l'indicateur 8 qui n'est armé que si les racines sont complexes.

Les racines x et y sont si ordonnées que $|x| \geq |y|$ sauf peut-être lorsque $|x|$ et $|y|$ sont identiques sur plus de neuf chiffres significatifs. Les racines sont aussi précises que si le coefficient c ayant été d'abord perturbé sur son 10^e chiffre significatif, l'équation perturbée aurait été résolue exactement et ses racines arrondies à 10 chiffres significatifs. Par conséquent, les racines calculées sont identiques aux racines de la quadratique données sur au moins cinq chiffres significatifs. Plus généralement, si les racines x et y sont semblables sur n chiffres significatifs pour n positif ≤ 5, elles sont correctes sur au moins $10 - n$ chiffres significatifs sauf en cas de dépassement de capacité supérieur ou inférieur.

Appuyez sur	Affichage
[g] [P/R]	
[f] CLEAR [PRGM]	000–
[f] [LBL] [A]	001–42,21,11
[ENTER]	002– 36
[g] [R↑]	003– 43 33
[×]	004– 20
[g] [LSTx]	005– 43 36

Appuyez sur	Affichage
[x≷y]	006- 34
[g] [R↑]	007- 43 33
[STO] [I]	008- 44 25
[g] [x^2]	009- 43 11
[-]	010- 30
[g] [TEST] 1	011-43,30, 1
[GTO] .9	012- 22 .9
[CHS]	013- 16
[$\sqrt{x}$]	014- 11
[f] [x≷] [I]	015-42, 4,25
[g] [TEST] 2	016-43,30, 2
[RCL] [-] [I]	017-45,30,25
[g] [TEST] 3	018-43,30, 3
[RCL] [+] [I]	019-45,40,25
[g] [TEST] 0	020-43,30, 0
[÷]	021- 10
[g] [LSTx]	022- 43 36
[g] [R↑]	023- 43 33
[+]	024- 10
[g] [RTN]	025- 43 32
[f] [LBL] .9	026-42,21, .9
[$\sqrt{x}$]	027- 11
[RCL] [I]	028- 45 25
[g] [R↑]	029- 43 33
[+]	030- 10
[x≷y]	031- 34
[g] [LSTx]	032- 43 36
[÷]	033- 10
[f] [I]	034- 42 25
[ENTER]	035- 36
[f] [Re≷Im]	036- 42 30
[CHS]	037- 16
[f] [Re≷Im]	038- 42 30
[g] [RTN]	039- 43 32
[g] [P/R]	

La méthode utilise $d = b^2 - ac$.

Si $d < 0$, les racines font partie d'une paire complexe conjuguée:

$$(b/a) \pm i\sqrt{-d}/a.$$

Si $d \geq 0$, les racines sont des nombres x et y réels calculés par :

$$s = b + \sqrt{d}\ \text{signe}\ (b)$$

$$x = s/a$$

$$y = \begin{cases} c/s & \text{si } s \neq 0 \\ 0 & \text{si } s \doteq 0. \end{cases}$$

Le calcul de s évite une compensation destructive.

Quand $a = 0 \neq b$, la plus grande racine x (qui devrait être ∞) rencontre une division par zéro (**Error 0**) qui peut être effacée en appuyant trois fois sur [R↓] pour exhiber la plus petite racine y correctement calculée. Mais quand les trois coefficients disparaissent, le message **Error 0** signale que les deux racines sont arbitraires.

Les résultats de plusieurs cas sont rassemblés ci-dessous.

	Cas 1	Cas 2	Cas 3	Cas 4
c	3	4	1	654,321
b	2	0	1	654,322
a	1	1	10^{-13}	654,323
Racines	Réelles	Complexes	Réelles	Réelles
	3	$0 \pm 2i$	2×10^{13}	0.9999984717
	1		0.5	0.9999984717

	Cas 5	Cas 6
c	46,152,709	12,066,163
b	735,246	987,644
a	11,713	80,841
Racines	Réelles	Complexes
	62.77179203	$12.21711755 \pm i0.001377461$
	62.77179203	

Les trois derniers cas montrent la sévérité de résultats de la perturbation sur le 10e chiffre significatif de tout coefficient de toute équation quadratique dont les racines coïncident presque. Les racines correctes dans ces cas sont les suivantes :

Cas 4 : 1 et 0.9999969434
Cas 5 : $62.77179203 \pm i8.5375 \times 10^{-5}$
Cas 6 : $12.21711755 \pm i0.001374514$.

En dépit des erreurs sur le cinquième chiffre significatif des résultats, le sous-programme "A" est suffisant pour presque toutes les applications d'équations quadratiques dans les domaines de l'ingénierie et de la recherche. Ses résultats sont corrects sur neuf chiffres significatifs pour la plupart des données, avec c, b et a représentables exactement à l'aide de seulement cinq chiffres significatifs ; et les racines calculées sont correctes sur au moins cinq chiffres significatifs dans tous les cas parce qu'elles ne peuvent pas être pires que si les données avaient été introduites avec des erreurs sur le 10e chiffre significatif. Néanmoins, certains lecteurs vont se sentir mal à l'aise avec des résultats calculés sur 10 chiffres significatifs mais corrects sur 5 seulement. Ne serait-ce que pour simplifier leur compréhension de la relation entre les données d'entrée et les résultats sortis, ils peuvent encore préférer des racines correctes sur neuf chiffres significatifs dans tous les cas.

Il existe des programmes qui, tout en tenant compte que de 10 chiffres significatifs pendant l'arithmétique, vont calculer correctement les racines de toute équation quadratique sur au moins neuf chiffres significatifs, quelle que soit la proximité de ces racines. Ces programmes calculés $d = b^2 - ac$ par quelque subterfuge équivalent au traitement de 20 chiffres significatifs chaque fois que b^2 et ac se "compensent" presque, mais ces programmes sont beaucoup plus longs et beaucoup plus lents que le petit sous-programme "A" donné précédemment. Le sous-programme "B" ci-dessous qui utilise l'un de ces subterfuges*, est un programme très court qui garantit neuf chiffres significatifs corrects sur un calculateur 10 chiffres. Il utilise les labels "B", ".7" et ".8", les registres R_0 et R_9 et le registre d'index. Pour l'utiliser, introduisez c [ENTER] b [ENTER] a, exécutez le sous-programme "B" et attendez, comme précédemment, vos résultats.

Appuyez sur	Affichage
[g] [P/R]	
[f] CLEAR [PRGM]	000–
[f] [LBL] [B]	001–42,21,12
[STO] [I]	002– 44 25
[R↓]	003– 33

* Le programme "B" exploite une propriété intéressante des touches [Σ−] et [Σ+] par laquelle certains calculs peuvent se faire sur 13 chiffres significatifs avant l'arrondi à 10 chiffres.

Appuyez sur	Affichage
[STO] 0	004- 44 0
[STO] 8	005- 44 8
[x⇄y]	006- 34
[STO] 1	007- 44 1
[STO] 9	008- 44 9
[f] [SCI] 2	009-42, 8, 2
[f] [LBL] .8	010-42,21, .8
[f] CLEAR [Σ]	011- 42 32
[RCL] 8	012- 45 8
[STO] 7	013- 44 7
[RCL] [÷] [I]	014-45,10,25
[g] [RND]	015- 43 34
[RCL] [I]	016- 45 25
[g] [Σ-]	017- 43 49
[RCL] 9	018- 45 9
[f] [x⇄] 7	019-42, 4, 7
[x⇄y]	020- 34
[RCL] 8	021- 45 8
[g] [Σ-]	022- 43 49
[R↓]	023- 33
[g] [Σ-]	024- 43 49
[RCL] 7	025- 45 7
[g] [ABS]	026- 43 16
[RCL] 9	027- 45 9
[g] [ABS]	028- 43 16
[g] [x≤y]	029- 43 10
[GTO] [B]	030- 22 12
[ENTER]	031- 36
[g] [R↑]	032- 43 33
[STO] 8	033- 44 8
[RCL] 7	034- 45 7
[STO] 9	035- 44 9
[g] [ABS]	036- 43 16
[EEX]	037- 26
2	038- 2
0	039- 0
[×]	040- 20
[RCL] 1	041- 45 1
[g] [ABS]	042- 43 16
[g] [x≤y]	043- 43 10

Appuyez sur	Affichage
GTO .8	044- 22 .8
f LBL B	045-42,21,12
f FIX 9	046-42, 7, 9
RCL 8	047- 45 8
g x^2	048- 43 11
STO 7	049- 44 7
RCL I	050- 45 25
RCL 9	051- 45 9
g $\Sigma-$	052- 43 49
RCL 7	053- 45 7
g TEST 2	054-43,30, 2
GTO .7	055- 22 .7
$\sqrt{x}$	056- 11
f $x \gtrless$ 0	057-42, 4, 0
g TEST 2	058-43,30, 2
RCL − 0	059-45,30, 0
g TEST 3	060-43,30, 3
RCL + 0	061-45,40, 0
f $x \gtrless$ 1	062-42, 4, 1
g TEST 0	063-43,30, 0
RCL ÷ 1	064-45,10, 1
RCL 1	065- 45 1
RCL ÷ I	066-45,10,25
g RTN	067- 43 32
f LBL .7	068-42,21, .7
CHS	069- 16
$\sqrt{x}$	070- 11
RCL ÷ I	071-45,10,25
ENTER	072- 36
CHS	073- 16
RCL 0	074- 45 0
RCL I	075- 45 25
÷	076- 10
$x \gtrless y$	077- 34
f I	078- 42 25
ENTER	079- 36
g R↑	080- 43 33
f I	081- 42 25
g RTN	082- 43 32
g P/R	

La précision de ce programme est phénoménale : meilleure que neuf chiffres significatifs même pour la partie imaginaire de racines complexes pratiquement indistinctes (comme lorsque $c = 4{,}877{,}163{,}849$ et $b = 4{,}877{,}262{,}613$ et $a = 4{,}877{,}361{,}379$) ; si les racines sont des entiers, réels ou complexes, et si $a = 1$, alors les racines sont calculées exactement (comme lorsque $c = 1{,}219{,}332{,}937 \times 10^1$, $b = 111{,}111.5$ et $a = 1$). Mais le programme est coûteux ; il utilise plus de deux fois plus de mémoire pour le programme et les données que le sous-programme "A", et prend beaucoup plus de temps pour réaliser une précision sur 9 chiffres significatifs au lieu de 5 dans quelques cas où cela n'est pas toujours important parce que les coefficients de l'équation quadratique peuvent difficilement être calculés exactement. Si l'un des coefficients c, b ou a est incertain de une unité sur son 10^e chiffre significatif, le sous-programme "B" en fait trop. Le sous-programme "B" doit être considéré comme un outil de luxe à n'utiliser que dans des circonstances exceptionnelles, laissant au sous-programme "A" la gestion des traitements de tous les jours.

Index alphabétique

Les numéros de page en gras renvoient aux pages principales.

E

M

N

O

P

Q

R

S

T

Hewlett-Packard France:
Société Anonyme au capital de 124 000 000 F, régie par les articles 118 à 150 de la loi sur les sociétés commerciales. RCS, Corbeil Essonnes B 709 805 030

Siège social Division commerciale d'Orsay: ZI de Courtabœuf
91947 Les Ullis Cedex, tél. (6) 907 78 25

Bureau commercial d'Aix-en-Provence: ZI Mercure B
Rue Berthelot, 13763 Les Milles Cedex, tél. (42) 59 41 02

Bureau commercial d'Alençon:
64, rue Marchand-Saillant, 61000 Alençon, tél. 16 (33) 29 04 42

Bureau commercial de Besançon:
28, rue de la République, 25000 Besançon, tél. (81) 83 16 22

Bureau commercial Blanc-Mesnil:
Rue de la Commune de Paris, BP 300, 93153 Le Blanc-Mesnil, tél. (1) 865 44 52

Bureau commercial de Bordeaux:
Avenue du Président-Kennedy, 33700 Mérignac, tél. (56) 34 00 84

Bureau commercial de Brest:
13, place Napoléon-III, 29000 Brest, tél. (98) 03 38 35

Bureau commercial d'Évry:
Tour Lorraine, Boulevard de France, 91035 Évry Cedex, tél. (6) 077 96 60

Bureau commercial de Lille:
Rue Van Gogh, Immeuble Péricentre, 59658 Villeneuve-d'ascq Cedex, tél. (20) 91 41 25

Bureau commercial de Lyon:
Chemin des Mouilles, BP 162, 69130 Écully, tél. (7) 833 81 25

Bureau commercial de Metz:
Garolor, ZAC d'Ennecy, 57640 Vigy, tél. (8) 771 20 22

Bureau commercial de Nantes:
Immeuble "Les 3 B", Nouveau chemin de la garde, ZAC de Bois-Briand, 44085 Nantes Cedex, tél. (40) 50 32 22

Bureau commercial d'Orléans::
125, rue du Faubourg-Bannier, 45000 Orléans, tél. (38) 68 01 63

Bureau de Paris-Porte Maillot:
15, boulevard de l'Amiral Bruix, 75782 Paris 16, tél. (1) 502 12 20

Bureau commercial de Pau:
124, boulevard Tourasse, 64000 Pau, tél. (59) 80 38 02

Bureau commercial de Rennes:
2, allée de la Bourgonnette, 35100 Rennes, tél. (99) 51 42 44

Bureau commercial de Rouen:
98, avenue de Bretagne, 76100 Rouen, tél. (35) 63 57 66

Bureau commercial de Strasbourg:
4, rue Thomas-Mann, BP 56, 67033 Strasbourg Cedex, tél. (88) 28 56 46

Bureau commercial de Toulouse:
Péricentre de Cépière, 20 chemin de la Cépière, 31081 Toulouse Cedex, tél. (61) 40 11 12

Bureau commercial de Valence:
9, rue Baudin, 26000 Valence, tél. (75) 42 76 16

Hewlett-Packard Belgium S.A./N.V.:
100, boulevard de la Woluwe, B-1200 Brussels, tél. (02) 762 32 00

Hewlett-Packard Schweiz AG:
Château bloc 19, CH-1219 Le Lignon-Genève, tél. (022) 96 03 22

Hewlett-Packard S.A., pour les pays du bassin méditerranéen, Afrique du Nord et Moyen-Orient:
Atrina Center, 32 Kiffisias Avenue Paradissos-Amaroussion, Athènes Grèce, tél. 8080 337/429/359/1741

Hewlett-Packard Canada Ltd:
17500 Trans Canada Highway South Service Road, Kirkland-Québec H9J2M5 Canada, tél. (514) 697 42 32

Hewlett-Packard S.A., Direction pour l'Europe:
150, route du Nant-d'Avril, P.O. Box - CH-1217 Meyrin 2, Genève Suisse

HEWLETT PACKARD

00015-90014 01-83-CP

Imprimé à Singapour