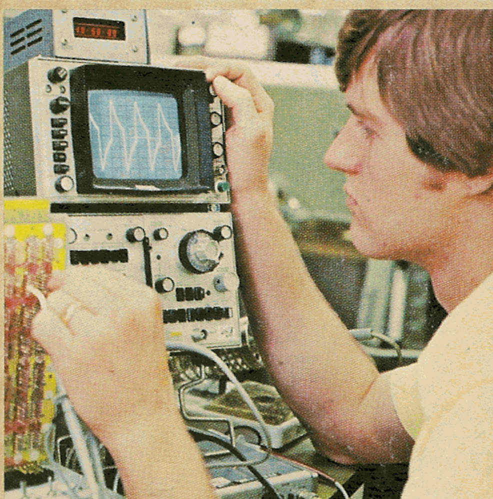
HEWLETT-PACKARD

HP-34C

Manuel d'utilisation et guide de programmation



Manuel d'utilisation
et
Guide de programmation
du
Calculateur scientifique
programmable HP-34C

April 1980

00034-90008 Rev B 04/80

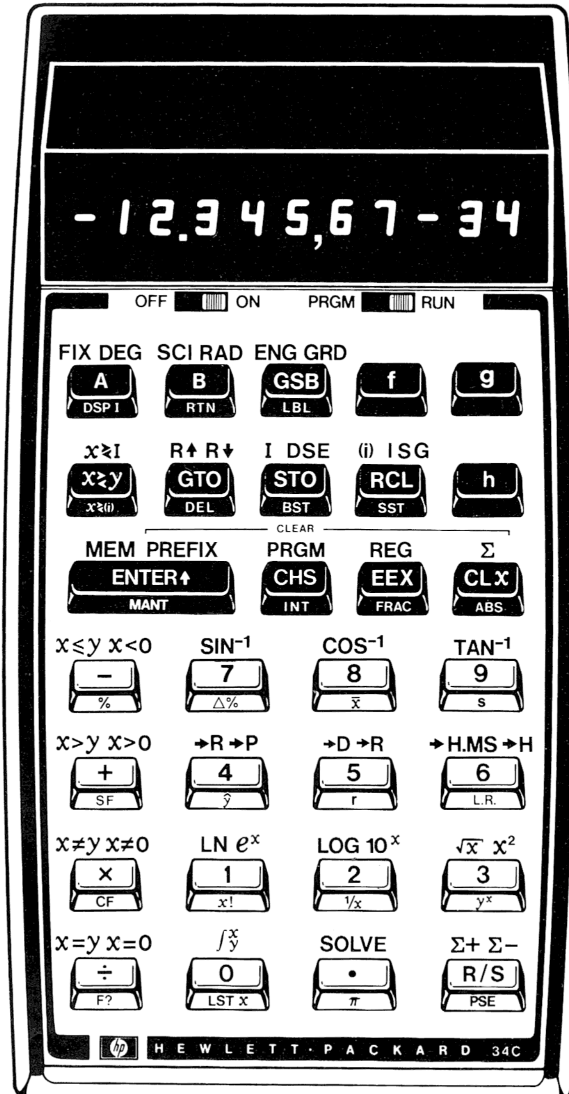
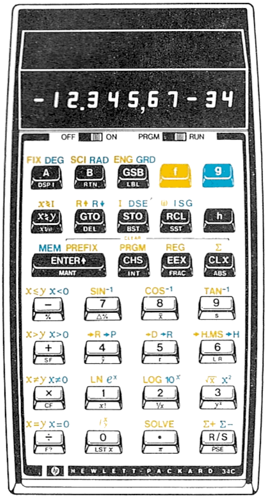

-12,345,67-34
OFF ON PRGM RUN
FIX DEG SCI RAD ENG GRD
A B GSB f g
DSP I RTN LBL
x≷I R↑ R↓ I DSE (i) ISG
x≷y GTO STO RCL h
x≷(i) DEL BST SST
CLEAR
MEM PREFIX PRGM REG Σ
ENTER↑ CHS EEX CLx
MANT INT FRAC ABS
x≤y x<0 SIN⁻¹ COS⁻¹ TAN⁻¹
− 7 8 9
% Δ% x̄ s
x>y x>0 →R →P →D →R →H.MS →H
+ 4 5 6
SF ŷ r L.R.
x≠y x≠0 LN eˣ LOG 10ˣ √x x²
× 1 2 3
CF x! 1/x yˣ
x=y x=0 ∫xy SOLVE Σ+ Σ−
÷ 0 • R/S
F? LST x π PSE
HEWLETT·PACKARD 34C

Table des matières

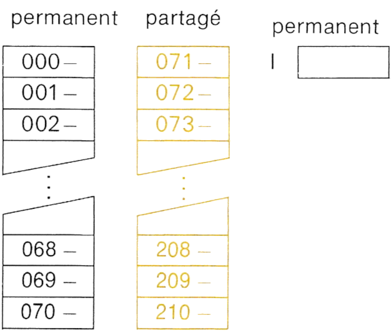
Le calculateur scientifique programmable HP-34C



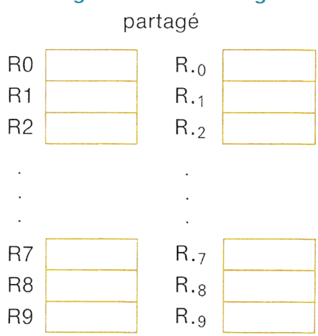
Pile opérationnelle

Hewlett-Packard 34C

Mémoire programme



Registres de stockage



La répartition de base de la mémoire correspond à 70 lignes de programme et à 20 registres de données, plus le registre I. Le calculateur convertit automatiquement au fur et à mesure des besoins chaque registre de stockage en sept lignes de programme. La conversion commence à $R._9$ et finit à R0.

LE HP-34C

Calculateur scientifique programmable

OFF ▢■ ON Commutateur de mise sous tension
PRGM ▢■ RUN Commutateur programme manuel.

INDEX DES TOUCHES DE FONCTION

Les touches du clavier permettent à l'utilisateur d'exécuter individuellement chaque fonction. Toutes les fonctions citées ci-après peuvent être exécutées au clavier ou comme instructions d'un programme.

Touches préfixe

[f] pressée avant une autre touche, sélectionne la fonction imprimée en jaune au-dessus de la touche
[g] pressée avant une autre touche, sélectionne la fonction imprimée en bleu au-dessus de la touche
[h] pressée avant une autre touche, sélectionne la fonction imprimée en noir sur la face inclinée de la touche

CLEAR [PREFIX] annule une séquence de touches partiellement introduites telles que [f], [f] [SCI], [g], [h], [h] [SF], [STO] [+], etc.

Introduction des nombres

[ENTER↑] copie le nombre affiché (Registre X) dans le registre Y
[CHS] change le signe de la mantisse ou de l'exposant du nombre affiché (Registre X).

[EEX] introduit l'exposant. Les nombres introduits après la pression de cette touche sont des puissances de 10
[0] à [9] touches numériques
[•] point décimal

Altération des nombres

[INT] prend la partie entière du nombre affiché
[FRAC] prend la partie décimale du nombre affiché
[ABS] prend la valeur absolue du nombre affiché

Manipulation des nombres

[R↓] exécute une permutation circulaire vers le bas des contenus de la pile opérationnelle
[R↑] exécute une permutation circulaire vers le haut, des contenus de la pile opérationnelle
[x⇄y] échange les contenus des registres X et Y
[CLx] annule le contenu du registre X

Stockage

[STO] suivi d'une adresse, stocke le nombre affiché dans le registre spécifié (R_0 - R_9, $R_{.0}$ à $R_{.9}$, I). Sert aussi pour exécuter des calculs arithmétiques en mémoire
[RCL] suivi d'une adresse, rappelle à l'affichage le contenu du registre spécifié (R_0 — R_9, $R_{.0}$ à $R_{.9}$, I)
[LST x] rappelle à l'affichage le contenu de X précédant la dernière opération

CLEAR REG annule les contenus de tous les registres de stockage (R_0 à R_9, $R_{.0}$ à $R_{.9}$, I)

Contrôle de l'affichage

FIX sélectionne la notation fixe
SCI sélectionne la notation scientifique
ENG sélectionne la notation ingénieur
DSP I spécifie le nombre de décimales affichées égal au contenu de I (0 à 9)
MANT affiche les dix chiffres significatifs de la mantisse tant que la pression est maintenue

Pourcentages

Δ% calcule la différence en pourcentage entre les contenus des registres X et Y
% calcule X % du contenu de Y

Mathématiques

−, +, ×, ÷ effectuent les opérations arithmétiques correspondantes
$\sqrt{x}$ calcule la racine carrée du nombre affiché
x^2 calcule le carré du nombre affiché
x! calcule la fonction factorielle x (x!) ou la fonction gamma (Γ (1 + x)) où x est le nombre affiché
1/x calcule l'inverse du nombre affiché
π affiche la valeur de (3,141592654)
$\int_y^x$ calcule l'intégrale finie $\int_y^x$ (x) dx, où l'expression f(x) est introduite en mémoire programme.
SOLVE calcule la ou les racines réelles de l'équation f(x) = 0 introduite en mémoire programme

Statistiques

CLEAR Σ annule les registres statistiques (R_0 à R_5)
Σ+ effectue des sommations des contenus de X et de Y dans les registres R_0 à R_5
Σ− corrige les sommations des contenus de X ou de Y dans les registres R_0 à R_5
$\bar{x}$ calcule la moyenne des valeurs x et y accumulées par Σ +
s calcule l'écart type des valeurs x et y accumulées par Σ +
$\hat{y}$ estimation linéaire. Calcule la valeur estimée de y pour un x donné par la méthode des moindres carrés
r coefficient de corrélation. Calcule le degré d'ajustement entre les valeurs x et y accumulées par Σ + et la fonction linéaire qui permet l'ajustement
L.R. régression linéaire. Calcule l'ordonnée à l'origine sur y et la pente de la droite en ajustant au mieux les valeurs x et y accumulées par Σ +. La valeur de l'ordonnée à l'origine sur y est placée dans le registre X et celle de la pente dans le registre Y.

Conversions polaires rectangulaires

→R convertit le module r dans X et l'argument (θ) dans y d'un vecteur, en coordonnées rectangulaires (abscisse dans X et ordonnée dans Y)
→P convertit les coordonnées rectangulaires (abscisse dans X et ordonnéee dans Y) en coordonnées polaires (module r dans X et argument (θ) dans y)

Trigonométrie

DEG sélectionne les angles en degrés

RAD sélectionne les angles en radians
GRD sélectionne les angles en grades
SIN, COS, TAN effectue la fonction correspondante sur la valeur affichée
SIN^{-1}, COS^{-1}, TAN^{-1} effectue la fonction correspondante sur la valeur affichée
→R convertit la valeur affichée de degrés en radians
→D convertit la valeur affichée de radians en degrés
→H.MS convertit la valeur affichée du système décimal en système sexagésimal
→H convertit la valeur affichée du système sexagésimal en système décimal

Registre de contrôle I

$x \gtrless I$ échange les contenus des registres X et I
$x \gtrless (i)$ échange les contenus du registre X et de celui dont l'adresse est le contenu de I
I registre de stockage pour les opérations d'incrémentation/décrémentation et pour les opérations indirectes
(i) commande d'opérations indirectes. Utilisée avec STO et RCL pour le stockage, le rappel de données et les opérations arithmétiques dans les registres indirects
DSP I affiche le nombre de décimales spécifié par le contenu du registre I
DSE décrémente la valeur du compteur et saute une ligne si la nouvelle valeur du compteur est égale ou inférieure à la valeur test spécifiée
ISG incrémente la valeur du compteur et saute une ligne si la nouvelle valeur du compteur est strictement supérieure à la valeur test spécifiée

Logarithmes et exponentielles

LN Calcule le logarithme népérien (base : 2,718281828) du nombre affiché.
e^x Calcule l'exponentielle du nombre affiché.
LOG Calcule le logarithme décimal du nombre affiché.
10^x Antilogarithme décimal.
y^x Fonction puissance.

INDEX DE PROGRAMMATION

Certaines touches décrites ci-dessous n'opèrent qu'en mode programme (PRGM).
MEM Affiche l'allocation mémoire courante.
A B Touches de programmation à définir par l'utilisateur pour l'étiquetage et l'exécution de programmes.
0-9 Labels. Précédé de , définissent le début d'un programme.
LBL Utilisé avec A-B ou 0-9, indique le début d'un programme ou d'un sous-programme.
GTO Utilisé avec A-B, 0-9 ou I ; au clavier, indique au calculateur de chercher séquentiellement dans la mémoire programme le label indiqué et de s'arrêter ; en cours de programme indique au calculateur de chercher séquentiellement dans la mémoire programme le label indiqué et de reprendre l'exécution à cet endroit.
GSB Utilisé avec A-B, 0-9 ou I ; au clavier indique au calculateur de chercher séquentiellement dans la

mémoire programme le label indiqué et d'y commencer l'exécution ; en cours de programme indique au calculateur de transférer l'exécution au sous-programme correspondant au label spécifié.

[GTO] [•] nn Positionne le calculateur à la ligne de programme spécifiée si elle existe.

[BST] Déplace le pointeur d'exécution d'une ligne en arrière.

[SST] Déplace le pointeur d'exécution d'une ligne en avant.

[DEL] En mode PRGM, permet d'enlever l'instruction affichée de la mémoire programme. Les instructions suivantes, sont déplacées d'une ligne en arrière.

CLEAR [PRGM] Efface toutes les instructions de la mémoire programme et place le pointeur à la ligne 000.

[PSE] PAUSE de une seconde et affichage du contenu du registre X.

[R/S] Lance l'exécution à l'emplacement courant du pointeur dans la mémoire programme. Arrête l'exécution si le programme tourne.

[RTN] Indique au pointeur d'exécution de retourner à la ligne 000 ou dans le cas d'un sous-programme, de retourner à la ligne appropriée.

[SF] Suivi d'un chiffre de 0 à 3, arme l'indicateur binaire spécifié.

[CF] Suivi d'un chiffre de 0 à 3, désarme l'indicateur binaire spécifié.

[F?] Si l'indicateur binaire spécifié est armé, le calculateur poursuit l'exécution à la ligne suivante ; s'il est désarmé, le calculateur saute une ligne avant de reprendre l'exécution.

[$x \leq y$] [$x > y$] [$x \neq y$] [$x = y$]
[$x < 0$] [$x > 0$] [$x \neq 0$] [$x = 0$]

Chaque test compare le contenu du registre x à celui de y ou à 0. Si la relation est vérifiée, le calculateur poursuit l'exécution à la ligne suivante ; si elle est fausse, le calculateur saute une ligne, avant de reprendre l'exécution.

Chapitre 1

PRÉSENTATION DU HP-34C

Vous avez acheté un calculateur scientifique programmable HP-34C à mémoire permanente et vous ne pourrez que vous féliciter de votre choix. Ce calculateur, qui utilise la notation polonaise inverse, est un instrument de calcul sans égal, faisant un jeu des opérations les plus complexes. Il est appréciable à plusieurs égards :

C'est un calculateur scientifique, doté d'un clavier dont chaque touche peut commander jusqu'à 4 fonctions distinctes, ce qui lui confère une grande puissance de calcul.

C'est une « machine faite pour résoudre vos problèmes ». En suivant les instructions pas-à-pas fournies dans les manuels d'application du HP-34C, vous pouvez entrer des dizaines de programmes choisis dans les différentes bibliothèques : mathématiques, ingénierie, statistiques, jeux, finance, et autres applications, et démarrer immédiatement vos calculs.

C'est un calculateur programmable « personnalisé » : de programmation et d'emploi faciles, il n'exige aucune connaissance spéciale ni d'étude des langages de programmation. Ses caractéristiques étonneront même les utilisateurs des ordinateurs les plus perfectionnés :

- Mémoire permanente : le calculateur y conserve programmes et données, même quand il est éteint.
- Affectation mémoire automatique : avec ses 70 lignes de mémoire programme de base et ses 20 registres mémoire convertibles automatiquement en 7 lignes de mémoire programme chacun, le calculateur peut disposer de 240 lignes de mémoire programme + 1 registre mémoire.
- Combinaison des codes des touches préfixes et des touches de fonctions permettant de gagner de la place dans chaque ligne de la mémoire programme.
- Dispositifs de mise au point facile des programmes.
- Branchements conditionnels et inconditionnels.
- Six niveaux de sous-programmes, quatre indicateurs binaires, douze labels réutilisables et facilement accessibles.

- Stockages direct et indirects, rappels, branchement et appel de sous-programmes.

- Opérations complexes de calcul de racines et d'intégration numérique avec les touches SOLVE et $\int_y^x$

Enfin, dernier avantage : le HP-34C, calculateur portable, peut fonctionner n'importe où grâce à ses batteries rechargeables.

Avant de lire ce manuel, si vous n'avez jamais utilisé de calculateur Hewlett-Packard et si la notation polonaise inverse ne vous est pas encore familière, vous pouvez consulter « La solution de vos problèmes avec votre calculateur Hewlett-Packard », brochure qui pourra être utile même aux initiés qui ne manqueront pas d'y découvrir de nouvelles possibilités.

En étudiant votre calculateur sous tous ses aspects, vous allez voir à quel point il est facile à utiliser, tant en mode d'utilisation manuelle qu'en mode automatique, sous contrôle de programmes.

SOLUTION MANUELLE DES PROBLÈMES

Pour commencer, il faut que vous sachiez résoudre un problème manuellement. Si vous avez des doutes à ce sujet, consultez la section « Mise en route » de « La solution de vos problèmes avec le calculateur Hewlett-Packard ».

Positionnez l'interrupteur du calculateur sur ON et le commutateur PRGM-RUN sur RUN. Appuyez sur f FIX 4 pour que votre format d'affichage soit conforme au format utilisé dans le reste de ce manuel.

Pour mettre en évidence le rapport étroit qui existe entre la solution manuelle d'un problème et la solution programmée, nous allons d'abord utiliser la solution manuelle pour résoudre un problème puis appliquer un programme au même problème en faisant varier les paramètres.

On calcule la surface d'une sphère d'après la formule $A = \pi d^2$.

A étant la surface de la sphère.
d le diamètre de la sphère et
π la valeur de pi (3,141592654).

Exemple : Ganymède, un des 12 satellites de Jupiter, a un diamètre de 3200 km. Avec votre calculateur, vous pouvez calculer manuellement la

Le format d'affichage utilisé dans ce manuel est FIX 4, sauf indication contraire.

surface de Ganymède. Il suffit d'appuyer sur la série de touches suivantes, après avoir positionné le commutateur PRGM-RUN sur RUN.

Appuyez sur	Affichage	
3 200	**3.200,**	Diamètre de Ganymède
g x^2	**10.240.000,00**	Carré du diamètre
h π	**3,1416**	Valeur de π.
×	**32.169.908,78**	Surface de Ganymède en km².

SOLUTION PROGRAMMÉE

Supposons maintenant que vous désiriez connaître la surface de chaque satellite, vous pourriez répéter douze fois le calcul précédent en faisant varier le diamètre (d). Mais il existe une méthode plus simple et plus rapide, qui vous évitera d'appuyer chaque fois sur les mêmes touches : vous écrivez un programme qui calculera la surface de n'importe quelle sphère à partir de son diamètre.

Pour calculer la surface d'une sphère à l'aide d'un programme, vous devez commencer par écrire ce programme, puis l'enregistrer dans le calculateur et enfin l'exécuter pour chaque calcul.

Rédaction du programme

Votre programme est déjà écrit. En effet, il n'est autre que les codes entrés par la succession de touches sur lesquelles vous avez appuyé pour résoudre le problème manuellement. A ces codes, il suffit d'ajouter un label et un code de retour pour marquer le début et la fin du programme.

Chargement du programme

Pour charger le programme dans le calculateur :

1. Mettez le commutateur PRGM-RUN sur PRGM (Programme).
2. Appuyez sur [f] CLEAR [PRGM] pour effacer la mémoire programme.
3. Appuyez en séquence sur les touches suivantes (les informations affichées sur l'écran lors du chargement d'un programme pourront vous être utiles plus tard, mais pour le moment, vous pouvez les ignorer).

Appuyez sur	
[h] [LBL] [A]	Marque le début du programme
[g] [x^2]	
[h] [π]	Mêmes touches qu'en mode manuel
[×]	
[h] [RTN]	Indique la fin du programme.

Le calculateur garde en mémoire cette séquence de pressions de touches.

Exécution du programme

Pour calculer la surface d'une sphère à partir de son diamètre à l'aide du programme :

1. Mettez le commutateur PRGM-RUN sur RUN.
2. Entrez la valeur du diamètre.
3. Appuyez sur [A] pour exécuter le programme.

Quand vous appuyez sur [A], le calculateur exécute automatiquement la séquence de pressions de touches qu'il a mémorisée et vous donne la même réponse qu'en mode manuel.

Reprenons l'exemple de Ganymède. Calculons sa surface sachant que son diamètre est égal à 3 200 km :

Appuyez sur	Affichage	
3 200	**3.200,**	
[A]	**32.169.908,78**	km²

Avec votre programme, vous pouvez calculer la surface des 12 satellites de Jupiter (ou d'une sphère quelconque), connaissant leur diamètre. Laissez le calculateur en mode RUN et entrez le diamètre de chaque sphère dont vous voulez connaître la surface, puis appuyez sur [A].

Calculons, par exemple, la surface d'Io dont le diamètre est de 2310 km :

Appuyez sur	Affichage	
2310 [A]	**16.763.852,56**	km²

Répétons le même calcul pour Europe qui a 1950 km de diamètre et pour Callisto, qui en a 3220.

Appuyez sur	Affichage	
1950 [A]	**11.945.906,07**	Surface d'Europe en km carrés
3220 [A]	**32.573.289,27**	Surface de Callisto en km carrés.

La programmation n'est pas plus difficile que ça! Le calculateur garde en mémoire une séquence de pressions de touches qu'il exécute ensuite à la demande. En fait, le HP-34C peut mémoriser jusqu'à 210 instructions différentes, correspondant souvent à deux ou trois pressions de touches.

LA MÉMOIRE PERMANENTE

Le HP-34C est doté d'une mémoire permanente — une technologie d'avant-garde dans le domaine des calculateurs de poche. Mémoire permanente signifie que si vous éteignez le calculateur, la mémoire programme et les 21 registres mémoire ne sont pas effacés et le mode d'affichage sélectionné reste en vigueur. Vous pouvez ainsi stocker un ou plusieurs programmes pendant des jours et des semaines.

La mémoire permanente vous permet donc, en particulier, de conserver les données, d'économiser les batteries et de personnaliser votre calculateur (dans le cas, par exemple, où vous utilisez 20 % de vos programmes pour résoudre 80 % de vos problèmes). Le gain de temps est considérable puisque vous n'avez plus besoin de réintroduire les programmes courants : ils sont stockés dans le calculateur. Réduisant le nombre de pressions de touches à effectuer, la mémoire permanente diminue considérablement les risques d'erreurs humaines au moment de l'entrée des données.

Mais peut-être l'avantage le plus important de la mémoire permanente est-il qu'elle vous permet de modifier ou de personnaliser votre calculateur. La manière la plus simple de personnaliser votre calculateur est la suivante : faites une liste des problèmes que vous rencontrez le plus fréquemment, classez-les par ordre de priorité, rédigez et conservez les programmes qui vous permettent de résoudre ces problèmes. Si vous rencontrez un ensemble de problèmes répétitifs, il vous suffit d'écrire un programme que vous réutiliserez chaque fois. Vous pouvez même conserver un ou deux programmes importants dans le calculateur.

La mémoire permanente vous permet également de conserver des données dans les 21 registres mémoire suivant l'affectation mémoire programme/registres mémoire actuelle. Constantes, sommations et résultats intermédiaires peuvent être rappelés chaque fois que vous en avez besoin. Et comme le mode d'affichage est également stocké en mémoire permanente, si vous éteignez votre HP-34C et que vous le rallumez ensuite, vous le retrouverez dans le mode dans lequel il était : FIX, SCI, ou ENG.

La mémoire permanente, enfin, prolonge la vie de vos batteries : elle conserve vos programmes pendant 1 mois et même davantage si vous n'utilisez pas votre calculateur. Et si vous l'utilisez, ayant moins de programmes à introduire, vos batteries s'useront moins vite.

Chapitre 2

CARACTÉRISTIQUES DU HP-34C

Les caractéristiques générales du HP-34C sont présentées dans le manuel « La solution de vos problèmes avec votre calculateur Hewlett-Packard ». Nous nous limiterons ici aux caractéristiques spéciales ou « nouveautés » du HP-34C.

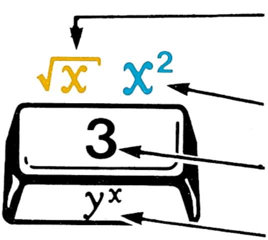
ORGANISATION DU CLAVIER

La plupart des touches du clavier possèdent trois ou quatre fonctions : la première fonction est indiquée par le symbole figurant sur le plat de la touche, la deuxième est inscrite en noir sur la face inclinée de la touche, la troisième et la quatrième sont inscrites respectivement en jaune et en bleu au-dessus de la touche.

Pour exécuter la fonction inscrite en noir sur la face de la touche, appuyez sur la touche préfixe noire [h], puis sur la touche correspondant à la fonction que vous voulez exécuter.

Pour exécuter la fonction inscrite en jaune au-dessus de la touche, appuyez sur la touche préfixe jaune [f], puis sur la touche correspondant à la fonction que vous voulez exécuter.

Pour exécuter la fonction inscrite en bleu au-dessus de la touche, appuyez sur la touche préfixe bleue [g], puis sur la touche correspondant à la fonction que vous voulez exécuter.



Pour exécuter cette fonction, appuyez sur [f], puis sur [3].

Pour exécuter cette fonction, appuyez sur [g], puis sur [3].

Pour exécuter cette fonction, appuyez uniquement sur 3.

Pour exécuter cette fonction, appuyez sur [h], puis sur [3].

Notez que sur toutes les touches à quatre fonctions, à l'exception de ENTER l'inscription jaune se trouve en haut à gauche de la touche et l'inscription bleue en haut à droite.

REGISTRE DE STOCKAGE ET MÉMOIRE PROGRAMME

En plus des quatre registres de la pile opérationnelle et du registre LAST X, le HP-34C est doté d'une mémoire de programme et de données partagée, commandée automatiquement.

Cette mémoire est répartie en 70 lignes de mémoire programme, 20 registres mémoire de données et un registre I. Les registres mémoire peuvent être automatiquement convertis en 7 lignes de mémoire programme chacun, en cas de besoin. Pour connaître l'affectation actuelle de la mémoire programme et des registres mémoire, il suffit d'appuyer sur g MEM*. Nous reviendrons sur ce sujet important au chapitre « Programmation ».

TOUCHES DE MODIFICATION DES NOMBRES

En plus de la touche CHS, le HP-34-C possède trois touches qui modifient les nombres : ABS, INT, et FRAC.

Valeur absolue

Dans certains calculs, il est intéressant de connaître la valeur absolue (ou module) d'un nombre. Pour afficher la valeur absolue d'un nombre placé dans le registre X, appuyez d'abord sur la touche h, puis sur la touche ABS (valeur absolue).

Pour calculer, par exemple, la valeur absolue de − 3* :

Appuyez sur	Affichage
3 CHS	**− 3,**
h ABS	**3,0000**

Partie entière d'un nombre

Pour extraire et afficher la partie entière d'un nombre, appuyez sur la touche préfixe h, suivie de la touche INT (integer = entier).

* Le registre I peut aussi servir de registre mémoire. Mais comme il joue un rôle spécial et qu'il ne peut être converti en lignes de mémoire programme, il n'apparaît pas parmi les registres mémoire et les lignes de mémoire programme appelées sur l'écran par les touches g MEM.

Pour n'afficher, par exemple, que la partie entière du nombre 111.222 :

Appuyez sur	Affichage	
111.222	**111,222**	
h INT	**111,0000**	Il ne reste que la partie entière
333.444 CHS	– **333,444**	
h INT	– **333,0000**	Ici aussi, il ne reste que la partie entière

Si l'on appuie sur h INT, la partie fractionnaire du nombre est remplacée par zéro. Le signe reste inchangé. Le nombre original est conservé dans le registre LAST X.

h LST x	– **333,4440**	Nombre d'origine

Partie fractionnaire d'un nombre

Pour extraire et n'afficher que la partie fractionnaire d'un nombre, appuyez sur la touche préfixe h, suivie de la touche FRAC (fraction).

Pour n'afficher, par exemple, que la partie fractionnaire de 555,666* :

Appuyez sur	Affichage	
555.666	**555,666**	
h FRAC	**0,6660**	Il ne reste que la partie fractionnaire du nombre
777.888	– **777,888**	
h FRAC	– **0,8880**	Ici aussi, il ne reste que la partie fractionnaire.

Si l'on appuie sur h FRAC, la partie entière du nombre est remplacée par zéro. Le signe reste inchangé. Le nombre entier est conservé dans le registre LAST X.

h LST x	– **777,8880**

FONCTIONS MATHÉMATIQUES

Factorielle

Si le nombre présent dans le registre X est un entier non négatif n, vous pouvez obtenir la factorielle de n : n!, en appuyant sur la touche x!, n! étant égal au produit des entiers compris entre 1 et n.

Cette fonction permet de résoudre rapidement et facilement les permutations et les combinaisons.

Exemple :

Le fabricant de « gadgets », Monsieur X, désire avoir une photo de ses produits pour son service publicité. De combien de manières la photographe peut-elle disposer ses huit sortes de gadgets.

Solution :

La solution est donnée par le calcul suivant :
$8! = 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$

Appuyez sur	**Affichage**	
8 [h] [x!]	**40.320,0000**	La photographe peut disposer ses gadgets de 40320 manières

Exemple :

Regardant à travers son objectif, la photographe s'aperçoit que pour saisir tous les détails de ses sujets avec sa caméra, elle doit se limiter à cinq gadgets. Combien d'ensembles de cinq gadgets peut-elle choisir parmi les huit ?

Solution

Le nombre d'ensembles est donné par la formule suivante :

$$\frac{8!}{(8-5)!\ 5!}$$

Appuyez sur	**Affichage**	
8 [h] [x!]	**40.320,0000**	
8 [ENTER↑]	**8,0000**	
5 [−]	**3,0000**	
[h] [x!]	**6,0000**	
5 [h] [x!]	**120,0000**	
[×]	**720,0000**	
[÷]	**56,0000**	Le photographe peut choisir 56 différents ensembles de gadgets.

Fonction gamma

La touche [x!] peut aussi servir à calculer la fonction Gamma, $\Gamma(x)$, utilisée dans certains problèmes de mathématiques et de statistiques. En appuyant sur [x!], on obtient $\Gamma(x + 1)$. Pour calculer la fonction Gamma d'un nombre quelconque, vous devez donc soustraire 1 du nombre puis, après avoir rangé le résultat dans le registre X, appuyer sur [h] [x!].

Exemple : Soit à calculer $\Gamma(2.7)$

Appuyez sur	Affichage	
2.7	**2,7**	Introduction du nombre
[ENTER↑] 1 [−]	**1,7000**	Soustraction de 1
[h] [x!]	**1,5447**	$\Gamma(2.7)$

Exemple

Soit à calculer $\Gamma(-2.7)$

Appuyez sur	Affichage	
2.7 [CHS]	− **2,7**	INtroduction du nombre
[ENTER↑] 1 [−]	− **3,7000**	Soustraction de 1
[h] [x!]	− **0,9311**	$\Gamma(-2.7)$

Comme $\Gamma(x)$ n'est pas défini quand x est un entier négatif ou nul, la valeur $\Gamma(x + 1)$ obtenue par [x!] n'est pas définie si x est un entier négatif. Si x s'approche de ces valeurs, la valeur de $\Gamma(x + 1)$ augmente jusqu'à l'infini. Comme votre calculateur HP-34C ne peut pas calculer au-delà de $9{,}999999999 \times 10^{99}$, il affiche un dépassement de capacité − 9,99999999 si vous appuyez sur [h] [x!] quand le registre X contient un entier négatif. Bien que lorsque x se rapproche d'un entier négatif, $\Gamma(x + 1)$ puisse être négative ou positive suivant la valeur de x, le calculateur ajoute toujours un signe moins au dépassement de capacité affiché si x est un entier négatif, ceci pour faire la différence avec le dépassement de capacité 9,99999999 affiché lorsque x prend des valeurs positives très grandes et que $\Gamma(x)$ croît à l'infini en restant toujours positive.

---

* La touche [x!] peut être utilisée aussi bien pour la fonction factorielle que pour la fonction Gamma, car lorsque x est un entier non négatif n, $\Gamma(x + 1) = \Gamma(n + 1) = n!$ La fonction Gamma peut être considérée comme une généralisation de la fonction factorielle, puisque le nombre stocké dans le registre X n'est pas obligatoirement un entier non négatif. D'autre part, la fonction factorielle peut être considérée comme un cas spécial de la fonction Gamma.

Différence en pourcentage

La touche [Δ%] permet de calculer le rapport différentiel — c'est-à-dire la différence en pourcentage dans le sens positif ou négatif entre deux nombres. Pour calculer un rapport différentiel.

1. Entrez le nombre de base (en principe, le premier chronologiquement).
2. Appuyez sur [ENTER↑]
3. Entrez le second nombre.
4. Appuyez sur [h] [Δ%]

La formule utilisée est : $\Delta\% = \dfrac{100(x-y)}{y}$

Si l'on utilise la séquence de touches précédente, un résultat positif indique une augmentation, et un résultat négatif une diminution.

Exemple :

La collection de monnaies de Silas Silversaver a été estimée à 475 dollars en 1974. En 1979, elle a été estimée à 735 dollars. Quel est le pourcentage d'augmentation de la collection de 1974 à 1979 ?

Appuyez sur	Affichage	
[f] [FIX] 4		Affichage en notation [FIX] 4
475 [ENTER↑]	**475,0000**	
735 [h] [Δ%]	**54,7368**	Pourcentage d'augmentation

FONCTIONS STATISTIQUES

Sommations

La touche [Σ+] effectue certains calculs importants de sommations et de produits à partir des valeurs introduites dans les registres X et Y. Les résultats sont automatiquement accumulés dans les registres mémoire R_0 à R_5. Avant d'entreprendre des calculs de sommation avec un nouvel ensemble de valeurs x et y, appuyez sur les touches [f] CLEAR [Σ] pour effacer le contenu des registres utilisés dans ces calculs. Puis, pour chaque paire de valeurs x et y données :

1. Introduisez la valeur y dans le registre X.
2. Appuyez sur ENTER↑ pour faire monter la valeur y dans le registre Y.
3. Introduisez la valeur x dans le registre X.
4. Appuyez sur f Σ+.

Si votre problème ne porte que sur une seule variable (x) au lieu de deux (x et y), la procédure est la même : commencez par effacer le contenu des registres mémoire de statistiques R_0 à R_5, puis, si le contenu du registre Y n'est pas nul, effacez-le également. (Si le registre Y contient un nombre non nul lors d'un calcul de s, r, L.R ou $\hat{y}$ avec une seule variable, une Error 3 risque d'apparaître). En appuyant sur f CLEAR REG vous effacerez non seulement les registres R_0 à R_5, mais aussi les registres R_6 à R_9, $R_{.0}$ à $R_{.9}$ et I. Par conséquent, si ces derniers contiennent des nombres que vous désirez conserver, appuyez sur les touches f CLEAR Σ au lieu de f CLEAR REG. Après avoir effacé les registres, procédez de la manière suivante pour chaque valeur de y donnée :

1. Introduisez le nombre dans le registre X.
2. Appuyez sur les touches f Σ+.

La séquence de touches f Σ+ déclenche la série d'opérations suivantes :

1. Addition du nombre introduit dans le registre X au contenu du registre mémoire R_1.
2. Addition du carré du nombre introduit dans le registre X au contenu du registre mémoire R_2.
3. Addition du nombre introduit dans le registre Y au contenu du registre mémoire R_3.
4. Addition du carré du nombre introduit dans le registre Y au contenu du registre mémoire R_4.
5. Multiplication du nombre introduit dans le registre Y par le nombre introduit dans le registre X et addition du produit au contenu du registre mémoire R_5.
6. Addition du nombre 1 au contenu du registre mémoire R_0 et copie du résultat — le nombre de paires de données (x, y) accumulées précédemment dans le registre X.

La séquence de touches f Σ+ a également pour effet de transférer dans le registre LAST X le nombre introduit dans le registre X. Celui du registre Y ne bouge pas.

En résumé, les sommations sont stockées dans les registres suivants de votre calculateur :

Registre	Contenu
R_0 R_1 R_2 R_3 R_4 R_5	n : nombre de paires de données accumulées Σx : sommation des valeurs de x Σx^2 : sommation des carrés des valeurs de x Σy : sommation des valeurs de y Σy^2 : sommation des carrés des valeurs de y Σxy : sommation des produits des valeurs de x et de y.

Certains ensembles de données comprennent des valeurs (x ou y) qui s'écartent relativement peu d'un certain nombre. Pour rendre les calculs statistiques plus précis, on peut alors n'introduire dans le calculateur que les différences entre chaque valeur et un nombre se rapprochant de la moyenne de ces valeurs. Dans ce cas, ce nombre doit être additionné au résultat du calcul de $\bar{x}$, $\hat{y}$ ou de l'ordonnée à l'origine de la droite de régression. Supposons, par exemple, que vous ayez pour x les valeurs 665999, 666000 et 666001. Vous allez entre − 1, 0 et 1. Si vous calculez ensuite x, ajoutez 666000 à la réponse. Si les valeurs données sont trop rapprochées, le calculateur ne peut pas toujours calculer s, r, L.R. ou $\hat{y}$ et, dans ce cas, affiche Error 3. Cette erreur n'apparaît pas si vous normalisez les données comme nous venons de le voir.

Remarque

Contrairement aux opérations arithmétiques effectuées dans les registres mémoire, les opérations [Σ+] et [Σ−] continuent normalement même si un dépassement de capacité (c'est-à-dire une valeur supérieure à $9{,}999999999 \times 10^{99}$) se produit dans l'un des registres mémoire R_0 à R_5. Aucune erreur (Error 1) n'est signalée.

Si vous avez besoin d'un cumul, vous pouvez sortir dans le registre X le contenu du registre mémoire désiré en appuyant sur la touche [RCL] suivie du numéro du registre correspondant. Si juste avant cette pression de touche, vous avez appuyé sur les touches [f] [Σ+] (ou [g] [Σ−]), le cumul est écrit au-dessus du nombre de paires de données (n) affiché.

Si vous voulez sortir les deux cumuls, Σx et Σy, appuyez sur [RCL] [f] [Σ+]. Σx est alors copié du registre R_1 dans le registre affiché X et Σy est copié du registre R_3 dans le registre Y. Si, juste avant ces pressions de touches, vous avez appuyé sur les touches [f] [Σ+], [g] [Σ−], [CLx], ou [ENTER↑], le contenu du registre Y monte d'abord dans le registre Z. Sinon, le contenu des deux registres, X et Y, monte d'abord dans les registres Z et T, respectivement.

Exemple : Calculez Σx, Σx^2, Σy, Σy^2 et Σxy pour les valeurs de x et y données ci-dessous :

y	7	5	9
x	5	3	8

Appuyez sur	**Affichage**	
f CLEAR Σ	**0,0000**	Effacement des registres mémoire. L'affichage montré suppose qu'il ne reste pas de résultats des calculs précédents.
7 ENTER↑	**7,000**	
5 f Σ+	**1,0000**	Cumul de la première paire n = 1
5 ENTER↑	**5,0000**	
3 f Σ+	**2,0000**	Cumul de la deuxième paire n = 2
9 ENTER↑	**9,0000**	
8 f Σ+	**3,0000**	Cumul de la troisième paire n = 3
RCL 1	**16,0000**	Somme des valeurs x du registre R_1.
RCL 2	**98,0000**	Somme des carrés des valeurs x du registre R_2.
RCL 3	**21,0000**	Somme des valeurs y du registre R_3.
RCL 4	**155,0000**	Somme des carrés des valeurs y du registre R_4.
RCL 5	**122,0000**	Somme des produits des valeurs x et y du registre R_5.
RCL 0	**3,0000**	Nombre de données (n = 3) du registre R_0.

Suppression et correction des données

En cas de fausse manœuvre, si vous n'avez pas encore appuyé sur les touches f Σ+, appuyez sur la touche CLx et introduisez la donnée correcte.

Si vous voulez changer une donnée ou si, après avoir appuyé sur f Σ+, vous vous rendez compte que vous avez introduit une donnée incorrecte, vous pouvez corriger l'erreur de sommation avec la touche Σ− :

1. Introduisez la paire de données incorrecte dans les registres X et Y (vous pouvez utiliser LST x pour sortir une seule donnée incorrecte dans le registre X).
2. Appuyez sur les touches g Σ− pour supprimer la donnée incorrecte.

3. Introduisez les valeurs correctes de x et de y. Si vous avez fait une erreur dans l'une des valeurs d'une paire de données (x, y), vous devez supprimer et rentrer les deux valeurs.

4. Appuyez sur les touches [f] [Σ+]

Supposons que dans l'exemple précédent la dernière paire de données (8,9) ait dû être (8,6),vous pouvez corriger l'erreur de cumul de la manière suivante :

Appuyez sur	Affichage	
9 [ENTER↑]	**9,0000**	Vous entrez à nouveau la valeur incorrecte de y
8	**8,**	Vous entrez à nouveau la valeur correcte de x
[g] [Σ−]	**2,0000**	Vous avez maintenant deux données (n)
6 [ENTER↑]	**6,0000**	Vous entrez la valeur correcte de y
8	**8,**	Vous entrez à nouveau la valeur de X.
[f] [Σ+]	**3,0000**	Vous avez à nouveau trois données.

Remarque

Bien que les touches [g] [Σ−] permettent de supprimer une paire incorrecte (x, y), elles ne suppriment pas les erreurs d'arrondi qui peuvent se produire au moment de l'addition de la paire au contenu d'un des registres mémoire R_1 à R_5. Les résultats obtenus risquent donc de ne pas être les mêmes que ceux obtenus si l'on avait entré la paire incorrecte (x, y) avec les touches [f] [Σ+] pour la supprimer ensuite avec les touches [g] [Σ−]. Mais la différence sera peu sensible, sauf si la paire incorrecte (x, y) est beaucoup plus grande que la paire correcte ; dans ce dernier cas, il est préférable de recommencer et de rentrer à nouveau les données (en faisant attention cette fois).

Moyenne

La touche [$\bar{x}$] permet de calculer la moyenne arithmétique des valeurs x et y cumulées dans les registres R_1 et R_3, respectivement.

Si vous appuyez sur les touches [h] [$\bar{x}$] :

1. le contenu des registres de la pile opérationnelle monte comme si vous aviez appuyé sur les touches [RCL] [f] [Σ+].

2. La moyenne des valeurs de x ($\bar{x}$) est calculée à partir des données cumulées dans les registres R_1 (Σx) et R_0 (n) suivant la formule :

$$\bar{x} = \frac{\Sigma x}{n}$$

La valeur obtenue pour $\bar{x}$ est affichée dans le registre X.

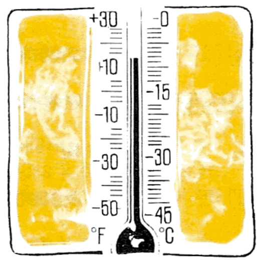
3. La moyenne des valeurs de y ($\bar{y}$) est calculée à partir des données cumulées dans les registres R_3 (Σy) et R_0 (n) suivant la formule :

$$\bar{y} = \frac{\Sigma y}{n}$$

La valeur obtenue pour $\bar{y}$ est stockée dans le registre Y de la pile.

Exemple

Voici un tableau des températures maximale et minimale des 7 jours d'une semaine hivernale à Fairbanks, en Alaska. Quelles sont les moyennes des températures maximales et minimales pour la semaine choisie ?



	dimanche	lundi	mardi	mercredi	jeudi	vendredi	samedi
Max.	6	11	14	12	5	− 2	− 9
Min.	− 22	− 17	− 15	− 9	− 24	− 29	− 35

Appuyez sur	Affichage	
f CLEAR Σ	**0,0000**	Effacement des registres mémoire. L'affichage montré suppose qu'il ne reste pas de résultats des calculs précédents.
6 ENTER↑ 22	**22,**	
CHS f Σ+	**1,0000**	Il y a maintenant 1 paire de données (n)
11 ENTER↑ 17	**17,**	
CHS f Σ+	**2,0000**	Il y a maintenant 2 paires de données (n)
14 ENTER↑ 15	**15,**	
CHS f Σ+	**3,0000**	

12 ENTER↑ 9	9,	
CHS f Σ+	4,0000	
5 ENTER↑ 24	24,	
CHS f Σ+	5,0000	
2 CHS ENTER↑	– 2,0000	
29 CHS f Σ+	6,0000	
9 CHS ENTER↑	– 9,0000	
35 CHS f Σ+	7,0000	Il y a maintenant 7 paires de données (n)
h $\bar{x}$	– 21,5714	Moyenne des températures minimales
x⇄y	5,2857	Moyenne des températures maximales

Écart type

Les touches h s permettent de calculer l'écart type (mesure de la dispersion par rapport à la moyenne) des données accumulées. Le HP-34C calcule les écarts types des valeurs x et y selon les formules s_x et s_y suivantes :

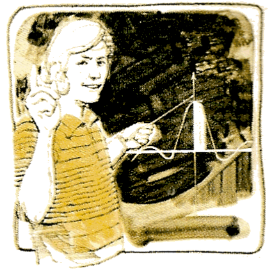
$$s_x = \sqrt{\frac{n\Sigma x^2 - (\Sigma x)^2}{n(n-1)}} \qquad s_y = \sqrt{\frac{n\Sigma y^2 - (\Sigma y)^2}{n(n-1)}}$$

Ces formules donnent les meilleures estimations des écarts types de la population par rapport à l'échantillon. Par convention, l'écart type donné par ces formules est appelé « écart type de l'échantillon ».

Si vous appuyez sur les touches h s :

1. Le contenu des registres de la pile opérationnelle monte comme sous l'action des touches RCL f Σ+.
2. L'écart type des valeurs de x (s_x) est calculé à partir des données accumulées dans les registres R_2 (Σx^2), R_1 (Σx) et R_0(n) selon la formule indiquée plus haut. La valeur obtenue par s_x est affichée dans le registre X.
3. L'écart type des valeurs de y (s_y) est calculé à partir des données accumulées dans les registres R_4 (Σy^2), R_3 (Σy) et R_0 (n) selon la formule indiquée plus haut. La valeur obtenue pour s_y est stockée dans le registre Y.

Exemple



M. Broidesnombres, jeune professeur de mathématiques à l'université de Mammouth, a mis au point un nouveau test pour mesurer les aptitudes des étudiants aux mathématiques. Pour évaluer sa qualité, il fait subir le test à 746 étudiants. Fatigué de corriger ses copies, M. Broidesnombres décide de choisir au hasard 8 parmi les 746 tests et d'estimer l'écart type de toutes les notes à partir de l'échantillon. Celui-ci se compose des notes suivantes : 79, 94, 68, 86, 82, 78, 83 et 89. Quel est l'écart type obtenu ?

Appuyez sur	Affichage	
CLx ENTER↑	**0,0000**	Effacement du registre affiché X et du registre Y.
f CLEAR Σ	**0,0000**	Effacement des registres mémoire.
79 f Σ+	**1,0000**	Introduction de la première note. Le calcul ne portant que sur une seule variable, il est inutile d'introduire une variable y dans le registre Y avec la touche ENTER↑
94 f Σ+	**2,0000**	
68 f Σ+	**3,0000**	
86 f Σ+	**4,0000**	
82 f Σ+	**5,0000**	
78 f Σ+	**6,0000**	
83 f Σ+	**7,0000**	
89 f Σ+	**8,0000**	Dernière note de l'échantillon.
h s	**7,8365**	Écart type estimé à partir de l'échantillon pour 746 étudiants.

Si vos données ne constituent pas seulement un échantillon d'une population mais représentent **toute** la population, l'écart type des données est l'écart type **vrai** de la population (σ). La formule utilisée pour le calcul de l'écart type **vrai** de la population diffère de celle de la fonction s d'un facteur de $[(n-1)/n]^{1/2}$. Cette différence est minime et peut être ignorée dans la plupart des cas. Mais si vous voulez calculer la valeur exacte de l'écart type de la population pour une population entière, il suffit de quelques pressions de touches sur votre HP-34C : appuyez sur f Σ+ pour additionner la moyenne ($\bar{x}$) des données aux données, puis sur h s. Vous obtiendrez l'écart type vrai de la population basé sur les données originales.

Exemple

Supposons que dans l'exemple précédent les données représentaient toutes les notes de l'examen final du cours de M. Broidesnombres sur les fonctions transcendantes. Comme c'est la première fois que M. Broidesnombres a donné ce cours, il veut juger de la qualité de ses examens en calculant l'écart type des notes attribuées aux étudiants. Il prend son calculateur, introduit les données, puis procède de la manière suivante :

Appuyez sur	Affichage	
h $\bar{x}$	**82,3750**	Moyenne des notes
f Σ+	**9,0000**	Addition de la moyenne aux données. Il y a neuf données au total.
h s	**7,3304**	Écart type de toutes les notes de l'examen final.

Régression linéaire

La régression linéaire est une méthode statistique consistant à chercher la droite qui ajuste le mieux une série de points et à montrer ainsi une relation entre deux variables. Après totalisation d'un groupe de points dans les registres R_0 à R_5, vous pouvez calculer les coefficients de l'équation linéaire $y = Ax + B$ à l'aide de la méthode des moindres carrés en appuyant sur h L.R. (il faut avoir introduit au moins deux points dans le calculateur, avant de pouvoir les ajuster par une droite de moindres carrés).

Pour utiliser la fonction régression linéaire sur votre HP-34C, entrez d'abord une série de points avec les touches f Σ+, puis appuyez sur h L.R.

Quand vous appuyez sur h L.R. :

1. Le contenu des registres de la pile monte comme sous l'action des touches RCL f Σ+.
2. La pente (A) de la droite des moindres carrés est calculée d'après l'équation :

$$A = \frac{n\Sigma xy - \Sigma x\Sigma y}{n\Sigma x^2 - (\Sigma x)^2}$$

   Elle est ensuite stockée dans le registre Y de la pile.

3. L'ordonnée à l'origine de la droite des moindres carrés est calculée d'après l'équation :

$$B = \frac{\Sigma y\Sigma x^2 - \Sigma x\Sigma xy}{n\Sigma x^2 - (\Sigma x)^2}$$

Elle apparaît à l'affichage

Pour afficher la valeur de A, il suffit d'appuyer sur la touche [$x \gtrless y$] qui permute le contenu des registres X et Y de la pile opérationnelle.

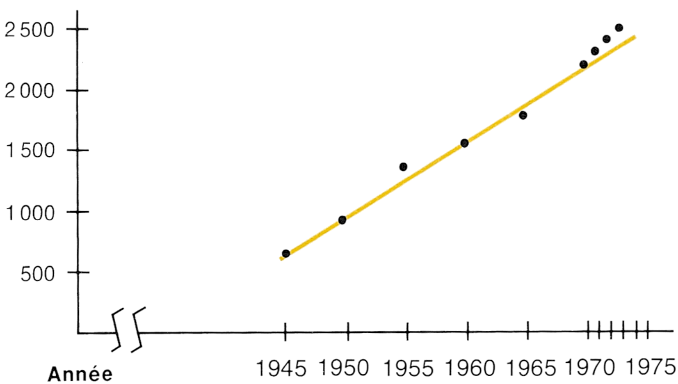
Exemple

M. Gusher, P.D.G. de la Compagnie pétrolière Gusher, veut connaître la pente et l'ordonnée à l'origine d'une droite de moindres carrés pour la consommation d'essence aux États-Unis depuis 1945. Il connaît les données du tableau suivant :

Demande en essence (millions de barils)	696	994	1 330	1 512	1 750	2 162	2 243	2 382	2 484
Année	1945	1950	1955	1960	1965	1970	1971	1972	1973

Solution

Gusher pourrait tracer la courbe suivante pour représenter la demande d'essence par rapport au temps :



Mais avec son calculateur, il lui suffit d'entrer ses données avec la touche [Σ+], puis d'appuyer sur les touches [h] [L.R.].

Appuyez sur	Affichage	
[f] [CLEAR] [Σ]	0,0000	Effacement des registres mémoire. L'affichage implique qu'il ne reste pas de résultats de calculs précédents.
696 [ENTER↑]	696,0000	
1945 [f] [Σ+]	1,0000	
994 [ENTER↑]	994,0000	
1950 [f] [Σ+]	2,0000	
1330 [ENTER↑]	1.330,0000	
1955 [f] [Σ+]	3,0000	
1512 [ENTER↑]	1.512,0000	
1960 [f] [Σ+]	4,0000	
1750 [ENTER↑]	1.750,0000	
1965 [f] [Σ+]	5,0000	
2162 [ENTER↑]	2.162,000	
1970 [f] [Σ+]	6,0000	
2243 [ENTER↑]	2.243,0000	
1971 [f] [Σ+]	7,0000	
2382 [ENTER↑]	2.382,0000	
1972 [f] [Σ+]	8,0000	
2484 [ENTER↑]	2.484,0000	
1973 [f] [Σ+]	9,0000	Toutes les paires de données ont été introduites.
[h] [L.R.]	–118.290,6295	Ordonnée à l'origine de la droite
[$x \rightleftarrows y$]	61,1612	Pente de la droite.

Estimation linéaire

Si les registres R_0 à R_5 contiennent des cumuls, vous pouvez estimer la valeur de y ($\hat{y}$) en introduisant une nouvelle valeur de x et en appuyant sur les touches [h] [$\hat{y}$].

Exemple

Ayant conservé les données des calculs précédents dans les registres R_0 à R_5, Gusher veut prévoir la demande d'essence des années 1980 et 2000. Il introduit les nouvelles valeurs de x et appuie sur [h] [$\hat{y}$].

Appuyez sur	Affichage	
1980 [h] [$\hat{y}$]	2.808,6264	Prévision de la demande en millions de barils pour l'année 1980
2000 [h] [$\hat{y}$]	4.031,8512	Prévision de la demande en millions de barils pour l'année 2000.

Coefficient de corrélation

La régression linéaire et l'estimation linéaire supposent qu'il existe entre les deux variables x et y une relation telle que les points qui les représentent puissent être assimilés à une ligne droite. La touche [r] (coefficient de corrélation) permet de calculer à quel degré les données sont assimilables à une droite. Le coefficient de corrélation peut varier entre $r = +1$ et $r = -1$. Si $r = +1$ les données coïncident exactement avec une droite dont la pente est positive. Si $r = -1$, elles coïncident avec une droite dont la pente est négative. Si $r = 0$, les données ne peuvent pas être ajustées par une droite.

Pour calculer le coefficient de corrélation des données de l'exemple précédent :

Appuyez sur	Affichage	
[h] [r]	**0,9931**	Les données sont pratiquement assimilables à une droite.

OPÉRATIONS SUR LES VECTEURS

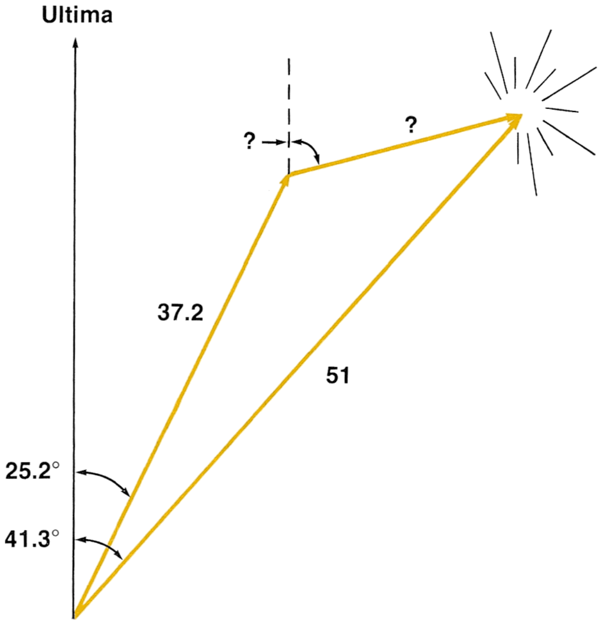
L'association des touches [Σ+] et [Σ−], d'une part, et [→R] et [→P], d'autre part, vous permet d'additionner ou de soustraire des vecteurs sur votre HP-34C.

Exemple

Le vaisseau spatial Félicité est sorti victorieux d'une bataille féroce contre le vaisseau spatial Thanatos en provenance de la planète ennemie Maldek. Mais son pilote automatique est cassé et son moteur principal bloqué à 37,2 méganewtons, dans un angle de 25,2° avec l'étoile Ultima. Consultant la carte des étoiles du vaisseau, le copilote constate la présence d'un vecteur entrée de l'hyperespace de 51 méganewtons à un angle de 41.3° d'Ultima. Sur quelle poussée et quel angle faut-il placer le moteur de secours pour aligner Félicité sur le vecteur entrée de l'hyperespace ?

Solution

Le vecteur poussée du moteur de secours est égal au vecteur entrée de l'hyperespace moins le vecteur poussée du moteur principal. On convertit les vecteurs en coordonnées rectangulaires à l'aide des touches [f] [→R] et l'on calcule leur différence à l'aide des touches [f] [Σ+] et [g] [Σ−]. On rappelle cette différence dans les registres X et Y avec les touches [RCL] [f] [Σ+]. Puis on convertit ces coordonnées rectangulaires du vecteur poussée du moteur de secours en coordonnées polaires à l'aide des touches [g] [→P] .



Appuyez sur	Affichage	
f CLEAR Σ	**0,0000**	Effacement de tous les registres mémoire (l'affichage suppose qu'il ne reste pas de résultats de calculs précédents).
g DEG		
41.3 ENTER↑	**41,3000**	Introduction de l'angle du vecteur entrée de l'hyperespace dans le registre Y.
51 f →R	**38,3145**	Introduction du module du vecteur entrée de l'hyperespace dans le registre X et conversion en coordonnées rectangulaires.
f Σ+	**1,0000**	Accumulation des coordonnées du vecteur entrée de l'hyperespace dans les registres R_1 et R_3.
25.2 ENTER↑	**25,2000**	Introduction de l'angle du vecteur poussée du moteur principal dans le registre Y.

37.2 [f] [→R]	**33,6596**	Introduction du module du vecteur poussée du moteur principal dans le registre X et conversion en coordonnées rectangulaires.
[g] [Σ−]	**0,0000**	Soustraction des coordonnées rectangulaires du vecteur poussée du moteur principal des coordonnées rectangulaires du vecteur entrée de l'hyperespace dans les registres R_1 et R_3 et transfert dans les registres X et Y.
[RCL] [f] [Σ+]	**4,6549**	Rappel dans les registres X et Y des coordonnées rectangulaires du vecteur poussée du moteur de secours introduites dans les registres R_1 et R_3.
[g] [→P]	**18,4190**	Conversion en coordonnées polaires. L'affichage montre le module en méganewtons du vecteur poussée du moteur de secours.
[x⇆y]	**75,3613**	Angle du vecteur poussée du moteur de secours.

Chapitre 3

NOTIONS DE PROGRAMMATION

QU'EST-CE QU'UN PROGRAMME ?

Un programme est une séquence de pressions de touches mise en mémoire dans le calculateur et exécutable indéfiniment par pression d'une seule touche. Le résultat obtenu est le même que si vous aviez réappuyé manuellement sur chaque touche de la séquence. La programmation du HP-34C ne nécessite aucune expérience dans ce domaine.

POURQUOI ÉCRIRE DES PROGRAMMES ?

Le programme sert à gagner du temps dans les calculs répétitifs. En effet, une fois que vous avez écrit la séquence de pressions de touches nécessaire à la résolution d'un problème et que vous l'avez enregistrée dans le calculateur, vous ne vous en occupez plus. Le calculateur reprendra cette séquence chaque fois que vous lui soumettrez le même problème. Après avoir vérifié que votre programme a été correctement enregistré, vous pouvez faire confiance à votre calculateur. Il exécutera plus fidèlement vos ordres que si vous preniez le risque de refaire toutes les manipulations vous-même. Il vous débarrassera de la routine et vous laissera libre de voguer dans de plus hautes sphères.

Avant de poursuivre, rappelons brièvement les avantages de la programmation du HP-34C.

- Un langage de programmation facile à comprendre.
- Douze labels réutilisables pour désigner les différents programmes et parties de programmes.
- Combinaison des différentes étapes du programme. Les instructions nécessitant la pression de plusieurs touches telles que f SIN ou STO + 1 n'occupent qu'une ligne de mémoire programme.
- Allocation mémoire automatique. Différentes combinaisons possibles entre 21 registres mémoire et 70 lignes de mémoire programme + 1 registre mémoire (le registre I) et 210 lignes de mémoire programme. La réaffectation de la mémoire s'effectue automatiquement à raison de 7 lignes de mémoire programme contre un registre mémoire.
- Possibilités de prises de décision pour les programmes plus compliqués.

- Facilité de correction et de modification des programmes.
- Six niveaux de sous-programmes et quatre indicateurs binaires simplifiant la programmation.
- Stockage indirect, rappel, branchement et appels de sous-programmes permettant le contrôle automatique des données, les décisions logiques et le contrôle du programme.
- Compteur/décompteur et contrôle de boucles.

Tous ces avantages vous offrent les moyens nécessaires pour attaquer avec confiance les problèmes les plus complexes.

TROIS MODES DE CALCUL

Votre calculateur HP-34C peut fonctionner en trois modes :
1. Mode RUN manuel
2. Mode PROGRAM
3. Mode RUN automatique

Mode RUN manuel

Les fonctions et opérations exposées dans la première partie de ce manuel et de « La solution de vos problèmes avec votre calculateur Hewlett-Packard » sont exécutées manuellement une à une. Ces fonctions, associées à la pile opérationnelle, rendent les calculs faciles.

Mode PROGRAM

En mode PROGRAM, les fonctions et opérations précédentes ne sont pas exécutées mais enregistrées dans une partie du calculateur appelée mémoire programme pour être exécutées ultérieurement. Pour passer en mode PROGRAM, il suffit de placer le commutateur PRGM-RUN sur PRGM. Toutes les opérations du clavier peuvent être enregistrées ainsi, sauf les suivantes :

[f] CLEAR [PRGM]	[f] CLEAR [PREFIX]	[g] [MEM]
[h] [SST]	[GTO] [•] nnn	[h] [MANT]
[h] [BST]	[h] [DEL]	[STO] [ENTER↑]

Ces dernières opérations, sauf [h] [MANT] et [STO] [ENTER↑] vous seront utiles pour entrer et éditer vos programmes.*

Mode RUN automatique

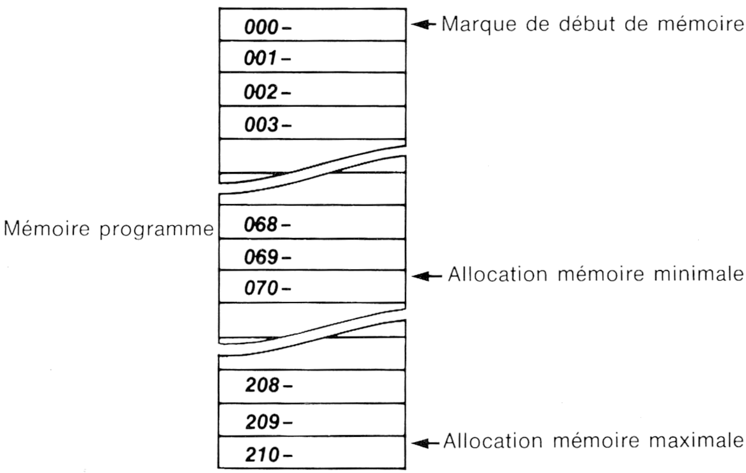
Vous avez vu qu'en mode PROGRAM le calculateur pouvait exécuter automatiquement une série d'opérations préalablement enregistrées en mémoire programme. En mode RUN automatique, ces opérations sont exécutées en séquence, à partir de la première, sur simple pression d'une touche. Toute la série d'opérations enregistrées est alors exécutée.

LA MÉMOIRE PROGRAMME

Vous avez vu dans le chapitre 1 qu'un programme était constitué d'une séquence de pressions de touches stockée dans la mémoire programme du calculateur et correspondant aux pressions de touches que vous devriez effectuer pour résoudre le problème manuellement. Si vous placez le commutateur PRGM-RUN sur PRGM, vous pouvez examiner ligne par ligne le contenu de la mémoire programme.

Appuyez sur [GTO] [•] 000 pour retourner au début de la mémoire programme. Si vous ne l'avez pas encore fait, placez le commutateur PRGM/RUN sur PRGM. Vous devriez avoir [000-] à l'affichage.

La mémoire programme comprend 70 à 210 lignes, ainsi qu'une marque de début de mémoire qui correspond au 000 que vous voyez sur l'écran. La mémoire programme est indépendante des registres de la pile opérationnelle, des registres LAST X et I et des registres mémoire disponibles.



*Les touches [h] [MANT] n'ont pas d'effet en mode PROGRAM, tandis que les touches [STO] [ENTER↑] déclenchent un contrôle automatique en mode PROGRAM et en mode RUN, effacent les registres de la pile et le registre LAST X et repositionnent le calculateur à la ligne 000 de la mémoire programme.

Lorsque le commutateur PRGM-RUN est sur la position PRGM, le nombre qui se trouve à gauche de l'écran indique le numéro de la ligne de mémoire programme sur laquelle est positionné le calculateur. Si vous appuyez sur [f] CLEAR [PRGM], [h] [LBL] [A], et sur les touches correspondant au début du programme de la surface de la lune (voir page 000), l'affichage devient :

Numéro de ligne **001 – 25,13,11**

Le nom de gauche, 001, indique que le calculateur se trouve maintenant sur la ligne 001 de la mémoire programme. Les autres nombres sont les codes des touches sur lesquelles vous avez appuyé et qui ont été stockés dans cette ligne de mémoire programme. Si vous appuyez sur [g] [x^2], votre affichage devient :

002 – 15 3

Le nombre de gauche, 002, indique que vous êtes maintenant à la deuxième ligne de la mémoire programme.

Chaque ligne de mémoire programme peut mémoriser une instruction, quel que soit le nombre de pressions de touches nécessaires à cette instruction. Une ligne de mémoire programme peut donc contenir une instruction telle que [CHS] qui correspond à une pression de touche ou une instruction du type [STO] [+] [6] (addition de la valeur affichée dans le registre X au contenu du registre 6) correspondant à trois pressions de touches.

Quel est le rapport entre les nombres affichés sur l'écran et les pressions de touches enregistrées à chaque instruction du programme ? Cette question nous mène à un nouveau chapitre de ce manuel : les codes des touches.

CODES DES TOUCHES

Reprenons les instructions du programme que nous venons d'introduire. appuyez sur [h] [BST]. Vous avez à l'affichage la première ligne du programme Surface de la lune :

Numéro de ligne ⟶ **001 – 25,13,11** ⟵ code de touche

Rappelons que le premier nombre ou code, 001, à gauche de l'affichage, désigne le numéro de ligne de la mémoire programme. Les deux chiffres suivants, 25, représentent la touche [h], 13, la touche [LBL] et 11, la touche [A]. Le premier de chaque paire de chiffres désigne la ligne du clavier où se trouve la touche. Le second indique la position de la touche dans cette ligne. Ainsi, 25 signifie que la touche se trouve en cinquième position dans la seconde ligne du calculateur. Il s'agit donc de la touche [h].

Chaque touche du clavier est donc représentée par un code deux chiffres, sauf pour les touches numériques 0 à 9. Celles-ci sont représentées en clair, comme le chiffre. Exemple : appuyez une fois sur [h] [SST]. La deuxième ligne du programme Surface de la lune, [g] [x^2], apparaît sur l'écran :

Numéro de ligne ⟶ 002 – 15 3 ⟵ code de touche

Nous savons que 002 est le numéro de la ligne de mémoire programme et 15 désigne la première ligne du clavier, touche cinq, c'est-à-dire la touche [g]. Comme la touche préfixe [g] est comprise dans cette séquence, le 3 désigne la fonction x^2 située sur la touche numérique [3]. En jargon informatique, [g] [x^2] est une « fonction secondaire » de la touche [3], tout comme l'astérisque est une fonction secondaire d'une touche de machine à écrire.

Les autres séquences de pressions de touches du programme Surface de la lune avec leur affichage respectif sont : (appuyez sur chacune de ces touches et vérifiez les codes affichés)

Appuyez sur	**Affichage**
[h] [π]	003– 25 73
[×]	004– 61
[h] [RTN]	005– 25 12

Ce programme, qui comprend 10 pressions de touches, n'occupe que cinq lignes de mémoire programme.

Problèmes

1. Quels sont les codes de touches des opérations suivantes ? [h] [1/x], [g] [GRD], [f] [→H.MS], [STO] [+] 1 ? (Réponses : 25 2, 15 13, 14 6, 23, 51, 1).

2. Combien de lignes de mémoire programme occuperaient les segments de programmes suivants ?
   a. 2 [ENTER↑] 3 [+].
   b. 10 [STO] 6 [RCL] [×].
   c. 100 [STO] 1 50 [STO] [×] 1 [RCL] 2 [g] [π] [×]
   (Réponses ; a. 4, b. 5, c. 10)

EFFACEMENT D'UN PROGRAMME

La mémoire permanente du calculateur a la propriété de sauvegarder les programmes en mémoire programme même si le calculateur est éteint. Pour effacer la mémoire programme, allumez le calculateur, placez le commutateur PRGM-RUN sur PRGM, et appuyez sur [f] CLEAR

[PRGM] Toutes les lignes de la mémoire programme occupées par un programme sont alors effacées et prêtes à recevoir de nouvelles instructions. Si le programme effacé occupait plus de 70 lignes de mémoire programme, celles qui dépassent les 70 lignes sont automatiquement réaffectées à des registres mémoire. Notez que si vous appuyez sur [f] CLEAR [PRGM] en mode RUN, le calculateur se repositionne sur la ligne 000 mais la mémoire programme n'est pas effacée.

Si le calculateur ne reçoit plus de courant (panne de batteries), toutes les instructions présentes en mémoire programme et toutes les données stockées dans les registres-mémoire peuvent être perdues. A la remise sous tension Error Pr apparaît à l'écran.

RÉDACTION D'UN PROGRAMME

Au début de ce manuel, vous avez écrit un programme qui vous permettait de calculer la surface d'une sphère connaissant le diamètre de cette sphère. Nous allons faire un autre programme pour apprendre à utiliser les autres possibilités du HP-34C.

Pour calculer manuellement la surface d'un cercle d'après la formule : $A = \pi r^2$

- Introduisez d'abord le rayon r.
- Élevez-le au carré en appuyant sur [g] [x^2]
- Affichez π en appuyant sur [h] [π].
- Multipliez enfin le carré du rayon par PI en appuyant sur [×].

Rappelez-vous qu'un programme n'est rien d'autre que la séquence de pressions de touches qui vous permettrait de résoudre le problème manuellement. Par conséquent, pour créer le programme qui va calculer la surface d'un cercle, utilisez les mêmes touches que pour résoudre le problème manuellement, notamment :

[g] [x^2]
[h] [π]
[×]

Vous allez donc charger ces pressions de touches dans la mémoire programme ainsi que deux autres séquences obligatoires : [h] [LBL] [A] et [h] [RTN]. [h] [LBL] [A] est une adresse de label et désigne le début du programme. [h] [RTN] désigne la fin du programme.

Début du programme

Le début d'un programme est identifié par une instruction [h] [LBL] (label), suivie de la lettre [A] ou [B] ou d'un chiffre 0 à 9. L'emploi de labels permet de charger simultanément plusieurs programmes ou segments de programmes différents et de les exécuter ensuite dans n'importe quel ordre.

Fin de programme

La fin d'un programme est identifiée par une instruction [h] [RTN] (return). Si, lors de l'exécution d'un programme, le calculateur rencontre une instruction [RTN], il retourne immédiatement à la ligne 000 et s'arrête (sauf, comme nous le verrons plus loin, s'il termine un sous-programme).

Remarque

> Si un programme en cours d'exécution arrive à la fin de la mémoire programme occupée, tout se passe comme s'il avait rencontré un [h] [RTN]. Si la dernière instruction que vous devriez mettre en mémoire programme occupée est une instruction [h] [RTN], vous pouvez donc l'éliminer et économiser ainsi la place d'une ligne en mémoire.

Si vous voulez arrêter un programme à une certaine ligne en mémoire sans retourner à la ligne 000, vous pouvez introduire un [R/S] à cette ligne. Lorsque le programme en cours d'exécution rencontre un [R/S] en mémoire programme, il s'arrête. Si vous passez du mode RUN en mode PRGM, vous verrez la ligne de mémoire programme qui suit l'instruction [R/S].

(Rappelez-vous que le calculateur retourne à la ligne 000 et s'arrête après l'exécution de la dernière instruction en mémoire programme si celle-ci n'est pas une instruction [R/S], [GSB], [GTO], ou [RTN] (retour d'un sous-programme). En général, il n'est donc pas nécessaire d'arrêter le dernier programme en mémoire par une instruction [R/S].

Le programme complet pour le calcul de la surface d'un cercle à partir de son rayon est :

[h] [LBL] [A]	Nom et indication du début du programme.
[g] [x^2]	Élévation du rayon au carré.
[h] [π]	Affichage de pi
[×]	Multiplication de r^2 par pi et affichage du résultat.
[h] [RTN]	Indication de la fin du programme. Arrêt.

CHARGEMENT D'UN PROGRAMME

Si le calculateur est positionné sur PRGM, les fonctions et opérations normalement exécutées quand vous appuyez sur les touches ne le sont pas. Elles sont stockées en mémoire programme en vue de leur exécution ultérieure. Toutes les opérations du clavier, sauf neuf, peuvent ainsi être chargées en mémoire programme en vue de leur exécution ultérieure.

Pour préparer le chargement d'un programme complet dans le calculateur :

1. Placez le commutateur PRGM-RUN sur PRGM
2. Appuyez sur [f] CLEAR [PRGM] pour effacer la mémoire programme.

Quand les chiffres 000 sont affichés à gauche de l'écran, le calculateur est au début de la mémoire programme. Les trois chiffres affichés à gauche quand le calculateur est en mode PRGM indiquent toujours la ligne de mémoire programme sur laquelle il se trouve.

Pour charger le programme calculant la surface d'un cercle, appuyez sur les touches suivantes :

[h] [LBL] [A]
[g] [x^2]
[h] [π]
[×]
[h] [RTN]

Première touche du programme :

Appuyez sur	**Affichage**
[h]	**000**–

Vous voyez que l'affichage de la mémoire programme n'a pas changé. Deuxième et troisième touches du programme :

Appuyez sur	**Affichage**
[LBL]	**000**–
[A]	**001**– **25.13.11**

Le numéro de la ligne de mémoire programme à gauche de l'affichage évolue à chaque fin d'opération sur cette ligne. Les codes de touches affichés à droite sur votre écran représentent une opération complète : [h] (code de touche 25), [LBL] (code de touche 13), [A] (code de touche 11). Rien n'est chargé en mémoire programme tant qu'une opération (correspondant à une, deux ou trois pressions de touches) n'est pas complète.

Chargez maintenant le reste du programme en appuyant sur les touches suivantes : (observez l'affichage des numéros de lignes et les codes de touches).

Appuyez sur	Affichage
[g] [x^2]	002– 15 3
[h] [π]	003– 25 73
[×]	004– 61
[h] [RTN]	005– 25 12

Le programme pour le calcul de la surface d'un cercle à partir de son rayon est maintenant chargé dans la mémoire programme du HP-34C. Notez que rien n'a pu être chargé dans la zone de début de la mémoire 000.

EXÉCUTION D'UN PROGRAMME

Les programmes sont exécutés en mode RUN automatique. Le commutateur PRGM-RUN étant positionné sur RUN, introduisez toutes vos données et appuyez sur la touche [A] ou [B] pour identifier votre programme.

Placez d'abord le commutateur PRGM-RUN sur la position RUN :

Appuyez sur	Affichage	
3 [A]	**28,2743**	centimètres carrés
6 [A]	**113,0973**	mètres carrés
9 [A]	**254,4690**	kilomètres carrés

Voyons maintenant comment le calculateur a exécuté ce programme.

Recherche du label

Quand vous avez placé le commutateur PRGM-RUN sur la position RUN, le calculateur s'est positionné sur la ligne 005 de la mémoire programme, dernière ligne que vous ayez remplie au moment de charger le programme. Quand vous avez appuyé sur la touche [A], le calculateur a commencé à explorer la mémoire de haut en bas, ligne par ligne, en partant de la ligne 005 pour y chercher une instruction [LBL] [A]. Pendant la recherche, il n'a pas exécuté d'instructions.

Comme la ligne 005 ne contenait pas d'instruction [h] [LBL] [A], et qu'aucune autre ligne de la mémoire programme n'était occupée, le HP-34C est retourné à la ligne 000 et s'est remis à explorer la mémoire programme de haut en bas. Ayant trouvé [h] [LBL] [A] dans la ligne 001, il a commencé à exécuter votre programme.

Exécution des instructions

Le calculateur exécute les instructions suivant leur ordre d'entrée : il commence par l'instruction [g] [x^2] qui se trouve dans la ligne 002, puis il exécute [h] [π] dans la ligne 003 et ainsi de suite jusqu'à ce qu'il rencontre une instruction [h] [RTN], ou une instruction [R/S] (run/stop) ou bien la fin de la mémoire programme occupée. Comme la ligne 005 contient une instruction [h] [RTN], le calculateur exécute cette instruction et retourne à la ligne 000, puis s'arrête. Il affiche alors le contenu du registre X.

En général, il est préférable d'utiliser [A] et [B] pour indiquer le début d'un programme et de réserver 0 à 9 à l'étiquetage des sous-programmes (que nous allons voir plus loin). En effet, [A] et [B] n'exigent qu'une seule pression de touche pour lancer l'exécution du programme, comme notre programme de la surface du cercle, tandis que les labels numériques demandent une touche supplémentaire, [GSB]. Les labels 0 à 9 sont préférables lorsque vous avez plusieurs programmes courts à introduire dans votre calculateur. Ils vous serviront alors à adresser chaque programme individuellement. Pour illustrer ce principe, nous allons charger et exécuter notre programme de calcul de la surface d'un cercle en prenant comme label 0 :

Placez le commutateur PRGM-RUN sur PRGM.

Appuyez sur	**Affichage**
[f] CLEAR [PRGM]	**000–**
[h] [LBL] 0	**001–25.13. 0**
[g] [x^2]	**002– 15 3**
[h] [π]	**003– 25 73**
[×]	**004– 61**
[h] [RTN]	**005– 25 12**

Remettez le commutateur PRGM-RUN sur la position RUN. Exécutez maintenant le programme en utilisant les valeurs 3, 6 et 9. Ce label ayant changé, appuyez sur [GSB] [0] au lieu de [A].

Appuyez sur	**Affichage**
[3] [GSB] 0	**28,2743**
[6] [GSB] 0	**113,0973**
[9] [GSB] 0	**254,4690**

Si vous tentez d'exécuter un label [LBL] qui ne figure pas sous la forme d'une instruction dans votre programme, le HP-34C affiche Error 4. Par exemple, si votre HP-34C ne contient que le programme de la surface d'un cercle que vous venez d'introduire, vous pouvez provoquer l'affichage de Error 4 en appuyant tout simplement sur une touche alphabétique.

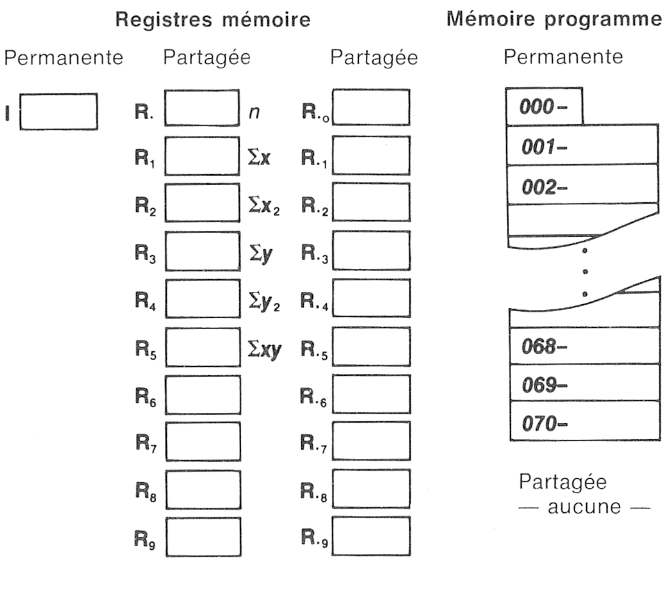
Appuyez sur	**Affichage**
[B]	**Error 4**

Pour effacer l'erreur, vous pouvez appuyer sur CLx ou sur toute autre touche, ou bien vous pouvez mettre le commutateur PRGM-RUN sur la position PRGM. Le calculateur reste positionné sur la ligne actuelle.

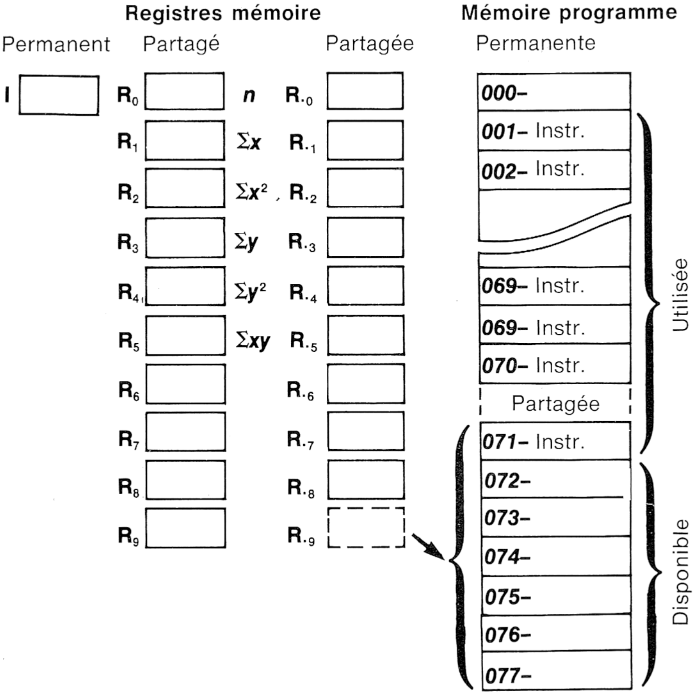
ALLOCATION MÉMOIRE AUTOMATIQUE

Conversion des registres de données en mémoire programme

L'allocation mémoire automatique qui consiste à convertir automatiquement, en cas de besoin, des registres mémoire en lignes de mémoire programme étend les possibilités de votre HP-34C. Vous commencez à programmer avec 70 lignes de mémoire programme et 20 registres mémoire (ainsi que le registre I décrit dans le chapitre 7). Pour les programmes inférieurs ou égaux à 70 lignes, l'allocation mémoire est la suivante :



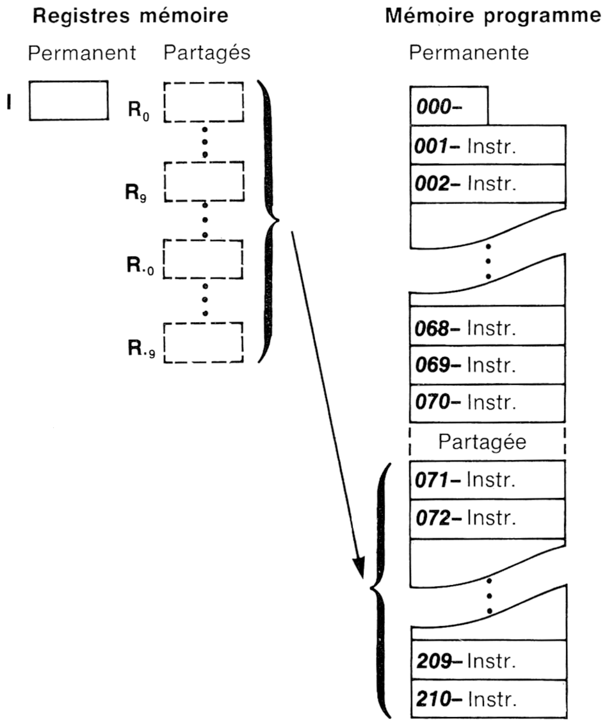
Quand vous entrez la 71e ligne de votre programme, le registre mémoire $R_{.9}$ est converti en 7 lignes de mémoire programme supplémentaires. Vous avez alors l'affectation mémoire suivante :



Quand vous enregistrez les 210 lignes de mémoire programme, les registres mémoire du calculateur se présentent comme suit :

La mémoire programme est distincte des quatre registres de la pile et du registre LAST X. Notez qu'à la place des 21 registres mémoire originaux (R_0 à R_9, $R_{.0}$ à $R_{.9}$ et I) nous n'avons plus que le registre non convertible I.

Que s'est-il passé avec les registres mémoire R_0 à R_9 et $R_{.0}$ à $R_{.9}$? Ils ont été convertis chacun en sept lignes de mémoire programme. Le tableau suivant montre l'affectation des lignes de mémoire programme à leurs registres mémoire respectifs.



$R_{.9} \rightarrow$ 071-077	$R_9 \rightarrow$ 141-147
$R_{.8} \rightarrow$ 078-084	$R_8 \rightarrow$ 148-154
$R_{.7} \rightarrow$ 085-091	$R_7 \rightarrow$ 155-161
$R_{.6} \rightarrow$ 092-098	$R_6 \rightarrow$ 162-168
$R_{.5} \rightarrow$ 099-105	$R_5 \rightarrow$ 169-175
$R_{.4} \rightarrow$ 106-112	$R_4 \rightarrow$ 176-182
$R_{.3} \rightarrow$ 113-119	$R_3 \rightarrow$ 183-189
$R_{.2} \rightarrow$ 120-126	$R_2 \rightarrow$ 190-196
$R_{.1} \rightarrow$ 127-133	$R_1 \rightarrow$ 197-203
$R_{.0} \rightarrow$ 134-140	$R_0 \rightarrow$ 204-210

Lorsque les 210 lignes de mémoire programme sont occupées, toute tentative d'insertion d'une instruction supplémentaire en un endroit quelconque de la mémoire fait apparaître Error 4. L'instruction est ignorée et les 210 lignes originales sont conservées intactes.

Ainsi, chaque fois que l'espace disponible en mémoire programme est rempli, toute nouvelle instruction entraîne la conversion automatique du registre mémoire inférieur en 7 lignes de mémoire programme. Par exemple, si les 77 premières lignes sont remplies et que l'on introduit une instruction dans la 78^e^, le registre $R_{.8}$ est converti en 7 lignes de mémoire programme supplémentaires (lignes 78 à 84) et ainsi de suite.

Remarque

La conversion des registres mémoire en lignes de mémoire programme s'effectue dans l'ordre de $R_{.9}$ à $R_{.0}$, puis de R_9 à R_0. Pour programmer les opérations [STO] et [RCL], il est donc recommandé d'utiliser les registres mémoire dans l'ordre inverse, c'est-à-dire en commençant par le registre R_0. On évite ainsi de commettre l'erreur d'acheminer [STO] et [RCL] vers des registres mémoire déjà convertis en lignes de mémoire programme. Notons aussi que le calculateur ne conserve pas les données stockées dans les registres lorsque ceux-ci sont convertis en lignes de mémoire programme.

Conversion de la mémoire programme en registres de données

Si vous appuyez sur les touches [f] CLEAR [PRGM] en mode PRGM, toute la mémoire programme partagée (lignes 071 à 210) est convertie en registres mémoire (R_0 à $R_{.9}$). La conversion mémoire programme — registres mémoire est également possible ligne par ligne, sans effacement de toute la mémoire programme. (Voir détails sur l'effacement des lignes de mémoire dans le chapitre 4, « Mise au point ».)

Utilisation de MEM

La fonction [MEM] (mémoire) intégrée à votre calculateur décrit l'affectation actuelle de la mémoire en mode programme ou non. Quand vous appuyez sur [g] [MEM], l'écran indique le nombre de lignes de programme encore disponibles avant de nécessiter la conversion d'un registre mémoire, ainsi que le nom du prochain registre-mémoire qui va être converti ($R_{.9}$ à $R_{.0}$, R_9 à R_0). Admettons qu'il y ait 44 lignes de mémoire programme occupées. Si vous appuyez sur [g] [MEM], vous aurez à l'affichage :

p−26 r−.9

Lignes à occuper avant conversion automatique d'un registre mémoire en 7 lignes de programme supplémentaires.

Prochain registre-mémoire à convertir

Si vous appuyez sur [g] [MEM] et que vous avez 173 lignes de mémoire programme occupées, vous obtiendrez l'affichage suivant :

p–02 r–4

Lignes à occuper avant conversion automatique d'un registre mémoire en 7 lignes de programme supplémentaires.

Prochain registre mémoire à convertir (c'est-à-dire registre-mémoire disponible de plus haut rang).

Si vous appuyez sur [g] [MEM] et que vous avez 205 lignes de mémoire programme occupées, l'affichage sera le suivant :

p–05 r–

Lignes encore disponibles avant occupation complète de la mémoire programme.

Il ne reste plus de registres à convertir.

Tant que vous appuyez sur [g] [MEM], l'affectation mémoire reste affichée. Quand vous relâchez la touche, l'ancien affichage réapparaît. Vous pouvez donc à tout instant connaître le nombre de lignes de programme disponibles ainsi que le nombre de registres pouvant recevoir des données. Le registre I étant un registre mémoire permanent qui possède des fonctions spéciales, il n'est pas concerné par la fonction MEM.

Remarque

Rappelons que les fonctions statistiques font appel aux registres R_0 à R_5. Si l'un ou plusieurs de ces six registres sont convertis en lignes de mémoire programme, toute tentative d'exécution de fonctions statistiques aboutira à l'affichage de Error 2.

RÉDACTION D'UN TROISIÈME PROGRAMME

Pour apprendre à connaître encore mieux les possibilités du HP-34C, nous allons écrire un troisième programme. Soit à calculer l'augmentation de volume d'un ballon sphérique en fonction de l'augmentation de son diamètre d'après la formule :

$$\text{Augmentation de volume} = 1/6\ \pi\ (d_1^3 - d_0^3)$$

d_0 étant le diamètre initial du ballon et d_1 son nouveau diamètre.
Si l'on introduisait d_0 dans le registre Y et d_1 dans le registre X, on pourrait résoudre le problème manuellement en appuyant sur les touches figurant dans la colonne de gauche ci-dessous. Les pressions de touches du programme sont les mêmes que les pressions de touches manuelles.

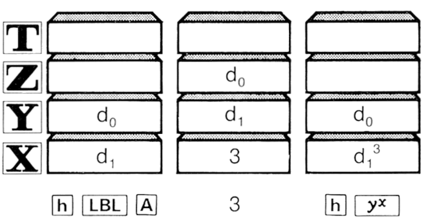
Placez le commutateur PRGM-RUN sur PRGM et continuez de la manière suivante :

Appuyez sur	Affichage	
f CLEAR PRGM	000–	
h LBL B	001–25.13.12	
3	002– 3	
h y^x	003– 25 3	Cube du nouveau diamètre
$x \rightleftarrows y$	004– 21	
3	005– 3	
h y^x	006– 25 3	Cube du diamètre initial
–	007– 41	Soustraction de $d_1^3 - d_0^3$
h π	008– 25 73	
×	009– 61	Multiplication par pi
6	010– 6	
÷	011– 71	Division par 6.
h RTN	012– 25 12	

Notez qu'aucune instruction ENTER↑ n'a été incluse pour séparer le numéro 3, ligne 002, des chiffres que vous allez entrer plus tard. Après l'instruction LBL, ENTER↑ ne provoque pas d'erreur ici, mais est inutile.

La raison est la suivante : lorsqu'un programme en cours exécute une instruction LBL, la pile est autorisée à monter. Par conséquent, l'introduction d'un nouveau nombre dans le registre X fait monter automatiquement la pile. Reprenons le dernier programme et introduisons d_0 dans le registre Y et d_1 dans le registre X. Voici ce qui se passe :

Appuyez sur **Registres de la pile**



(d'autres cas de montée de la pile sont décrits en Annexe E, « Montée de la pile » et LAST X).

Exemple

Calculez l'augmentation de volume d'un ballon sphérique si le diamètre augmente de 30 à 35 mètres.

Appuyez sur	Affichage	
30 [ENTER↑]	**30,0000**	Introduisez le diamètre initial dans le registre Y.
35 [B]	**8.312,1306**	Introduisez le nouveau diamètre dans X et exécutez le programme. La réponse est affichée en mètres cubes.

ARRÊTS ET PAUSES DANS UN PROGRAMME

Deux touches ont été prévues pour interrompre le programme : [R/S] (run/stop) et [PSE] (pause). La première permet d'arrêter l'exécution du programme pour introduire des données. La seconde permet d'effectuer une pause pour afficher les résultats, puis de reprendre automatiquement le programme.

Arrêts programmés

La fonction [R/S] (run/stop) peut être utilisée soit automatiquement (comme instruction dans un programme) soit manuellement, par pression d'une touche au clavier :

Si vous appuyez sur [R/S] :

1. Si un programme est en cours, il s'arrête.
2. Si un programme est arrêté ou n'a pas encore été lancé et que le calculateur est en mode RUN, l'exécution démarre. Elle commence à la première ligne de mémoire programme qui suit l'instruction [R/S].

   Si vous appuyez sur [R/S] en mode RUN et que vous maintenez la touche enfoncée, l'écran affiche le numéro de ligne actuel et le code de touche — si vous relâchez la touche, l'exécution commence à la ligne affichée.

Vous pouvez ainsi utiliser l'instruction [R/S] pour arrêter l'exécution d'un programme quand vous voulez introduire des données. La touche [R/S] vous sert ensuite à relancer le programme.

Exemple

Universal Tins, une fabrique de conserves, utilise des boîtes cylindriques de différentes tailles. Calculons le volume des différentes boîtes en notant d'abord la surface de la base de chacune d'elle.

Le programme ci-dessous calcule la surface de la base de chaque boîte, puis s'arrête. Une fois que vous avez noté les résultats, le programme repart pour calculer le volume, d'après la formule :

$$\text{Volume} = \text{surface de la base} \times \text{hauteur} = \pi r^2 \times h$$

Le rayon (r) et la hauteur (h) de la boîte doivent être introduits dans les registres X et Y, respectivement, avant l'exécution du programme.

Pour enregistrer ce programme, placez le commutateur PRGM-RUN sur PRGM puis entrez les instructions suivantes du programme :

Appuyez sur	**Affichage**	
[f] CLEAR [PRGM]	000–	Effacement de la mémoire programme et affichage de la ligne 000.
[h] [LBL] [A]	001– 25.13.11	
[g] [x^2]	002– 15 3	Élévation du rayon au carré.
[h] [π]	003– 25 73	Introduction de pi dans X.
[×]	004– 61	Calcul de la surface de la base.
[R/S]	005– 74	Arrêt pour noter la surface.
[×]	006– 61	Calcul du volume final.
[h] [RTN]	007– 25 12	

Placez le commutateur PRGM-RUN sur RUN. A l'aide du programme, complétez le tableau ci-dessous :

Hauteur	Rayon	Surface de la base	Volume
25	10	?	?
8	4,5	?	?

Appuyez sur	Affichage	
25 [ENTER↑]	**25,0000**	Introduction de la hauteur dans Y. Introduction du rayon dans X et calcul de la surface.
10 [A]	**314,1593**	Arrêt du programme pour affichage de la surface.
[R/S]	**7.853,9816**	Calcul du volume de la première boîte.
8 [ENTER↑]	**8,000**	Introduction de la hauteur dans Y. Introduction du rayon dans X et calcul de la surface.
4.5 [A]	**63,6173**	Arrêt du programme pour affichage de la surface.
[R/S]	**508,9380**	Calcul du volume de la deuxième boîte.

La hauteur étant introduite dans le registre Y et le rayon dans le registre X, vous pouvez calculer la surface de la base de la boîte en appuyant sur [A] en mode RUN automatique. Le programme s'arrête dès qu'il rencontre une instruction [R/S]. Vous devez alors appuyer sur [R/S] pour calculer le volume de la boîte. Après ce calcul, le programme se repositionne sur 000 et s'arrête.

Pause

L'instruction [h] [PSE] au milieu d'un programme interrompt momentanément l'exécution du programme pour afficher les résultats obtenus. La reprise du programme est automatique. La pause dure environ 1 seconde mais elle peut être prolongée par plusieurs instructions [h] [PSE] successives.

Pour montrer l'utilisation de [h] [PSE] dans un programme, nous allons modifier le programme du volume des cylindres de l'exemple précédent. Dans le nouveau programme, nous afficherons rapidement la surface de la base avant de calculer le volume. Cet exemple va également montrer qu'en programmation il peut exister différentes solutions pour résoudre le même problème.

Pour entrer le programme, positionnez le commutateur PRGM-RUN sur PRGM et appuyez sur [f] CLEAR [PRGM] afin d'effacer la mémoire programme et afficher la ligne 00. Entrez ensuite les instructions du programme.

Appuyez sur :	Affichage	
f CLEAR PRGM	000–	
h LBL A	001–25 13 11	
g x^2	002– 15 3	Élévation au carré du rayon placé dans le registre X.
h π	003– 25 73	Introduction de pi dans X.
×	004– 61	Calcul de la surface de la base.
h PSE	005– 25 74	Pause d'une seconde pour afficher la surface de la base.
×	006– 61	Calcul du volume de la boîte.
h RTN	007 25 12	

Comme dans l'exemple précédent, la hauteur et le rayon doivent avoir été introduits au préalable dans les registres Y et X respectivement. Les instructions ayant été chargées en mémoire, positionnez le commutateur PRGM-RUN sur ▮▯ RUN et remplissez le tableau suivant à l'aide du nouveau programme :

Hauteur	Rayon	Surface de la base	Volume
20	15	?	?
10	5	?	?

Appuyez sur :	Affichage	
20 ENTER↑	20,0000	Introduction de la hauteur dans Y.
		Introduction du rayon dans X et calcul.
15 A	706,8583	Affichage de la surface de la base pendant une seconde.
	14.137,1669	Arrêt du programme, affichage du volume.
10 ENTER↑	10,0000	Introduction de la 2e hauteur dans Y et du rayon dans X. Calcul.
5 A	78,5398	Affichage de la surface de la base pendant une seconde.
	785,3982	Arrêt du programme, affichage du volume.

Arrêts imprévus

Il se peut qu'une erreur se soit glissée dans votre programme et arrête son exécution. Votre calculateur peut par ailleurs s'arrêter au milieu d'un programme pour plusieurs raisons :

Exécution de [h] [RTN]

Chaque fois qu'il exécute [h] [RTN] dans un programme, le calculateur retourne à la ligne 000 et s'arrête. A la rencontre d'une instruction [RTN] à la fin d'un sous-programme, le calculateur retourne au point de branchement original (voir détails chapitre 6).

Rencontre de la fin de la mémoire programme

Si la dernière instruction présente en mémoire programme n'est pas une instruction [GTO], [GSB], [RTN] ou [R/S], et n'est pas dans un sous-programme, le programme exécuté va rencontrer la fin de la mémoire programme occupée. Il retourne alors immédiatement à la ligne 000 et s'arrête.

Pression d'une touche quelconque

La pression d'une touche quelconque arrête l'exécution du programme. Il faut donc éviter d'appuyer sur une touche pendant l'exécution d'un programme. Par contre, au milieu d'une séquence de chiffres, le calculateur ne s'arrête jamais. Si vous appuyez sur une touche pendant qu'un programme introduit un nombre dans le registre X, tout le nombre est écrit et le programme exécute la ligne suivante avant de s'arrêter.

Vous pouvez relancer un programme à l'aide de la touche [R/S] en mode RUN. Quand vous appuyez sur [R/S], le programme reprend à l'endroit où il s'est arrêté comme si rien ne s'était passé.

Arrêts dûs à une erreur

Si le calculateur tente d'exécuter une opération qui est source d'erreur (voir « Indications d'erreurs », Annexe D) pendant l'exécution d'un programme, le programme s'arrête immédiatement et le calculateur affiche le mot Error accompagné d'un chiffre. Pour connaître le numéro de ligne et le code de touche de l'instruction en erreur, vous pouvez passer en mode PRGM.

Calculs avec dépassement de capacité

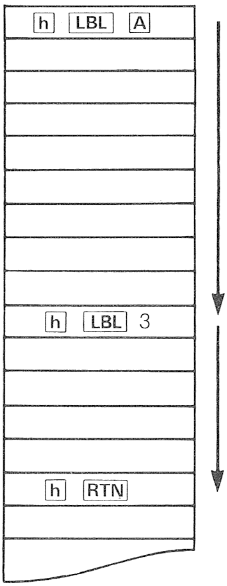
L'affichage vous renseigne à tout instant sur les raisons d'un arrêt éventuel du programme. Si le programme s'arrête parce que le résultat d'un calcul effectué dans le registre X est un nombre supérieur à $9.999999999 \times 10^{99}$, l'écran n'affiche que des 9 et un signe. Pour connaître la raison de l'arrêt, il suffit alors de passer en mode PRGM et de regarder le code de touche affiché sur l'écran.

Si l'opération arithmétique lancée dans un registre mémoire dépasse la capacité du registre, le calculateur s'arrête et affiche Error 1. Le nombre présent dans le registre concerné reste inchangé. Quand vous effacez le message d'erreur, l'écran affiche le nombre précédent.

Si le résultat d'un calcul est un nombre inférieur à $1.000000000 \times 10^{-99}$, le nombre est remplacé par un zéro et le programme en cours se poursuit normalement. On appelle cela un dépassement inférieur de capacité.

LABELS

Les labels (A, B, 0 à 9) utilisés dans votre programme servent d'adresses. Ils indiquent le début ou le point de reprise de l'exécution du programme. A l'intérieur d'un programme, le label est ignoré et le programme se poursuit normalement. Par exemple, dans le segment de programme ci-dessous, si vous appuyez sur A, l'exécution commence à h LBL A et continue en descendant dans la mémoire programme, saute l'instruction h LBL 3, poursuit jusqu'à RTN et après exécution de RTN, retourne à la ligne 000 et s'arrête.



Début du programme... si vous appuyez sur A.

Déroulement du programme...

avec saut de h LBL 3 qui est « passante »...

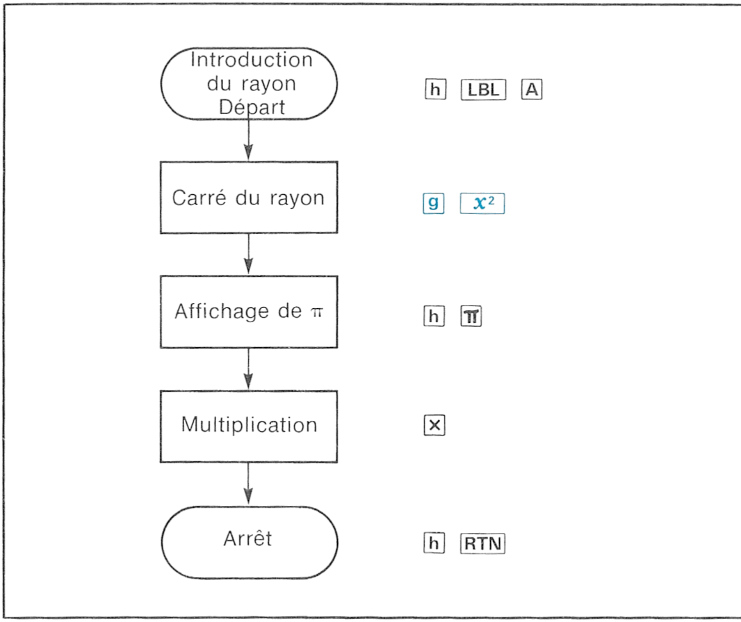
jusqu'à RTN qui provoque un retour à la ligne 000 et arrêt.

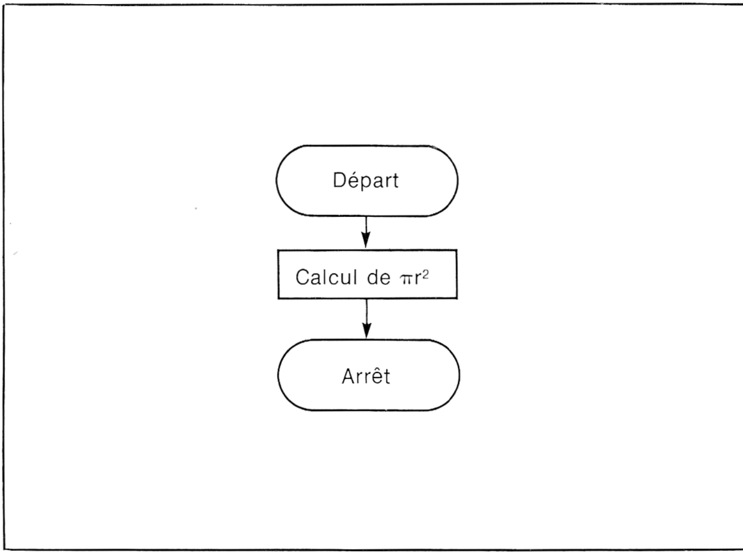
ORGANIGRAMMES

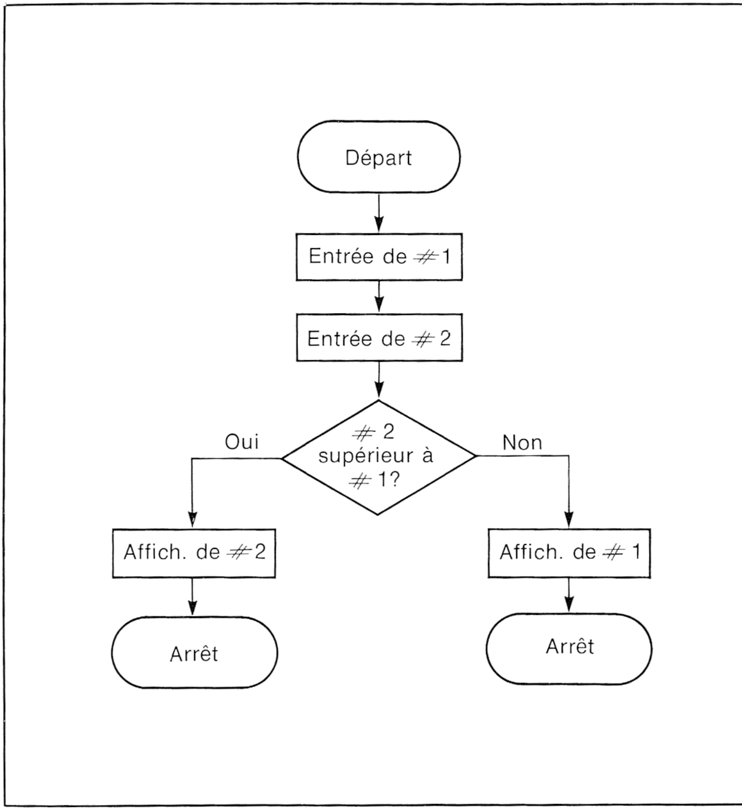
Nous allons abandonner un moment la programmation proprement dite pour parler d'un outil particulièrement utile : l'organigramme.

L'organigramme est une esquisse de la démarche suivie pour résoudre un problème. Il n'est pas toujours aisé de rédiger un programme qui contient 210 instructions, surtout si vous avez l'intention de le charger entièrement sans interruptions. L'organigramme vous permet de découper un programme en plusieurs groupes successifs d'instructions et indique, comme une carte routière, le chemin à suivre.

Un organigramme peut être simple ou détaillé. L'organigramme suivant représente les opérations effectuées pour calculer la surface d'un cercle d'après la formule $A = \pi r^2$. En regard de chaque case, se trouvent les instructions du programme.




Départ
Calcul de πr²
Arrêt



Une fois que l'organigramme est tracé, vous remplacez chacun de ses éléments par le groupe d'instructions qu'il représente. Quand vous exécutez le programme chargé dans le calculateur, si # 2 est supérieur à # 1, la réponse à la question « # 2 supérieur à # 1 » est OUI. Le programme suit donc la voie de gauche, affiche # 1 et s'arrête. Si, au contraire, la réponse à la question est NON, le programme suit la voie de droite et affiche ≠1. Nous verrons plus loin les nombreuses instructions qui vous permettront de faire prendre à votre calculateur des décisions logiques.

Dans ce manuel, vous apprendrez progressivement, à l'aide des organigrammes illustrant les exemples et problèmes choisis pour vous faire connaître les nombreuses possibilités de votre calculateur, à tracer vos propres organigrammes pour créer, rédiger, mettre au point vos programmes et éliminer les erreurs.

PROBLÈMES

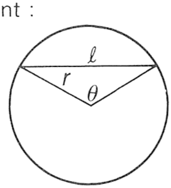
Voici quatre exemples de programmation qui vous permettront de mettre en pratique les connaissances acquises jusqu'ici. Vous trouverez les solutions aux problèmes quelques pages plus loin. Mais essayez d'abord de chercher vos propres solutions, l'exercice vous sera d'autant plus utile. Et n'oubliez pas qu'en programmation, il n'existe pas qu'une seule solution à un problème.

1. En suivant l'exemple du programme de calcul de la surface d'un cercle à partir de son rayon, rédigez et chargez un programme calculant le rayon r d'un cercle connaissant sa surface A, d'après la formule $r = \sqrt{A/\pi}$. Mettez-vous en mode PRGM et appuyez sur [f] CLEAR [PRGM] pour effacer la mémoire programme. Encadrez votre programme avec [h][LBL][A] et [h] [RTN]. Après avoir chargé le programme, exécutez-le pour calculer le rayon de cercles ayant des surfaces 28,27 cm², 113,10 m² et 254,47 km².

   Réponses : 3 cm, 6 m et 9 km.

2. Écrivez un programme pour calculer la longueur d'une corde l, sous-tendue par un angle θ dans un cercle de rayon r à l'aide de l'équation $l = 2r\text{Sin}\,\frac{\theta}{2}$ identifier ce nouveau programme avec [h] [LBL] [B] et utilisez-le pour remplir le tableau suivant :

r (mètres)	θ	l
25	30	?
50	45	?
100	90	?



Introduisez d'abord le rayon r et l'angle θ dans cet ordre.

Réponses : 12,94 m, 38,27 m, 141,42 m.

Si ce programme vous pose des difficultés, reprenez le chapitre : « Rédaction d'un troisième programme ».

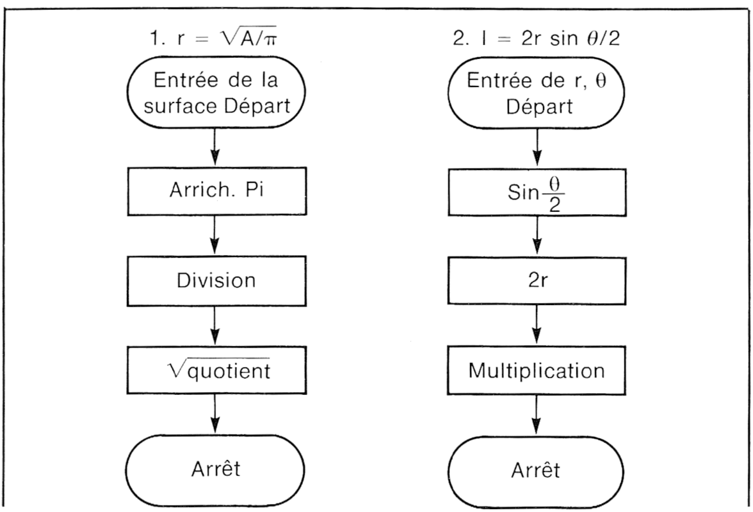
3. Écrivez et chargez un programme convertissant des degrés centigrades en degrés Fahrenheit, suivant la formule F = 1,8 C + 32. Marquez le début et la fin du programme avec h LBL 0 et h RTN et exécutez-le pour convertir − 40, 0 et + 72 degrés centigrades en degrés Fahrenheit.

   Réponses : − 40 °F, 32 °F, 161,60 °F

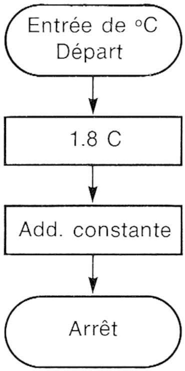
4. Après avoir exécuté le programme 3, créez un programme de conversion Fahreheint/centigrades suivant la formule C = (F − 32) 5/9. Encadrez ce programme avec h LBL 1 et h RTN. Exécutez-le pour reconvertir les degrés Fahrenheit en degrés centigrades.

Si vous avez écrit et chargé les programmes 3 et 4, vous pouvez maintenant convertir toutes les températures de centigrades en Fahrenheit en appuyant sur GSB 0 et de Fahrenheit en centigrades en appuyant sur GSB 1. Si vous avez des doutes concernant l'utilisation de GSB et des labels 0 à 9 pour l'adressage sélectif des programmes, reportez-vous au chapitre « Exécution des instructions ».

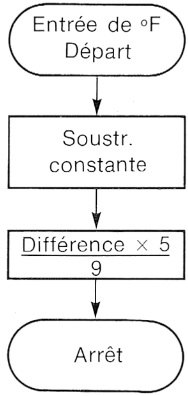
Organigrammes des programmes 1, 2, 3, 4



3. $F = 1.8\ C + 32$



4. $C = (F - 32)\ 5/9$



Solutions des problèmes 1, 2, 3, 4

	Appuyez sur	Affichage
Rayon d'un cercle	[h] [LBL] [A]	001–25 13 11
	[h] [π]	002– 25 73
	[÷]	003– 71
	[f] [√x]	004 14 3
	[h] [RTN]	005– 25 12
Longueur d'une corde	[h] [LBL] [B]	001–25 13 12
	2	002– 2
	[÷]	003– 71
	[f] [SIN]	004– 14 7
	[×]	005 61
	2	006– 2
	[×]	007– 61
	[h] [RTN]	008– 25 12
Conversion de degrés centigrades en degrés	[h] [LBL] 0	001–25 13 0
	1	002– 1
	[•]	003– 73
	8	004– 8

Fahrenheit	[×]	005–	61
	3	006–	3
	2	007–	2
	[+]	008–	51
	[h] [RTN]	009–	25 12
Conversion	[h] [LBL] 1	010–25	13 1
de degrés	3	011–	3
Fahrenheit	2	012–	2
en degrés	[−]	013–	41
centigrades	5	014–	5
	[×]	015–	61
	9	016–	9
	[÷]	017–	71
	[h] [RTN]	018–	25 12

Solutions des séquences de touches

1. Surface [A] = rayon
2. Rayon [ENTER↑] θ [B] = longueur de la corde
3. C [GSB] [0] = F
4. F [GSB] [1] = C

Bravo ! Vous commencez à connaître votre calculateur.

TECHNIQUES DE PROGRAMMATION

Dans certains calculs, la même variable revient plusieurs fois au cours du même programme. Bien que votre calculateur vous offre dans ce cas plusieurs solutions, la meilleure est toujours celle qui vous fait économiser le plus de temps et de place en mémoire. Nous allons comparer deux manières de résoudre le problème de la récurrence des variables. Prenons le polynôme suivant :

$$f(x) = x^4 + 3x^3 - x^2 + 4x - 1$$

La variable x revient quatre fois : x^4, x^3, x^2, x. Il faudra donc quatre puissances de x pour calculer f(x). Votre tâche consistera alors à écrire un programme qui décrit mathématiquement f(x) **et** qui vous fournit une copie de x chaque fois que vous en avez besoin au cours de l'exécution du programme. Pour cela, vous pouvez soit mettre en mémoire une copie de la variable et la rappeler en cas de besoin, soit remplir la pile de copies de la variable avant l'exécution du programme. Cette seconde méthode est préférable car :

1. Il est facile de prévoir dans votre programme la mise en mémoire d'une copie de la variable prête à être utilisée directement ou par l'intermédiaire d'une instruction [x⇄y]. (Rappelons que chaque fois que la pile

descend, le registre T recopie le dernier nombre qu'elle contenait avant de descendre.) Vous économisez ainsi des lignes de mémoire programme puisque les instructions [STO] et [RCL] sont inutiles.

2. Vous pouvez disposer d'un registre mémoire supplémentaire.
3. Cette méthode de remplissage de la pile est pratique pour l'évaluation des expressions polynomiales en général et dans la plupart des cas d'utilisation de [SOLVE] et [$\int_x^y$] (voir plus loin).

Imaginons un programme évaluant l'expression $x^4 + 3x^3 - x^2 + 4x - 1$ à l'aide de la méthode de remplissage de la pile. Examinons les instructions ligne par ligne et voyons la relation entre chaque instruction et la pile. La valeur de x est supposée présente dans la pile au départ du programme.

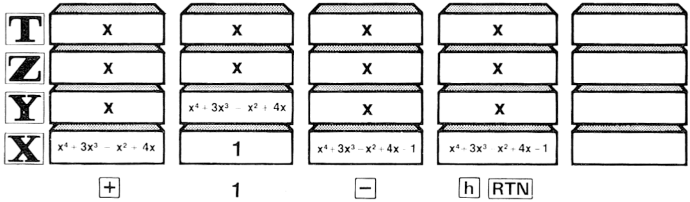
Registres de la pile

Appuyez sur : **Affichage :**

T	x	x	x	x	x
Z	x	x	x	x	x^4
Y	x	x	x	x^4	x
X	x	4	x^4	x	3
	[h] [LBL] [A]	4	[h] [y^x]	[$x \gtrless y$]	3

T	x	x	x	x	x
Z	x	x^4	x	x	x
Y	x^4	x^3	x^4	x	$x^4 + 3x^3$
X	x^3	3	$3x^3$	$x^4 + 3x^3$	x
	[h] [y^x]	3	[$\times$]	[$+$]	[$x \gtrless y$]

T	x	x	x	x	x
Z	x	x	x	$x^4+3x^3-x^2$	x
Y	$x^4 + 3x^3$	x	$x^4+3x^3-x^2$	x	$x^4+3x^3-x^2$
X	x^2	$x^4+3x^3-x^2$	x	4	$4x$
	[g] [x^2]	[$-$]	[$x \gtrless y$]	4	[$\times$]



Notez qu'il reste des copies de la variable dans la pile après l'exécution du programme. Si, pour une raison quelconque, vous voulez réafficher une copie de la variable après avoir évalué f(x) à cette variable, il suffit d'appuyer sur [x⇄y].

Exercez-vous maintenant à évaluer l'expression avec différentes valeurs de x après avoir introduit le programme.

Mettez le commutateur PRGM-RUN sur PRGM.

Appuyez sur :	Affichage
[f] CLEAR [PRGM]	000–
[h] [LBL] [A]	001–25 13 11
4	002– 4
[h] [y^x]	003– 25 3
[x⇄y]	004– 21
3	005– 3
[h] [y^x]	006– 25 3
3	007– 3
[×]	008– 61
[+]	009– 51
[x⇄y]	010– 21
[g] [x^2]	011– 15 3
[−]	012– 41
[x⇄y]	013– 21
4	014– 4
[×]	015– 61
[+]	016– 51
1	017 1
[−]	018– 41
[h] [RTN]	019– 25 12

Mettez le commutateur PRGM-RUN sur ▮ RUN et calculez l'expression en prenant pour x les valeurs suivantes : 1, 2, 7.1935 17.5.

Appuyez sur :	**Affichage**	
1 [ENTER↑] [ENTER↑] [ENTER↑]	**1,0000**	Introduction de la variable dans la pile
[A]	**6,0000**	f(x)
2 [ENTER↑] [ENTER↑] [ENTER↑]	**2,0000**	Introduction de la variable dans la pile
[A]	**43,000**	f(x)
7.1935 [ENTER↑] [ENTER↑] [ENTER↑]	**7,1935**	Introduction de la variable dans la pile
[A]	**3.770,4359**	f(x)
17.5 [f] [LN] [ENTER↑] [ENTER↑] [ENTER↑]	**2,8622**	Introduction de la variable dans la pile
[A]	**139,7118**	f(x)

Utilisation de la méthode de Horner

La méthode utilisée plus haut pour écrire le programme est logique et facile et elle donne les résultats désirés. Mais il existe une méthode mathématique, dite « Méthode d'Horner », qui a l'avantage d'être encore plus simple et plus courte.

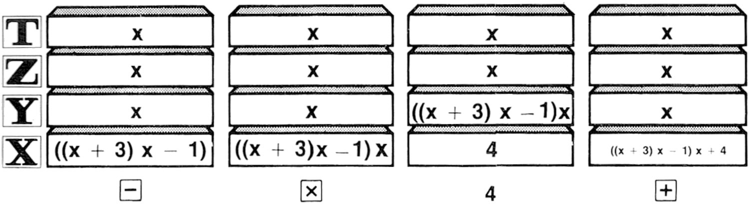
La méthode de Horner réarrange le polynôme $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0$ pour éliminer les exposants supérieurs à zéro. L'expression est donc formulée comme une série d'opérations arithmétiques portant sur la variable x et les coefficients a_n, a_{n-1}, $\ldots a_1$, a_0. Si nous appliquons, par exemple, la méthode d'Horner à l'expression polynominale que nous venons de calculer :

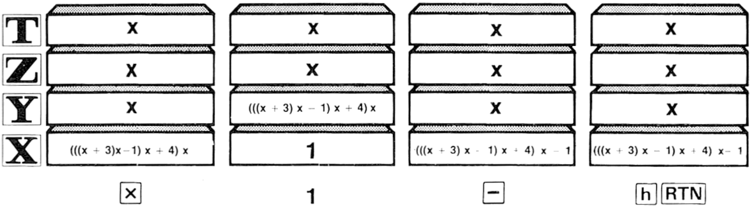
$x^4 + 3x^3 - x^2 + 4x - 1$
$(X^3 + 3x^2 - x + 4)\, x - 1$
$((x^2 + 3x - 1)\, x + 4)\, x - 1$
$(((x + 3)\, x - 1)\, x + 4)\, x - 1$

Grâce à cette méthode de simplification, nous pouvons réécrire le programme précédent en remplaçant les six opérations arithmétiques et les trois opérations exponentielles par seulement sept opérations artihmétiques.

Appuyez sur : **Registres de la pile**

T	x	x	x	x	x
Z	x	x	x	x	x
Y	x	x	x	x	(x + 3) x
X	x	3	(x + 3)	(x + 3) x	1
	[h] [LBL] [B]	3	[+]	[×]	1





Ce programme n'occupe que 13 lignes en mémoire, donc six lignes de moins que le programme précédent. Introduisez-le dans le calculateur et reprenez les mêmes exemples que précédemment :

Mettez le commutateur PRGM-RUN sur PRGM ▮▮▮. Si vous n'avez pas exécuté d'autres instructions depuis les dernières évaluations de f(x), vous aurez 000 à l'affichage. Sinon, appuyez sur [GTO] [•] 000 (voir détails sur [GTO] plus loin).

Appuyez sur :	**Affichage**
[h] [LBL] [B]	001−25, 13, 12
3	002− 3
[+]	003− 51
[×]	004− 61
1	005− 1
[−]	006− 41
[×]	007− 61
4	008− 4
[+]	009− 51
[×]	010− 61
1	011− 1
[−]	012− 41
[h] [RTN]	013− 25 12

Mettez le commutateur PRGM-RUN sur ▮▯ RUN et évaluez l'expression en prenant pour x les mêmes valeurs que précédemment.

Appuyez sur :	Affichage	
1 [ENTER↑] [ENTER↑] [ENTER↑]	**1,0000**	Introduction de la variable dans la pile
[B]	**6,0000**	f(x)
2 [ENTER↑] [ENTER↑] [ENTER↑]	**2,0000**	Introduction de la variable dans la pile
[B]	**43,0000**	f(x)
7.1935 [ENTER↑] [ENTER↑] [ENTER↑]	**7,1935**	Introduction de la variable dans la pile
[B]	**3.770,4359**	f(x)
17.5 [f] [LN] [ENTER↑] [ENTER↑] [ENTER↑]	**2,8622**	Introduction de la variable dans la pile
[B]	**139,7118**	f(x)

Avez-vous noté un changement ? Le temps de calcul de chaque variable a été réduit de 2 secondes par rapport au programme [A].

Autres applications

Vous avez vu que la méthode de remplissage de la pile offrait un moyen simple et utile pour évaluer une expression contenant plusieurs fois la même variable. En appliquant la méthode d'Horner, quand c'est possible, on peut en outre gagner du temps et de la place. Nous verrons plus loin que le remplissage automatique de la pile dans le cas des opérations de calcul de racines et d'intégration numérique avec [SOLVE] et [$\int_y^x$] représente un autre atout de votre HP-34C, sur le plan des performances et de la facilité de programmation.

Problèmes

A l'aide de la technique de remplissage de la pile et de la méthode d'Horner, écrivez les programmes d'évaluation des expressions suivantes en prenant pour x les valeurs 1,5, 3,73 et −4,25 et exécutez ces programmes :

1. $2\ x^5 - x^4 + 2\ x^2 + x + 1$
2. $0{,}97\ \sin^3 x + 0{,}04\ \sin^2 x - 1{,}73\ \sin x - 1$

Réponses :

1. 17,1250 ; 1.283,0102 ; 3.006,5371
2. − 1,0452 ; − 1,1121 ; − 0,8720

MISE AU POINT DU PROGRAMME

Le clavier du calculateur HP-34C possède plusieurs touches qui permettent de modifier les étapes d'un programme chargé en mémoire sans avoir à réenregistrer ce dernier.

Vous savez qu'il y a neuf fonctions qui ne peuvent pas être enregistrées dans la mémoire programme. Sept d'entre elles sont des fonctions de mise au point et de manipulation du programme qui vous permettent de faire des modifications et des corrections sur vos programmes.

OPÉRATIONS NON ENREGISTRABLES

CLEAR [PRGM] est une opération du clavier qui ne peut être enregistrée en mémoire programme. Si vous appuyez sur [f] CLEAR [PRGM] en mode PRGM, la mémoire programme s'efface et le calculateur se repositionne au début de la mémoire (ligne 000). En mode RUN, cette même séquence de touches provoque un retour à la ligne 000. Notez que [f] CLEAR [PRGM] ne change pas le mode trigonométrique RAD ou GRD en mode DEG.

[SST] (single step) est également une opération non enregistrable. Si vous appuyez sur [h] [SST] en mode PRGM, le calculateur recherche et affiche la ligne de programme suivante sans exécuter d'instruction. Si vous appuyez sur [h] [SST], en mode RUN, le calculateur recherche également la ligne de programme suivante et l'affiche. Mais quand vous relâchez la touche [SST], le calculateur exécute l'instruction mémorisée dans cette ligne.

[BST] (back step) est une opération non enregistrable utilisée en modes PRGM et RUN pour rechercher et afficher la ligne précédente de la mémoire programme. En mode RUN, si vous relâchez [BST], le calculateur réaffiche l'ancien contenu du registre X, sans exécuter d'instructions.

CLEAR [PREFIX] est une opération non enregistrable utilisée après une pression de touche préfixe ([f], [g] ou [h]) pour annuler la touche. Elle annule aussi toutes les pressions de touche d'une instruction incomplète telle que [f] [SCI] ou [STO] [+]. Elle n'a aucun effet sur une instruction complète (par ex. [f] [SCI] 5, [STO] [+] 1 etc.).

[GTO] (go to) [•] nnn est une instruction non enregistrable qui permet de se placer sur un numéro de ligne donné. (Mais si elle est suivie d'un label numérique (0 à 9), elle peut être enregistrée comme une instruction du programme — voir détails plus loin.) En mode PRGM ou en mode RUN, GTO • suivi d'un numéro de ligne à trois chiffres, positionne le calculateur sur la ligne portant ce numéro. Aucune instruction n'est exécutée. Si le calculateur est en mode RUN, vous pouvez vérifier s'il est positionné sur cette ligne en déplaçant rapidement le commutateur de RUN sur PRGM. L'opération GTO • nnn est particulièrement utile en mode PRGM pour se rendre en un point quelconque de la mémoire programme occupée pour y effectuer des vérifications ou des modifications.

Remarque :

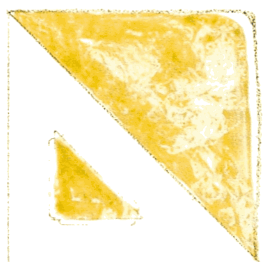
Toute tentative d'exécution d'une instruction GTO • nnn pour se positionner sur une ligne de mémoire programme non occupée ou ne résultant pas d'une conversion d'un registre mémoire est une opération illicite provoquant un signal d'erreur 4.

DEL (delete) est une opération non enregistrable qui permet d'annuler des instructions en mémoire programme. En mode PRGM, la pression de h DEL provoque l'annulation de l'instruction placée dans la ligne de mémoire programme actuelle et la remontée d'une ligne de toutes les instructions suivantes dans la mémoire programme. En mode RUN, la séquence de touches h DEL n'a pas d'effet.

MEM (memory) est une fonction qui permet d'afficher l'allocation actuelle de la mémoire en mode programme ou non (revoir chapitre 3, « Notions de programmation »).

Nous allons appliquer maintenant les fonctions de mise au point à la vérification et à la modification d'un programme.

Programme « théorème de Pythagore »



Le programme suivant calcule l'hypoténuse d'un triangle rectangle, suivant la formule $c = \sqrt{a^2 + b^2}$.

Les pressions de touches sont les mêmes que pour le calcul manuel de c, en supposant que les valeurs des côtés a et b aient été introduites dans les registres X et Y de la pile.

Pour charger le programme, placez le commutateur sur PRGM. Appuyez ensuite sur f CLEAR PRGM afin d'effacer de la mémoire programme tous les programmes précédents et repositionner le calculateur sur la ligne 000. Enfin, chargez le programme suivant :

Appuyez sur :	Affichage	
[h] [LBL] [A]	001– 25 13 11	
[g] [x^2]	002– 15 3	b^2
[$x \rightleftarrows y$]	003– 21	
[g] [x^2]	004– 15 3	a^2
[+]	005– 51	$a^2 + b^2$
[f] [$\sqrt{x}$]	006– 14 3	$\sqrt{a^2 + b^2}$
[h] [RTN]	007– 25 12	Fin du programme ; le calculateur retourne à la ligne 000 et s'arrête.

Remettez le calculateur en mode ▮▯ RUN. Exécutez le programme. Calculez, par exemple, l'hypoténuse d'un triangle rectangle dont les côtés a et b sont égaux à 22 et 9 m. (Vous pouvez entrer ces données dans n'importe quel ordre.)

Appuyez sur :	Affichage	
22 [ENTER↑]	22,0000	
9	9,	
[A]	23,7697	Longueur du côté c en mètres.

Pour calculer l'hypoténuse d'un triangle rectangle dont les côtés a et b sont égaux à 73 et 99 km :

Appuyez sur :	Affichage	
73 [ENTER↑]	73,0000	
99	99,	
[A]	123,0041	Longueur du côté c en km.

EXÉCUTION PAS A PAS DU PROGRAMME

En mode RUN, vous pouvez exécuter le programme ligne par ligne en appuyant sur les touches [h] [SST] (single-step). Pour exécuter pas à pas le programme de calcul de l'hypoténuse d'un triangle rectangle ayant des côtés de 73 à 99 km :

Appuyez sur :	Affichage	
73 [ENTER↑]	73,0000	
99	99,	Initialisation du programme pour cet ensemble de données avant exécution.

Pour exécuter l'instruction suivante, appuyez sur [h] [SST], maintenez cette pression pour afficher le code de touche de la nouvelle instruction, puis relâchez-la pour exécuter l'instruction. (Rappelons que l'instruction [h][RTN] dans la ligne 007 repositionne le calculateur sur la ligne 000 après la dernière exécution du programme).

Appuyez sur :	**Affichage**	
[h] [SST]	**001–25, 13, 11**	Affichage du code de touche de de [h][LBL][A] quand [SST] est enfoncée
	99,0000	Exécution de [h] [LBL] [A] au relâchement de [SST].

Notons que vous n'étiez pas obligé d'appuyer sur [A]. En mode d'exécution pas à pas, le programme débute à la ligne à laquelle vous vous trouvez au moment d'appuyer sur [h] [SST] : ici 001.

Continuez à exécuter le programme en réappuyant sur [h] [SST]. Maintenez les touches enfoncées pour afficher le code de touche de l'instruction suivante. Relâchez-les pour exécuter l'instruction.

Appuyez sur :	**Affichage**	
[h] [SST]	**002– 15 3**	Code de touche pour [x^2]
	9801,0000	Exécution.

Exécutez de même la 3e instruction [$x \rightleftarrows y$] du programme.

Appuyez sur :	**Affichage**	
[h] [SST]	**003– 21**	Code de touche de [$x \rightleftarrows y$] .
	73,0000	Exécution.

Allez jusqu'au bout du programme. Après avoir exécuté [h] [RTN] dans la ligne 007, le calculateur retournera à la ligne 000 (nous verrons plus loin d'autres détails concernant le fonctionnement de [RTN]). Le résultat affiché en fin d'exécution du programme est exactement le même que si le programme avait été exécuté automatiquement.

Appuyez sur :	**Affichage**	
[h] [SST]	**004– 15 3**	
	5.329,0000	
[h] [SST]	**005– 51**	
	15.310,0000	
[h] [SST]	**006– 14 3**	
	123,0041	
[h] [SST]	**007– 25 12**	
	123,0041	Résultat final.

Remarque

> [h] [SST] ne franchira pas les limites de la mémoire programme occupée. A partir de la dernière ligne de la mémoire occupée, cette séquence de touches aura pour effet de vous ramener sur la ligne 000, en mode RUN comme en mode PRGM. En mode RUN, l'affichage original reste inchangé. En mode PRGM l'écran affiche 000, c'est-à-dire le début de la mémoire programme.

La touche [SST] associée à la touche préfixe [h] permet donc, en mode RUN, de créer et de corriger des programmes. Pour modifier les programmes, en mode PRGM, on se sert des touches [GTO] [•] 000, [SST], [BST] et [GTO] [•] nnn.

MODIFICATION D'UN PROGRAMME

Vous allez maintenant modifier le programme du théorème de Pythagore pour que le contenu du registre X soit automatiquement affiché à certains endroits du programme. Pour cela, vous allez insérer l'instruction [h] [PSE] : le programme s'arrête pour vous permettre de lire le contenu du registre X pendant 1 seconde, puis il reprend. (Voir détails sur [PSE] plus loin.) Voici le programme que vous venez d'exécuter :

Appuyez sur :	Affichage	
[h] [LBL] [A]	001–25 13 11	
[g] [x^2]	002– 15 3	
[$x \gtrless y$]	003– 21	Nous allons insérer une instruction [h] [PSE] après chacune de ces instructions
[g] [x^2]	004– 15 3	
[+]	005– 51	
[f] [$\sqrt{x}$]	006– 14 3	
[h] [RTN]	007– 25 12	

VISUALISATION SANS EXÉCUTION

Vous pouvez utiliser [SST] en mode PRGM pour avancer pas à pas jusqu'à la ligne désirée sans exécuter le programme. Si vous passez en mode PRGM, le calculateur se repositionne sur la ligne 000 de la mémoire programme après avoir exécuté l'instruction [h] [RTN]. Une pression simple sur [h] [SST] le place sur la ligne 001 et le contenu de cette ligne est affiché sur l'écran. Aucune instruction n'est exécutée.

Mettez le commutateur PRGM-RUN sur PRGM.

Appuyez sur :	Affichage	
	000–	Ligne 000 de la mémoire programme
[h] [SST]	**001–25 13 11**	

Le calculateur est donc positionné sur la ligne 001 de la mémoire programme. Si vous entrez une fonction enregistrable, elle sera chargée dans la ligne suivante de la mémoire programme, soit la ligne 002, et toutes les instructions suivantes descendront d'une ligne dans la mémoire.

Ainsi, pour charger l'instruction [h] [PSE] qui va provoquer une pause et l'affichage du contenu du registre X :

Appuyez sur :	Affichage
[h] [PSE]	**002– 25 74**

Que s'est-il passé en mémoire programme lorsque vous avez chargé cette instruction ? Quand vous avez appuyé sur [h] [PSE] à la ligne 001, le contenu de la mémoire programme a été modifié ainsi :

Avant...	**Après...**	
001 [h] [LBL] [A]	001 [h] [LBL] [A]	
002 [g] [x^2]	002 [h] [PSE]	Insertion de l'instruction [h] [PSE] ici.
003 [$x \gtrless y$]	003 [g] [x^2]	
004 [g] [x^2]	004 [$x \gtrless y$]	
005 [+]	005 [g] [x^2]	Toutes les instructions suivantes descendent d'une ligne.
006 [f] [$\sqrt{x}$]	006 [+]	
007 [h] [RTN]	007 [f] [$\sqrt{x}$]	
	008 [h] [RTN]	

L'insertion d'une instruction dans le programme a fait descendre toutes les instructions suivantes d'une ligne en mémoire.

RETOUR A LA LIGNE 000

Au moment de sa mise en fonction, le HP-34C retourne automatiquement à la ligne 000. La séquence de touches [f] CLEAR [PRGM] en mode PRGM produit le même effet de retour à la ligne 000 avec effacement de toutes les instructions en mémoire programme. Pour garder les programmes en mémoire, vous pouvez retourner à la ligne 000 en appuyant sur [GTO] [•] 000 en mode PRGM ou en mode RUN et sur [h] [RTN] ou [f] CLEAR [PRGM] en mode RUN.

Retournez à la ligne 000 avec le programme du théorème de Pythagore chargé en mémoire programme :

Appuyez sur :	**Affichage**
GTO • 000	**000–**

BRANCHEMENT A UNE LIGNE DÉTERMINÉE

Passer de la ligne 000 à une autre ligne dans la mémoire programme demande parfois un grand nombre de pressions de touches h SST et beaucoup de temps. Il existe donc un moyen plus simple : GTO • nnn. Cette instruction vous branche directement à n'importe quelle ligne de la mémoire, de la même façon que GTO • 000 saute à la ligne 000.

Quand vous appuyez sur GTO • nnn, le calculateur saute immédiatement à la ligne spécifiée par nnn, sans exécuter d'instruction. En mode RUN, l'affichage reste inchangé. En mode PRGM, la ligne de mémoire occupée (numéro de ligne et codes de touches) adressée par nnn apparaît sur l'écran. La recherche d'un label ou l'exécution du programme commencera à cette ligne en mode RUN. Par contre, le chargement de nouvelles instructions en mode PRGM commencera à la ligne suivante.

Par exemple, pour ajouter une instruction h PSE au programme et afficher le contenu du registre X après addition des carrés par l'instruction chargée dans la ligne 006, appuyez sur GTO (go to) suivi d'un point décimal et du nombre à trois chiffres correspondant au numéro de ligne approprié. Appuyez ensuite sur h PSE pour placer cette instruction dans la ligne suivante de la mémoire programme. N'oubliez pas qu'une telle insertion décale toutes les instructions qui suivent d'une ligne en mémoire. Pour ajouter l'instruction h PSE à la suite de l'instruction + actuellement chargée dans la ligne 006, mettez le calculateur en mode PRGM, puis :

Appuyez sur :	**Affichage**
GTO • **006**	**006– 51**
h PSE	**007– 25 74**

Quand l'instruction h PSE est chargée dans la ligne 007, celle qui se trouvait dans cette ligne de la mémoire descend dans la ligne 008 faisant descendre toutes les instructions d'une ligne.

L'insertion de l'instruction [h] [PSE] après la ligne 006 a modifié l'état de la mémoire programme :

Avant...		**Après...**	
001 [h] [LBL] [A]		001 [h] [LBL] [A]	
002 [h] [PSE]		002 [h] [PSE]	
003 [g] [x^2]		003 [g] [x^2]	
004 [$x \gtrless y$]		004 [$x \gtrless y$]	
005 [g] [x^2]		005 [g] [x^2]	
006 [+]		006 [+]	
007 [f] [$\sqrt{x}$]		007 [h] [PSE]	Insertion de [h] [PSE]
008 [h] [RTN]	→	008 [f] [$\sqrt{x}$]	Toutes les instructions qui suivent descendent d'une ligne.
	→	009 [h] [RTN]	

INSERTION D'INSTRUCTIONS DANS DE LONGS PROGRAMMES

Quand les 70 premières lignes de la mémoire programme sont occupées, le calculateur convertit automatiquement les registres mémoire en mémoire programme disponible en remplaçant chaque fois un registre par 7 lignes de mémoire. La réaffectation de la mémoire disponible s'effectue ainsi par blocs de 7 lignes. Donc, si les 70 premières lignes de mémoire programme sont occupées, l'insertion d'une nouvelle instruction en un point quelconque entraîne la conversion automatique d'un registre mémoire (R_9 en l'occurence) en 7 lignes supplémentaires de mémoire programme disponible, la dernière instruction du programme étant stockée dans la 71^e ligne. On a donc 77 lignes de programme disponibles et 71 occupées. (Voir « Affectation de la mémoire partagée ».) Si 77 lignes sont occupées, l'insertion d'une nouvelle instruction entraîne à nouveau la conversion d'un registre mémoire (R_8) en 7 lignes de mémoire programme, etc. Si les 210 lignes de mémoire programme sont occupées, le calculateur n'accepte plus d'instructions. Dans ce cas, toute nouvelle tentative d'insertion entraîne l'affichage d'une erreur 4 et la mémoire programme reste inchangée. (Rappel : pendant le chargement d'un programme long, les touches [g] [MEM] vous permettent de connaître à chaque instant l'état d'affectation des lignes de programme et des registres mémoire.)

RETOUR EN ARRIÈRE PAS A PAS DANS UN PROGRAMME

La touche [BST] (back step) permet de remonter dans un programme pour le mettre au point, tant en mode RUN qu'en mode PRGM. Lorsque vous appuyez sur [h] [BST], le calculateur revient d'une ligne en arrière. S'il est en mode RUN, il affiche la ligne précédente tant que vous maintenez la touche [BST] enfoncée. Dès que vous relâchez la touche, l'ancien contenu du regitre X est réaffiché. En mode PRGM, vous pouvez voir à chaque instant le numéro de ligne et le code de touche de l'instruction. Dans les deux modes, aucune instruction n'est exécutée.

Remarque

Quand votre HP-34C est au début de la mémoire programme (ligne 000), vous pouvez le positionner sur la dernière ligne de la mémoire programme occupée en appuyant sur [h] [BST]. Cette opération est particulièrement utile lorsque vous voulez vérifier la longueur d'un programme existant ou commencer à charger un nouveau programme ou sous-programme à la suite d'un programme ou sous-programme présent en mémoire.

Il vous reste une instruction [h] [PSE] à ajouter au programme du théorème de Pythagore : l'instruction [h] [PSE] qui suit l'instruction [x⇄y] chargée dans la ligne 004 de la mémoire programme. Si vous venez de charger une instruction [h] [PSE] dans la ligne 007 comme indiqué plus haut, le calculateur est positionné sur la ligne 007 de la mémoire programme. Vous pouvez revenir à la ligne 004 avec la touche [BST], puis insérer l'instruction [h] [PSE] dans la ligne 005. Pour commencer, mettez-vous en mode PRGM :

Appuyez sur :	Affichage	
	007– 25 74	Positionnement initial à la ligne 008.
[h] [BST]	**006– 51**	Retour d'une ligne en mémoire programme par pression simple de [h] [BST].

Continuez à revenir en arrière dans la mémoire avec la touche [BST] jusqu'à ce que le calculateur affiche la ligne 004 :

Appuyez sur :	Affichage
[h] [BST]	**005– 15 3**
[h] [BST]	**004– 21**

Comme vous voulez insérer l'instruction [h] [PSE] à la suite de l'instruction [x⇄y] qui occupe actuellement la ligne 004, vous devez d'abord vous positionner sur la ligne 004. L'instruction que vous allez entrer en appuyant sur les touches [h] [PSE] sera donc chargée, suivant le principe énoncé plus haut, dans la ligne 005 de la mémoire programme et toutes les instructions suivantes seront décalées d'une ligne.

Appuyez sur :	Affichage
[h] [PSE]	**005– 25 74**

Vous avez fini de modifier le programme du théorème de Pythagore. Vous pouvez maintenant examiner le contenu du registre X en différents points de l'exécution du programme. Votre nouveau programme se présente comme suit :

Appuyez sur :	Affichage
h LBL A	**001–25 13 11**
h PSE	**002– 25 74**
g x^2	**003– 15 3**
$x \gtrless y$	**004– 21**
h PSE	**005– 25 74**
g x^2	**006– 15 3**
+	**007– 51**
h PSE	**008– 25 74**
f $\sqrt{x}$	**009– 14 3**
h RTN	**010– 25 12**

Si vous voulez, vous pouvez utiliser les touches h SST en mode PRGM pour vérifier si le programme que vous avez dans votre calculateur correspond bien au programme ci-dessus.

EXÉCUTION DU PRORAMME MODIFIÉ

Pour exécuter le programme du théorème de Pythagore, il suffit de mettre le calculateur en mode RUN, d'introduire les valeurs des côtés a et b et d'appuyer sur A. Le calculateur affiche alors le contenu du registre X (côté b), élève le côté b au carré, permute les contenus des registres X et Y et réaffiche le contenu du registre X (côté **a** cette fois). Puis le calculateur élève le côté **a** au carré, additionne b^2 et a^2 et réaffiche le contenu de X une troisième fois ($a^2 + b^2$)... Enfin, il calcule la valeur de l'hypoténuse, retourne à la ligne 000 et s'arrête.

Par exemple, pour calculer l'hypoténuse d'un triangle rectangle dont les côtés sont égaux à 22 et 9 m :

Mettez le calculateur sur RUN.

Appuyez sur :	Affichage	
22 ENTER↑	**22,0000**	
9 A	**23,7697**	Après trois affichages du contenu du registre X pendant l'exécution du programme, le résultat final est affiché en mètres.

Repassez le même programme en remplaçant 22 m et 9 m par 73 km et 99 km.

(**Réponse :** 123,0041 km.)

SUPPRESSION D'INSTRUCTIONS

Pour modifier ou corriger un programme, il faut souvent supprimer une instruction. Cette opération consiste à éliminer l'instruction affichée par une pression sur la touche non enregistrable [h] [DEL] (delete) en mode PRGM. (En mode RUN, [DEL] n'a d'autre effet que d'annuler la touche préfixe [h].) Lorsque vous supprimez une instruction dans la mémoire programme avec la touche [DEL], toutes les instructions qui suivent en mémoire programme remontent d'une ligne. Le calculateur affiche alors la ligne qui précède celle qui contenait l'instruction supprimée.

Supposons que vous vouliez modifier le programme du théorème de Pythagore actuellement enregistré dans le calculateur de telle façon que le registre X ne soit affiché qu'une fois, pour la somme des carrés. Vous devez pour cela supprimer les instructions [h] [PSE] qui occupent les lignes 002 et 005 de la mémoire programme. Pour supprimer ces instructions, positionnez-vous sur ces lignes en faisant [h] [SST], [h] [BST] ou [GTO] [•] nnn, puis appuyez sur [h] [DEL]. Supprimez l'instruction [h] [PSE] de la ligne 002 en passant d'abord en mode PRGM puis :

Appuyez sur :	**Affichage**	
[GTO] [•] 002	**002– 25 74**	Affichage de la ligne 002.
[h] [DEL]	**001– 25 13 11**	Suppression de l'instruction de la ligne 002 et positionnement du calculateur sur la ligne 001.

Vérifiez en faisant [h] [SST] si l'instruction [h] [PSE] a été supprimée et si les instructions suivantes sont remontées d'une ligne.

Appuyez sur :	**Affichage**	
[h] [SST]	**002– 15 3**	L'instruction qui se trouvait dans la ligne 003 est montée dans la ligne 002 et toutes les instructions suivantes sont montées d'une ligne quand vous avez appuyé sur [h] [DEL].

Quand vous étiez sur la ligne 002 et que vous avez appuyé sur [h] [DEL], la mémoire a été modifiée :

Avant...	**Après...**	
001 [h] [LBL] [A]	001 [h] [LBL] [A]	← Une instruction a été supprimée ici.
002 [h] [PSE]	002 [g] [x^2]	
003 [g] [x^2]	003 [$x \gtrless y$]	
004 [$x \gtrless y$]	004 [h] [PSE]	
005 [h] [PSE]		

006 g x^2	→ 005 g x^2	Toutes ces instructions montent d'une ligne.
007 +	→ 006 +	
008 f $\sqrt{x}$	→ 007 f $\sqrt{x}$	
009 h PSE	→ 008 h PSE	
010 h RTN	→ 009 h RTN	

Pour supprimer l'instruction h PSE qui occupe actuellement la ligne 004, vous pouvez utiliser la touche SST et redescendre à cette ligne, puis effacer l'instruction en faisant h DEL.

Appuyez sur :	**Affichage**	
h SST	**003– 21**	
h SST	**004– 25 74**	
h DEL	**003– 21**	L'instruction h PSE est effacée de la ligne 004 et la ligne 003 est affichée. Les instructions suivantes montent d'une ligne en mémoire.

Si vous avez modifié le programme de la manière indiquée, le contenu du registre X est affiché une seule fois, juste après le calcul de la somme des carrés. Le programme calcule ensuite la valeur de l'hypoténuse et s'arrête.

Mettez le calculateur en mode RUN et exécutez le programme des triangles rectangles en prenant pour les côtés a et b 17 et 34 m. Après avoir affiché le registre X (somme des carrés = 1.445,0000 m), le calculateur exécute le reste du programme et s'arrête en affichant l'hypoténuse : 38,0132 m. Prenez ensuite pour les côtés a et b 550 et 740 perches. Après avoir affiché le registre X (somme des carrés = 850.100,0000 perches), le calculteur exécute le reste du programme et s'arrête en affichant l'hypoténuse : 922,0087 perches.

Pour remplacer une instruction par une autre, il suffit de vous positionner sur la ligne de programme désirée, d'appuyer sur h DEL pour supprimer l'instruction actuelle et d'entrer la nouvelle instruction.

Suppression d'instructions dans des programmes plus longs

Quand vous supprimez des instructions dans un programme occupant plus de 70 lignes, le processus de conversion automatique de registres mémoire en lignes de programme est inversé. Dans un programme de 78 lignes, par exemple, la suppression d'une instruction entraîne la reconversion immédiate des lignes de programme 78 à 85 en registre mémoire ($R_{.8}$) (voir « Affectation automatique de la mémoire partagée »).

Les fonctions de mise au point intégrées dans votre calculateur permettent un accès rapide et facile à un point quelconque du programme. Si un programme s'arrête par suite d'une erreur ou d'un dépassement de capacité, il suffit de passer en mode PRGM pour voir à l'affichage le numéro de ligne et le code de touche de l'opération en cause. Si vous soupçonnez tout un segment de votre programme, vous pouvez vous positionner sur ce segment par une opération [GTO] [•] nnn, puis en vous servant de la touche [SST], reprendre le programme ligne par ligne en mode RUN pour surveiller tout changement d'état dans le calculateur au fur et à mesure de l'exécution du programme.

PROBLÈMES

1. Seule l'opération [g] [→P] permet de calculer l'hypoténuse ou côté c d'un triangle rectangle dont les côtés a et b ont été introduits dans les registres X et Y. Nous allons donc remplacer les instructions [x^2], [$x⇄y$], [x^2], [+], [PSE] et [$\sqrt{x}$] dans le programme du théorème de Pythagore par l'instruction unique [g] [→P] :

   a. Faites [GTO] [•] nnn et [h] [SST] pour vérifier que le programme du théorème de Pythagore contient bien les instructions suivantes :

Appuyez sur :	Affichage	
[h] [LBL] [A]	001– 25 13 11	
[g] [x^2]	002– 15 3	Toutes ces instructions sont remplacées par une instruction
[$x⇄y$]	003– 21	
[g] [x^2]	004– 15 3	
[+]	005– 51	
[h] [PSE]	006– 25 74	
[f] [$\sqrt{x}$]	007– 14 3	
[h] [RTN]	008– 25 12	

   b. Appuyez sur [GTO] [•] nnn pour vous positionner sur la ligne 007 qui contient la dernière ligne à supprimer dans le programme.

   c. Effacez les instructions des lignes 007, 006, 005, 004, 003 et 002 à l'aide des touches [h] [DEL] en mode PRGM.

   d. Chargez l'instruction [g] [→P] dans la ligne 002.

   e. Vérifiez si le programme modifié correspond à celui-ci :

[h] [LBL] [A]	001– 25 13 11
[g] [→P]	002– 15 4
[h] [RTN]	004– 25 12

f. Mettez-vous en mode ▮▯ RUN et exécutez le programme du triangle rectangle pour calculer l'hypoténuse d'un triangle de 73 et 112 m de côtés.

(**Solution :** 133,6899 m.)

2. Le programme suivant calcule la valeur acquise par des placements, suivant la formule : FV = PV $(1 + i)^n$ où PV représente la valeur initiale du capital, FV sa valeur acquise, n le nombre de périodes et i le taux d'intérêt. En supposant que PV soit placé dans le registre Y et n dans le registre X, nous aurions le programme suivant pour un taux d'intérêt constant de 7,5 % par an (0,075 pour 1) :

[f] CLEAR [PRGM]
[h] [LBL] [B]
[f] [FIX] [2]
1
[•]
0
7
5
[x⇄y]
[h] [y^x]
[×]
[h] [RTN]

a. Chargez le programme dans le calculateur.

b. Calculez la valeur acquise pour 1 000 F placés pendant 5 ans (réponse : 1.435,63 F).
et pour 2 300 F placés pendant 4 ans (réponse : 3 071,58 F).

c. Modifiez le programme en faisant varier le taux d'intérêt de 7,5 à 8 % par an.

d. Calculez avec le nouveau programme la valeur acquise pour 500 F placés pendant 4 ans et pour 2 000 F placés pendant 10 ans. (Réponses : 680,24 et 4 317,85 F.)

3. Le programme suivant calcule la durée de chute d'un objet tombant d'une hauteur donnée (la résistance de l'air est considérée comme négligeable).

Après avoir introduit la hauteur h dans le registre X, on appuie sur la touche [B] et le calculateur affiche le temps de chute t en secondes. La formule employée est la suivante :

$$t = \sqrt{\frac{2h}{9.8\ \mathrm{m/s^2}}}$$

a. Annulez tous les anciens programmes, remettez le calculateur en mode d'affichage FIX 4 et chargez le programme suivant :

Appuyez sur :	Affichage	
f CLEAR PRGM	000–	
h LBL B	001–	25, 13, 11
2	002–	2
×	003–	61
9	004–	9
•	005–	73
8	006 –	8
÷	007 –	71
f $\sqrt{x}$	008 –	14 3
h RTN	009 –	25 12

b. Calculez la durée de chute d'une pierre tombant du haut de la Tour Eiffel, (300,51 m) et d'un dirigeable évoluant à une hauteur de 1 000 m. (Réponses : 7,8313 secondes ; 14,2857 secondes).

c. Modifiez le programme pour qu'il convienne à des hauteurs exprimées en pieds. La formule devient :

$$t = \sqrt{\frac{2\,h}{32{,}1740 \text{ ft/s}^2}}$$

d. Utilisez ce programme pour connaître la durée de chute d'une pierre depuis un barrage haut de 550 pieds et depuis le gratte-ciel du World Trade Center de New York, haut de 1 350 pieds (Réponses : 5,8471 secondes et 9,1607 secondes).

BRANCHEMENTS, DÉCISIONS ET INDICATEURS BINAIRES

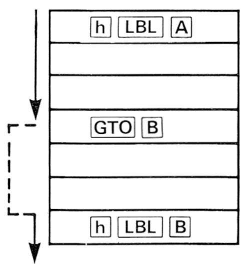
BRANCHEMENTS INCONDITIONNELS ET BOUCLES

Nous avons vu que [GTO] [•] nnn pouvait être utilisé depuis le clavier comme instruction non enregistrable pour accéder à une ligne déterminée en mémoire programme occupée. [GTO] peut également être utilisé comme instruction dans un programme, mais, pour être enregistrable, cette instruction doit être suivie d'un label [A] ou [B] ou 0 à 9 (ou bien d'un [I] comme nous verrons plus tard).

Lors de l'exécution d'un programme, si le calculateur rencontre une instruction [GTO] [B], par exemple, il s'arrête immédiatement et se met à rechercher le label spécifié dans l'instruction en explorant la mémoire programme de haut en bas. Dès qu'il trouve une instruction [h] [LBL] [B], il reprend l'exécution du programme.

L'instruction [GTO] suivie d'un label permet donc de se brancher en un point quelconque du programme.

Branchement à l'instruction [h] [LBL] [B] suivante.



Ce cas d'utilisation de l'instruction [GTO] est appelé **branchement inconditionnel.** Le branchement s'effectue toujours de l'instruction [GTO] au label spécifié. (Nous verrons plus loin qu'un [GTO] peut être soumis à une condition et déterminer un branchement **conditionnel** qui dépend du résultat d'un test).

Un branchement peut être utilisé pour créer une «boucle» dans un programme. Le programme suivant, par exemple, calcule et affiche les racines carrées de nombres entiers successifs à partir de 1. Le calculateur continue le calcul jusqu'à ce que l'on appuie sur [R/S] pour l'arrêter ou jusqu'à un dépassement de capacité.

Pour enregistrer ce programme, placez le commutateur PRGM-RUN sur PRGM. Videz la mémoire programme et retournez à la ligne 000 en appuyant sur [f] CLEAR [PRGM].

Appuyez sur	Affichage	
[h] [LBL] [A]	001–25, 13, 11	
0	002– 0	
[STO] 1	003– 23 1	
[h] [LBL] 0	004–25, 13, 0	
1	005– 1	
[STO] [+] 1	006–23, 51, 1	1 est additionné au contenu de R_1.
[RCL] 1	007– 24 1	Rappel du contenu de R_1.
[h] [PSE]	008– 25 74	Affichage du nombre actuel.
[f] [$\sqrt{x}$]	009– 14 3	
[h] [PSE]	010– 25 74	Affichage de la racine carrée du nombre actuel.
[GTO] 0	011– 22 0	Retour à [h] [LBL] 0.
[h] [RTN]	012– 25 12	

Pour exécuter le programme, placez le commutateur PRGM-RUN sur la position RUN et appuyez sur [A]. Le programme commence à afficher la liste des entiers et de leurs racines carrées et continue jusqu'à ce que vous appuyiez sur [R/S] ou jusqu'à ce que la capacité du calculateur soit dépassée.

Le principe de cette opération est le suivant : quand vous appuyez sur [A], le calculateur parcourt la mémoire programme à la recherche d'une instruction [h] [LBL] [A] marquant le début du programme. Il exécute alors cette instruction, ainsi que les suivantes en séquence jusqu'à la ligne 011 qui contient l'instruction [GTO] 0. Il part alors à la recherche de l'instruction [LBL] 0, qu'il va trouver dans la ligne 004 et le programme reprend à cette instruction [LBL] 0. (Notez que l'adresse qui suit l'instruction [GTO] dans un programme n'est pas un numéro de ligne mais un label). Chaque fois que le calculateur exécute l'instruction [GTO] 0 à la ligne 011, le programme reprend donc à l'instruction [LBL] 0 de la ligne 004 et le calculateur tourne ainsi dans une « boucle », ajoutant 1 à chaque tour au contenu du registre mémoire R_1 et affichant le nombre obtenu et sa racine carrée.

Les boucles sont couramment utilisées et très utiles en programmation, car elles offrent la possibilité de calculer et de modifier automatiquement des données et de refaire indéfiniment des calculs sur ces données.

Les branchements inconditionnels permettent d'établir des boucles comme nous venons de le voir, ou de se brancher d'un point quelconque d'un programme vers un autre, défini par un label. A la rencontre d'un [GTO], le programme en cours d'exécution s'arrête, laissant place à une exploration systématique de la mémoire programme pour rechercher le label spécifié et reprend à l'instruction contenant ce label.

En mode RUN, [GTO] peut aussi servir à rechercher un label sans exécuter le programme en mémoire. Si vous faites [GTO] ([A], [B] ou n) au clavier, le calculateur va au label spécifié et s'arrête. Cette manœuvre est utile si vous voulez examiner ou mettre au point des lignes de mémoire après un certain label, sans les exécuter.

Problèmes

1. Le programme suivant calcule le carré du nombre contenu dans le registre R_1 et effectue une pause pour afficher le résultat après chaque évolution. Enregistrez le programme, commutateur sur PRGM, et exécutez-le plusieurs fois en mode RUN. (La réponse sera toujours 1.000). Modifiez ensuite le programme en remplaçant l'instruction [h] [RTN] à la ligne 010 par une instruction [GTO] 1 et en insérant une instruction [h] [LBL] 1 à la suite de l'instruction [STO] 1 ligne 003. Vous établissez ainsi une boucle de tabulation des nombres entiers de 1 en 1 et de leurs carrés. Pour charger le programme original avant modification, mettez le commutateur PRGM-—RUN sur PRGM, puis :

Appuyez sur	Affichage
[f] CLEAR [PRGM]	000–
[h] [LBL] [B]	001–25, 13, 12
0	002– 0
[STO] 1	003– 23 1
1	00 – 1
[STO] [+] 1	005–23, 51, 1
[RCL] 1	006– 24 1
[h] [PSE]	007– 25 74
[g] [x^2]	008– 15 3
[h] [PSE]	009– 25 74
[h] [RTN]	010– 25 12

Mettez le commutateur PRGM-RUN sur RUN et vérifiez le programme sous sa forme originale. Après avoir introduit les modifications suggérées, repassez le programme pour créer une table de carrés.

2. A l'aide de l'organigramme de la page suivante, créez un programme de tabulation des valeurs acquises FV par un capital PV à intérêts composés, en accroissant la durée d'une période (incrément) à chaque exécution. La formule employée est :

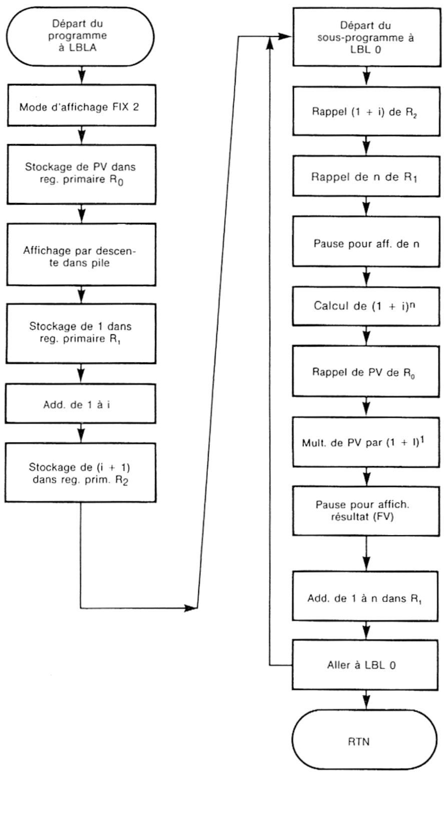
$$FV = PV\ (1 + i)^n$$

FV étant la valeur acquise, ou future
PV la valeur présente
i le taux d'intérêt sous forme décimale
n le nombre de périodes (années, en général)

On suppose que i a été introduit dans le registre Y de la pile et PV dans le registre X avant l'exécution du programme.

Après avoir rédigé et chargé le programme, exécutez-le avec un taux d'intérêt de 6 % (vous posez .06) et un placement PV de 1 000 F (Réponses : 1 an = 1060 ; 2 ans = 1 123,60 ; 3 ans = 1 191, 02 etc.). Le programme calcule :

$FV = 1\,000\,(1 + 0.06)^n$
avec un n croissant de 1 en 1 jusqu'à ce que vous appuyiez sur R/S ou n'importe quelle autre touche ou jusqu'à ce qu'il y ait dépassement de capacité. Essayez ce programme pour d'autres valeurs de i et de PV.



Solution proposée pour le 2e problème :

2. h LBL A	
f FIX 2	Mode d'affichage préconisé.
STO 0	Mise en mémoire de PV.
g R↓	Affichage de i.
1	
STO 1	Mise en mémoire de la valeur initiale (1) de n.
+	Addition de 1 à i.
STO 2	Mise en mémoire de (1 + i)
h LBL 0	
RCL 2	Rappel de (1 + i).
RCL 1	Rappel de n
h PSE	Affichage de n
h y^x	Calcul de $(1 + i)^n$
RCL 0	Rappel du PV stocké dans R_0.
×	Calcul du nouveau FV.
h PSE	Affichage du résultat (FV)
1	
STO + 1	Addition de i à n dans R_1.
GTO 0	Branchement inconditionnel.
h RTN	Fin de programme

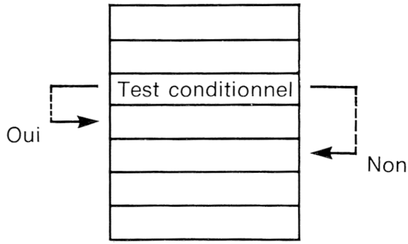
TEST ET BRANCHEMENTS CONDITIONNELS

Votre HP-34C dispose des moyens nécessaires pour prendre des décisions au milieu d'un programme : les opérations conditionnelles. Celles-ci sont incorporées dans le programme sous forme d'instructions.

Il y a 9 tests de comparaison :

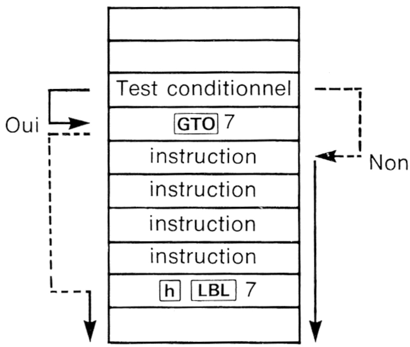
f $x \leq y$	la valeur du registre X est-elle égale ou inférieure à celle du registre Y ?
f $x > y$	la valeur du registre X est-elle supérieure à celle du registre Y ?
f $x \neq y$	la valeur du registre X est-elle différente de celle du registre Y ?
f $x = y$	la valeur du registre X est-elle égale à celle du registre Y ?
g $x < 0$	la valeur du registre X est-elle inférieure à 0 ?
g $x > 0$	la valeur du registre X est-elle supérieure à 0 ?
g $x = 0$	la valeur du registre X est-elle égale à 0 ?
g $x \neq 0$	la valeur du registre X est-elle différente de 0 ?
h F? **n**	l'indicateur binaire n est-il positionné ? (voir détails plus loin).

Chaque test de comparaison pose une question au calculateur. Si la réponse est oui, le calculateur passe à la ligne suivante en mémoire progamme. Si la réponse est non, il saute la ligne suivante. Exemple :



Après avoir fait le test de comparaison, le calculateur exécute donc l'instruction suivante si la condition est vérifiée.

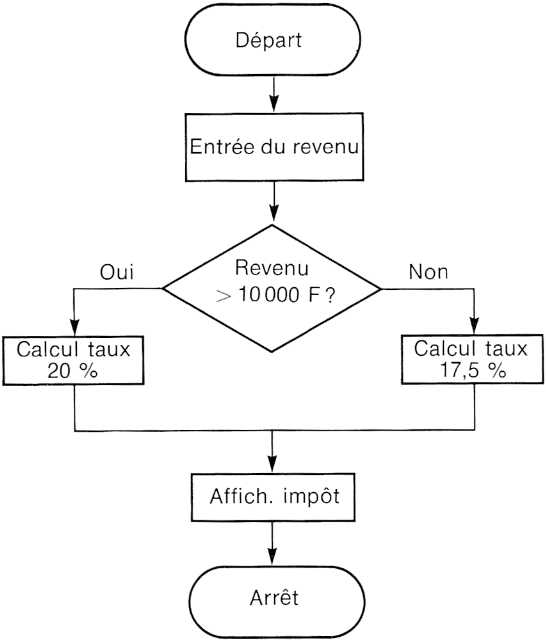
N'importe quelle instruction peut suivre le test de comparaison, mais la plus fréquente est l'instruction $\boxed{\text{GTO}}$ qui branche l'exécution du programme vers une autre ligne de la mémoire programme si la condition est vérifiée.



Exemple

Un comptable veut calculer le montant des impôts à payer par chacun de ses clients sachant que ceux qui gagnent moins de 10 000 F par an paieront 17,5 % et ceux qui gagnent plus de 10 000 F par an paieront 20 % d'impôts. Il veut utiliser un programme avec des branchements conditionnels.

L'organigramme du programme aura l'allure suivante :



Pour introduire le programme, mettez le commutateur PRGM-RUN sur PRGM.

Appuyez sur	Affichage	
f CLEAR PRGM	000–	
h LBL A	001–25 13 11	
EEX	002– 33	Introduction du montant de 10 000 F dans le registre Y.
4	003– 4	
x⇄y	004– 21	
f x>y	005– 14 51	Si le revenu est supérieur à 10 000 F branchement au segment de programme identifié par le label B.
GTO B	006– 22 12	

1	**007– 1**	Taux d'imposition pour ce segment de programme : 17,5 %
7	**008– 7**	
[•]	**009– 73**	
5	**010– 5**	
[GTO] 1	**011– 22 1**	
[h] [LBL] [B]	**012–25 13 12**	
2	**013– 2**	Taux d'imposition pour ce segment de programme : 20 %
0	**014– 0**	
[h] [LBL] 1	**015–25, 13, 1**	
[h] [%]	**016– 25 41**	
[h] [RTN]	**017– 25 12**	

Calculez les impôts à payer sur des revenus de 15 000 et 7 500 F.

Mettez le commutateur PRGM-RUN sur RUN.

Appuyez sur	**Affichage**	
15 000 [A]	**3.000,00**	Montant de l'impôt.
7 500 [A]	**1.312,50**	Montant de l'impôt.

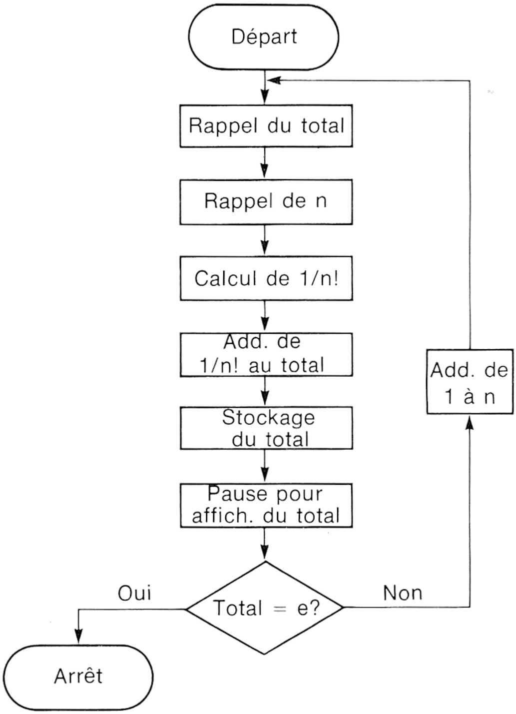
Pour calculer les impôts des autres clients, il ne reste plus qu'à introduire le montant de leur revenu et appuyer sur [A]. Le calculateur détermine automatiquement la tranche d'imposition du client et calcule le montant de l'impôt.

Une décision logique peut aussi intervenir dans une boucle. Les boucles que nous avons vues jusqu'à présent étaient des boucles « infinies » dans lesquelles le calculateur restait enfermé, réexécutant indéfiniment les mêmes instructions jusqu'à dépassement de ses capacités ou jusqu'à une intervention externe par pression de la touche [R/S] ou d'une autre touche.

Vous pouvez avoir recours à une décision logique pour sortir d'une boucle. Une instruction de branchement conditionnel fera sortir le calculateur de la boucle après un nombre déterminé de tours de boucle ou après avoir atteint le résultat voulu.

Exemple

Vous savez que votre HP-34C contient une valeur de e, qui est la base des logarithmes népériens. (Vous pouvez afficher cette valeur en appuyant sur 1 [g] [e^x].) Le programme suivant montre qu'on peut utiliser 1/n! pour vérifier que la série e = 1/0! + 1/1! + 1/2! + ... + 1/n! se rapproche de la valeur de e. Après chaque tour de boucle, la dernière approximation est affichée et comparée à la valeur de e enregistrée dans le calculateur. Si les deux valeurs concordent, le calculateur sort de la boucle et arrête le programme.



Pour enregistrer le programme, mettez le commutateur PRGM-RUN sur PRGM ▥▥■.

Appuyez sur :	**Affichage**
[f] CLEAR [PRGM]	000–
[h] [LBL] [A]	001– 25 13 11
[RCL] 1	002– 24 1
[RCL] 0	003– 24 0
[h] [x!]	004– 25 1
[h] [1/x]	005– 25 2
[+]	006– 51

[f] [FIX] 9	007–	14	11	9
[STO] 1	008–		23	1
[h] [PSE]	009–		25	74
1	010–			1
[g] [e^x]	011–		15	1
[f] [$x=y$]	012–		14	71
[h] [RTN]	013–		25	12
1	014–			1
[STO] [+] 0	015–	23	51	0
[GTO] [A]	016–		22	11

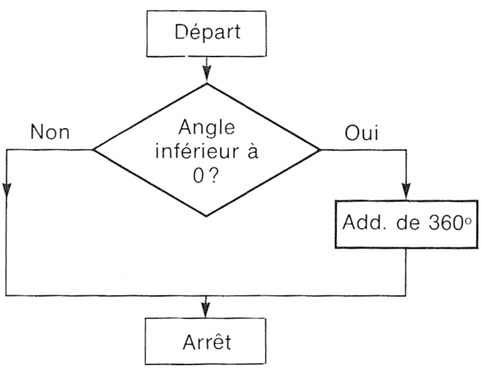
Mettez le commutateur PRGM/RUN sur RUN.
Effacez les registres mémoire, puis appuyez sur [A] pour exécuter le programme.

Appuyez sur :	**Affichage**	
[f] CLEAR [REG]	**1.312,50**	Remise à 0 de tous les registres mémoire (la valeur affichée est celle de l'exemple précédent).
[A]	**2,718281828**	

Vous voyez que le calculateur tourne dans la boucle jusqu'à ce qu'il y ait concordance entre l'approximation de e et la valeur de e enregistrée dans le calculateur. Quand la condition [$x=y$] de la ligne 012 est enfin vérifiée, le calculateur sort de la boucle.

Problèmes

1. Écrivez un programme qui teste l'angle et, si l'angle est négatif, le convertisse en un angle positif équivalent. Utilisez pour cela un test de comparaison et si l'angle est négatif, ajoutez-lui 360° pour le rendre positif. L'organigramme ci-dessous vous aidera à écrire le programme.

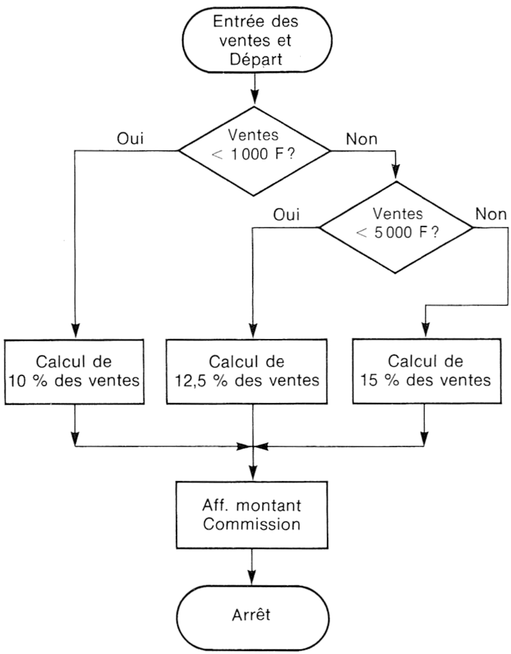


2. Aidez-vous de cet organigramme pour écrire le programme qui permet à un représentant de calculer ses commissions d'après le barème suivant :

   10 % des ventes jusqu'à 1 000
   12,5 % des ventes de 1 000 à 5 000
   15 % des ventes au-delà de 5 000.

   Le programme devra afficher le montant des ventes et celui des commissions.

Chargez votre programme et exécutez-le. Pour des ventes de 500 F, 1 000 F, 1 500 F, 5 000 F et 6 000 F, vous devez obtenir 50 F, 125 F, 187,50 F, 625 F et 900 F de commissions.



Solutions

1. **Appuyez sur :** **Affichage**

Appuyez sur	Affichage
f CLEAR PRGM	000–
h LBL A	001– 25 13 11
g $x<0$	002– 15 41
GTO 0	003– 22 0
h RTN	004– 25 12
h LBL 0	005– 25 13 0
3	006– 3
6	007– 6
0	008– 0
+	009– 51
h RTN	010– 25 12

Enregistrez le programme, mettez le commutateur PRGM-RUN sur RUN et le mode d'affichage sur FIX 2. Introduisez l'angle et appuyez sur A.

2. **Appuyez sur :** **Affichage**

Appuyez sur	Affichage
f CLEAR PRGM	000–
h LBL A	001– 25 13 11
EEX	002– 33
3	003– 3
f $x>y$	004– 14 51
GTO 0	005– 22 0
5	006– 5
×	007– 61
$x \rightleftarrows y$	008– 21
f $x \leq y$	009– 14 41
GTO 1	010– 22 1
1	011– 1
5	012– 5
h %	013– 25 41
h RTN	014– 25 12
h LBL 1	015– 25 13 1
1	016– 1
2	017– 2
•	018– 73
5	019– 5
h %	020– 25 41
h RTN	021– 25 12
h LBL 0	022–25, 13, 00
$x \rightleftarrows y$	023– 21
1	024– 1

0	025–	0
h %	026–	25 41
h RTN	027–	25 12

Enregistrez le programme. Mettez le commutateur PRGM-RUN sur RUN et le mode d'affichage sur FIX 4. Entrez les nombres et appuyez sur A.

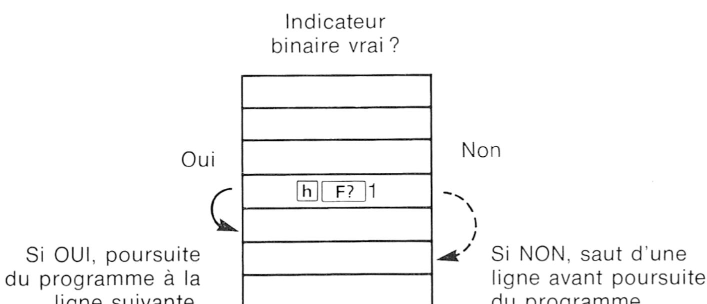
INDICATEURS BINAIRES

Il existe un autre moyen de test à l'intérieur d'un programme : les indicateurs binaires. L'indicateur binaire est un dispositif de la mémoire qui peut être positionné à 1 (vrai) ou à 0 (faux). Le programme en cours d'exécution peut donc tester l'indicateur binaire et prendre une décision en fonction de son positionnement.

Le HP-34C dispose de quatre indicateurs binaires : F0, F1, F2, F3. Pour positionner un indicateur binaire (à 1), on utilise l'instruction SF (set flag) suivie de la touche numérique 0, 1, 2 ou 3 correspondant à l'indicateur binaire concerné. Pour positionner l'indicateur 3, par exemple, appuyez sur h SF 3.

Les indicateurs binaires sont remis à 0 par l'instruction CF (clear flag), suivie de la touche numérique appropriée. Pour remettre à zéro l'indicateur binaire 3, appuyez sur : h CF 3.

Dans le test des indicateurs binaires, la question est posée par l'instruction F? (est-ce que l'indicateur est vrai ?) suivie de la touche numérique 0, 1, 2, 3 spécifiant l'indicateur à tester. Si l'indicateur est positionné (condition VRAIE vérifiée) au moment de l'instruction h F? n, le calculateur exécute la ligne suivante. S'il est à zéro, le calculateur saute la ligne suivante et poursuit l'exécution du programme.



Un indicateur binaire positionné par une instruction [h] [SF] n reste positionné jusqu'à ce qu'il soit remis à zéro par l'une des conditions suivantes :

1. exécution d'une instruction [h] [CF] n
2. extinction du calculateur

Utilisation des indicateurs binaires

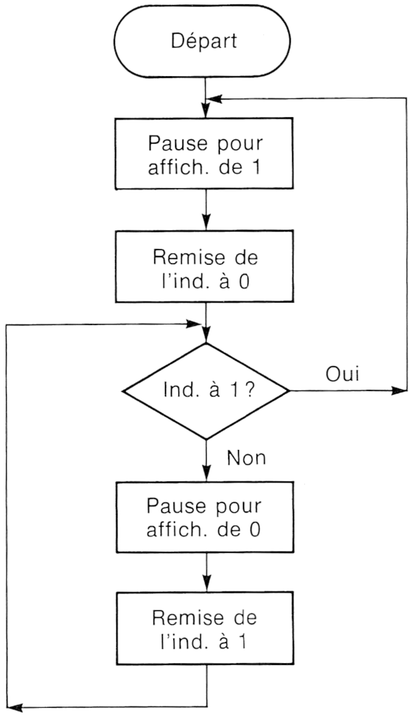
Tout comme les tests de comparaison de x par rapport à y et de x par rapport à 0, les indicateurs binaires vous permettent de sauter ou d'exécuter des lignes déterminées en mémoire programme. Mais, alors que les tests x/y et x/0 impliquent des comparaisons, les indicateurs binaires indiquent simplement au calculateur si une opération ou un type d'opération a été exécutée ou non.

---

* Notez qu'un indicateur binaire positionné par une instruction [h] [SF] n ne peut être remis à zéro par [f] CLEAR [PRGM], mais que [STO] [ENTER] (auto-test) remet tous les indicateurs binaires à zéro.

Exemple

Le programme suivant contient une boucle infinie illustrant le fonctionnement d'un indicateur binaire. Le programme affiche tour à tour des 1 et des 0 en changeant l'état de l'indicateur, modifiant ainsi le résultat du test dans la ligne 006 à chaque itération. Ce programme pourrait être représenté par l'organigramme suivant :



Pour ce programme, il faut avoir au préalable mis 0 dans le registre R_0 et le nombre 1.111111111 dans le registre R_1.

Mettez le commutateur PRGM-RUN sur PRGM ▮.

Appuyez sur :	Affichage	
f CLEAR PRGM	000–	
h LBL A	001–25 13 11	
RCL 1	002– 24 1	Rappel et affichage des 1 placés dans le registre R_1.
h PSE	003– 25 74	
h CF 0	004–25, 61, 0	Remise à zéro de l'indicateur F0.
h LBL B	005–25 13 12	
h F? 0	006–25 71 0	Test de l'indicateur F0.
GTO A	007– 22 11	Si positionné (vrai), saut à LBL A.
RCL 0	008– 24 0	Sinon, rappel et affichage des 0 placés dans le registre R_0, positionnement de l'indicateur F0 et saut à LBL B
h PSE	009– 25 74	
h SF 0	010–25 51 0	
GTO B	011– 22 12	
h RTN	012– 25 12	

Mettez le commutateur sur ▮▯ RUN, chargez les registres mémoire R_0 et R_1, puis exécutez le programme.

Appuyez sur :	Affichage	
f FIX 9	0,000000000	
0 STO 0	0,000000000	
1.111111111	1,111111111	
STO 1	1,111111111	Des 1 ou des 0 exclusivement.
A	1,111111111	
	0,000000000	

Pour arrêter l'exécution du programme, appuyez sur R/S (ou sur toute autre touche).

Déroulement du programme

Une fois que vous avez initialisé le programme en mettant 0 dans le registre R_0 et uniquement des 1 dans le registre R_1, vous lancez le programme en appuyant sur A. A la suite des instructions RCL 1 et h PSE dans les lignes 002 et 003, le calculateur effectue une pause pour afficher les 1 du registre R_1. L'instruction h CF 0 dans la ligne 004 remet l'indicateur F0 à zéro. (Cet indicateur étant déjà à zéro au lancement du programme, il garde le même état.)

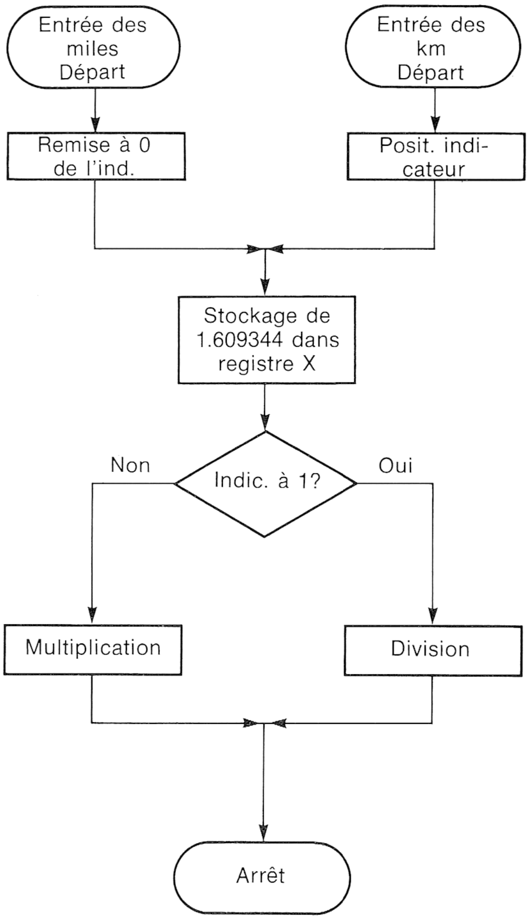
Après LBL A, ne rencontrant pas de RTN, le programme poursuit jusqu'à LBL B dans la ligne 005 et effectue le test h F? 0 dans la ligne 006. L'instruction h F? 0 déclenche un test de positionnement de l'indicateur F0 (F0 positionné ?). Celui-ci ayant déjà été remis à zéro, la réponse au test est NON, le programme saute donc une ligne et continue à l'instruction [RCL] 0, ligne 008. Après [h] [PSE] dans la ligne 009, il effectue une pause pour afficher les zéros du registre R_0. L'instruction [h] [SF] 0, ligne 010, positionne l'indicateur F0 et l'instruction [GTO] [B] ligne 011 branche le programme sur [LBL] [B].

L'indicateur F0 étant maintenant positionné, la réponse au test [h] [F?] 0 (F0 vrai ?) est OUI et le calculateur exécute [GTO] [A], ligne 008, qui est la ligne qui suit le test. Après une nouvelle pause pour affichage des uns, l'indicateur est remis à zéro et le programme continue indéfiniment, affichant tour à tour des 1 et des 0 jusqu'à ce que vous l'arrêtiez par une pression de touche au clavier.

Problème

Un mile est égal à 1,609344 km. A l'aide de l'organigramme suivant, créez et chargez un programme qui calculera les distances en miles (segment de programme à identifier par [LBL] [A]) et en kilomètres (segment à identifier par [LBL] [B]). Servez-vous d'un indicateur binaire pour faire effectuer au programme soit une multiplication soit une division d'une unité de mesure dans l'autre. (Attention : [h] [1/x] [×] donne le même résultat que [÷].)

Mettez le calculateur sur le mode d'affichage FIX 4. Puis exécutez le programme pour convertir 26 miles en kilomètres et 1 500 m (1,5 km) en miles. (Réponses : 41,8429 km ; 0,9321 miles.)

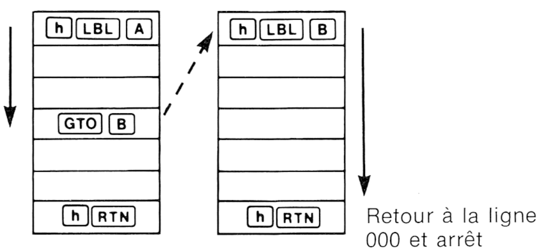


SOUS-PROGRAMMES

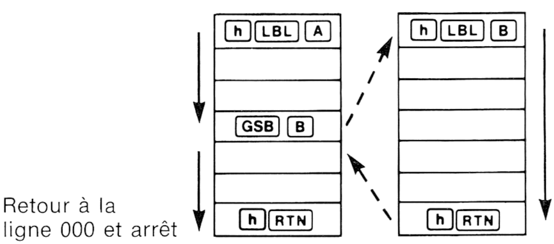
Il n'est pas rare que des séquences d'instructions réapparaissent plusieurs fois à l'intérieur d'un même programme. Ces séquences d'instructions récurrentes peuvent être exécutées sous forme de sous-programme. Un sous-programme est appelé par l'instruction GSB (go to subroutine), suivie d'une adresse de label (A, B ou 0 à 9), ou bien, comme nous allons le voir plus loin, par GSB I.

L'instruction GSB branche le programme sur le sous-programme spécifié par l'adresse du label, de la même manière qu'une instruction GTO. Mais contrairement à ce qui se passe avec cette dernière, le retour du programme à la rencontre d'un RTN (return) s'effectue à l'instruction qui suit GSB. L'exécution continue alors en séquence vers le bas de la mémoire.

Branchement



Sous-programme



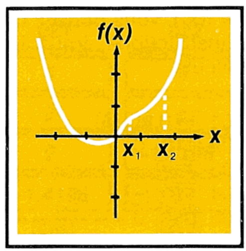
L'illustration ci-dessus met en évidence la différence fondamentale entre un [GTO] et un [GSB]. Dans le schéma de gauche, la pression de [A] démarre le programme qui exécute les instructions en séquence de haut en bas de la mémoire. A la rencontre d'une instruction [GTO] [B], il cherche le [LBL] [B] suivant et continue à partir de cette instruction jusqu'à la rencontre d'un [RTN]. Après exécution du [RTN], il retourne à la ligne 000 et s'arrête.

Cependant, si le programme en cours rencontre une instruction [GSB] [B] (go to subroutine B), comme dans le schéma de droite, il descend dans la mémoire à la recherche de l'instruction [LBL] [B] suivante et repart. A la rencontre de [RTN] (return), un nouveau branchement a lieu, mais cette fois vers la ligne qui suit l'instruction ayant appelé le sous-programme ([GSB] [B]), et l'exécution reprend.

Ainsi, la seule différence entre un branchement à un sous-programme et un branchement normal est le point de retour après un [RTN]. Avec [GTO], le programme retourne à la ligne 000 et s'arrête ; avec [GSB], le programme repart à une instruction antérieure et continue jusqu'à la rencontre d'un nouvel [RTN] ou jusqu'à l'intervention d'une pression de touche ([R/S]).

Exemple :

Soit à calculer la pente moyenne de la courbe de f(x) entre x_1 et x_2 d'après la formule :
$f(x) = x^2 - \ln(x^2 + e^{-x})$



Solution :

La pente moyenne de f(x) est donnée par :

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$= \frac{\left[x_2^2 - \ln(x_2^2 + e^{-x_2})\right] - \left[x_1^2 - \ln(x_1^2 + e^{-x_1})\right]}{x_2 - x_1}$$

Notez que la solution de l'expression $x^2 - \ln(x^2 - e^{-x})$ se calcule en deux temps.

Le programme suivant vous permet d'introduire les valeurs de x_1 et x_2 et de calculer la pente moyenne en appuyant sur A.

h LBL A		x⇆y
STO 0		ENTER↑
x⇆y		CHS
STO − 0		g e^x
ENTER↑		h LST x
CHS		g x^2
g e^x	Calcul de $f(x_2)$ →	+
h LST x		f LN
g x^2		CHS
+		x⇆y
f LN	Ces segments de programme sont identiques	g x^2
CHS		+
x⇆y		+
g x^2	← Calcul de $f(x_1)$	RCL 0
+		÷
CHS		h RTN

Le segment de programme utilisé pour le calcul de $f(x_2)$ ayant un grand tronçon commun avec le segment de programme utilisé pour le calcul de $f(x_1)$, vous pouvez créer un sous-programme pour exécuter ce groupe d'instructions. Vous ferez appel à ce sous-programme pour les deux calculs.

001 h LBL A	013 h LBL 0
002 STO 0	014 ENTER↑
003 x⇆y	015 CHS
004 STO − 0	016 g e^x
005 GSB 0	017 h LST x
006 CHS	018 g x^2
007 x⇆y	019 +
008 GSB 0	020 f LN
009 +	021 CHS
010 RCL 0	022 x⇆y
011 ÷	023 g x^2
012 h RTN	024 +
	025 h RTN

Le programme ayant été modifié, si vous appuyez sur A après avoir introduit x_1 dans le registre Y et x_2 dans le registre X, l'exécution démarre à l'instruction h LBL A, ligne 001. A la rencontre de l'instruction GSB 0 dans la ligne 005, le programme se branche sur l'instruction h LBL 0, ligne 013 et calcule la valeur de $f(x_1)$. Prenons x_1 égal à 2 et x_2 égal à 3. Voici ce qui se passe dans la pile au moment du calcul de f(x) :

Appuyez sur : Registres de la pile

	001	002	003	004	005
T					
Z					
Y	2	2	3	3	3
X	3	3	2	2	2
	h LBL A	STO 0	$x \rightleftarrows y$	STO − 0	GSB 0
	(x_2)	(x_2)	(x_1)	(x_1)	(x_1)

T					3
Z		3	3	3	2
Y	3	2	2	2	0.1353
X	2	2	−2	0.1353	−2
	h LBL 0	ENTER	CHS	g e^x	h LST x
	(x_1)	(x_1)	$(-x_1)$	(e^{-x_1})	$(-x_1)$

	018	019	020	021	022
T	3	3	3	3	3
Z	2	3	3	3	3
Y	0.1353	2	2	2	−1.4196
X	4	4.1353	1.4196	−1.4196	2
	g x^2	+	f LN	CHS	$x \rightleftarrows y$
	(x_1^2)	$(x_1^2 + e^{-x_1})$	$(\ln(x_1^2 + e^{-x_1}))$	$(-\ln(x_1^2 + e^{-x_1}))$	

	023	024	025
T	3	3	3
Z	3	3	3
Y	−1.4196	3	3
X	4	2.5804	2.5804
	g x^2	+	h RTN
	(x_1^2)	$(f(x_1))$	retour au programme

A la ligne 025, le calculateur retourne au programme principal qui exécute l'instruction qui suit la dernière instruction GSB. A la rencontre de l'instruction [GSB] 0 dans la ligne 008, le programme saute à l'instruction [h] [LBL] 0, ligne 013.

Suite du programme :

Appuyez sur : Registres de la pile

	006	007	008	013	014
T	3	3	3	3	3
Z	3	3	3	3	-2.5804
Y	3	-2.5804	-2.5804	-2.5804	3
X	-2.5804	3	3	3	3
	[CHS]	[x⇄y]	[GSB] 0	[h] [LBL] 0	[ENTER↑]
	$(-f(x_1))$	$(-f(x_1)$ dans la pile		Début de sous-programme	

	015	016	017	018	019
T	3	3	-2.5804	-2.5804	-2.5804
Z	-2.5804	-2.5804	3	3	-2
Y	3	3	0.0498	0.0498	3
X	-3	0.0498	-3	9	9.0498
	[CHS]	[g] [e^x]	[h] [LST x]	[g] [x^2]	[+]
	$(-x_2)$	(e^{-x_2})	$(-x_2)$	(x_2^2)	$(x_2^2 + e^{-x_2})$

	020	021	022	023	024
T	-2.5804	-2.5804	-2.5804	-2.5804	-2.5804
Z	-2.5804	-2.5804	-2.5804	-2.5804	-2.5804
Y	3	3	-2.2027	-2.2027	-2.5804
X	2.2027	-2.2027	3	9	6.7973
	[f] [LN]	[CHS]	[x⇄y]	[g] [x^2]	[+]
	$(\ln(x_2^2 + e^{-x_2}))$	$(-\ln(x_2^2 + e^{-x_2}))$	(x_2)	(x_2^2)	$(f(x_2)$

	025
T	-2.5804
Z	-2.5804
Y	-2.5804
X	6.7973
	[h] [RTN]
	Retour au programme

Après un second passage du sous-programme sous [LBL] 0 pour calculer f(x_2), le calculateur arrive à l'instruction [h] [RTN], ligne 025 et retourne à l'instruction qui suit la dernière instruction [GSB] 0 dans le programme principal. f(x_2) est dans le registre X, − f(x_1) dans les registres Y, Z et T.

Appuyez sur : Registres de la pile

	009	010	011	012
T	-2.5804	-2.5804	-2.5804	-2.5804
Z	-2.5804	-2.5804	-2.5804	-2.5804
Y	-2.5804	4.2168	-2.5804	-2.5804
X	4.2168	1	4.2168	4.2168
	[+]	[RCL] 0	[÷]	[h] [RTN]
	$(f(x_2) - f(x_1)$	$(x_2 - x_1)$	$(f(x_2) - f(x_1) \div (x_2 - x_1))$	Fin de programme

Introduisez le programme et essayez les solutions suivantes. Mettez le commutateur PRGM RUN sur PRGM.

Appuyez sur :	**Affichage**
[f] CLEAR [PRGM]	000–
[h] [LBL] [A]	001– 25 13 11
[STO] 0	002– 23 0
[x⇄y]	003– 21
[STO] [−] 0	004– 23 41 0
[GSB] 0	005– 13 0
[CHS]	006– 32
[x⇄y]	007– 21
[GSB] 0	008– 13 0

[+]	009–		51
[RCL] 0	010–	24	0
[÷]	011–		71
[h] [RTN]	012–	25	12
[h] [LBL] 0	013–	25 13	0
[ENTER↑]	014–		31
[CHS]	015–		32
[g] [e^x]	016–	15	1
[h] [LST x]	017–	25	0
[g] [x^2]	018–	15	3
[+]	019–		51
[f] [LN]	020–	14	1
[CHS]	021–		32
[$x \leftrightarrows y$]	022–		21
[g] [x^2]	023–	15	3
[+]	024–		51
[h] [RTN]	025–	25	12

Mettez le commutateur PRGM-RUN sur RUN. Calculez la pente moyenne de f(x) entre les couples de points suivants : (0, 0.5) ; (0.55, 1.15) ; (1.25, 1.75).

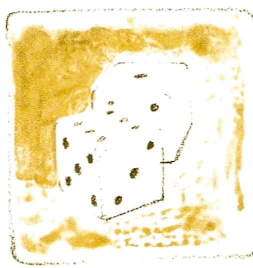
Réponses : 0,8097 ; 0,6623 ; 1,8804.

UTILISATION DES SOUS-PROGRAMMES

Les sous-programmes vous offrent une grande souplesse en programmation. Un sous-programme peut contenir une boucle ou peut être exécuté au milieu d'une boucle. Il peut être utilisé à la fois comme sous-programme et comme une partie de programme : cet emploi est courant et permet de gagner de la place.

Exemple

Le programme ci-dessous simule une partie de dés. Le calculateur simule d'abord le jet du premier dé et affiche le résultat (nombre entier de 1 à 6), puis il simule le jet du second dé et affiche le résultat (nombre entier de 1 à 6), enfin il additionne les deux résultats pour donner le total.



Le cœur du programme est un générateur de nombres aléatoires (ou plutôt pseudo-aléatoires), d'abord exécuté comme sous-programme puis comme le reste du programme. Après avoir introduit le premier chiffre ou « semence », vous appuyez sur [A] et le sous-programme [h] [LBL] 2 exécuté en tant que sous-programme calcule et affiche le chiffre du premier dé. Celui du second dé est calculé ensuite par le même sous-programme mais considéré cette fois comme partie intégrante du programme. Le programme utilise ensuite le résultat obtenu comme nouvelle « semence » pour une série de jets de dés.

Pour introduire le programme, mettez le commutateur sur PRGM.

Appuyez sur :	Affichage	
[f] CLEAR [PRGM]	000–	
[GTO] 1	001– 22 1	
[h] [LBL] [A]	002– 25 13 11	
[STO] 0	003– 23 0	
[h] [LBL] 1	004– 25 13 1	
0	005– 0	
[STO] 1	006– 23 1	
[GSB] 2	007– 13 2	[h] [LBL] 2 est d'abord exécuté comme sous programme
[h] [LBL] 2	008– 25 13 2	
[RCL] 0	009– 24 0	
9	010– 9	
9	011– 9	
7	012– 7	
[×]	013– 61	
[h] [FRAC]	014– 25 33	
[STO] 0	015– 23 0	
6	016– 6	
[×]	017– 61	
1	018– 1	[h] [LBL] 2 est exécuté comme le reste du programme.
[+]	019– 51	
[h] [INT]	020– 25 32	
[f] [FIX] 0	021– 14 11 0	
[h] [PSE]	022– 25 74	
[STO] [+] 1	023– 23 51 1	
[RCL] 1	024– 24 1	
[h] [RTN]	025– 25 12	Retour à la ligne 008 si [LBL] 2 est exécuté comme un sous-programme ou à la ligne 000 si [LBL] 2 est exécuté comme le reste du programme.

Revenez en mode RUN et jetez les dés. Pour jeter les dés, introduisez la semence (nombre décimal $0 < n < 1$), puis appuyez sur [A]. Le calculateur affiche le chiffre du premier dé, puis celui du second, et enfin, quand le programme s'arrête, le total des deux dés. Pour faire un autre jet, appuyez sur [R/S]. Le programme utilise le dernier nombre comme nouvelle semence.

Vous pouvez aussi jouer avec un ami. Si votre premier total de dés est 7 ou 11, vous avez gagné. Si c'est un autre nombre, ce nombre devient votre pari. Vous continuez à jouer jusqu'à ce que le total soit égal à votre pari (dans ce cas, vous avez également gagné) ou jusqu'à obtenir 7 ou 11 (vous avez perdu et vous passez la main). Exécutez le programme.

Appuyez sur :	Affichage	
.2315478	**0,2315478**	Semence
[A]	**10,**	Votre pari est 10
[R/S]	**8,**	Raté!
[R/S]	**5,**	Encore raté!
[R/S]	**7,**	Vous avez perdu!

Maintenant, recommencez en prenant le dernier nombre comme semence :

Appuyez sur :	Affichage	
[R/S]	**8,**	Votre pari est 8
[R/S]	**8,**	Cette fois-ci, vous avez gagné!

Avant de continuer, remettez l'affichage sur quatre positions décimales :

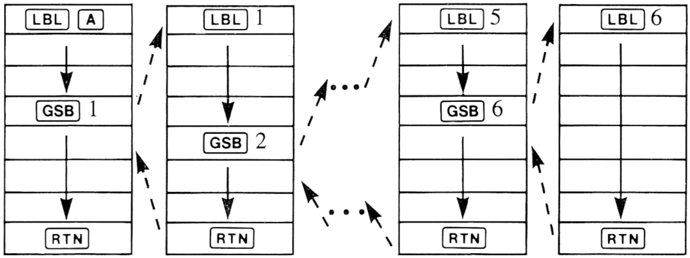
Appuyez sur :	Affichage
[f] [FIX] 4	**8,0000**

LIMITES DES SOUS-PROGRAMMES

Un sous-programme peut en appeler un autre qui peut, à son tour, en appeler un troisième. Les branchements de sous-programmes ne sont limités que par le nombre de retours pouvant rester en attente dans le calculateur, en d'autres termes le nombre de niveaux d'attente de sous-programmes. Le HP-34C possède six niveaux d'attente représentés par le schéma ci-dessous :

Six niveaux d'attente

Programme principal

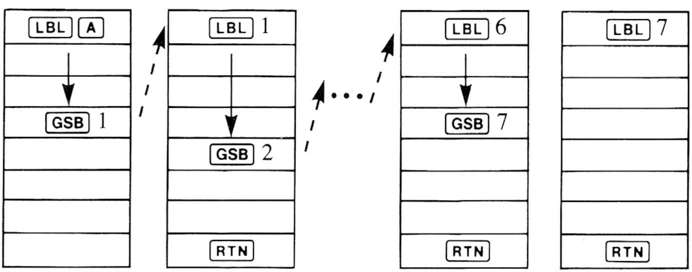


FIN

Le calculateur peut retourner au programme à partir du sixième niveau d'attente. Mais si vous tentez d'appeler un sous-programme au septième niveau, le calculateur s'arrête et affiche une erreur 8.

Six niveaux d'attente seulement

Programme principal



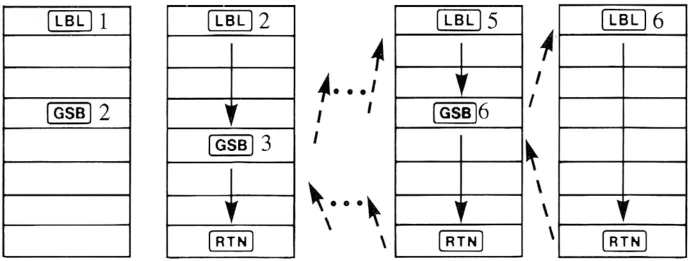
Arrêt et
aff. de
l'erreur 8

Par contre, le calculateur peut toujours exécuter des instructions RTN (retour à la ligne 000 et arrêt), qui ne sont pas des sous-programmes. Si vous appuyez sur GTO ou GSB avec un label A, B ou 0 à 9, toutes les instructions RTN en attente sont perdues.

Appuyez sur GSB 2

Programme principal

Départ



Arrêt et
retour
à 000

Notez qu'en mode PRGM, l'exécution pas à pas d'un programme contenant des sous-programmes suit le même ordre que l'exécution automatique du programme.

Utilisation de h RTN à la fin de la mémoire programme occupée

Dans les exemples de programmation du manuel d'utilisation du HP 34C, la dernière ligne de la mémoire programme occupée contient une instruction h RTN. Cette instruction a été placée dans cette ligne, d'une part, pour bien marquer la fin du programme et, d'autre part, pour montrer l'effet de RTN sur l'exécution du programme. Cependant, cette instruction n'est pas obligatoire et son absence n'a aucun effet sur l'exécution du programme. Car, si la dernière instruction stockée dans la mémoire programme n'est pas h RTN, le programme se comporte comme si la dernière instruction introduite en mémoire était suivie d'une instruction h RTN. En d'autres termes, lorsque le programme arrive à la fin de la mémoire occupée et qu'il ne trouve pas d'instruction h RTN :

1. S'il exécute un sous-programme, il retourne à la ligne qui suit la dernière instruction GSB.

2. S'il n'exécute pas de sous-programme, il retourne à la ligne 000 et s'arrête.

Si la dernière ligne de la mémoire occupée contient une instruction GSB, le calculateur exécute le sous-programme indiqué, retourne à la ligne 000 et s'arrête.

Notons que la recherche du label, déclenchée par les instructions GTO et GSB s'effectue toujours en avant dans la mémoire programme. Ceci permet donc d'utiliser plusieurs fois le même label dans un programme.

Exemple : Soit à calculer la valeur de l'expression $\sqrt{x^2 + y^2 + z^2 + t^2}$ avec un programme identifié par LBL A. Le même label marque le début d'un sous-programme à l'intérieur du programme. Avant d'exécuter le programme, on introduit x, y, z et t dans la pile, puis on appuie sur A.

Mettez le commutateur PRGM-RUN sur PRGM

Appuyez sur :	Affichage
f CLEAR PRGM	000–
h LBL A	001– 25 13 11
g x^2	002– 15 3
GSB A	003– 13 11
GSB A	004– 13 11
GSB A	005– 13 11
f $\sqrt{x}$	006– 14 3
h RTN	007– 25 12
h LBL A	008– 25 13 11
$x \gtrless y$	009– 21
g x^2	010– 15 3
+	011– 51
h RTN	012– 25 12

Mettez le commutateur PRGM-RUN sur RUN et entrez l'ensemble de variables suivant :

x = 4,3 ; y = 7,9 ; z = 1,3 ; t = 8

Appuyez sur :	Affichage
8 ENTER↑	8,000
1.3 ENTER↑	1,3000
7.9 ENTER↑ 4.3 A	12,1074

chapitre 7

PROGRAMMATION ÉTENDUE

REGISTRE I

Le registre I est l'un des outils de programmation les plus intéressants de votre HP 34C. Non seulement il sert de registre de stockage et de rappel des données mais il peut aussi être utilisé en combinaison avec certaines instructions pour remplir les fonctions suivantes :

- Incrémentation et décrémentation du contenu du registre I d'une valeur spécifiée pour contrôle de boucle ou autres fonctions.
- Contrôle indirect de l'adresse du registre mémoire spécifiée dans [STO], [RCL] et arithmétique dans les registres mémoire.
- Contrôle indirect du label spécifié dans [GTO] et [GSB].
- Contrôle indirect du nombre de chiffres affichés en formats [FIX] [SCI] [ENG].
- Saut à une ligne quelconque de la mémoire programme occupée.

Stockage d'un nombre dans le registre I

Pour stocker un nombre dans le registre I, on utilise la séquence de pressions de touches [STO] [f] [I]. Exemple : pour stocker un 7 dans le registre I, mettez d'abord le commutateur PRGM-RUN sur ■▥ RUN, puis

Appuyez sur :	**Affichage**
7 [STO] [f] [I]	**7,0000**

Pour rappeler dans le registre X un nombre placé dans le registre I, on utilise la séquence [RCL] [f] [I].

Appuyez sur :	**Affichage**	
[CLx]	**0,0000**	
[RCL] [f] [I]	**7,0000**	Rappel d'une copie du nombre stocké dans I.

Permutation des contenus de X et de I

De la même manière que l'opération [$x \gtrless y$], l'opération [f] [$x \gtrless I$] (permutation de x et de I) permute le contenu du registre X et celui du registre I.

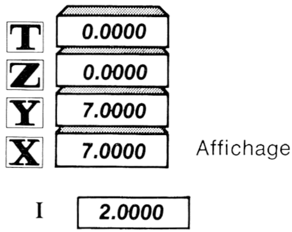
Introduisons par exemple le nombre 2 dans le registre X et permutons le contenu de X avec la valeur introduite dans I, dans l'exemple précédent.

Appuyez sur :	Affichage	
2	2,	
f x≷I	7,000	Permutation des contenus des registres X et I.

Quand vous avez appuyé sur x≷y, le contenu de la pile et celui du registre I ont été permutées :

Avant...

Après...



Pour remettre les contenus respectifs des registres X et I à l'endroit initial :

Appuyez sur :	Affichage
f x≷I	2,0000

Incrémentation et décrémentation de I

Il existe deux autres fonctions, très utiles en programmation, qui permettent de modifier le contenu du registre I : ISG (increment, then skip if greater) et DSE (decrement, then skip if less than or equal). Les deux fonctions ont des compteurs internes permettant de contrôler l'exécution d'une boucle ainsi que les opérations d'adressage séquentiel qui seront traitées plus loin dans cette section.

Les fonctions ISG et DSE se basent sur un nombre stocké dans le registre I et interprété d'une certaine manière. On appelle ce nombre **nombre de contrôle de boucle.** Il a généralement le format suivant :

nnnnn.xxxyy

Un nombre de contrôle de boucle est considéré comme trois entiers distincts dont

± nnnnn est la valeur actuelle du compteur,
xxx est la valeur de référence du compteur et
yy est la valeur de l'incrément

La partie nnnnn du nombre indique au HP 34C que vous voulez compter le nombre de passages dans la boucle à partir de ce nombre. La valeur par défaut est zéro. nnnnn peut comprendre de 1 à 5 chiffres.

La partie xxx du nombre indique au HP 34C que vous voulez vous arrêter de compter à ce nombre. Elle doit toujours avoir trois positions numériques. (Par exemple, si xxx est égal à 10, on indique 010.)

La partie yy du nombre de contrôle de boucle indique au calculateur la manière de compter. La valeur actuelle du compteur nnnnn est incrémentée ou décrémentée de la valeur de yy. La valeur par défaut de yy est l'unité. Le calculateur compte alors de 01 en 01. La valeur yy doit être donnée sous forme de deux chiffres (par ex., 02, 03, 55).

Incrémentation avec saut si supérieur à

A chaque exécution, ISG incrémente nnnnn de yy, puis vérifie si nnnnn est supérieur à xxx. Si oui, le calculateur, saute la ligne suivante du programme.

Supposons que le registre I contienne le nombre de contrôle de boucle 100.20001. L'instruction ISG commence à compter à partir de 100. Chaque fois que le programme exécute ISG, la partie nnnnn du nombre de contrôle de boucle est incrémentée de I.

Contenu du registre I = 100.20001
Exécution de l'instruction ISG :
  Le compte part à 100.
  nnnnn est incrémenté de 1
  Vérification de nnnnn pour voir si sa valeur est supérieure à 200.

Après une exécution ou passage dans la boucle contenant ISG, le registre I contient 101.20001. Après 10 exécutions ou passages dans la boucle, le registre I contient 110.20001. Après chaque incrémentation, ISG vérifie si la valeur actuelle du compteur nnnnn est supérieure à 200 (xxx). Si oui, le calculateur saute la ligne qui suit l'instruction ISG. Nous allons voir dans un moment l'utilité de ces sauts de ligne dans un programme.

Décrémentation avec saut si égal (ou inférieur à)

A chaque exécution, DSE décrémente nnnnn de yy, puis vérifie si nnnnn est égal (ou inférieur) à xxx. Si oui, le HP 34C saute la ligne suivante du programme.

Supposons que le registre I contienne le nombre 100.01001. L'instruction DSE commence à compter à partir de 100. A chaque exécution de DSE, la partie nnnnn du nombre de contrôle de boucle est décrémentée de 1.

Contenu du registre I = 100.01001

Exécution de [DSE] :
Le compte part à 100.
nnnnn est décrémenté de 1.
Vérification de xxx pour voir si sa valeur est égale (ou inférieure) à 10.

Après une exécution ou passage dans la boucle, le registre I contient 99.01001. Après 10 exécutions ou passages dans la boucle, le registre I contient 90.01001. Après chaque décrémentation, [DSE] vérifie si la valeur nnnnn du compteur est inférieure ou égale à 010 (xxx). Si oui, le calculateur saute la ligne suivante du programme.

Exemple : Le programme suivant illustre le fonctionnement de l'instruction [ISG]. Il contient une boucle qui fait une pause pour afficher la valeur actuelle du registre I et l'instruction [ISG] qui contrôle le nombre de passages dans la boucle ainsi que la valeur du carré du nombre. Le programme crée une table des carrés des nombres pairs de 2 à 50.

Mettez le commutateur PRGM-RUN sur PRGM.

Appuyez sur :	Affichage	
[f] CLEAR [PRGM]	000–	
[h] [LBL] [A]	001– 25 13 11	Label du programme.
[f] [FIX] [5]	002– 14 11 5	Valeur actuelle du compteur
[2]	003– 2	(nnnnn).
[•]	004– 73	
[0]	005– 0	
[5]	006– 5	Valeur de réference du compteur
[0]	007– 0	(xxx).
[0]	008– 0	Valeur de l'incrément.
[2]	009– 2	
[STO] [f] [I]	010– 23 14 23	Stockage du nombre de contrôle de boucle dans I.
[h] [LBL] [1]	011– 25 13 1	Début de la boucle.
[RCL] [f] [I]	012– 24 14 23	Rappel du nombre placé dans I.
[h] [INT]	013– 25 32	Extraction de la partie entière.
[h] [PSE]	014– 25 74	Pause pour affichage de l'entier.
[g] [x^2]	015– 15 3	Carré du nombre.
[h] [PSE]	016– 25 74	Affichage du carré du nombre.
[g] [ISG]	017– 15 24	Incrémentation de [I] par 2 et vérification du compte pour voir s'il n'est pas supérieur au nombre final (50). Dans l'affirmative, saut de la ligne suivante du programme.
[GTO] [1]	018– 22 1	Rebouclage sur le label 1.
[h] [RTN]	019– 25 12	Arrêt du programme.

Exécutez le programme : mettez le commutateur PRGM-RUN sur RUN et appuyez sur [A].

Appuyez sur :	**Affichage**	
[A]	**2,00000**	Dès le début de l'exécution, le
	4,00000	HP 34C fait une pause pour afficher le nombre à élever au
	4,00000	carré, puis une autre pour
	16,00000	afficher le carré du nombre. Quand le compteur de boucle atteint 50, le programme
	50,00000	s'arrête.
	2.500,00000	

Déroulement du programme

1. Sous le label [A], le nombre 2.05002 est stocké dans le registre I comme nombre de contrôle de boucle. Il a le format conventionnel nnnnn xxx yy.

nnnnn	xxx	yy
(0000)2	050	02
Compte actuel	Référence	Incrément

2. Sous le label [1], le déroulement est le suivant : après affichage de 2 et 4 (carré de 2), le compte actuel du registre [I], 00002 (nnnnn) est incrémenté de 02 (yy). Le nouveau contenu du registre I est :

nnnnn	xxx	yy
(0000)4	050	02
Compte actuel	Référence	Incrément

Ce nouveau contenu est comparé à la valeur de référence (xxx). Comme le compte ne dépasse pas la valeur de référence, le calculateur passe à la ligne suivante, GTO 1 et reprend le même processus pour le nombre suivant.

3. Au bout du 25e nombre pair (2 à 50), le compte du registre I dépasse 50. Le calculateur saute alors une ligne après [g] [ISG] à la ligne 17. Il saute donc l'instruction [GTO] [1] à la ligne 18 et exécute l'instruction [RTN] à la ligne 19, puis retourne à la ligne 000 et s'arrête.

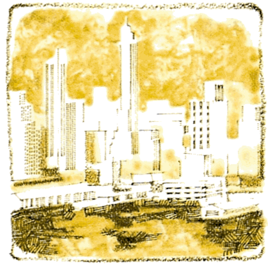
A la fin du programme, appuyez sur [RCL] [f] [I]. Le registre I affiche le contenu suivant :

52.050.02

Compte actuel (nnnnn) — Valeur de référence (xxx) — Incrément (yy)

Nous allons écrire maintenant un second programme qui utilise la fonction [DSE]. Le compte du registre I a le même format nnnnn. xxxyy qu'avec [ISG], mais au lieu d'être incrémenté, il est décrémenté.

L'île de Manhattan fut vendue en 1624 pour la somme symbolique de 24 dollars. Le programme suivant calcule la valeur acquise théorique de cette somme sur un certain nombre d'années si elle avait été placée sur un compte bancaire rapportant 6 % par an. Les années sont stockées dans le registre I comme nombre de contrôle de boucle et l'instruction [DSE] sert à compter le nombre d'itérations dans la boucle.



Mettez le commutateur PRGM-RUN sur PRGM. L'instruction [RTN] dans la ligne 019 du programme précédent a repositionné le calculateur sur la ligne 000. Pour ajouter le programme suivant à la fin de la mémoire programme occupée, appuyez sur [h] [BST] (ou [GTO] [•] 019) afin de retourner à la ligne 019.

Appuyez sur :	Affichage	
[h] [BST]	019– 25 12	Dernière ligne du programme précédent.
[h] [LBL] [B]	020–25, 13, 12	Label du nouveau programme
[f] [FIX] 2	02– 14, 11, 2	Stockage de la valeur
[STO] [f] [I]	022–23, 14, 23	nnnnn.xxxyy dans le registre I.
1	02– 1	
6	02 – 6	Première année
2	025– 2	
4	026– 4	
[+]	027– 51	Dernière année
[STO] 0	028– 23 0	Stockage de la dernière année
2	029– 2	
4	030– 4	Somme originale
[h] [LBL] 2	031–25, 13, 2	Début de la boucle
1	032– 1	
0	033– 0	Calcul de la valeur acquise par
6	034– 6	an
[h] [%]	035– 25 41	

[g] [DSE]	036–	15 23	← Décrémentation du compte actuel nnnnn et comparaison avec la valeur de référence xxx.
[GTO] 2	037–	22 2	
[RCL] 0	038–	24 0	
[h] [PSE]	039–	25 74	
[x⇄y]	040–	21	Si nnnnn > xxx, retour à [LBL] 2.
[h] [RTN]	04 –	25 12	Si nnnnn ⩽ xxx, affichage de la dernière année, de la valeur acquise finale et arrêt.

Entrez le nombre d'années désiré (nombre de contrôle de boucle). Appuyez sur [B] pour stocker cette valeur dans le registre I et exécuter le programme.

Appuyez sur :	**Affichage**	
5	**5,**	Nombre de contrôle de boucle : nnnnn = 5, xxx = 000, yy = 00 (valeur par défaut : 01)
[B]	**32,12**	Au bout de cinq ans, en 1629, le compte aurait valu 32,12 dollars.
15	**15,**	Nombre de contrôle de boucle : nnnnn = 15, xxx = 000, yy = 00 (valeur par défaut : 01).
[B]	**57,52**	Au bout de 15 ans, en 1639, le compte aurait valu 57,52 dollars.

Déroulement du programme

Quand vous introduisez le nombre d'années et appuyez sur [B], les années sont stockées dans le registre I et servent de nombre de contrôle de boucle (nnnnn. xxxyy).

nnnnn	xxx	yy
(0000)5	000	00
Compte actuel	Valeur de référence	Uncrément (valeur par défaut : 01)

Notez que si la valeur de référence est 000 et l'incrément est 01, il est inutile de les entrer.

Le nombre de contrôle de boucle est ensuite ajouté à la première année. Le total représente la dernière année, qui est stockée dans R_0. Vous introduisez le capital initial. A chaque passage dans la boucle, ce capital augmente de 6 %. L'instruction DSE retranche 1 du registre I. Si le nombre de contrôle de boucle stocké dans le registre I n'est pas nul, le calculateur retourne à LBL 2 et réexécute la boucle.

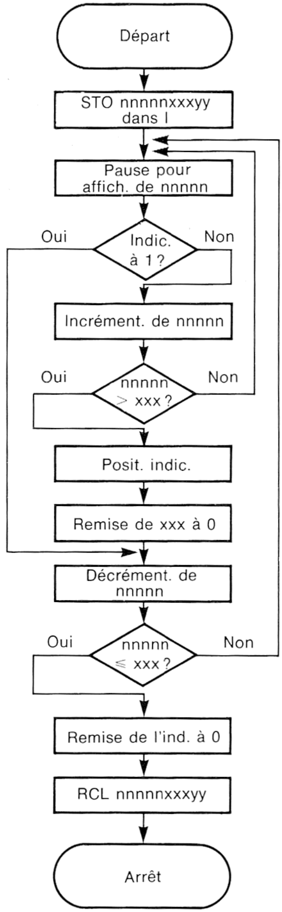
Quand le nombre de contrôle de boucle stocké dans le registre I devient nul (nnnnn = xxx), le programme saute l'instruction GTO 2, ligne 37 et reprend à l'instruction RCL 0, ligne 38. La dernière année et la valeur finale du capital apparaissent l'une après l'autre et le programme s'arrête.

Limites pour ISG et DSE

Les instructions ISG et DSE permettent d'incrémenter et de décrémenter tous les nombres affichables par le HP-34C. Cependant la partie décimale du nombre de contrôle de boucle est affectée par la partie qui dépasse les cinq chiffres du compte actuel nnnnn du registre I. Prenons, par exemple, le nombre 99.950,50055. Incrémenté par ISG, il devient 100.005,5006. L'incrément est égal à 55. Le nouveau nombre (100.005,50055) ne pouvant pas être entièrement affiché, la partie décimale du nombre a été arrondie. Comme le calculateur prend un incrément de deux chiffres (yy), l'incrément suivant sera 60 et non pas 55, et quand le nombre devient 999.945,5006, le nombre suivant est 1.000.005,501 en arrondissant la partie décimale. Puisque l'incrément yy n'est pas spécifié, le calculateur prend ensuite par défaut la valeur 01 au lieu de rester à 60.

Problèmes

1. Écrivez un programme comptant de 0 à une limite donnée en utilisant la fonction ISG, puis décomptant de la limite à 0 en utilisant cette fois la fonction DSE. Aidez-vous de l'organigramme de la page suivante.



UTILISATION DU REGISTRE I POUR LE CONTRÔLE INDIRECT

Vous avez vu comment le contenu du registre I pouvait être modifié à l'aide des instructions [STO], [x⋛I], [ISG] et [DSE]. Ce contenu peut aussi servir à contrôler certaines opérations concernant l'affichage, les registres mémoire, les branchements et les sous-programmes. Nous allons d'abord voir quelques généralités sur ces opérations, puis nous les reprendrons en détail, une à une.

DSP I (display I) spécifie à l'aide d'un nombre stocké dans le registre I le nombre de positions décimales apparaissant sur l'écran.

$x \gtrless (i)$ (X exchange indirect) permute le contenu du registre affiché X avec celui du registre mémoire disponible, adressé par la valeur absolue du nombre stocké dans le registre I.

STO f (i) (store indirect) stocke la valeur affichée dans le registre mémoire adressé par la valeur absolue du nombre présent dans le registre I.

RCL f (i) (recall indirect) rappelle le contenu du registre mémoire adressé par la valeur absolue du nombre présent dans le registre I.

STO (+, ×, − ou ÷) (i) (arithmétique indirecte dans les registres mémoire) effectue des opérations arithmétiques sur le contenu du registre mémoire adressé par la valeur absolue du nombre présent dans le registre I.

GTO f I (go to label I) branche le programme en cours d'exécution sur le premier label correspondant au nombre positif présent dans le registre I. La recherche du label s'effectue en séquence vers le bas de la mémoire programme.

GTO f I (go to line I) branche le programme en cours d'exécution sur le numéro de ligne occupé correspondant à la valeur absolue du nombre négatif présent dans le registre I.

GSB f I (go to label I subroutine) branche le programme en cours d'exécution sur le premier sous-programme correspondant au nombre positif présent dans le registre I. La recherche du sous-programme s'effectue en séquence vers le bas de la mémoire programme. A la rencontre d'un RTN, le programme retourne à la ligne qui suit l'instruction GSB et continue.

GSB f I (go to line I subroutine) branche le programme en cours d'exécution sur le numéro de ligne correspondant à la valeur absolue du nombre négatif présent dans le registre I. L'exécution se poursuit comme pour un sous-programme normal.

Dans toutes ces opérations, si le nombre présent dans le registre I ne convient pas, le calculateur affiche un message d'erreur. D'autre part, dans ces opérations de contrôle de l'affichage, des registres mémoire ou du programme, le calculateur ne retient que la partie entière du nombre présent dans I. Ainsi, le nombre 12,99041276 stocké dans le registre I est conservé tel quel dans le registre I mais dans les opérations de contrôle décrites plus haut, le calculateur le considère comme un 12.

Vous voyez donc que l'utilisation du registre I joint à certaines fonctions offre d'intéressantes possibilités de calcul et de programmation. Étudions maintenant de plus près les opérations indirectes.

Contrôle indirect de l'affichage

Le nombre présent dans le registre I permet de contrôler le nombre de positions décimales affichées sur l'écran. Le nombre affiché sous l'action de [h] [DSP I] est arrondi au nombre de positions décimales spécifié par la valeur présente dans le registre I. (Cet arrondi n'apparaît qu'à l'affichage puisque le calculateur conserve intégralement la valeur de 10 × 10 chiffres élevés à une puissance de deux chiffres.)

L'opération qui vient d'être mentionnée est particulièrement utile à l'intérieur d'un programme mais elle peut également être exécutée manuellement à partir du clavier. Exécutez, par exemple, l'opération suivante en mode RUN :

Appuyez sur :	**Affichage**	
[CLx] [f] [FIX] 4	**0,0000**	Effacement de l'affichage ;
[STO] [f] [I]	**0,0000**	format FIX. Introduction de 0dans le
9.123456789	**9,123456789**	registre I.
[h] [DSP I]	**9,**	Spécification du format FIX par la valeur 0 présente dans le registre I.
[g] [ISG]	**9,**	Incrémentation à 1 de la valeurde I.
[h] [DSP I]	**9,1**	Spécification du format FIX par la valeur présente dans le registre I.
[g] [ISG]	**9,1**	Incrémentation à 2 de la valeur
[h] [DSP I]	**9,12**	de I. Spécification du format FIX par la valeur présente dans le registre I.

Exemple : le programme suivant calcule et affiche un exemple de format d'affichage FIX pour chaque position décimale possible. Il utilise une boucle contenant une instruction [DSE] pour changer automatiquement le nombre de positions décimales.

Pour introduire le programme, mettez le commutateur PRGM-RUN sur PRGM

Appuyez sur :	**Affichage**
[f] CLEAR [PRGM]	**000–**
[h] [LBL] [A]	**001–25, 13, 11**

9	002– 9
STO f I	003–23, 11, 23
h LBL 0	004–25, 13, 0
h DSP I	005– 25 11
RCL f I	006–24, 14, 23
h PSE	007– 25 74
g DSE	008– 15 23
GTO 0	009– 22 0
g $x>0$	010– 14 51
GTO 0	011– 22 0
h RTN	012– 25 12

Pour afficher en notation fixe toutes les positions décimales possibles, passez en mode RUN, puis :

Appuyez sur :	Affichage
A	9,000000000
	8,00000000
	7,0000000
	6,000000
	5,00000
	4,0000
	3,000
	2,00
	1,0
	0,
	0,

Pour afficher en notation scientifique ou « ingénieur » toutes les positions décimales possibles, remplacez le 9 de la ligne 002 par un 6 et passez du mode SCI en mode ENG en appuyant sur f SCI ou f ENG et un chiffre quelconque de 0 à 7*. Puis, appuyez sur A comme dans l'exemple précédent.

Mettez le commutateur PRGM-RUN sur PRGM.

Appuyez sur :	Affichage
GTO • 002	002– 9
h DEL	001–25, 13, 11
6	002– 6

* Les pressions de touches f SCI ou f ENG 8 ou 9 se traduisent automatiquement par un f SCI 7 ou f ENG 7 dans la mémoire programme.

Mettez le commutateur PRGM - RUN sur RUN.

Appuyez sur :	**Affichage**	
f SCI 4	**0,0000 00**	Format SCI normal
f ENG 4	**0,0000 00**	Format ENG normal
A	**6,000000 00**	
	5,00000 00	
	4,0000 00	
	3,000 00	
	2,00 00	
	1,0 00	
	0, 00	
	0, 00	

Si le registre I* contient un nombre quelconque inférieur à 0, l'opération h DSP I donne le même nombre de chiffres à l'affichage que si le registre I contenait des 0. Si le registre I contient un nombre supérieur à 0, l'opération h DSP I donne le même nombre de chiffres à l'affichage que si le registre contenait des 9. Notons qu'en modes SCI et ENG si le nombre présent dans le registre I est supérieur à 6, les chiffres affichés à droite du point décimal ne sont jamais supérieurs à 6, sans compter l'exposant de 2 chiffres. (Rappelons, cependant, que SCI ou ENG 7 arrondissent l'affichage à un chiffre de plus que SCI ou ENG 6.)

Exécutez l'opération suivante :

Appuyez sur	**Affichage**	
f FIX 4	**0,0000**	Format FIX normal.
1.999999999	**1,999999999**	
STO f I	**2,0000**	Arrondi à la valeur de la dernière instruction de format.
h DSP I	**2,0**	DSP I ne lit que la partie entière de la valeur stockée dans I.
.9852	**0,9852**	
STO f I	**1,0**	Arrondi à la valeur de la dernière instruction de format.
h DSP I	**1,**	Une valeur $<$ 1 donne le même résultat qu'une valeur nulle.
19	**19,**	

* Lors d'une opération ∫, si le registre I contient un nombre compris entre − 6 et + 9, DSP I prend ce nombre comme paramètre automatique du calcul de ∫ (voir détails dans la section 9).

STO f I	**19,**	S'il y a deux chiffres à gauche du point décimal, la présence d'une valeur > 9 dans le registre I donne le même résultat qu'une valeur égale à 8 ou 9.
h DSP I	**19,00000000**	
CHS STO f I	**−19,00000000**	Stockage d'un nombre négatif dans I.
h DSP I	**−19,**	La présence d'un nombre négatif dans le registre I donne le même résultat qu'un nombre positif < 1.
f SCI 4	**−1,9000 −01**	Affichage SCI normal
1.1111119 ENTER↑	**1,1111 00**	
7 STO f I x⇄y	**1,1111 00**	
h DSP I	**1,111111 00**	Affichage arrondi à 7 pos. décimales
6 STO f I x⇄y	**1,111111**	
h DSP I	**1,111112 00**	Affichage arrondi à 6 pos. décimales

Permutation de X et (i)

La fonction x⇄(i) permet de permuter le contenu du registre affiché X et celui d'un registre mémoire disponible quelconque adressé indirectement par la valeur absolue d'un nombre quelconque compris entre − 21 et 21 stocké dans le registre I.

Les nombres entiers de 0 à ± 9 correspondent aux registres mémoire R_0 à R_9. Ceux de ± 10 à ± 19 aux registres $R_{.0}$ à $R_{.9}$. Si le registre I contient ± 20, x⇄(i) adresse le registre I lui-même.

Le tableau suivant résume le système d'adressage :

	adresse de [(i)]		adresse de [(i)]
R_0	0	$R_{.0}$	10
R_1	1	$R_{.1}$	11
R_2	2	$R_{.2}$	12
R_3	3	$R_{.3}$	13
R_4	4	$R_{.4}$	14
R_5	5	$R_{.5}$	15
R_6	6	$R_{.6}$	16
R_7	7	$R_{.7}$	17
R_8	8	$R_{.8}$	18
R_9	9	$R_{.9}$	19
		I	20

Mettez d'abord l'affichage sur FIX 4 et effacez le registre X et tous les registres mémoire.

Appuyez sur :	**Affichage**
f FIX 4 CLx	**0,0000**
f CLEAR REG	**0,0000**

Essayez ensuite les exemples suivants en utilisant x⪤(i) pour stocker 1.234 dans les registres R_3, R_5 et I.

Appuyez sur :	**Affichage**	
3 f x⪤I	**0,0000**	Permutation des contenus des registres X et I.
1.2345 h x⪤(i)	**0,0000**	Permutation des contenus des registres X et R_3 en prenant pour adresse l'entier 3 placé dans I.
RCL 3	**1,2345**	Rappel d'une copie du contenu de R_3.
15 f x⪤I	**3,0000**	Permutation des contenus des registres X et I.
x⪤y	**1,2345**	Permutation des contenus des registres X et Y.
h x⪤(i)	**0,0000**	Permutation des contenus des registres X et $R_{.5}$ en prenant pour adresse l'entier 15 placé dans I.
RCL • 5	**1,2345**	Rappel d'une copie du contenu de $R_{.5}$
f CLEAR REG	**1,2345**	Effacement de tous les registres mémoire.
15.3974 CHS	**–15,3974**	
f x⪤I	**0,0000**	Permutation des contenus des registres X et I
x⪤y	**1,2345**	Permutation des contenus des registres X et Y.
h x⪤(i)	**0,0000**	Permutation des contenus des resgistres X et $R_{.5}$ en prenant pour adresse la partie entière de la valeur absolue de – 15,3971 placée dans I.
RCL • 5	**1,2345**	Rappel d'une copie du contenu de $R_{.5}$
20 f x⪤I	**– 15,3974**	Permutation des contenus des registres X et I.
x⪤y	**1,2345**	Permutation des contenus des registres X et Y.

h x⇄(i)	**20,0000**	Permutation des contenus des registres X et I en prenant pour adresse l'entier placé dans I.
RCL f I	**1,2345**	Rappel d'une copie du contenu de I.
f CLEAR REG	**1,2345**	Effacement de tous les registres mémoire.
CLx ENTER↑ ENTER↑ ENTER↑	**0,0000**	Effacement de tous les registres de la pile.

Stockage et rappel indirects

Le registre I peut servir à adresser indirectement les 21 registres mémoire lors d'une opération STO ou RCL.

Quand vous appuyez sur STO f (i), la valeur affichée est stockée dans le registre mémoire adressé par le nombre placé dans le registre I. Les registres mémoire peuvent être adressés de la même manière par RCL f (i) et par les opérations arithmétiques STO + (i), STO − (i), STO × (i) et STO ÷ (i). (Si vous avez oublié le fonctionnement normal des registres mémoire ou les opérations arithmétiques dans les registres mémoire, retournez au chapitre « Stockage et rappel des nombres » de « La solution de vos problèmes avec votre calculateur Hewlett-Packard »).

Avec STO f (i), RCL f (i) ou toute opération arithmétique dans les registres mémoire utilisant la fonction (i), le registre I peut contenir les mêmes valeurs positives ou négatives de 0 à 20 qu'avec x⇄(i).

En utilisant le calculateur manuellement, il vous sera facile de voir comment l'association de STO f (i) ou RCL f (i) et du registre I permet d'adresser les différents registres mémoire.

Mettez le commutateur PRGM-RUN sur RUN

Appuyez sur	Affichage	
CLx h FIX 4	**0,0000**	
f CLEAR REG	**0,0000**	Effacement de tous les registres mémoire, y compris I.
5 STO f I	**5,0000**	Stockage du nombre 5 dans le registre I.
1.23 STO f (i)	**1,2300**	Stockage du nombre 1,23 dans le registre mémoire adressé par le nombre placé dans I, c.à.d. le registre R_5.
19 STO f I	**19,0000**	Stockage du nombre 19 dans le registre I.

85083 [STO] [f] [(i)]	**85.983,0000**	Stockage du nombre 85083 dans le registre mémoire $R_{.9}$ adressé par le nombre présent dans I, c.à.d. 19.
12 [STO] [f] [I]	**12,0000**	Stockage du nombre 12 dans
77 [EEX] 43	**77, 43**	le registre I.
[STO] [f] [(i)]	**7,7000 44**	Stockage du nombre 7.7×10^{44} dans le registre mémoire adressé par le nombre présent dans I, c.à.d. le registre $R_{.2}$.

Pour rappeler les nombres stockés dans un registre, vous pouvez utiliser la touche [RCL] (recall) suivie du nombre correspondant à l'adresse du registre. Cependant, si le nombre présent dans le registre I adresse le registre mémoire désiré, vous pouvez rappeler le contenu de ce registre avec [RCL] [f] [(i)]

Appuyez sur	Affichage	
[RCL] 5	**1,2300**	Rappel du contenu du registre mémoire R_5 dans le registre X.
[RCL] [f] [(i)]	**7,7000 44**	Comme le registre I contient toujours le nombre 2, cette opération rappelle le contenu du registre mémoire $R_{.2}$ adressé par 12.

En changeant le nombre présent dans le registre I, vous changez l'adresse spécifiée par [STO] [f] [(i)] ou [RCL] [f] [(i)]. Par exemple :

Appuyez sur	Affichage	
19 [STO] [f] [I]	**19,0000**	
[RCL] [f] [(i)]	**85.083,0000**	Rappel du contenu du registre mémoire $R_{.9}$ dans le registre X.
5 [STO] [f] [I]	**5,0000**	
[RCL] [f] [(i)]	**1,2300**	Rappel du contenu du registre mémoire R_5 dans le registre X.

Les opérations arithmétiques dans les registres mémoire portent sur le contenu du registre adressé par I avec [STO] [+] [(i)], [STO] [−] [(i)], [STO] [×] [(i)] et [STO] [÷] [(i)]. Notez que dans dans ces quatre opérations, la touche préfixe [f] est inutile.

Appuyez sur	Affichage	
1 [STO] [+] [(i)]	**1,0000**	Addition de 1 au nombre présent dans le registre mémoire (R_5) actuellement adressé par le registre I.

Appuyez sur	Affichage	
RCL f (i)	2,2300	Rappel du nombre placé dans R_5.
2 STO × (i)	2,0000	Multiplication par 2 du contenu de R_5.
RCL f (i)	4,4600	Rappel du résultat de R_5.
CLx	0,0000	Effacement de l'affichage.
RCL 5	4,4600	Rappel direct du contenu de R_5.

Remarque :

Les intructions concernant les opérations arithmétiques effectuées dans les registres R_0 à R_9 peuvent être introduites comme opérations de stockage direct ou indirect. Par contre, pour les registres $R_{.0}$ à $R_{.9}$ et le registre I, seul le stockage indirect est admis.

Naturellement, la meilleure façon d'utiliser le registre I comme adresse pour STO et RCL est à l'intérieur d'un programme.

Exemple : Le programme suivant utilise une boucle pour placer le nombre représentant son adresse dans les registres mémoire R_0 à R_9 et $R_{.0}$ à $R_{.9}$. A chaque itération, le programme s'arrête pour afficher la valeur actuelle de I. Arrivé à 20, il sort de la boucle pour l'intermédiaire de l'instruction g ISG, retourne à la ligne 000 et s'arrête.

Pour introduire le programme, mettez le commutateur PRGM - RUN sur PRGM.

Appuyez sur :	Affichage	
f CLEAR PRGM	000–	
h LBL A	001–25, 13, 11	
•	002– 73	
0	003– 0	
1	004– 1	Nombre de contrôle de boucle.
9	005– 9	
STO f I	006–23, 14, 23	Stockage du nombre de contrôle de boucle.
h LBL 1	007–25, 13, 1	Valeur entière actuelle de I.
RCL f I	008–24, 14, 23	
h INT	009– 25 32	Stockage dans registre mémoire adressé par (i).
STO f (i)	010–23, 14, 24	
h PSE	011– 25 74	Pause pour affichage de la valeur actuelle de I.

[g] [ISG]	012–	15 24	Addition de 1 à la valeur présente dans le registre I et comparaison avec la valeur de référence du compteur (019).
[GTO] 1	013–	22 1	Si I ≤ 19, nouvelle itération
[h] [RTN]	014–	25 12	Si I > 19, retour à la ligne 000 et arrêt.

Mettez le commutateur PRGM - RUN sur RUN

La première étape consiste à mettre des 0 dans le registre I. Puis, le programme rappelle la valeur actuelle du registre I (nombre de contrôle de boucle) et stocke la partie entière de ce nombre dans le registre d'adresse correspondante. Par exemple, si le registre I contient le nombre 17.020, le programme rappelle ce nombre et stocke la partie entière, 17, dans le registre de stockage indirect ($R_{(17)}$) adressé par 17. A chaque passage dans la boucle, le registre I est incrémenté et le résultat utilisé par l'instruction [STO] [f] [(i)] à la fois comme donnée et comme adresse. Quand le nombre stocké dans le registre I arrive à 20, le programme sort de la boucle et s'arrête.

Pour exécuter le programme :

Appuyez sur :	**Affichage**
[A]	**0,0000**
	1,0000
	2,0000
	.
	.
	.
	19,0000

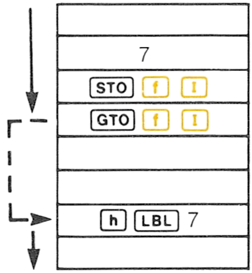
Notez que le contenu du registre I a été incrémenté à 20.0190

Appuyez sur :	**Affichage**
[RCL] [f] [I]	**20,0190**

Contrôle indirect des branchements et sous-programmes

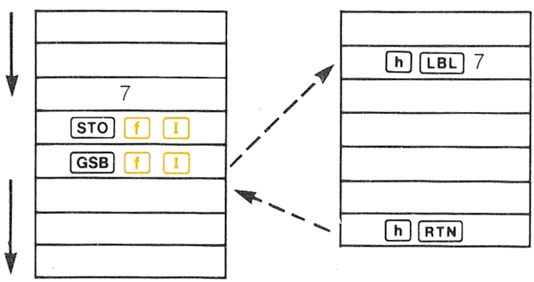
Le registre I permet également d'adresser des segments de programmes, des sous-programmes et même des programmes entiers.

Pour adresser un segment de programme à l'aide du registre I, on utilise une instruction [GTO] [f] [I]. Lorsqu'un programme en cours d'exécution rencontre une instruction [GTO] [f] [I], il se branche en avant sur le LBL adressé par le nombre placé dans le registre I. Ainsi, si le registre I contient un 7, quand le programme rencontre GTO f I, il avance dans la mémoire programme jusqu'à l'instruction LBL 7 suivante, puis reprend.



Vous pouvez également introduire l'instruction GTO f I manuellement, au clavier, si vous voulez positionner le calculateur sur le label adressé dans le registre I et l'arrêter.

Pour adresser un sous-programme à l'aide du registre I, on utilise une instruction GSB f I. L'exécution de cette instruction au cours d'un programme branche le programme sur le LBL spécifié et lance le sous-programme. A la rencontre d'un RTN, le sous-programme retourne à l'instruction qui suit l'instruction GSB f I, et le programme reprend. Supposons, par exemple, que le registre I contienne un 7. GSB f I provoque l'exécution du sous-programme défini par LBL 7 et RTN



Vous pouvez aussi faire GSB f I au clavier pour exécuter le programme ou sous-programme adressé par le nombre contenu dans le registre I, puis arrêter.

Le système d'adressage par l'intermédiaire du registre I est facile à retenir et est identique pour GTO f I et GSB f I. Si le registre I contient des 0 ou un nombre positif de 1 à 9, GTO f I adresse LBL 0 à 9. Si le nombre placé dans I est un nombre positif égal à 10 ou 11, il adresse LBL A ou LBL B. Tableau d'adressage des labels :

Nombre contenu dans le registre I	Label adressé par GTO f I ou GSB f I
0	h LBL 0
1	h LBL 1
2	h LBL 2
3	h LBL 3
4	h LBL 4
5	h LBL 5
6	h LBL 6
7	h LBL 7
8	h LBL 8
9	h LBL 9
10	h LBL A
11	h LBL B

Rappelons que les valeurs des adresses de labels stockées dans le registre I doivent être comprises entre 0 et 12 (nous verrons plus loin que les nombres négatifs provoquent des branchements) et que le calculateur ne retient comme adresse que la partie entière du nombre présent dans I.

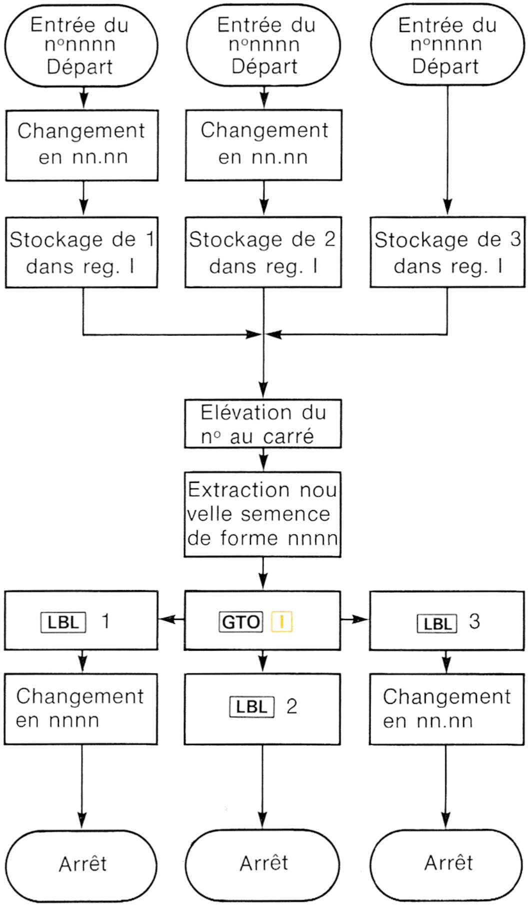
Exemple : Pour générer des nombre pseudo-aléatoires dans un programme, il existe une méthode qui consiste à prendre un nombre (la « semence »), l'élever au carré, extraire le milieu du carré obtenu et l'élever au carré, etc. Ainsi, le carré d'une semence de 5182 est égal à 26853124. Le générateur de nombres aléatoires extrait les quatre chiffres du milieu, 8531, et les élève au carré. Au bout de plusieurs passages dans la boucle, il aura généré plusieurs nombres aléatoires*.

Le programme suivant utilise l'instruction GTO f I pour introduire une semence de quatre chiffres sous l'une des trois formes suivantes : nnnn, .nnnn ou nn.nn. Le programme calcule le carré de la semence, puis tronque le résultat. Les quatre chiffres résultants sont affichés sous la même forme que la semence originale : nnnn, .nnnn, nn.nn.

L'organigramme de ce programme a l'allure suivante :

---

* En fait, ces nombres ne sont pas vraiment aléatoires. Une fois que plusieurs de ces nombres « pseudo-aléatoires » ont été générés par la méthode du carré de la partie centrale, ils commencent à se comporter de façon très systématique et non aléatoire. La véritable génération de nombres aléatoires ne nous intéresse pas dans ce manuel.



L'utilisation de GTO f I permet de sélectionner, en fonction du format de la semence, les opérations exécutées sur le nombre qui suit le segment primaire du programme.

En stockant 1, 2 ou 3 dans le registre I suivant le format de la semence, le programme sélectionne la forme du résultat généré par le segment primaire du programme. Bien que le programme choisi ici s'arrête après chaque résultat, il serait facile de créer une boucle à plusieurs itérations pour augmenter ainsi l'apparent caractère aléatoire des résultats.

Pour introduire le programme, mettez le commutateur RUN - PRGM sur PRGM ▯▮.

f CLEAR PRGM	000 –	
h LBL 4	001 – 25, 13, 4	
EEX	002 – 33	Transformation de nnnn en nn.nn
2	003 – 2	
÷	00 – 71	
1	005 – 1	Introduction de 1 dans le registre X pour stockage dans I.
GTO 7	006 – 22 7	
h LBL 5	007 – 25, 13, 5	
EEX	008 – 33	Transformation de .nnnn en nn.nn
2	009 – 2	
×	010 – 61	
2	011 – 2	Introduction de 2 dans le registre X pour stockage dans I.
GTO 7	012 – 22 7	
h LBL 6	013 – 25, 13, 6	
3	014 – 3	Introduction de 3 dans le registre X pour stockage dans I.
h LBL 7	015 – 25, 13, 7	
STO f I	016 – 23, 14, 23	Stockage de l'adresse de l'opération ultérieure dans I.
$x \gtrless y$	017 – 21	Transfert de nn.nn dans le registre X
g x^2	018 – 15 3	Carré de nn. nn
EEX	019 – 33	Troncature des deux derniers chiffres du carré.
2	020 – 2	
×	021 – 61	
h INT	022 – 25 32	
EEX	023 – 33	Troncature des deux premiers chiffres du carré.
4	024 – 4	
÷	025 – 71	
h FRAC	026 – 25 33	
GTO f I	027 – 22, 14, 23	Branchement au segment de programme opérationnel.
h LBL 1	028 – 25, 13, 1	Forme du résultat : nnnn
EEX	029 – 33	
4	030 – 4	
×	031 – 61	

f FIX 0	**032–14, 11, 0**	
h RTN	**033– 25 12**	
h LBL 2	**034–25, 13, 2**	Forme du résultat : .nnnn
f FIX 4	**035–14, 11, 4**	
h RTN	**036– 25 12**	
h LBL 3	**037–25; 13, 3**	Forme du résultat : nn.nn
EEX	**038– 33**	
2	**039– 2**	
×	**040– 61**	
f FIX 2	**041–14, 11, 2**	
h RTN	**042– 25 12**	

Nous aurions aussi pu stocker les chiffres pour 100 (c.à.d. EEX 2) et les rappeler pour les utiliser dans les lignes 02-03, 08-09, 19-20 et 38-39, mais nous avons préféré ce programme plus direct pour montrer l'utilisation de l'instruction GTO f I.

Quand vous introduisez une semence de quatre chiffres sous l'une des formes indiquées, une adresse (1,2 ou 3) est placée dans le registre R_0. L'instruction GTO f I, ligne 27, utilise cette adresse pour se brancher sur le segment de programme approprié afin que le nouveau nombre aléatoire ait la même forme que la semence originale.

Prenons comme semences 5182, .5182 et 51.82. Pour exécuter le programme, mettez le calculateur sur RUN.

Appuyez sur	**Affichage**	
5182 GSB 4	**8,531**	Nombre aléatoire généré sous la forme désirée.
.5182 GSB 5	**0,8531**	
51.82 GSB 6	**85,31**	

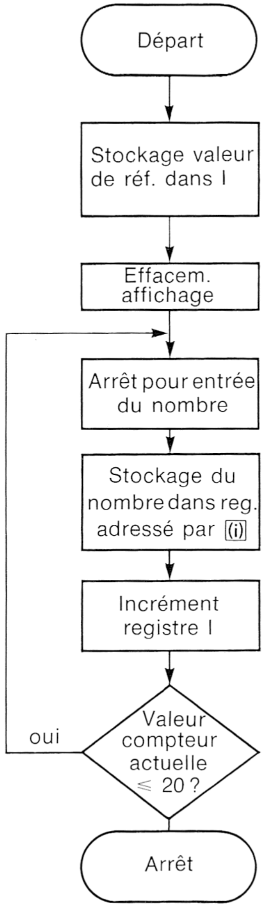
Le programme génère un nombre aléatoire sous la même forme que la semence que vous avez introduite. Pour que ce nombre devienne à son tour la semence (simulant le fonctionnement d'un véritable générateur de nombres alétoires qui se servirait d'une boucle pour réduire l'apparente prévisibilité de chaque nombre successif), continuez à appuyer sur GSB et sur la touche correspondant au label désiré :

Appuyez sur	**Affichage**
GSB 6	**77,79**
GSB 6	**51,28**
GSB 6	**29,63**

En modifiant légèrement le programme, vous pourriez remplacer l'instruction GTO f I par l'instruction GSB f I.

Problème

Créez et chargez un programme utilisant [ISG] et [STO] [f] [(i)] et qui vous permet d'entrer une série de valeurs au cours d'arrêts successifs. Ces valeurs devront être stockées suivant leur ordre d'entrée dans les registres mémoire R_0 à R_9, $R_{.0}$ à $R_{.9}$ et [I]. Aidez-vous de l'organigramme suivant :



Branchements et sous-programmes avec adressage par numéro de ligne

Si le registre I contient un nombre négatif, les instructions GTO f I et GSB f I vous permettent de vous brancher sur n'importe quel numéro de ligne en mémoire programme occupée.

Vous savez que, dans un programme en cours d'exécution, ces instructions déclenchent la recherche du LBL adressé par le nombre positif présent dans I, recherche effectuée de haut en bas de la mémoire. Le programme poursuit à cette adresse. Cependant, si le nombre présent dans le registre I est négatif, au lieu de rechercher un label, le calculateur se branche sur le numéro de ligne de mémoire programme occupée correspondant à la valeur absolue du nombre négatif présent dans I. Ceci permet d'effectuer des branchements quand tous les labels sont épuisés ou lorsqu'on désire exécuter une partie seulement d'un sous-programme ou d'un programme sans utiliser de label supplémentaire.

Exemple : Dans la section de mémoire programme ci-dessous, le registre I contient la valeur − 35. Au moment de l'exécution de l'instruction GTO f I, ligne 047, le programme saute immédiatement à la ligne 035 et poursuit.

	h y^x
034–	3
035–	STO 3
036–	4
037–	5
038–	g R↓
039–	h SF 0
040–	h RTN
041–	h LBL B
042–	f LOG
043–	3
044–	5
045–	CHS
046–	STO f I
047–	GTO f I
048–	g TAN^{-1}

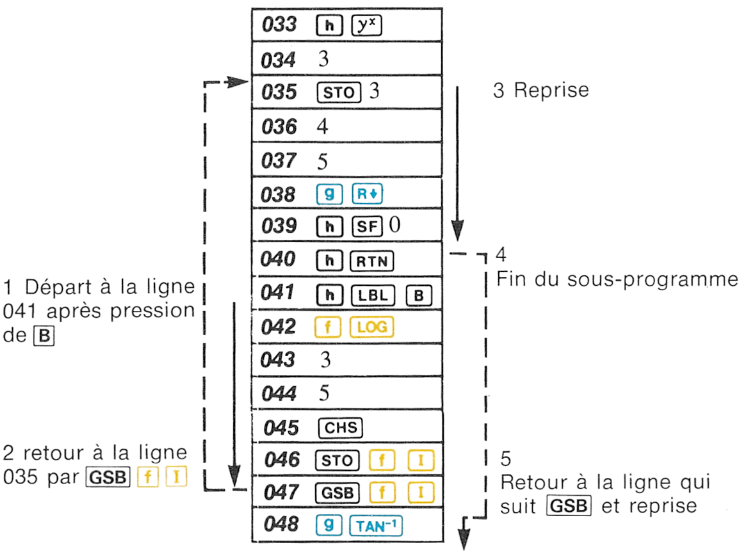
Comme le registre I contient − 35, l'instruction GTO f I branche sur la ligne 035.

Si GTO f I est exécutée au milieu d'un programme, le programme poursuit jusqu'à l'instruction RTN ou R/S suivante, puis s'arrête. Si les instructions mentionnées plus haut sont chargées dans le calculateur et que vous appuyez sur B, les lignes 041 à 047 sont exécutées dans l'ordre. Puis, le programme revient en arrière et reprend à la ligne 035 pour exécuter 036, 037, etc., jusqu'à la rencontre de RTN dans la ligne 040. Il s'arrête alors et retourne à la ligne 000.

Notons que la pression de GTO f I produit les mêmes résultats que l'exécution de cette instruction au milieu d'un programme, mais le calculateur s'arrête au numéro de ligne spécifié sans poursuivre le programme.

Si le registre I contient un nombre négatif, GSB f I branche également sur la ligne de mémoire programme occupée correspondant à la valeur absolue du nombre négatif présent dans I. Cependant, comme dans le cas de GSB avec des labels, les instructions suivantes sont exécutées comme un sous-programme. A la rencontre du RTN suivant, le programme retourne donc à l'instruction qui suit l'instruction GSB f I.

La section de mémoire programme ci-dessous montre le fonctionnement de GSB f I. La pression de B provoque le stockage de − 35 dans le registre I. Au moment de l'exécution de GSB f I, ligne 047, le programme retourne à la ligne 035 et poursuit. A la rencontre de l'instruction RTN, ligne 040, il saute à la ligne 040 et poursuit.

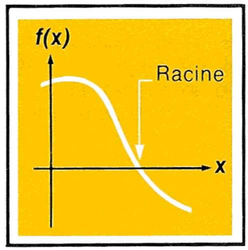


GSB f I permet, comme GTO f I, de sauter à une ligne déterminée de la mémoire programme sans exécuter tout le programme. La pression de GSB f I provoque un saut à la ligne spécifiée par la valeur absolue du nombre négatif contenu dans I et le départ du programme à cette ligne. Mais contrairement à ce qui se passe lorsque l'instruction GSB f I est exécutée au milieu d'un programme, RTN provoque un retour à la ligne 000 et l'arrêt du programme.

RECHERCHE DES RACINES D'UNE ÉQUATION

Dans bien des applications, la solution d'un problème passe par la résolution d'une équation de la forme f(x) = 0*.

Il faut donc chercher les valeurs réelles de x qui satisfont l'équation. Chaque valeur de x est appelée racine de l'équation f(x) = 0 et zéro de la fonction f(x). Pour beaucoup de problèmes, les racines d'une équation peuvent être déterminées de façon analytique par manipulation algébrique. Lorsque cela n'est pas possible, des techniques numériques permettent de calculer ces racines. La fonction [SOLVE] du HP-34C utilise une technique numérique particulièrement performante et vous permet de calculer les racines pour de nombreux types d'équations.



UTILISATION DE [SOLVE]

Règles de base :

1. introduction d'un sous-programme évaluant la fonction f(x) pour zéro. Ce sous-programme doit commencer par une instruction [R] [LBL] suivie de 0, 1, 2, 3, [A] ou [B] et doit utiliser la valeur de f(x) contenue dans le registre X ;

2. introduction dans les registres X et Y de deux estimations initiales de la racine cherchée, séparées par [ENTER↑]. Ces estimations indiquent simplement au calculateur l'intervalle pour x dans lequel il est susceptible de trouver une racine de f(x) = 0 ;

---

* En fait, toute équation d'une seule variable peut être exprimée sous cette forme. Par exemple, f(x) = a est équivalent à f(x) − a = 0 et f(x) = g(x) = f(x) − g(x) = 0.

3. pression de [f] [SOLVE] suivi du label du sous-programme. Le calculateur cherche alors le zéro de la fonction et affiche le résultat. Si la fonction égale zéro pour plus d'une valeur, le calculateur s'arrête dès que [SOLVE] a trouvé une des valeurs. Pour trouver les autres valeurs, il vous faudra introduire des estimations différentes et recommencer le calcul.

Immédiatement avant le calcul effectif, [SOLVE] place une valeur de x dans les registres X, Y, Z et T. Cette valeur est ensuite utilisée par votre sous-programme pour calculer f(x). La pile étant entièrement remplie par la valeur x, le nombre est en permanence disponible.

Exemple : utilisez [SOLVE] pour calculer les valeurs de x pour lesquelles :

$$f(x) = x^2 - 3x - 10 = 0.$$

Réécrivez f(x) à l'aide de la méthode de Horner (fin du chapitre 3) pour une programmation plus simple :

$$f(x) = (x - 3)\,x - 10 = 0.$$

Placez le commutateur PRGM - RUN sur PRGM et introduisez votre sous-programme d'évaluation de f(x) :

Appuyez sur	Affichage	
[f] CLEAR [PRGM]	*000–*	Efface la mémoire pro-
[h] [LBL] 0	*001– 25, 13, 0*	gramme
3	*002– 3*	Débute par l'instruction [LBL]
[−]	*003– 41*	
[×]	*004– 61*	
1	*005– 1*	
0	*006– 0*	
[−]	*007– 41*	
[h] [RTN]	*008– 25 12*	

Placez ensuite le commutateur PRGM - RUN sur RUN. Introduisez les deux estimations initiales dans les registres X et Y. Testez les valeurs 0 et 10 pour une racine positive.

Appuyez sur	**Affichage** [1]	
0 ENTER↑	*0,0000*	
10	*10,*	estimations

Vous pouvez alors calculer les racines en appuyant sur f SOLVE 0. Le calculateur n'affiche pas la réponse immédiatement car il utilise un processus itératif pour estimer la racine. L'algorithme analyse votre fonction en l'échantillonnant par répétitions successives de votre sous-programme. Le calcul d'une racine prend en général de 30 secondes à deux minutes ; le processus peut néanmoins être parfois plus long.

Appuyez sur f SOLVE 0 et attendez la réponse

Appuyez sur	**Affichage**
f SOLVE 0	*5,000*

Après que la fonction SOLVE ait calculé et affiché la racine, vous pouvez vérifier que la valeur affichée est bien une racine de $f(x) = 0$ en visualisant la pile. Vous avez vu précédemment que le registre X contient la racine désirée. Le registre Y contient l'estimation précédente, qui doit être très proche de la racine calculée. Le registre Z contient la valeur de votre fonction pour la racine affichée.

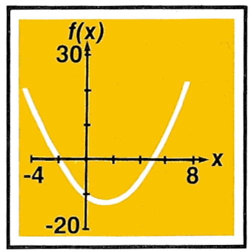
Appuyez sur	**Affichage**	
g R↓	*5,0000*	estimation précédente
g R↓	*0,0000*	valeur de la fonction pour la racine calculée, $f(x) = 0$

1. Appuyez sur f FIX 4 pour obtenir des affichages identiques. Le format d'affichage n'influe pas sur la fonction SOLVE .

Les équations quadratiques comme celles de cet exemple peuvent avoir deux racines. Introduisez deux nouvelles estimations pour chercher la seconde racine. Prenez par exemple 0 et − 10 pour chercher une racine négative.

Appuyez sur	Affichage	
0 ENTER↑	**0,0000**	Estimations initiales
10 CHS	**−10,**	
f SOLVE 0	**−2,0000**	Seconde racine
g R↓	**−2,0000**	Estimation précédente de la racine
g R↓	**0,0000**	Valeur de f(x) pour la seconde racine

Nous avons donc trouvé deux racines de f(x) = 0. Notez qu'il aurait été possible de résoudre cette équation quadratique algébriquement, le résultat aurait été le même.



Graphe de $f(x) = x^2 - 3x - 10$

L'intérêt de la touche SOLVE est plus évident lorsque la racine d'une équation ne peut pas être trouvée algébriquement.

Exemple : Un objet est jeté en l'air à une vitesse de 50 mètres/seconde. Si la hauteur qu'il atteint est exprimée par
$h = 50000(1\text{-}e^{-t/20})\text{-}200\ t$,

combien de temps lui faudra-t-il pour retomber au sol ? Dans cette équation, h est la hauteur en mètres et t le temps en secondes.

Solution : elle consiste à chercher la valeur positive de t pour laquelle h =0.

Mettez le commutateur PRGM - RUN sur PRGM et introduisez un sous-programme calculant la hauteur.

Appuyez sur	**affichage**	
[h] [LBL] [A]	**001–25, 13, 11**	Départ à l'instruction [LBL]
2	**002– 2**	
0	**003– 0**	
[÷]	**004– 71**	
[CHS]	**005– 32**	
[g] [e^x]	**006– 15 1**	
[CHS]	**007– 32**	
1	**008– 1**	
[+]	**009– 51**	
5	**010– 5**	
0	**011– 0**	
0	**012– 0**	
0	**013– 0**	
[×]	**014– 61**	
[x⇄y]	**015– 21**	Introduction de la valeur de x dans le registre X.
2	**016– 2**	
0	**017– 0**	
0	**018– 0**	
[×]	**019– 61**	
[–]	**020– 41**	
[h] [RTN]	**021– 25 12**	

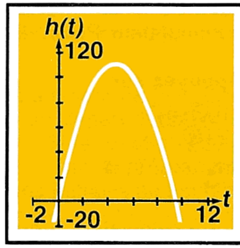
Mettez le commutateur PRGM-RUN sur RUN. Introduisez deux estimations initiales du temps (par exemple 5 et 6 secondes) et exécutez [SOLVE].

Appuyez sur	**Affichage**	
5 [ENTER↑]	**5,0000**	Estimations initiales
6	**6,**	
[f] [SOLVE] [A]	**9,2843**	Racine recherchée.

Vérifiez la racine en examinant les registres Y et Z.

Appuyez sur	**Affichage**	
[g] [R↓]	**9,2843**	Estimation précédente de la racine
[g] [R↓]	**0,0000**	Solution de la fonction : h = 0

L'objet retombe donc au sol 9.2843 secondes après avoir été jeté en l'air.



CAS DE RACINE INEXISTANTE

Il arrive qu'une équation n'ait pas de racine (c'est-à-dire qu'il n'existe pas de valeur réelle où x résout l'équation). Dans ce cas, il est évident que votre HP-34C ne pourra pas trouver de racine et il affiche une erreur 6.

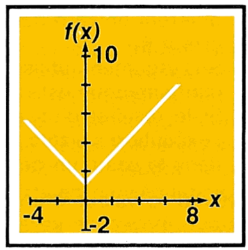
Exemple : Prenons l'équation

$$|x| = -1$$

qui ne peut être résolue puisque la fonction ayant une valeur absolue n'est jamais négative. Exprimons cette équation sous la forme voulue :

$$|x| + 1 = 0$$

Graphe de $f(x) = |x| + 1$



et essayons d'utiliser SOLVE pour trouver une solution. Mettons le commutateur PRGM-RUN sur PRGM et introduisons le sous-programme.

Appuyez sur	Affichage	
h LBL 1	001–25, 13, 1	Départ du sous-programme à l'instruction LBL
h ABS	002– 25 34	
1	003– 1	
+	004– 51	
h RTN	005– 25 12	

Comme la valeur d'une fonction absolue est minimale au voisinage d'un argument nul, choisissez des estimations initiales dans cette région, par exemple 1 et − 1. Essayez ensuite de chercher la racine. Mettez le commutateur PRGM-RUN sur RUN :

Appuyez sur	Affichage	
1 [ENTER↑]	**1,0000**	Estimations initiales
1 [CHS]	**−1,**	
[f] [SOLVE] 1	**Error 6**	Cet affichage indique qu'il n'y a pas de racine.

Votre calculateur a donc abandonné ses calculs quand il a vu que, dans le domaine de x qui lui avait été indiqué, il n'existait pas de racine de f (x) = 0. L'erreur 6 ne signale pas une tentative d'opération illicite ; elle ne fait qu'indiquer qu'il n'y a pas de racine à l'endroit présumé (le calculateur se base pour cela sur les estimations qui lui ont été fournies).

Si le HP-34C arrête ses calculs et affiche un message d'erreur, une des quatre conditions suivantes est vérifiée :

- Si, malgré plusieurs itérations, la valeur trouvée pour la fonction spécifiée reste constante, l'exécution s'arrête sur une erreur 6.
- Si plusieurs échantillonnages indiquent que la fonction a un minimum différent de zéro dans la région examinée, l'exécution s'arrête sur une erreur 6.
- Si l'argument utilisé par votre sous-programme dans une opération mathématique est incorrect, l'exécution s'arrête sur une erreur 0.
- Si le résultat d'un calcul dépasse une valeur de $9.999999999 \times 10^{99}$, l'exécution s'arrête avec affichage de 9 plus un signe (ou de l'erreur 1 dans le cas d'un dépassement de capacité d'un registre).

Quand la valeur d'une fonction est constante, le sous-programme n'a aucun moyen de savoir si cette valeur tend vers zéro. Ceci peut être le cas d'une fonction dont les 10 chiffres de poids fort **sont** constants (par exemple, quand son graphe se réduit à une asymptote horizontale non nulle) ou d'une fonction ayant une zone locale plate relativement large par rapport au domaine des valeurs de x examinées.

Quand la fonction atteint un minimum différent de zéro, le sous-programme a effectué logiquement une séquence d'échantillonnages où la valeur de la fonction continuait à décroître. Mais il n'a pas trouvé de valeur de x à laquelle le graphe de la fonction touchait ou coupait l'axe des x.

Les deux derniers cas mettent en évidence un défaut éventuel du sous-programme plutôt qu'une limite du programme de calcul des racines. Il est parfois possible d'éviter des problèmes en partant d'estimations qui concentrent la recherche dans une région où ces problèmes ne surviennent pas. Cependant, le programme [SOLVE] étant très « entreprenant », il ira faire ses recherches dans une région assez vaste. Il est donc préférable de tester ou de rectifier les arguments suspects avant d'effectuer une opération (par exemple, en utilisant ABS avant $\sqrt{x}$). On peut également réduire les variables pour éviter d'avoir des nombres trop grands.

Le succès de l'opération de SOLVE dépend principalement de la nature de la fonction analysée et des estimations fournies au départ. Il ne suffit pas qu'une racine existe pour que SOLVE la trouve. Si la fonction f(x) a une asymptote horizontale non nulle ou un minimum local, le programme ne trouvera la racine de f(x) = 0 que si les estimations initiales ne limitent pas la recherche à l'une de ces zones non productives — et si la racine existe réellement.

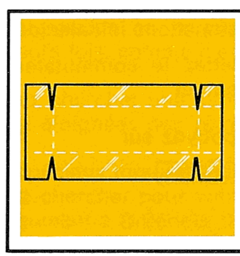
CHOIX DES ESTIMATIONS INITIALES

Lorsque vous cherchez la racine d'une équation avec SOLVE, les deux estimations initiales que vous fournissez au calculateur déterminent le point de départ de la recherche des valeurs de la variable x. En général, vous avez d'autant plus de chances de trouver une racine que vous comprenez mieux la fonction que vous analysez. Des estimations intelligentes et réalistes facilitent considérablement l'opération.

Le choix des estimations initiales repose sur plusieurs facteurs.

Si le domaine des valeurs dans lequel il est convenable de trouver une solution pour la variable x est limité, il vaut mieux choisir les estimations initiales dans ce domaine. Une équation applicable à un problème réel a souvent, en dehors de la solution recherchée, des racines qui n'ont physiquement pas de sens. Ceci est dû au fait que l'équation analysée ne convient qu'entre certaines limites de la variable. Il faudra tenir compte de cette restriction et interpréter les résultats en conséquence.

Si vous connaissez le comportement de la fonction f(x) lorsqu'elle varie pour différentes valeurs de x, vous pouvez prendre des estimations initiales proches du zéro de la fonction. Vous pouvez aussi éviter les domaines où x pose des problèmes tels que ceux où la fonction prend une valeur relativement constante ou ceux où la fonction passe par un minimum.



Exemple : On veut fabriquer une boîte sans couvercle ayant un volume de 7,5 décimètres cubes à l'aide d'un morceau de tôle rectangulaire de 4 décimètres x 8 décimètres de côtés. Comment faut-il plier la tôle (une forme haute est préférable à une forme basse) ?

Solution : Il s'agit de trouver la hauteur de la boîte (c'est-à-dire la quantité à replier le long des quatre côtés) donnant le volume spécifié. Si x est la hauteur (ou la quantité à replier), la longueur de la boîte est égale à (8-2x) et sa largeur est (4-2x). Le volume V est donné par la formule :

$V = (8-2x)(4-2x)x$

En développant l'expression et en utilisant la méthode d'Horner, cette équation peut être ramenée à :

$V = 4((x - 6)x + 8)x$

Pour trouver V = 7.5, il faut calculer les valeurs de x pour lesquelles :

$f(x) = 4((x - 6)x + 8) \times - 7.5 = 0$

Mettez le commutateur PRGM-RUN sur PRGM ▯■ et introduisez un sous-programme calculant f(x).

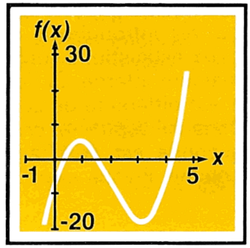
Appuyez sur	Affichage	
[h] [LBL] 3	001–25, 13, 3	Départ à l'instruction [LBL]
6	002– 6	
[−]	003– 41	
[×]	004– 61	
8	005– 8	
[+]	006– 51	
[×]	007– 61	
4	008– 4	
[×]	009– 61	
7	010– 7	
[•]	011– 73	
5	012– 5	
[−]	013– 41	
[h] [RTN]	014– 25 12	

Avec le volume indiqué, on pourrait fabriquer soit une boîte haute et mince, soît une boîte basse et large. Comme on préfère la boîte haute, les estimations initiales de la hauteur devront être plus grandes. Mais comme la tôle n'a que 4 décimètres de largeur, il n'est physiquement pas possible de choisir des hauteurs supérieures à 2. On prendra donc des estimations initiales de 1 et 2.

Mettez le commutateur PRGM-RUN sur ■▯ RUN et recherchez la hauteur.

Appuyez sur	Affichage	
1 [ENTER↑]	1,0000	Estimations initiales.
2	2,	
[f] [SOLVE] 3	1,5000	Hauteur recherchée.
[g] [R↓]	1,5000	Estimation précédente.
[g] [R↓]	0,0000	Solution de f(x).

En prenant une hauteur de 1,5 décimètre, on obtient une boîte de 5,0 × 1,0 × 1,5 décimètre.



Si vous ne tenez pas compte de la limite supérieure de la hauteur et que vous partez d'estimations de 3 et 4 décimètres (toujours inférieures à la largeur), vous obtiendrez une hauteur de 4,2026 décimètres, donc une racine qui n'a physiquement pas de sens. Si, au contraire vous partez d'estimations faibles, telles 0 et 1 décimètre, vous obtiendrez une hauteur de 0,2974 décimètre, qui correspondrait à une boîte courte et large.

Graphe de $f(x) = (8-2x)(4-2x)$

Pour étudier le comportement d'une fonction, vous pouvez facilement évaluer la fonction à une ou plusieurs valeurs de x si votre sous-programme est en mémoire programme. Pour cela, introduisez la valeur de x dans le registre X puis appuyez sur ENTER ENTER ENTER pour remplir la pile. Calculez la valeur de la fonction en appuyant sur A, B ou bien sur GSB suivi du label de votre fonction. Les valeurs calculées peuvent être représentées par un graphe donnant la courbe de la fonction. Cette procédure est particulièrement utile dans le cas d'une fonction dont vous ignorez le comportement. Une fonction apparemment simple peut avoir une courbe qui présente des variations inattendues. Une racine proche d'une variation localisée peut être difficile à trouver sauf si vous partez d'estimations au voisinage de la racine.

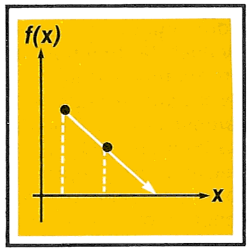
Si vous n'avez aucune idée de la nature de la fonction ni de l'endroit de sa solution, vous pouvez procéder par tâtonnements. Le succès de l'opération dépend en partie de la fonction elle-même. La méthode par tâtonnements réussit souvent, mais pas toujours.

- Si vous partez de deux estimations positives ou négatives relativement faibles et que la courbe de la fonction n'a pas d'asymptote horizontale, le programme recherchera la solution la plus positive ou la plus négative (à moins que la fonction n'oscille plusieurs fois, comme c'est le cas des fonctions trigonométriques).
- Si vous avez déjà résolu la fonction, vous pouvez chercher une autre solution en prenant des estimations relativement éloignées des zéros connus.
- Beaucoup de fonctions ont des comportements spéciaux lorsque leurs arguments se rapprochent de zéro. Vous pouvez chercher pour votre fonction des valeurs de x pour lesquelles tout argument à l'intérieur de la fonction s'annule, puis prendre des estimations égales à ou proches de ces valeurs.

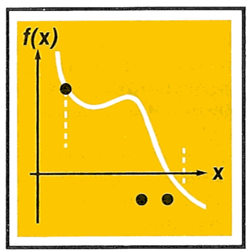
Bien qu'en général les deux estimations initiales soient différentes, vous pouvez introduire la même estimation dans les registres X et Y. Si les deux estimations sont identiques, le calculateur en crée une seconde de manière interne. Si l'estimation unique introduite dans les registres X et Y est différente de zéro, la seconde diffère de la première par une unité dans le septième chiffre significatif. Si votre estimation est nulle, la seconde est prise égale à 1×10^{-7}. L'opération se poursuit ensuite comme si vous aviez introduit deux estimations.

Principe de fonctionnement de SOLVE

Pour bien utiliser SOLVE, quelques connaissances de base sur le fonctionnement de l'algorithme sont nécessaires.



En recherchant la solution (0) d'une fonction donnée, l'algorithme applique à la fonction deux ou trois estimations précédentes pour approcher la forme de la courbe de la fonction. Cette forme lui sert à prédire une nouvelle valeur ou estimation où la courbe pourrait intercepter l'axe des ×. Le sous-programme évalue ensuite la fonction avec cette nouvelle valeur. Cette procédure est reprise plusieurs fois par l'algorithme de SOLVE.



Si deux estimations donnent des résultats de signes opposés, l'algorithme comprend que la courbe de la fonction doit intercepter l'axe des x en un point au moins de l'intervalle entre ces estimations. L'algorithme resserre systématiquement l'intervalle jusqu'à ce qu'il trouve la racine de l'équation.

On obtient une racine lorsque la fonction calculée a une valeur nulle ou si deux estimations dont la différence entre les chiffres de poids faibles (dizaines) est inférieure à deux ou trois unités donnent des résultats de signes opposés.

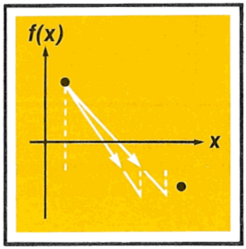
Dans ce cas, l'exécution est arrêtée et l'estimation est affichée.

Comme nous l'avons vu plus haut, l'apparition d'autres situations dans le processus itératif indique que l'équation ne peut être résolue. Dans ces cas, il n'est logiquement pas possible de prédire une nouvelle valeur rapprochant la fonction de 0. Il en résulte une erreur 6.

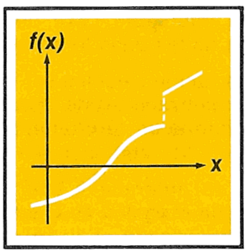
Notez que les estimations que vous fournissez au départ servent à lancer le processus de « prévision ». Si les estimations sont bien choisies, les prédictions seront plus exactes et par conséquent la solution plus facile à trouver.

L'algorithme $\boxed{\text{SOLVE}}$ trouvera **toujours** une racine, en admettant qu'elle existe, si l'une des quatre conditions suivantes est remplie :

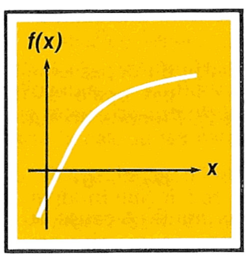
- Deux estimations quelconques donnent des résultats de signes opposés



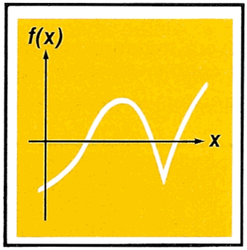
- La fonction est monotone, ce qui signifie que *f (x)* décroît ou croît chaque fois que x croît.



- La courbe de la fonction est totalement convexe ou totalement concave.



- Les seuls minima et maxima locaux du graphe de la fonction apparaissent entre deux zéros adjacents de la fonction.



On suppose par ailleurs que l'algorithme SOLVE n'est pas interrompu par une opération incorrecte ni par un dépassement de capacité.

PRÉCISION DES RÉSULTATS

La touche SOLVE effectue des calculs de racine précis. Si l'on applique le résultat affiché en fin d'opération à la fonction calculée, on obtient un résultat nul ou un nombre de 10 chiffres pratiques adjacent au point où la courbe de la fonction intercepte l'axe des x. Cette racine a une précision de deux ou trois unités dans le dixième chiffre significatif.

Dans la plupart des cas, la racine calculée est une estimation précise de la racine théorique (infiniment précise) de l'équation. Cependant, il peut arriver que le calculateur donne un résultat apparemment différent de la valeur théorique prévue.

Si le résultat d'un calcul est inférieur à une valeur de $1{,}000000000 \times 10^{-99}$, il est considéré comme égal à 0. C'est ce que l'on appelle un dépassement inférieur de capacité. S'il se produit un dépassement inférieur de capacité dans un domaine de valeurs de x et que ce dépassement influe sur la valeur de la fonction, la racine trouvée dans ce domaine risque d'être imprécise. Par exemple, l'équation

$$x^4 = 0$$

a une racine à $x = 0$. A cause du dépassement inférieur de capacité, la racine obtenue par SOLVE est égale à 1,5060 – 25 (en partant d'estimations égales à 1 et 2). Prenons un autre exemple et considérons l'équation

$$1/x^2 = 0$$

dont la racine a une valeur infinie. A cause du dépassement inférieur de capacité, la racine obtenue par SOLVE est égale à 3,1707 E 49 (en partant d'estimations de 10 et 20). Dans chacun de ces exemples, l'algorithme a trouvé une valeur de x à laquelle la fonction calculée s'annule. Ces résultats sont donc faciles à interpréter si l'on connaît les effets d'un dépassement inférieur de capacité.

La précision d'un résultat peut parfois être mise en cause par une erreur d'arrondi qui consiste à tronquer à 10 chiffres significatifs un nombre infiniment précis. Si votre sous-programme demande une extrême précision pour calculer correctement la fonction dans un certain domaine de valeurs de x, le résultat fourni par SOLVE risque d'être faux. Ainsi, l'équation

$$| x^2 - 5 | = 0$$

a une racine à $x = \sqrt{5}$. Comme il n'existe pas de nombre à 10 chiffres **exactement** égal à $\sqrt{5}$, SOLVE entraine une erreur 6 (quelles que soient les estimations initiales) puisque la fonction n'est jamais égale à zéro ou change de signe. D'autre part, l'équation

$$[(|x| + 1) + 10^{15}]^2 = 10^{30}$$

n'a pas de racines puisque le côté gauche de l'équation est toujours supérieur au côté droit. Mais à cause de l'arrondi effectué dans le calcul de

$$f(x) = [(| x | + 1) + 10^{15}]^2 - 10^{30},$$

on trouve une racine de 1.000 avec des estimations initiales de 1 et 3. Si vous savez qu'il s'agit d'une erreur d'arrondi, vous pouvez en tenir compte dans l'évaluation des résultats obtenus et peut-être réécrire la fonction afin d'en réduire les effets.

Dans de nombreux cas pratiques, les paramètres d'une équation — ou l'équation elle-même — ne sont que des **approximations.** Les paramètres physiques ont une précision (ou imprécision) intrinsèque. Les représentations mathématiques des processus physiques ne sont que des modèles de ces processus, dont la précision est fonction de la véracité des hypothèses sur lesquelles ils reposent. Mais vous pouvez tirer profit de ces imprécisions si vous en êtes conscient. En concevant votre sous-programme de telle manière qu'il vous rende une fonction nulle lorsque la valeur calculée est négligeable en pratique, vous pouvez en général réduire considérablement le temps de calcul des racines par SOLVE — et plus particulièrement dans le cas où les calculs seraient longs.

Exemple : Un lanceur de disque lance d'habitude son disque à des distances de 105 mètres. Aujourd'hui, il bat son record et l'envoie à 107 mètres. Combien de temps faut-il à son disque pour atteindre 107 mètres ?

Solution : La solution recherchée est la valeur de t pour laquelle h = 107. Le sous-programme de l'exemple précédent calcule la hauteur du disque. Ce sous-programme peut vous servir à calculer

$$f(t) = h(t) - 107.$$

Mettez le commutateur PRGM-RUN sur PRGM et introduisez le sous-programme calculant f(t).

Appuyez sur	Affichage	
h LBL B	001–25, 13, 12	Départ à l'instruction LBL
GSB A	002– 13 11	Calcul de h(t)
1	003– 1	Calcul de h (t) – 107
0	004– 0	
7	005– 7	
−	006– 41	
h RTN	007– 25 12	

Mettez maintenant le commutateur PRGM-RUN sur RUN. Pour trouver le premier temps t auquel la hauteur est égale à 107 mètres, prenez comme estimations initiales 0 et 1.

Appuyez sur	Affichage	
0 ENTER↑	0,0000	Estimations initiales
1	1,	
f SOLVE B	4,1718	Racine recherchée
g R↓	4,1718	Estimation précédente de la
g R↓	0,0000	racine Solution de f(t)

Le disque met donc 4.1718 secondes pour atteindre une hauteur d'exactement 107 mètres (et le calculateur met environ 1 minutes pour trouver cette solution).

Supposons que la fonction h(t) ne soit précise qu'au mètre près. Vous pouvez modifier votre sous-programme de manière qu'il vous donne f(t) = 0 lorsque la valeur calculée de f(t) est inférieur à 0,5 mètres. Mettez le commutateur PRGM-RUN sur RUN et introduisez les changements suivants dans votre sous-programme :

Appuyez sur	Affichage	
GTO • 006	006– 41	Ligne précédant l'instruction RTN
h ABS	007– 25 34	Valeur de f(t)
•	008– 73	Précision
5	009– 5	

f x>y	010–	14 51	0 si précision > valeur
CLx	011–	34	
g x=0	012–	15 61	Établissement de f(t) si valeur différente de 0.
h LST x	013–	25 0	

Mettez le commutateur PRGM-RUN sur RUN.

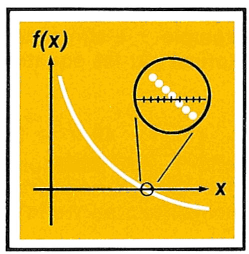
0 ENTER↑	0,0000	
1	1,	
f SOLVE B	4,0681	Racine recherchée
g R↓	4,0681	Estimation précédente de la racine.
g R↓	0,0000	Solution de f(t) modifiée.

Au bout de 4,0681 secondes, le disque atteint une hauteur de 107 ± 0,5 mètres. Cette solution est correcte bien que différente du résultat précédent, compte tenu de l'incertitude de l'équation de la hauteur, et elle est trouvée en deux fois moins de temps que la précédente.

INTERPRÉTATION DES RÉSULTATS

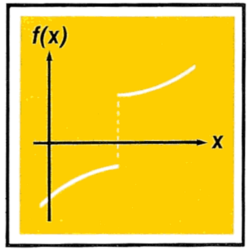
Les nombres placés par SOLVE dans les registres X, Y et Z vous permettent d'évaluer les résultats de la recherche d'une racine pour votre équation*. Ces résultats sont significatifs même si vous n'avez pu trouver de racine.

Quand SOLVE trouve une racine de la fonction donnée, les valeurs de la racine et de la fonction sont placées dans les registres X et Z. En principe, la fonction doit avoir une valeur nulle. Mais elle peut aussi avoir une valeur différente de zéro — et ceci est acceptable — indiquant que sa courbe coupe apparemment l'axe des x à une distance infinitésimale de la racine calculée. Dans la plupart des cas, la fonction aura une valeur relativement proche de zéro.

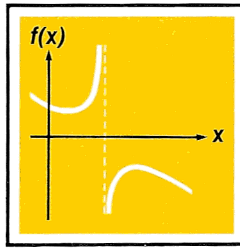


* Le nombre placé dans le registre T est celui qui est resté dans le registre Y à l'issue de l'exécution de votre sous-programme de calcul de fonction. En général, ce nombre n'a pas d'intérêt.

Un autre cas où SOLVE trouve une racine pour une fonction différente de zéro mérite d'être souligné. Si la courbe de la fonction présente une discontinuité à travers l'axe des x, SOLVE donne comme racine une valeur de x adjacente à la discontinuité. Cette solution est raisonnable puisqu'un grand changement dans la valeur de la fonction entre deux valeurs adjacentes de x peut résulter d'une transition continue très rapide. Comme l'algorithme est incapable de résoudre ce problème, il affiche la racine et la soumet à votre interprétation.



Une fonction peut avoir un **pôle** où sa valeur tend vers l'infini. Si la fonction change de signe à un pôle, la valeur correspondante de x paraît être une racine possible de l'équation comme pour toute autre discontinuité traversant l'axe des x. Mais pour ce type de fonctions, la valeur de la fonction placée dans le registre Z et correspondant à la racine trouvée sera relativement grande. Si la valeur de x au pôle de la fonction correspond **exactement** à 10 chiffres, le sous-programme essaye cette valeur et s'arrête prématurément sur une indication d'erreur ou de dépassement de capacité. Dans ce cas, l'opération SOLVE reste inachevée. On peut éviter cet ennui en plaçant judicieusement une condition dans le sous-programme.



Exemple : L'effort de cisaillement exercé sur une poutre peut être exprimé par les équations suivantes :

$$Q = \begin{cases} 3x^3 - 45x^2 + 350 & \text{pour } 0 < x < 10 \\ 1000 & \text{pour } 10 \leqslant x < 14 \end{cases}$$

Q étant l'effort de cisaillement en newtons et x la distance d'une extrémité en mètres. Écrivez un sous-programme calculant l'effort de cisaillement pour une valeur quelconque de x. Utilisez SOLVE pour trouver le point où l'effort de cisaillement est nul.

Solution : l'équation de l'effort de cisaillement en prenant x compris entre 0 et 10 est plus facile à programmer si on la simplifie par la méthode d'Horner :

$Q = (3x - 45)x^2 + 350$ pour $0 < x < 10$

Mettez le commutateur PRGM-RUN sur PRGM et entrez le sous-programme :

Appuyez sur	Affichage	
f CLEAR PRGM	000–	Effacement de la mémoire programme.
h LBL 2	001–25, 13, 2	Départ de l'instruction LBL
1	002– 1	Test du domaine de X
0	003– 0	
f x≤y	004– 14 41	
GTO 9	005– 22 9	Branchement si $x \geqslant 10$.
CLx	006– 34	
3	007– 3	
×	008– 61	
4	009– 4	
5	010– 5	
−	011– 41	
×	012– 61	
×	013– 61	
3	014– 3	
5	015– 5	
0	016– 0	
+	017– 51	
h RTN	018– 25 12	
h LBL 9	019–25, 13, 9	
EEX	020– 33	
3	021– 3	
h RTN	022– 25 12	

Mettez maintenant le commutateur PRGM-RUN sur RUN. Prenez comme estimations initiales 7 et 14 pour commencer à l'extrémité extérieure de la poutre et cherchez un point où l'effort de cisaillement est nul.

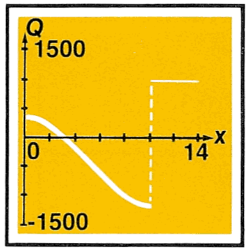
Appuyez sur	Affichage	
7 ENTER	7,0000	Estimations initiales
14	14,	
f SOLVE 2	10,0000	Racine possible.
g R↓ g R↓	1000,0000	Effort non nul.

La valeur élevée de l'effort pour la racine trouvée par SOLVE indique la présence d'une discontinuité. Celle-ci correspond à un point de la poutre où l'effort passe brusquement du négatif au positif. Commencez à l'autre extrémité de la poutre (avec des estimations de 0 et 7) et réutilisez SOLVE.

Appuyez sur	affichage	
0 ENTER	**0,0000**	Estimations initiales
7	**7,**	
f SOLVE 2	**3,1358**	Racine possible
g R↓ g R↓	**2,0000 −07**	Effort négligeable

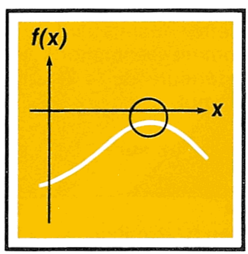
La poutre est donc soumise à un effort au cisaillement égal à 0 à environ 3,1358 mètres, effort qui subit un changement brusque à 10,000 mètres.

Graphe de **Q** en fonction de **x.**



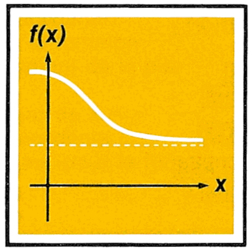
Si aucune racine n'a pu être trouvée et que l'erreur 6 est affichée, vous pouvez appuyer sur n'importe quelle touche pour effacer l'affichage et voir à quelle valeur de la racine la fonction se rapprochera le plus de zéro. En ré-examinant les nombres placés dans les registres Y et Z, vous pouvez souvent déterminer la nature de la fonction au voisinage de la racine estimée et partir de cette information.

Si l'algorithme arrête sa recherche près d'un minimum local de la valeur de la fonction, effacez l'affichage de l'erreur 6 et observez les nombres placés dans les registres X, Y et Z en descendant dans la pile. Si la valeur de la fonction sauvegardée dans le registre Z est relativement proche de 0, il est possible que vous ayiez trouvé une racine pour votre équation — le nombre rendu dans le registre X peut être un nombre à 10 chiffres très proche d'une racine théorique. Vous pouvez étudier de plus

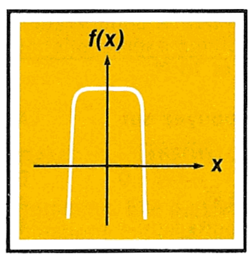


près ce minimum possible en descendant dans la pile jusqu'à ce que les estimations soient retournées dans les registres X et Y, puis ré-exécuter SOLVE en prenant ces nombres comme estimations initiales. Si vous avez effectivement trouvé un minimum, l'erreur 6 réapparaît et le nombre présent dans le registre X sera pratiquement le même qu'avant, bien que peut-être plus proche du minimum réel.

Certes, vous êtes libres d'utiliser SOLVE pour trouver un minimum local de la valeur de la fonction. Mais dans ce cas, la recherche devra se limiter à la région du minimum et n'oubliez pas que SOLVE cherchera à tout prix à trouver une solution de la fonction.



Si l'algorithme arrête sa recherche et affiche une erreur 6 parce qu'il est sur une asymptote horizontale (cas où la fonction a une valeur constante pour un grand domaine de valeurs de x), les estimations placées dans les registres X et Y sont en général très différentes. Le nombre placé dans le registre Z est la valeur de l'asymptote possible. Si vous refaites SOLVE en prenant comme estimations initiales les nombres retournés dans les registres X et Y, une asymptote horizontale risque à nouveau de provoquer une erreur 6, mais avec d'autres nombres dans les registres X et Y. La valeur de la fonction dans le registre Z sera alors la même que précédemment.



Si l'erreur 6 est affichée par suite d'une recherche concentrée dans une zone locale « plate » de la fonction, les estimations placées dans les registres X et Y seront relativement rapprochées ou extrêmement faibles. Refaites SOLVE en prenant comme estimations initiales les nombres contenus dans les registres X et Y (ou peut-être deux nombres plus éloignés l'un de l'autre). Si la valeur de la fonction ne correspond pas à un minimum ou à une valeur constante, l'algorithme poursuit sa recherche pour trouver un résultat plus significatif.

Exemple : Étudiez le comportement de la fonction

$f(x) = 3 + e^{-|x|/10} - 2e^{x^2 e^{-|x|}}$

Mettez le commutateur PRGM-RUN sur PRGM et entrez un sous-programme pour calculer f(x).

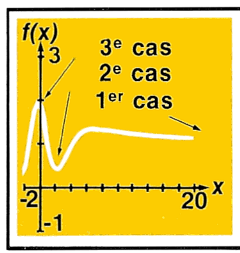
Appuyez sur	Affichage	
h LBL 0	001 – 25, 13, 0	Départ à l'instruction LBL
h ABS	002 – 25 34	
CHS	003 – 32	
g e^x	004 – 15 1	
$x \gtrless y$	005 – 21	Introduction de la valeur de x dans le registre X
g x^2	006 – 15 3	
×	007 – 61	
g e^x	008 – 15 1	
2	009 – 2	
×	010 – 61	
CHS	011 – 32	
$x \gtrless y$	012 – 21	Introduction de la valeur de x dans le registre X.
h ABS	013 – 25 34	
CHS	014 – 32	
1	015 – 1	
0	016 – 0	
÷	017 – 71	
g e^x	018 – 15 1	
+	019 – 51	
3	020 – 3	
+	021 – 51	
h RTN	022 – 25 12	

Mettez le commutateur PRGM - RUN sur RUN et exécutez SOLVE avec comme estimations initiales l'une seulement des valeurs suivantes : 10, 1 et 10^{-20}.

Appuyez sur	Affichage	
10 ENTER↑	10,0000	Estimation unique
f SOLVE 0	Error 6	
CLx	455,4335	Meilleure valeur de x
g R↓	48.026.721,85	Valeur précédente.
g R↓	1,0000	Valeur de la fonction
f R↑ f R↑	455,4335	Rétablissement de la pile

f SOLVE 0	**Error 6**	
CLx	**48.088.400,66**	Autre valeur de x.
g R↓ g R↓	**1,0000**	Même valeur pour la fonction (asymptote)
1 ENTER↑	**1,0000**	Estimation unique
f SOLVE 0	**Error 6**	
CLx	**2,1213**	Meilleure valeur de x
g R↓	**2,1471**	Valeur précédente
g R↓	**0,3788**	Valeur de la fonction
f R↑ f R↑	**2,1213**	Rétablissement de la pile
f SOLVE 0	**Erreur 6**	
CLx	**2,1213**	Même valeur de x
g R↓ g R↓	**0,3788**	Même valeur pour la fonction (minimum).
EEX CHS 20 ENTER↑	**1.0000 – 20**	Estimation unique
f SOLVE 0	**Error 6**	
CLx	**1,0000 – 20**	Meilleure valeur de x
g R↓	**1,1250 – 20**	Valeur précédente
g R↓	**2,0000**	Valeur de la fonction
f R↑ f R↑	**1,0000 – 20**	Rétablissement de la pile
f SOLVE 0	**Error 6**	
CLx	**1,1250 – 20**	Autre valeur de x
g R↓	**1,5626 – 16**	Valeur précédente
g R↓	**2,0000**	Même valeur pour la fonction.

Dans chacun de ces trois cas, SOLVE a commencé par chercher une racine dans le sens suggéré par la courbe autour de l'estimation initiale. Pour une estimation de 10, SOLVE a trouvé l'asymptote horizontale (valeur de 1,000). Pour 1, SOLVE a trouvé un minimum de 0,3788 à x = 2,1213. Pour 10^{-20}, la fonction était constante (à une valeur de 2.0000) pour le petit domaine de valeurs de x échantillonné.
Graphe de f (x)



UTILISATION DE **SOLVE** DANS UN PROGRAMME

L'opération SOLVE peut être intégrée dans un programme. Elle doit être immédiatement précédée de l'introduction des estimations initiales dans les registres X et Y. Elle s'arrête après avoir introduit une valeur de x dans le registre X et la valeur correspondante de la fonction dans le registre Z. Si la valeur de x est une racine, le programme passe à la ligne suivante. Sinon, il saute la ligne suivante. L'instruction [SOLVE] vérifie si la valeur de x est une racine et réagit suivant la règle de la condition vérifiée. Le programme peut prendre des mesures si la recherche de la racine a échoué, par exemple : partir de nouvelles estimations ou modifier un paramètre de la fonction.

La mise en œuvre de l'instruction [SOLVE] dans un programme fait appel à l'un des six niveaux d'attente de retours dans le calculateur. Comme le sous-programme appelé par [SOLVE] utilise un autre retour, il ne reste que quatre niveaux d'attente. D'autre part, si la fonction [SOLVE] est exécutée manuellement, à partir du clavier, elle n'utilise pas de niveau d'attente, de sorte qu'il en reste cinq pour les sous-programmes exécutés à l'intérieur du sous-programme appelé par [SOLVE]. Rappelons que si tous les six niveaux d'attente ont été utilisés, un nouvel appel à un sous-programme provoquera une erreur 8.

RESTRICTION D'EMPLOI DE **SOLVE**

La seule restriction à l'emploi de [SOLVE] est que [SOLVE] ne peut être utilisée de façon récursive, ce qui signifie que vous ne pouvez pas employer [SOLVE] dans un sous-programme appelé en cours d'exécution de [SOLVE]. Dans cette situation, le calculateur s'arrête et affiche erreur 5.

Cependant, il est possible de combiner [SOLVE] avec [∫] et d'exploiter ainsi certaines particularités de ces touches. Un exemple de cette utilisation combinée des touches est donné en Annexe A.

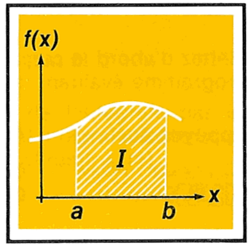
Pour plus de détails...

Consultez l'Annexe A, « particularités d'utilisation de [SOLVE] » qui présente d'autres techniques et cas d'utilisation de [SOLVE] tels que :

- Utilisation de [SOLVE] avec des polynômes.
- Recherche de plusieurs racines.
- Recherche des limites locales d'une fonction.
- Limitation du temps d'estimation.
- Utilisation de [SOLVE] avec [∫].

Chapitre 9

INTÉGRATION NUMÉRIQUE



Beaucoup de problèmes dans le domaine des mathématiques, des sciences pures et des sciences appliquées font appel à des calculs de l'intégrale définie d'une fonction. Pour une fonction f(x) et un intervalle d'intégration de à b, l'intégrale peut être exprimée mathématiquement par la formule

$$I = \int_a^b f(x)\, dx$$

La valeur I peut généralement être interprétée géométriquement comme l'aire d'une zone bornée par le graphe de f(x), l'axe des abscisses et les bornes x = a et x = b.

Lorsqu'une intégrale est difficile, voire impossible, à évaluer par des méthodes analytiques, elle peut être calculée à l'aide de techniques numériques. Ceci n'était possible au début qu'avec des programmes compliqués. Aujourd'hui, avec votre calculateur de poche HP-34C, une simple pression de touche suffit : la touche [∫] (integrate) effectue les intégrations numériques autrefois réservées aux gros calculateurs.

UTILISATION DE [∫]

Les principales règles d'utilisation de [∫] sont :

1. Introduction d'un sous-programme évaluant la fonction f(x) à intégrer. Ce sous-programme doit commencer par l'instruction [h] [LBL] suivie de 0, 1, 2, 3, [A] ou [B] et doit placer la valeur de f(x) dans le registre X.
2. Introduction de la borne inférieure de l'intervalle d'intégration (a) dans le registre affiché X, puis pression de [ENTER↑] pour la monter dans le registre Y.
3. Introduction de la borne supérieure de l'intervalle d'intégration (b) dans le registre X.
4. Pression de [f] [∫] suivie du label du sous-programme.

Exemple : La fonction de Bessel de premier type de degré 0, utilisée dans certains problèmes de physique et de sciences appliquées, peut-être exprimée par la formule suivante :

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin \theta)\, d\theta$$

Calculez

$$J_0(1) = \frac{1}{\pi} \int_0^{\pi} \cos(\sin \theta)\, d\theta$$

Mettez d'abord le commutateur PRGM-RUN sur PRGM et entrez un sous-programme évaluant la fonction f(θ) = cos (sin θ).

Appuyez sur	Affichage	
f CLEAR PRGM	000 –	Effacement de la mémoire programme.
h LBL 0	001 – 25, 13, 0	Départ du sous-programme à l'instruction LBL. Une valeur de θ est supposée présente dans le registre X.
f SIN	002 – 14 7	Calcul du sin θ
f COS	003 – 14 8	Calcul du cos (sin θ).
h RTN	004 – 25 12	

Remettez le commutateur PRGM-RUN sur RUN et introduisez la borne inférieure de l'intervalle d'intégration dans le registre Y et la borne supérieure dans le registre X. Dans ce cas particulier, vous devez aussi spécifier le mode radians pour les fonctions trigonométriques.

Appuyez sur	Affichage	
0 ENTER↑	0,0000	Introduction de la borne inférieure, 0, dans le registre Y.
h π	3,1416	Introduction de la borne supérieure π, dans le registre X.
g RAD	3,1416	Mode radians pour les fonctions trigonométriques.

Vous pouvez maintenant appuyer sur f ∫ 0 pour calculer l'intégrale. Comme pour SOLVE, vous devez attendre le résultat quelques instants. En effet, votre HP-34C utilise un algorithme itératif compliqué pour le calcul des intégrales. En quelques mots, cet algorithme évalue f(x), la fonction à intégrer, et un grand nombre de valeurs de × entre les bornes de l'intervalle d'intégration. A chacune de ces valeurs, il évalue la fonction en exécutant le sous-programme que vous avez écrit à cet effet. Rappelez-vous qu'avec certains programmes et sous-programmes que vous avez exécutés dans les chapitres précédents, la réponse ne vous parvenait qu'au bout de quelques secondes, ce qui est court si l'on pense que le calculateur doit ré-exécuter le sous-programme de nombreuses fois. Pour le calcul des intégrales, le calculateur met généralement 30 secondes à 2 minutes, et parfois davantage. Nous verrons plus loin comment réduire ce temps de calcul, mais pour le moment, appuyez sur [f] [∫] 0 et reposez-vous (ou continuez à lire) pendant que votre HP-34C fait le travail pour vous.

Appuyez sur	Affichage	
[f] [∫] 0	**2,4040**	$= \int_0^\pi \cos(\sin \theta)\, d\,\theta$

En général, il convient de multiplier la valeur de l'intégrale par les constantes éventuellement présentes en dehors de l'intégrale. Dans le cas présent, il faut multiplier l'intégrale par $1/\pi$ pour obtenir J_0 (1) :

Appuyez sur	Affichage	
[h] [π]	**3,1416**	
[÷]	**0,7652**	$J_0(1)$

Avant d'appeler le sous-programme qui évalue f(x), l'algorithme [∫], tout comme l'algorithme [SOLVE], met les valeurs de x dans les registres X, Y, Z et T. Comme chaque registre de la pile contient la valeur de x, votre sous-programme peut calculer sur ce nombre sans avoir à le rappeler d'un registre mémoire. C'est ce que nous allons voir dans les deux exemples suivants.

Remarque

Vous pouvez utiliser le sous-programme que vous avez écrit pour [∫] uniquement pour évaluer la fonction pour une certaine valeur de x. Si la valeur de x utilisée doit être prélevée plusieurs fois dans la pile, introduisez-la manuellement dans la pile en appuyant sur [ENTER↑] [ENTER↑] [ENTER↑] avant d'exécuter le sous-programme. Aussi, comme le calculateur met la valeur de x dans tous les registres de la pile, tous les nombres qui s'y trouvaient seront-ils remplacés par x. Par conséquent, si la pile contient des résultats intermédiaires dont vous aurez besoin après le calcul de l'intégrale, stockez ces nombres dans des registres mémoire d'où vous les rappellerez plus tard.

Exemple : La fonction de Bessel de premier type de degré 1 peut être exprimée sous la forme :

$$J_1(x) = \frac{1}{\pi}\int_0^{\pi} \cos(\theta - x \sin\theta)\, d\theta$$

Calculez
$$J_1(1) = \frac{1}{\pi}\int_0^{\pi} \cos(\theta - \sin\theta)\, d\theta$$

Mettez d'abord le commutateur PRGM - RUN sur PRGM et entrez un sous-programme évaluant la fonction $f(\theta) = \cos(\theta - \sin\theta)$.

Appuyez sur	Affichage	
[h] [LBL] 1	**001 – 25, 13, 1**	Départ du sous-programme à l'instruction [LBL]
[f] [SIN]	**002 – 14 7**	Calcul de sin θ.
[–]	**003 – 41**	Comme l'algorithme [∫] met une valeur de θ dans le registre Y avant d'exécuter ce sous-programme l'opération [–] consiste pour le moment à calculer (θ – sin θ).
[f] [COS]	**004 – 14 8**	Calcul de cos (θ – sin θ).
[h] [RTN]	**005 – 25 12**	

Remettez le commutateur PRGM-RUN sur RUN et introduisez les bornes de l'intervalle d'intégration dans les registres X et Y. Mettez-vous en mode trigonométrique radians, puis appuyez sur [f] [∫] 1 pour calculer l'intégrale. Enfin, multipliez l'intégrale par $1/\pi$ pour calculer $J_1(1)$.

Appuyez sur	Affichage	
0 [ENTER↑]	**0,0000**	Introduction de la borne inférieure dans le registre Y.
[h] [π]	**3,1416**	Introduction de la borne supérieure dans le registre X.
[g] [RAD]	**3,1416**	Mode trigonométrique radians. (Ceci est inutile si vous n'avez pas éteint votre calculateur ni changé de mode trigonométrique depuis la dernière fois.)
[f] [∫] 1	**1,3825**	$= \int_0^{\pi} \cos(\theta - \sin\theta)\, d\theta$
[h] [π] [÷]	**0,4401**	$J_1(1)$

Exemple : Dans certains problèmes sur les théories de transimission (par exemple, la transmission d'impulsions par des réseaux théoriques), l'intégrale calculée (parfois appelée **l'intégrale du sinus**) a la forme suivante :

$$Si(t) = \int_0^t \frac{\sin x}{x} dx$$

Calculez Si(2)

Mettez d'abord le commutateur PRGM - RUN sur PRGM et entrez un sous-programme évaluant la fonction f(x) = (sin x) /x*.

Appuyez sur	Affichage	
[h] [LBL] 2	**001 – 25, 13, 2**	Départ du sous-programme à l'instruction [LBL]
[f] [SIN]	**002 – 14 7**	Calcul de sin x.
[x⇄y]	**003 – 21**	Comme l'algorithme [∫] met une valeur de x dans le registre Y avant d'exécuter ce sous-programme, l'opération [x⇄y] consiste ici à retourner x dans le registre X et à mettre sin x dans le registre Y.
[÷]	**004 – 71**	Division de sin x par x.
[h] [RTN]	**005 – 15 12**	

Remettez le commutateur PRGM - RUN sur RUN et introduisez les bornes de l'intervalle d'intégration dans les registres X et Y. Mettez-vous en mode trigonométrique radians, puis appuyez sur [f] [∫] 2 pour calculer l'intégrale.

Appuyez sur	Affichage	
0 [ENTER↑]	**0,0000**	Introduction de la borne inférieure dans le registre Y.
2	**2,**	Introduction de la borne supérieure dans le registre X.

* Si le calculateur tentait d'évaluer f(x) = (sin x)/x pour x = 0, la borne inférieure de l'intervalle d'intégration, il provoquerait l'affichage d'une erreur 0, indiquant une tentative de division par zéro, et l'intégrale ne pourrait pas être calculée. Cependant, en principe, l'algorithme [∫] n'évalue *pas* les fonctions aux deux bornes de l'intervalle d'intégration, il *peut* donc calculer l'intégrale d'une fonction indéfinie en ces points. Ce n'est que lorsque les bornes de l'intervalle d'intégration sont très rapprochées, ou lorsque le nombre de points échantillonnés est extrêmement élevé, que l'algorithme évalue la fonction aux bornes de l'intervalle d'intégration.

g RAD	**2,0000**	Mode trigonométrique radians (Ceci est inutile si vous n'avez pas éteint votre calculateur et si vous n'avez pas changé de mode trigonométrique depuis la dernière fois).
f ∫ 2	**1,6054**	Si (2).

PRÉCISION DE ∫

La précision de l'intégrale d'une fonction quelconque dépend de celle de la fonction elle-même. Ainsi, la précision d'une intégrale calculée avec ∫ est limitée par celle de la fonction calculée par votre sous-programme*. Pour indiquer au calculateur la précision recherchée, prenez un format d'affichage qui ne dépasse pas le nombre de chiffres précis contenus dans les valeurs prises par la fonction**. Si vous indiquez un nombre de chiffres inférieur, le calcul de l'intégrale se fera plus rapidement***, mais alors la fonction ne sera supposée précise qu'au nombre de chiffres indiqués dans le format d'affichage. Nous verrons plus loin comment déterminer la précision de l'intégrale. Mais d'abord quelques mots sur le format d'affichage !

Vous vous rappelez que le HP-34C possède trois formats d'affichage : FIX, SCI et ENG. Le choix du format est le plus souvent une question de préférence personnelle puisque, pour beaucoup d'intégrales, les résultats obtenus seront pratiquement identiques avec les trois formats (à condition que le nombre de chiffres soit indiqué correctement, compte tenu de l'ordre de grandeur de la fonction). Comme le format SCI convient le mieux au calcul de la plupart des intégrales, nous allons choisir ce format dans la suite du texte.

---

* Il est possible que le calcul des intégrales de certaines fonctions ayant des caractéristiques particulières, telles que des pics ou des oscillations très rapides, manque de précision. **Cette possibilité est toutefois très faible.** Les caractéristiques générales des fonctions susceptibles de soulever des problèmes, ainsi que les techniques utilisées pour les résoudre, sont décrites en Annexe B.

** La précision d'une fonction dépend de considérations telles que la précision de constantes empiriques présentes dans la fonction ainsi que des erreurs d'arrondi dans les calculs. Ces points seront repris en détail en Annexe B.

***Voir explications en Annexe B.

Remarque : Rappelons que une fois le format d'affichage écart établi (FIX), (SCI), ou (ENG), vous pouvez toujours modifier nombre de chiffres affichés par l'introduction d'un nombre dans le registre I, suivie de la pression de h DSP I, comme décrit dans la section 00. Ce procédé est particulièrement utile dans le cas de l'exécution de ∫ au milieu d'un programme et est indispensable dans le cas particulier décrit en Annexe B « calcul des intégrales avec le maximum de précision ».

Comme la précision d'une intégrale est limitée par celle de la fonction indiquée dans le format d'affichage, le calculateur ne peut pas calculer exactement la valeur d'une intégrale, il ne peut que *l'approcher*. Le HP-34C met l'incertitude* de l'approximation d'une intégrale dans le registre Y pendant qu'il introduit l'approximation dans le registre X. Pour connaître la précision d'une approximation, vous pouvez vérifier son incertitude en appuyant sur x⇄y.

Exemple : Calculez l'intégrale de J_1 (1) en prenant comme format d'affichage SCI 2.

Appuyez sur	Affichage	
0 ENTER	**0,0000**	Introduction de la borne inférieure dans le registre Y.
h π	**3,1416**	Introduction de la borne supérieure dans le registre X.
g RAD	**3,1416**	Mode trigonométrique sur radians. (Ceci est inutile si vous n'avez pas éteint votre calculateur ou si vous n'avez pas changé de mode trigonométrique depuis la dernière fois que vous étiez sur radians)
f SCI 2	**3,14 00**	Format d'affichage sur SCI 2.
f ∫ 1	**1,38 00**	Approximation de l'intégrale en SCI 2.
x⇄y	**1,88 – 03**	Incertitude de l'approximation en SCI 2.

* Il n'existe pas d'algorithme d'intégration numérique capable de calculer la différence exacte entre son approximation et l'intégrale réelle. Celui de votre HP-34C calcule une « limite supérieure » de cette différence, qui est **l'incertitude** de l'approximation. Par exemple : si l'intégrale Si(2) est égale à 1.6054 ± 0.0001, l'approximation de l'intégrale est égale à 1.6054 et son incertitude est égale à 0.0001. Cela signifie que si nous ne connaissons pas la différence exacte entre l'intégrale réelle et son approximation nous savons du moins que cette différence ne dépasse pas 0.00001.

L'intégrale est égale à 1.38 ± 0.00188. Comme l'incertitude n'affecte pas l'approximation avant la troisième position décimale, vous pouvez considérer que tous les chiffres affichés sont précis. Mais en général, il est difficile de prévoir dans une approximation le nombre de chiffres qui ne seront pas affectés par son incertitude. Cela dépend de la fonction à intégrer, des bornes de l'intervalle d'intégration et du format d'affichage.

Si l'incertitude d'une approximation est supérieure aux tolérances choisies, vous pouvez la réduire en indiquant un nombre de chiffres supérieur dans le format d'affichage et reprendre l'approximation.

Quand vous reprenez une approximation, il est inutile de rentrer à nouveau les bornes de l'intervalle d'intégration dans les registres X et Y. En effet, lors du calcul d'une intégrale, plusieurs opérations parallèles de stockage sont effectuées dans les registres : stockage de l'approximation et de l'incertitude dans les registres X et Y, stockage de la borne supérieure de l'intervalle d'intégration dans le registre Z et de la borne inférieure dans le registre T. Pour transférer ensuite ces bornes dans les registres X et Y, il suffit d'appuyer sur [g] [R↓] [g] [R↓].

Exemple : L'approximation de l'intégrale de J_1 (1) doit avoir une précision de quatre positions décimales au lieu de deux.

Appuyez sur	Affichage	
[f] [SCI] 4	1,8826 – 03	Format d'affichage sur [SCI] 4.
[g] [R↓] [g] [R↓]	3,1416 00	Descente dans la pile jusqu'à apparition de la borne supérieure dans le registre X.
[f] [∫] 1	1,3825 00	Approximation de l'intégrale en [SCI] 4.
[x⇄y]	1,7091 – 05	Incertitude de l'approximation en [SCI] 4.

L'incertitude indique que cette approximation est précise à au moins quatre positions décimales. Notez qu'en format [SCI] 4, l'incertitude est égale au centième de celle d'une approximation en format [SCI] 2. En général, l'incertitude d'une approximation de [∫] décroît d'un facteur de 10 par chiffre supplémentaire dans le format d'affichage.

Dans l'exemple précédent, l'incertitude indiquait que l'approximation ne pouvait être correcte qu'à quatre positions décimales. Mais si nous affichons rapidement les 10 chiffres de l'approximation et que nous les comparons à la valeur réelle de l'intégrale (qui est en fait une approximation dont on sait que la précision porte sur un nombre suffisant de positions décimales), nous voyons que l'approximation est finalement plus précise que ne le laisserait supposer son incertitude.

Appuyez sur	Affichage	
[x⇄y]	**1,3825 00**	Affichage de l'approximation
[h] [MANT]	**1382459676**	Les 10 chiffres de l'approximation.

La valeur de cette intégrale, qui a une précision de huit positions décimales, est égale à 1,38245969. L'approximation du calculateur a une précision de **sept** positions décimales au lieu de quatre. En fait, comme l'incertitude d'une approximation est calculée avec beaucoup de prudence, **dans la plupart des cas, l'approximation obtenue sera plus précise que ne l'indique son incertitude.** Mais il est généralement impossible de prévoir la précision d'une approximation, tout ce qu'on sait, c'est que la différence entre l'approximation et l'intégrale n'est pas supérieure au nombre placé dans le registre Y.

La précision et l'incertitude de [∫] seront reprises de plus près dans l'Annexe B.

UTILISATION DE [∫] DANS UN PROGRAMME

[∫] peut figurer comme instruction dans un programme à condition que ce ne soit pas [∫] qui ait appelé le programme (comme sous-programme). En d'autres termes, [∫] ne peut pas être utilisé de manière récurrente, comme, par exemple, pour le calcul de plusieurs intégrales ; toute tentative dans ce sens donne lieu à une erreur 5. Cependant, [∫] **peut** figurer comme instruction dans un sous-programme appelé par [SOLVE]. Nous en verrons un exemple dans l'Annexe A.

L'utilisation de [∫] comme instruction dans un programme fait appel aux six niveaux d'attente du calculateur. Comme le sous-programme appelé par [∫] utilise un autre retour, il ne reste que quatre niveaux de retour. D'autre part, si la fonction [∫] est commandée à partir du clavier, elle n'utilise aucun niveau de retour, de sorte qu'il en reste cinq pour les sous-programmes exécutés à l'intérieur du programme à appel de [∫]. Rappelons que si tous les six niveaux d'attente sont occupés, tout nouvel appel à un sous-programme provoque une erreur 8.

Pour plus de détails...

Cette section vous a fourni les bases nécessaires à la bonne utilisation de [∫] dans de nombreux cas pratiques. L'annexe B « particularités sur

$\boxed{\int}$ » vous fournira quelques détails supplémentaires concernant des cas particuliers d'utilisation de $\boxed{\int}$. Cette annexe couvre les points suivants :

- Fonctionnement de $\boxed{\int}$.
- Précision incertitude et temps de calcul.
- Précision de la fonction à intégrer.
- Incertitude et format d'affichage.
- Calcul des intégrales avec un maximum de précision.
- Calcul de l'approximation actuelle d'une intégrale.
- Risques d'erreurs de sortie.
- Facteurs de prolongement du temps de calcul.

PARTICULARITÉS DE SOLVE

Le chapitre 8 donne les bases nécessaires à l'utilisation de l'algorithme SOLVE. Cette annexe présente des considérations supplémentaires concernant certaines particularités d'utilisation de SOLVE.

SOLVE ET LES POLYNÔMES

Dans de nombreux cas pratiques, les fonctions appelées **polynômes** servent à représenter des processus physiques ou des fonctions mathématiques plus complexes. Les polynômes sont faciles à comprendre et peuvent être structurés de manière à présenter un grand nombre de caractéristiques mathématiques.

Un polynôme de degré **n** peut être représenté par la formule

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

Cette fonction a au maximum n valeurs réelles de x pour lesquelles la fonction est résolue. On peut connaître la limite du nombre de solutions **positives** de cette fonction en comptant le nombre de changements de signes des coefficients si l'on examine le polynôme de gauche à droite. De même, il est possible de connaître la limite du nombre de solutions **négatives** en examinant la nouvelle fonction obtenue si l'on remplace le x du polynôme original par $-x$. Si le nombre de solutions positives ou négatives réelles obtenu est inférieur à la limite, la différence sera exprimée par un nombre pair. (Ces relations sont connues sous la Loi des signes de Descartes).

Considérons, par exemple, la fonction polynomiale de degré 3

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

Cette fonction ne peut pas avoir plus de trois solutions réelles. Elle a au maximum deux solutions positives réelles (notez les changements de signe du premier au second terme et du troisième au quatrième) et une solution négative réelle (obtenue pour $f(-x) = -x^3 - 3x^2 + 6x + 8$).

Pour programmer une fonction polynômiale, la meilleure technique consiste à la réécrire sous une forme légèrement différente utilisant une multiplication imbriquée. Cette technique est parfois appelée « méthode d'Horner ». A titre d'illustration, récrivons la fonction de l'exemple précédent :

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

Cette représentation est plus facile à programmer et à exécuter que la forme originale, d'autant plus que la pile contient la valeur de x dans les quatre registres.

Exemple : Durant l'hiver 78, l'explorateur Jean-Claude Coulerre, isolé au pôle Nord dans son bivouac, a commencé à scruter l'horizon en direction du Sud pour voir le soleil réapparaître. Coulerre savait qu'il ne verrait pas le soleil avant le début de mars lorsqu'il atteindrait une déclinaison de 5º18′S. Quel jour du mois de mars et à quelle heure l'explorateur verra-til son attente récompensée ?

Solution : Le temps t où le soleil atteindra une déclinaison de 5º18′S peut être calculé à l'aide de l'équation suivante :

$$D = a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$$

D étant la déclinaison en degrés, t le temps en jours et

$$\begin{aligned} a_4 &= 4{,}2725 \times 10^{+8} \\ a_3 &= -\,1{,}9931 \times 10^{-5} \\ a_2 &= 1{,}0229 \times 10^{-3} \\ a_1 &= 3{,}7680 \times 10^{-1} \\ a_o &= -\,8{,}1806. \end{aligned}$$

Cette équation est valable pour $1 \leqslant t < 32$, représentant le mois de mars de l'année 1978.

Convertissez d'abord 5°18′S en degrés décimaux en appuyant sur 5.18 CHS g →H pour obtenir – **5,3000** en mode d'affichage FIX 4. (Pour les besoins du calcul, les latitudes sud sont exprimées en nombres négatifs.)

La solution au problème de Coulerre est donnée par la valeur de t satisfaisant

$$-\,5{,}3000 = a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$$

Exprimée sous la forme admise par SOLVE, l'équation devient

$$0 = a_4t^4 - a_3t^3 + a_2t^2 - a_1t - 2{,}8806$$

où le dernier terme constant comprend maintenant la valeur de la déclinaison. Appliquant la méthode d'Horner, la fonction à résoudre devient :

$$f(1) = (((a_4 t + a_3)\ t + a_2)\ t + a_1)\ t - 2{,}8806$$

Pour abréger le sous-programme, stockez et rappelez les constantes dans les registres correspondant à l'exposant de t. Mettez le commutateur PRGM - RUN sur PRGM et introduisez le sous-programme :

Appuyez sur	Affichage	
f CLEAR PRGM	000 –	Effacement de la mémoire programme.
h LBL A	001 – 25, 13, 11	Départ à l'instruction LBL
RCL 4	002 – 24 4	
×	003 – 61	
RCL 3	004 – 24 3	
+	005 – 51	
×	006 – 61	
RCL 2	007 – 24 2	
+	008 – 51	
×	009 – 61	
RCL 1	010 – 24 1	
+	011 – 51	
×	012 – 61	
RCL 0	013 – 24 0	
+	014 – 51	
h RTN	015 – 25 12	

Mettez maintenant le commutateur PRGM - RUN sur RUN et introduisez les cinq coefficients :

Appuyez sur	Affichage*	
4,2725 EEX CHS 8	4,2725 –08	
STO 4	4,2525 –08	Coefficient de t^4.
1,9931 CHS EEX	– 1,9931 –05	
CHS 5		
STO 3	– 1,9931 –05	Coefficient de t^3.
1,0229 EEX CHS 3	1,0229 –03	
STO 2	0,0010	Coefficient de t^2.
3,7680 EEX CHS 1	3,7680 –01	
STO 1	0,3768	Coefficient de t.
2,8806 CHS STO 0	– 2,8806	Terme constant.

* Appuyez sur f FIX 4 pour les affichages de cette annexe.

Comme la solution cherchée doit être comprise entre 1 et 32, introduisez ces deux valeurs initiales. Puis, calculez la racine à l'aide de SOLVE.

Appuyez sur	Affichage	
1 ENTER	1,0000	Estimations initiales
32	32,	
f SOLVE A	7,5137	Racine trouvée.
g R↓	7,5137	Même estimation qu'avant
g R↓	0,0000	Valeur de la fonction
f R↑ f R↑	7,5137	Rétablissement de la pile.

Il s'agit donc du 7 mars. Convertissez la partie fractionnaire du nombre en heures décimales, puis en heures, minutes et secondes.

Appuyez sur	Affichage	
h FRAC	0,5137	Partie fractionnaire du jour.
24 ×	12,3293	Heures décimales.
f →H.MS	12,1945	Heures, minutes, secondes.

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

En examinant la fonction de Coulerre f(t), vous voyez qu'elle peut avoir quatre racines réelles — trois positives et une négative. Cherchez d'autres racines positives à l'aide de SOLVE en partant d'estimations positives plus élevées.

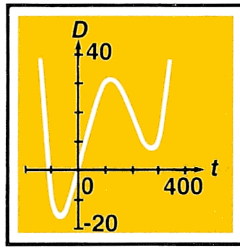
Appuyez sur	Affichage	
1000 ENTER 1100	1.100,	Deux estimations positives plus élevées.
f SOLVE A	Error 6	Racine inexistante.
CLx	278,4497	Dernière estimation essayée.
g R↓	276,7942	Estimation précédente.
g R↓	7,8948	Valeur de la fonction différente de 0,
f R↑ f R↑	278,4497	Remise de la pile à l'état initiale
f SOLVE A	Error 6	A nouveau, racine inexistante.
CLx	278,4398	Estimation à peu près identique
g R↓	278,4497	Estimation précédente.
g R↓	7,8948	Même valeur de la fonction

Vous avez trouvé un minimum local positif au lieu d'une racine. Cherchez maintenant une racine négative.

Appuyez sur	Affichage	
1000 [CHS] [ENTER]	**−1.000,0000**	Deux nouvelles estimations plus élevées.
1100 [CHS]	**−1,100**	
[f] [SOLVE] [A]	**−108,9441**	Racine négative.
[g] [R↓]	**−108,9441**	Même estimation qu'avant.
[g] [R↓]	**1,6000 −08**	Valeur de la fonction.

Il est inutile de chercher davantage : vous avez trouvé toutes les racines possibles. La racine négative n'a pas de sens puisqu'elle est en dehors du domaine dans lequel l'approximation de la déclinaison est valide. Le graphe de la fonction confirme les résultats que vous avez trouvés.

Graphe de *D* en fonction de *t*.



RECHERCHE DE PLUSIEURS RACINES

Beaucoup d'équations ont plusieurs racines. Aussi est-il utile de préciser quelques techniques de calcul pour ce type d'équations.

La méthode la plus simple pour calculer plusieurs racines consiste à les rechercher dans les différents domaines de x susceptibles d'en contenir. Les estimations initiales indiquent le premier domaine de x à examiner. Cette méthode est bonne et a été utilisée dans toute la section 8 « Recherche des racines d'une équation ».

Une méthode plus complexe est la **déflation.** Cette méthode est applicable à une fonction d'une équation, dont les caractéristiques rendent la recherche de toutes les racines par [SOLVE] difficile. Elle consiste à « éliminer » les racines d'une équation, ce qui oblige à modifier l'équation de manière que les premières racines trouvées ne soient plus des racines, mais que les autres restent des racines.

Si une fonction f(x) est nulle à $x = a$, la nouvelle fonction $\frac{f(x)}{(x-a)}$ ne s'approche pas de zéro dans cette zone (si a est une racine simple de $f(x)= 0$). Vous pouvez utiliser cette information pour éliminer une racine connue en ajoutant simplement quelques lignes de programme à la fin du sous-programme de calcul de la fonction. Ces lignes devront soustraire la racine connue (arrondie à 10 chiffres significatifs) de la valeur de x et diviser la différence par la valeur de la fonction. La racine sera souvent simple et la nouvelle fonction écartera SOLVE de la racine connue.

D'autre part, la racine peut être multiple. Une racine multiple est une racine qui réapparaît plusieurs fois dans ce sens qu'à la valeur de cette racine, le graphe de la fonction f(x) coupe l'axe des x et sa pente (ainsi que peut-être les dérivées de degré supérieur suivantes) est égale à zéro. Si la racine connue d'une équation est une racine multiple, la division par le facteur indiqué plus haut ne suffit pas pour l'éliminer. Ainsi l'équation

$$f(x) = x\,(x - a)^3 = 0$$

a une racine multiple à $x = a$ (avec une multiplicité de 3). Cette racine n'est pas éliminée par une division de f(x) par $(x - a)$, mais elle peut l'être par une division par $(x - a)^3$.

Exemple : En appliquant la méthode de la déflation, calculez les racine de

$$60\,x^4 - 944\,x^3 + 3003\,x^2 + 6171\,x - 2890 = 0$$

Grâce à la méthode d'Horner, cette équation peut être réduite à

$$(((60\,x - 944)\,x + 3003)\,x + 6171)\,x - 2890 = 0$$

Mettez le commutateur PRGM-RUN sur PRGM. Entrez le sous-programme d'évaluation du polynôme.

Appuyez sur	**Affichage**
h LBL 2	**001 – 25, 13, 2**
6	**002 – 6**
0	**003 – 0**
×	**004 – 61**
9	**005 – 9**
4	**006 – 4**
4	**007 – 4**
−	**008 – 41**
×	**009 – 61**
3	**010 – 3**
0	**011 – 0**
0	**012 – 0**
3	**013 – 3**
+	**014 – 51**
×	**015 – 61**
6	**016 – 6**
1	**017 – 1**
7	**018 – 7**

1	**019 –**	**1**
[+]	**020 –**	**51**
[×]	**021 –**	**61**
2	**022 –**	**2**
8	**023 –**	**8**
9	**024 –**	**9**
0	**025 –**	**0**
[−]	**026 –**	**41**
[h] [RTN]	**027 –**	**25 12**

Mettez le commutateur PRGM-RUN sur RUN . Introduisez deux grandes estimations initiales (telles que − 10 et − 20) et calculez la racine la plus négative à l'aide de [SOLVE]

Appuyez sur	Affichage	
10 [CHS] [ENTER]	**−10,0000**	Estimations initiales.
20 [CHS]	**−20**	
[f] [SOLVE] 2	**−1,6667**	Première racine.
[STO] 0	**−1,667**	Mémorisation de la racine pour la déflation
[g] [R↓] [g] [R↓]	**4,0000 − 06**	Valeur de la fonction dans le voisinage de zéro.

Mettez le commutateur PRGM-RUN sur PRGM . Ajoutez à votre sous-programme les instructions destinées à éliminer la racine que vous venez de trouver.

Appuyez sur	Affichage	
[GTO] [•] 026	**026 – 41**	Ligne précédant l'instruction RTN.
[x⇄y]	**027 – 21**	Introduction de x dans le registre X.
[RCL] 0	**028 – 24 0**	Division par (x − a), *a* étant une racine.
[−]	**029 – 41**	
[÷]		

Remettez le commutateur PRGM - RUN sur RUN . Calculez la racine suivante en prenant les mêmes estimations initiales.

Appuyez sur	Affichage	
10 [CHS] [ENTER]	**− 10,0000**	Mêmes estimations initiales.
20 [CHS]	**− 20,**	

f SOLVE 2	0,4000	Seconde racine.
STO 1	0,4000	Mémorisation de la racine pour la déflation
g R↓ g R↓	0,0000	Valeur de la fonction après déflation.

Mettez le commutateur PRGM - RUN sur PRGM et modifiez le sous-programme afin d'éliminer la seconde racine.

Appuyez sur	Affichage	
GTO • 030	030– 71	Ligne précédant l'instruction RTN.
x⇄y	031– 21	Introduction de x dans le registre X.
RCL 1	032– 24 1	Déflation pour la seconde racine.
−	033– 41	
÷	034– 71	

Mettez le commutateur PRGM - RUN sur RUN . Calculez la racine suivante en reprenant les mêmes estimations initiales.

Appuyez sur	Affichage	
10 CHS ENTER↑	−10,0000	Mêmes estimations initiales.
20 CHS	−20,	
f SOLVE 2	8,4999	Troisième racine.
STO 2	8,4999	Mémorisation de la racine pour la déflation.
g R↓ g R↓	−1,0929 −07	Valeur de la fonction après déflation dans le voisinage de zéro.

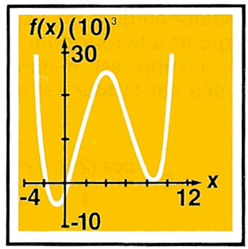
Mettez le commutateur PRGM - RUN sur PRGM et modifiez le sous-programme afin d'éliminer la troisième racine.

Appuyez sur	Affichage	
GTO • 034	034 – 71	Ligne précédant l'instruction RTN.
x⇄y	035 – 21	Introduction de x dans le registre X.
RCL 2	036 – 24 2	Déflation pour la troisième racine.
−	037 – 41	
÷	038 – 71	

Mettez le commutateur PRGM-RUN sur RUN et calculez la quatrième racine.

Appuyez sur	Affichage	
10 CHS ENTER↑	−10,0000	Mêmes estimations initiales
20 CHS	−20,	
f SOLVE 2	8,5001	Quatrième racine.
STO 3	8,5001	Mémorisation de la racine pour référence.
g R↓ g R↓	−0,0009	Valeur de la fonction après déflation dans le voisinage de zéro.

En partant chaque fois des mêmes estimations, vous avez donc trouvé quatre racines pour cette équation qui constitue un polynôme de degré 4. Mais les deux dernières racines sont très voisines et n'en forment en fait qu'une seule (avec une multiplicité de 2). C'est pourquoi cette racine n'a pas été éliminée quand vous avez essayé la déflation à cette racine. (A cause d'une erreur d'arrondi, la fonction originale a des valeurs positives et négatives très faibles lorsque **x** est compris entre 8,4999 et 8,5001. Quand **x** = 8,5, la fonction est nulle.)



Graphe de l'exemple du polynôme

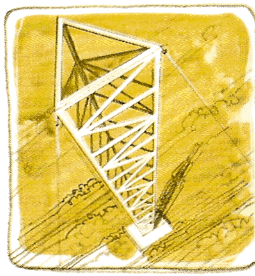
En général, la multiplicité de la racine à éliminer n'est pas connue d'avance. Si, après une tentative d'élimination de la racine, SOLVE retrouve cette racine, plusieurs solutions sont possibles :

- Prendre différentes estimations initiales avec la fonction **défléchie** pour chercher une racine différente.
- Refaire une déflation afin d'éliminer une racine multiple. Si la multiplicité de la racine n'est pas connue, cette opération devra sans doute être répétée plusieurs fois.
- Examiner le comportement de la fonction **défléchie** à des valeurs de **x** voisine de la racine connue. Si les valeurs obtenues pour la fonction coupent l'axe des **x** sans discontinuité, il doit exister une autre racine ou une plus grande multiplicité.
- Analyser algébriquement la fonction originale. Son comportement pourra être analysé avec des valeurs de **x** voisines de la racine connue. (La multiplicité d'une racine peut être indiquée, par exemple, par la représentation d'une suite de Taylor.)

Utilisation de la dérivée

La méthode classique de recherche des maxima et des minima locaux du graphe d'une fonction consiste à utiliser 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 nulle représentent des limites locales possibles de la fonction. (Bien que moins couramment pour des fonctions dont le comportement est bon, il est possible d'avoir des limites à des valeurs de **x** où la dérivée est infinie ou indéfinie.) Si vous pouvez exprimer la dérivée d'une fonction sous une forme fermée, vous pouvez chercher avec SOLVE le point où la dérivée est nulle — montrant où la fonction passe par un maximum ou un minimum.

Exemple : Pour construire une tour d'émission radiophonique, on veut connaître l'angle par rapport à la tour pour lequel l'intensité relative de champ est minimale. L'intensité relative créée par la tour est donnée par la formule :



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

E étant l'intensité relative de champ, **h** la hauteur de l'antenne en longueur d'ondes et θ l'angle par rapport à la verticale en radians. La hauteur h est égale à 0.6.

Solution : L'angle recherché est un angle pour lequel la dérivée de l'intensité par rapport à θ est nulle.

Pour économiser de la place en mémoire et du temps de calcul, stockez les constantes suivantes dans des registres et rappelez-les au moment voulu :

$R0 = 2\pi h$	dans le registre R_0
$R1 = \cos(2\pi h)$	dans le registre R_1.
$R2 = 1/[1 - \cos(2\pi h)]$	dans le registre R_2.

La dérivée de l'intensité **E** par rapport à l'angle θ est donnée par la formule

$$\frac{dE}{d\theta} = R^2 \left[R0 \sin(R0 \cos\theta) - \frac{\cos(R0 \cos\theta) - R1}{\sin\theta \tan\theta} \right].$$

Mettez le commutateur PRGM-RUN sur PRGM ▯▮ et entrez un sous-programme pour calculer la dérivée.

Appuyez sur	Affichage
f CLEAR PRGM	000 –
h LBL 0	001 – 25, 13, 0
f COS	002 – 14, 8
RCL 0	003 – 24, 0
×	004 – 61
f COS	005 – 14 8
RCL 1	006 – 24 1
−	007 – 41
x⇄y	008 – 21
f SIN	009 – 14 7
÷	010 – 71
x⇄y	011 – 21
f TAN	012 – 14 9
÷	013 – 71
CHS	014 – 32
x⇄y	015 – 21
f COS	016 – 14 8
RCL 0	017 – 24 0
×	018 – 61
f SIN	019 – 14 7
RCL 0	020 – 24 0
×	021 – 61
+	022 – 51
RCL 2	023 – 24 2
×	024 – 61
h RTN	025 – 25 12

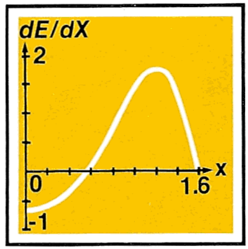
Remettez le commutateur PRGM - RUN sur RUN . En mode radians, calculez et stockez les trois constantes.

Appuyez sur	Affichage	
g RAD	0,0000	Mode radians (après effacement préalable de l'affichage.)
2 h π ×	6,2832	
.6 × STO 0	3,7699	Constante **PO.**
f COS STO 1	– 0,8090	Constante **R1.**
CHS 1 +	1,8090	
h 1/x STO 2	0,5528	Constante **R2.**

L'intensité relative de champ est maximale pour un angle de 90° (perpendiculaire à la tour). Pour trouver le minimum, prenez comme estimations initiales des angles plus voisins de zéro, tels que les équivalents de 10° et 60° en radians.

Appuyez sur	**Affichage**	
10 [g] [→R]	**0,1745**	
60 [g] [→R]	**1,0472**	Estimations initiales
[f] [SOLVE]	**0,4899**	Angle donnant une pente nulle.
[g] [R↓] [g] [R↓]	– **5,5279** – **10**	Pente à l'angle spécifié
[f] [R↑] [f] [R↑]	**0,4899**	Rétablissement de la pile
[f] [→D]	**28,0680**	Angle en degrés.

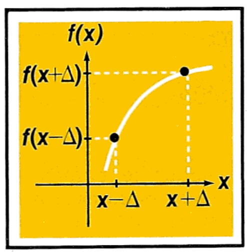
L'intensité relative de champ est minimale pour un angle de 28.0680° de la verticale.



Utilisation d'une pente approchée

La dérivée d'une fonction peut aussi être approchée numériquement. Si vous échantillonnez une fonction en deux points relativement voisins de **x** (x + Δ et x – Δ), vous pouvez calculez une pente moyenne du graphe de la fonction

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



Graphe de dE/dx en fonction de x

La précision de cette approximation dépend de l'incrément Δ et de la nature de la fonction. Les approximations de la dérivée sont d'autant plus précises que Δ est plus faible mais si la valeur de Δ devient trop faible, il existe des risques d'imprécision dus à l'arrondi. Une valeur de x (comprise entre x + Δ et x – Δ) à laquelle la pente est nulle peut être une limite locale de la fonction.

Exemple : Soit à résoudre l'exemple précédent sans utiliser l'équation de la dérivée dE/dθ.

Solution : Calculez l'angle où la dérivée (déterminée numériquement) de l'intensité E est nulle.

Mettez le commutateur PRGM-RUN sur PRGM et introduisez deux sous-programmes : un pour estimer la dérivée de l'intensité et l'autre pour évaluer la fonction de l'intensité E. Le sous-programme suivant calcule la pente entre θ + 0.001 et θ − 0.001 radians (domaine équivalent à environ 0.1°).

Appuyez sur	Affichage	
h LBL A	001 – 25, 13, 11	Évaluation de E pour θ + 0.001
EEX	002 – 33	
CHS	003 – 32	
3	004 – 3	
+	005 – 51	
ENTER↑	006 – 31	
GSB B	007 – 13 12	
x⇄y	008 – 21	Évaluation de E pour θ − 0.001
EEX	009 – 33	
CHS	010 – 32	
3	011 – 3	
−	012 – 41	
ENTER↑	013 – 31	
GSB B	014 – 13 12	
−	015 – 41	
2	016 – 2	
EEX	017 – 33	
CHS	018 – 32	
3	019 – 3	
÷	020 – 71	
h RTN	021 – 25 12	
h LBL B	022 – 25, 13, 12	
f COS	023 – 14 8	
RCL 0	024 – 24 0	
×	025 – 61	
f COS	026 – 14 8	
RCL 1	027 – 24 1	
−	028 – 41	
x⇄y	029 – 21	
f SIN	030 – 14 7	
÷	031 – 71	
RCL 2	032 – 24 2	
×	033 – 61	
h RTN	034 – 25 12	

Mettez le commutateur PRGM - RUN sur RUN. Dans l'exemple précédent, le calculateur était en mode radians et les trois constantes ont été stockées dans les registres 0, 1 et 2. Introduisez les mêmes estimations initiales que précédemment et exécutez SOLVE.

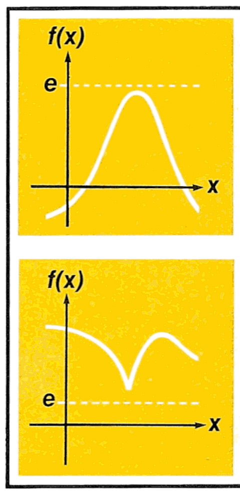
Appuyez sur	Affichage	
10 g →R	**0,1745**	Estimations initiales
60 g →R	**1,0472**	
f SOLVE A	**0,4899**	Angle donnant une pente nulle.
g R↓ g R↓	**0,0000**	Pente correspondant à l'angle spécifié
f R↑ f R↑	**0,4899**	Rétablissement de la pile
ENTER↑ ENTER↑ B	**– 0,2043**	Utilisation du sous-programme de la fonction pour calculer l'intensité minimale.
x⇄y	**0,4899**	Rappel de la valeur de 0.
f →D	**28,0679**	Angle en degrés.

Cette approximation numérique de la dérivée indique une intensité de champ minimale de – 0,2043 à un angle de 28.0679°. (Cet angle diffère de la première solution de 0.0001°).

Utilisation d'estimations répétées

Une troisième méthode est possible si les autres échouent. Elle est plus lente car elle suppose l'emploi répété de la touche SOLVE et elle ne nécessite pas le calcul d'une valeur de Δ comme la méthode précédente. Pour chercher une limite locale de la fonction f(x), on définit une nouvelle fonction

g(x) = f(x) – e



e étant un nombre légèrement supérieur à la valeur limite estimée pour f(x). Si e est correctement choisi, g(x) **s'approche** de zéro au voisinage de la limite de f(x) mais sans devenir **égal** à zéro. Avec SOLVE, vous pouvez analyser g(x) au voisinage de la limite. Le résultat doit être Error 6.

- L'affichage d'Error 6 signifie que le nombre contenu dans le registre X est une valeur de x proche de la limite. Le nombre contenu dans le registre Z indique la différence entre e et la valeur limite de f(x). Changez e pour le rapprocher de la valeur limite (mais sans le confondre avec cette valeur). Puis, à l'aide de [SOLVE], examinez la nouvelle valeur de g(x) au voisinage de la valeur de x que vous avez trouvée. Répétez cette opération jusqu'à ce que les valeurs successives de x tendent à se confondre.

- Si vous avez trouvé une racine de g(x), le nombre e n'est **pas** au-delà de la valeur limite de f(x) ou bien [SOLVE] a trouvé une autre région où f(x) est égale à e. Rechoisissez e dans le voisinage de la valeur limite de f(x) — mais sans dépasser cette limite — et refaites [SOLVE]. Il est également possible de modifier g(x) afin d'éliminer la racine éloignée.

Exemple : Soit à résoudre l'exemple précédent en calculant la dérivée de l'intensité relative de champ E.

Solution : Le sous-programme calcule E, les constantes ayant déjà été introduites dans les exemples précédents.

Mettez le commutateur PRGM-RUN sur PRGM. Entrez le sous-programme qui soustrait une valeur limite estimée de l'intensité de champ *E*. Stockez la valeur limite dans un registre afin de pouvoir la changer manuellement, le cas échéant.

Appuyez sur	Affichage	
[h] [LBL] 1	001 – 25, 13, 1	Départ à l'instruction LBL
[GSB] [B]	002 – 13 12	Calcul de E.
[RCL] 9	003 – 24 9	Soustraction de la valeur limite
[−]	004 – 41	estimée
[h] [RTN]	005 – 25 12	

Mettez le commutateur PRGM-RUN sur RUN. Estimez la valeur minimale de l'intensité en échantillonnant manuellement la fonction.

Appuyez sur	Affichage	
10 [g] [→R]	0,1745	
[ENTER↑] [B]	– 0,1029	
30 [g] [→R]	0,5236	Echantillonnage de la fonction
[ENTER↑] [B]	– 0,2028	à 10°, 30° 50°...
50 [g] [→R]	0,8727	
[ENTER↑] [B]	0,0405	

En partant de ces exemples, reprenez les calculs en partant d'une valeur limite de − 0.25 et d'estimations, en radians, d'environ 10° et 30° pour SOLVE

Appuyez sur	Affichage	
.25 CHS STO 9	− 0,2500	Stockage de la valeur limite estimée
.2 ENTER↑	0,2000	Estimations initiales
.6	0,6	
f SOLVE 1	Error 6	Racine inexistante
CLx STO 4	0,4849	Stockage de l'estimation de θ.
g R↓ STO 5	0,4698	Stockage de l'estimation précédente de θ ?
g R↓	0,0457	Distance à la limite.
.9 ×	0,0411	Changement d'estimation de la
STO + 9	0,0411	limite en la prenant égale à 90 % de la distance.
RCL 4	0,4849	Rappel de l'estimation de θ.
ENTER↑ ENTER↑ B	− 0,2043	Calcul de l'intensité E.
CLx	0,0000	Rappel de l'autre estimation
RCL 5	0,4698	de θ, en gardant la première dans le registre Y.
f SOLVE 1	Error 6	Racine inexistante
CLx STO 4		
x≷y	0,4893	Estimation précédente de θ.
x≷y	0,4898	Rappel de l'estimation de θ.
ENTER↑ ENTER↑ B	− 0,2043	Calcul de l'intensité *E*.
x≷y	0,4898	Rappel de la valeur de *θ*.
f →D	28,0660	Angle en degrés.

La seconde itération donne deux estimations de *θ* dont la quatrième position décimale diffère. Les intensités de champ *E* pour les deux itérations sont égales à quatre positions décimales. Si l'on s'arrête ici, on obtient une intensité de champ de − 0,2043 à un angle de 28,0660°. (La différence entre cet angle et les solutions précédentes est d'environ 0,002°).

LIMITE DU TEMPS D'ESTIMATION

Il existe deux moyens de limiter le temps de calcul d'une racine par SOLVE : le comptage des itérations et l'indication d'une tolérance.

Comptage des itérations

Lors de la recherche d'une racine, SOLVE échantillonne la fonction au moins une dizaine de fois, et parfois une centaine et même davantage (mais en s'arrêtant toujours d'elle-même). Comme le sous-programme de la fonction est exécuté une fois pour chaque analyse d'une estimation, il peut compter et limiter le nombre d'itérations. Il peut faire appel pour cela à l'instruction ISG qui accumulera le nombre d'itérations dans le registre I. Si, avant de lancer l'opération SOLVE, vous stockez un nombre correct dans le registre I, le sous-programme peut interrompre l'algorithme SOLVE quand la limite est dépassée.

Spécification d'une tolérance

Vous pouvez réduire le temps de calcul d'une racine en imposant une tolérance au calcul de votre fonction. Le sous-programme fournira une solution pour la fonction si la valeur obtenue est inférieure à une tolérance donnée. L'algorithme SOLVE s'arrêtera alors et affichera la racine estimée. La tolérance indiquée devra correspondre à une valeur non négligeable dans la pratique et à la précision du calcul. Cette technique permet d'éliminer le temps nécessaire à un affinement inutile de l'estimation.

UTILISATION CONJOINTE DE SOLVE ET DE $\int_y^x$

Exemple : Pour un signal radio à modulation de phase, l'amplitude de la porteuse est proportionnelle à J_0 (x), fonction de Bessel de premier type de degré zéro où x est l'indice de modulation. Quel est le plus petit indice de modulation auquel la porteuse disparaît (c'est-à-dire que son amplitude devient nulle) ?

Solution : L'indice recherché est la valeur de x pour laquelle

$$J_0(x) = \int_0^{\pi} \frac{\cos(x \sin \theta)}{\pi} \, d\theta = 0.$$

Vous pouvez calculer cette valeur avec SOLVE et utiliser $\int_y^x$ pour calculer la fonction J_0 (x).

L'approximation de $J_0(x)$ calculée par $\int_y^x$ a une incertitude qui est rendue dans le registre Y. Quand la valeur de J_0 (x) est inférieure à cette incertitude, on peut considérer que J_0 (x) est égale à zéro. Grâce à cette méthode, vous pouvez éviter à SOLVE de rechercher une précision inutile.

Mettez le commutateur PRGM-RUN sur PRGM. Entrez un sous-programme calculant J_0 (x) et un autre qui calcule la fonction à intégrer.

Appuyez sur	Affichage	
f CLEAR PRGM	000 –	Effacement de la mémoire programme.
h LBL A	001 – 25, 13, 11	Départ à l'instruction LBL.
STO 0	002 – 23 0	Stockage de l'argument x.
0	003 – 0	Bodes de l'intervalle d'intégration.
h π	004 – 25 73	
f ∫y 3	005 – 14, 72, 3	Calcul de J_0 (x).
h ABS	006 – 25 34	Valeur de J_0(x).
f x≤y	007 – 14 41	Retour à 0 si J_0 (x) ≤ incertitude.
CLx	008 – 34	
g x≠0	009 – 15 61	Rétablissement de J_0 (x) si sa valeur est différente de zéro.
h LSTx	010 – 25 0	
h RTN	011 – 25 12	Calcul de la fonction à intégrer
h LBL 3	012 – 25, 13, 3	
f SIN	013 – 14 7	
RCL 0	014 – 24 0	
×	015 – 61	
f COS	016 – 14 8	
h π	017 – 25 73	
÷	018 – 71	
h RTN	019 – 25 12	

Pour réduire le temps de calcul de la racine, mettez-vous d'abord en mode d'affichage SCI 0 pour l'intégration. Après avoir trouvé une solution approchée, indiquez une précision d'intégration plus grande (en mode SCI 3). Puis laissez SOLVE chercher la racine en utilisant la fonction plus précise. Cette méthode permet de gagner beaucoup de temps en évitant d'opérer avec une grande précision sur des valeurs de x éloignées de la racine.

Mettez le commutateur PRGM-RUN sur RUN et procédez de la manière suivante. Rappelez-vous que SOLVE échantillonne plusieurs fois votre fonction et que ∫y demande parfois plus d'une minute pour évaluer une intégrale. C'est pourquoi, il faut compter de 3 à 6 minutes pour exécuter SOLVE dans les exemples suivants.

Appuyez sur	Affichage	
f SCI 0	0, 00	Indiquez la précision de ∫y. (après effacement préalable des données affichées).

g RAD	**0, 00**	
0 ENTER	**0, 00**	Estimations initiales à analyser au voisinage de 0.
1	**1,**	
f SOLVE A	**2, 00**	Racine recherchée.
f SCI 3	**2,480 00**	Augmentation de la précision de $\int_y^x$.
2.4 ENTER	**2,400 00**	Estimations initiales proches de la première approximation.
2.5	**2,5**	
f SOLVE A	**2,405 00**	Racine recherchée.
f FIX 4	**2,4049**	Format d'affichage FIX 4.
g R↓ g R↓	**0,0000**	J° (x) inférieure à l'incertitude.
g R↓	**0,0001**	Incertitude de $\int_y^x$.

Un indice de modulation de 2,4040 supprime au moins 99,99 % de l'amplitude de la porteuse (son amplitude est donc inférieure à 0,0001 du maximum).

Annexe B

PARTICULARITÉS DE $\boxed{\int_y^x}$

Le chapitre 9 a présenté des généralités permettant d'utiliser correctement $\boxed{\int_y^x}$ dans la plupart des applications. Cette annexe présente quelques particularités de $\boxed{\int_y^x}$ qui pourront vous intéresser si vous utilisez souvent cette fonction.

PRINCIPE DE FONCTIONNEMENT DE $\boxed{\int_y^x}$

L'algorithme de $\boxed{\int_y^x}$ calcule l'intégrale d'une fonction f(x) en calculant la moyenne pondérée des valeurs de la fonction pour un grand nombre de valeurs de x (appelées **points échantillons**) dans l'intervalle d'intégration. La précision des résultats obtenus par ce système d'échantillonnage est fonction du nombre de points considérés : elle est d'autant plus grande que les points sont plus nombreux. Si f(x) pouvait être évaluée avec un nombre infini de points, l'algorithme pourrait — en négligeant les limites imposées par l'imprécision de la fonction calculée f(x) — fournir une réponse exacte.

L'évaluation de la fonction avec un nombre infini d'échantillons serait très longue (en fait, éternelle). Heureusement, cela est inutile car la précision maximale de l'intégrale calculée est subordonnée à la précision des valeurs obtenues pour la fonction. En n'utilisant qu'un nombre fini d'échantillons, l'algorithme peut calculer une intégrale dont la précision est satisfaisante compte tenu de l'imprécision inhérente à f(x).

L'algorithme de $\boxed{\int_y^x}$ ne considère au début que quelques points d'échantillonnage et fournit des approximations relativement peu précises. Si la précision de ces approximations ne répond pas encore à celle de f(x), l'algorithme est itéré (repris) avec un plus grand nombre de points d'échantillonnage. Si le résultat n'est toujours pas satisfaisant, une seconde itération est effectuée avec le double de points d'échantillonnage et ainsi de suite jusqu'à ce que l'approximation obtenue ait le maximum de précision compte tenu de l'incertitude inhérente à f(x).

L'incertitude de l'approximation finale est un nombre dérivé du format d'affichage indiquant l'incertitude de la fonction*. A la fin de chaque itération, l'algorithme compare l'approximation obtenue au cours de cette

* Le rapport entre le format d'affichage, l'incertitude de la fonction et l'incertitude de l'approximation et son intégrale est décrit plus loin dans cette annexe.

itération aux approximations obtenues lors des deux itérations précédentes. Si la différence entre l'une de ces trois approximations et les deux autres est inférieure à l'incertitude de l'approximation finale, l'algorithme s'arrête mettant l'approximation actuelle dans le registre X et son incertitude dans le registre Y.

L'algorithme de $\boxed{\int_y^x}$ est conçu de telle manière qu'il est extrêmement peu probable que l'erreur produite dans les trois approximations successives — c'est-à-dire la différence entre l'intégrale et l'approximation — soit supérieure à la différence qui existe entre les approximations elles-mêmes. L'erreur produite dans l'approximation finale sera donc inférieure à son incertitude**. Ainsi, bien que nous ne puissions pas connaître l'erreur de l'approximation finale, nous pouvons être certains qu'elle est inférieure à l'incertitude affichée. L'incertitude de l'approximation constitue la limite supérieure de la différence entre l'approximation et l'intégrale réelle.

PRÉCISION, INCERTITUDE ET TEMPS DE CALCUL

La précision d'une approximation par $\boxed{\int_y^x}$ ne change pas obligatoirement si l'on augmente de un le nombre de chiffres spécifié par le format d'affichage. De même, le temps nécessaire au calcul d'une intégrale ne change pas forcément avec le format d'affichage.

Exemple : La fonction de Bessel de premier type de degré 4 peut être exprimée sous la forme :

$$J_4(x) = 1/\pi \int_0^\pi \cos(4\theta - x \sin\theta)\, d\theta$$

Calculer l'intégrale en $J_4(1) = 1/\pi \int_0^\pi \cos(4\theta - \sin\theta)\, d\theta$.

Mettez le commutateur PRGM-RUN sur PRGM ▮ et entrez un sous-programme qui évalue la fonction $f(\theta) = \cos(4\theta - \sin\theta)$.

Appuyez sur	**Affichage**
f CLEAR PRGM	000 –
h LBL 0	001 – 25, 13, 0
4	002 – 4
×	003 – 61

** A condition que f(x) ait une continuité suffisante, critère que nous allons exposer avec plus de détails plus loin dans cette annexe.

x⇄y	004 –	21
f SIN	005 –	14 7
−	006 –	41
f COS	007 –	14 8
h RTN	008 –	25 12

Remettez le commutateur PRGM-RUN sur RUN et introduisez les limites de l'intégration dans les registres X et Y. Mettez-vous en mode radians, format d'affichage sur SCI 2. Enfin, appuyez sur f ∫y 0 pour estimer l'intégrale.

Appuyez sur	Affichage	
0 ENTER↑	**0,0000**	Introduisez la borne inférieure dans le registre Y.
h π	**3,1416**	Introduisez la borne supérieure dans le registre X.
g RAD	**3,1416**	Mode trigonométrique en radians. (Cette étape est inutile si vous n'avez pas éteint votre calculateur ou changé de mode trigonométrique depuis la dernière fois que vous avez utilisé le mode radians).
f SCI 2	**3,14 00**	Mettez le format d'affichage sur SCI 2.
f ∫y 0	**7,79 – 03**	Approximation de l'intégrale dans SCI 2.
x⇄y	**1,45 – 03**	Incertitude de l'approximation de SCI 2.

L'incertitude indique que les chiffres affichés comme approximation peuvent ne pas contenir de chiffres susceptibles d'être considérés précis. En fait, comme la dernière approximation de la section 9, celle-ci est plus précise que ne le laisse supposer son incertitude.

Appuyez sur	Affichage	
x⇄y	**7,79 – 03**	Affichage de l'approximation.
h MANT	**7785820888**	Tous les 10 chiffres de l'approximation de SCI 2.

La valeur de cette intégrale dont la précision s'étend à cinq chiffres significatifs est $7{,}7805 \times 10^{-3}$. Dans cette approximation, l'erreur est donc égale à environ $(7{,}7858 - 7{,}7805) \times 10^{-3} = 5{,}3 \times 10^{-6}$. Cette erreur est nettement inférieure à l'incertitude $1{,}45 \times 10^{-3}$. L'incertitude n'est qu'une **limite supérieure** de l'erreur d'approximation ; l'erreur réelle sera en général plus faible.

Calculons maintenant l'intégrale en SCI 3 et comparons la précision de l'approximation obtenue avec celle de SCI 2.

Appuyez sur	Affichage	
f SCI 3	**7,786 −03**	Format d'affichage SCI 3.
g R↓ g R↓	**3,142 00**	Descente dans la pile jusqu'à apparition de la borne supérieure dans le registre X.
f $\int_y^x$ 0	**7,786 −03**	Approximation de l'intégrale en SCI 3.
x⇄y	**1,448 −04**	Incertitude de l'approximation de SCI 3.
x⇄y	**7,786 −03**	Affichage de l'approximation.
h MANT	**7785820888**	Tous les 10 chiffres de l'approximation de SCI 3.

Les 10 chiffres des approximations en SCI 2 et en SCI 3 sont identiques : la précision de l'approximation en SCI 3 n'est pas meilleure qu'en SCI 2, bien que l'incertitude en SCI 3 soit inférieure à celle de SCI 2. Pourquoi ? Rappelez-vous que la précision d'une approximation dépend essentiellement du nombre d'échantillons auxquels la fonction f(x) a été évaluée. L'algorithme $\int_y^x$ est itéré avec un nombre croissant d'échantillons jusqu'à ce que la différence entre trois approximations successives soit inférieure à l'incertitude qui est un nombre dérivé du format d'affichage. Après une itération donnée, la différence entre les estimations peut être tombée en-dessous de l'incertitude au point où, même si l'on réduisait l'incertitude par un facteur de 10, celle-ci serait encore supérieure à la différence. Dans ce cas, si l'on réduisait l'incertitude en augmentant de un les chiffres spécifiés dans le format d'affichage, l'algorithme n'aurait pas à considérer d'autres échantillons et l'approximation obtenue serait identique à celle obtenue avec une incertitude plus grande.

Si vous avez calculé les deux approximations précédentes sur votre calculateur, vous avez pu noter que le temps de calcul était le même en SCI 3 et en SCI 2. En effet, le temps de calcul de l'intégrale d'une fonction donnée dépend du nombre d'échantillons auxquels la fonction doit être évaluée pour obtenir une approximation dont la précision soit acceptable. En ce qui concerne l'approximation en SCI 3, l'algorithme n'avait pas à examiner plus d'échantillons qu'en SCI 2, par conséquent, le temps de calcul de l'intégrale n'était pas plus long.

Cependant, si l'on augmente le nombre de chiffres affichés, la fonction devra être évaluée avec des échantillons supplémentaires ce qui prolongera le temps de calcul de l'intégrale. Calculons maintenant la même intégrale en SCI 4 :

Appuyez sur	Affichage	
f SCI 4	**7,7858 −03**	Format d'affichage SCI 4.
g R↓ g R↓	**3,1416 00**	Descente dans la pile jusqu'à apparition de la borne supérieure dans le registre X.
f $\int_y^x$ 0	**7,7807 −03**	Approximation de l'intégrale en SCI 4.

Cette approximation a pris environ deux fois plus de temps qu'en SCI 3 ou SCI 2. L'algorithme devait cette fois-ci évaluer la fonction avec le double environ d'échantillons pour obtenir une approximation dont la précision était satisfaisante. Notez, cependant, que votre patience a été récompensée puisque la précision obtenue est supérieure de presque deux chiffres à celle que vous avez obtenue avec la moitié d'échantillons.

Les exemples précédents montrent qu'en reprenant l'approximation d'une intégrale avec différents formats d'affichage on peut espérer obtenir une réponse plus précise mais qu'elle ne l'est pas obligatoirement. C'est la fonction qui décide et le seul moyen de savoir si la précision peut être améliorée est d'essayer.

D'autre part, la précision ne peut être améliorée qu'aux dépens du temps de calcul qui devient deux fois plus long. Il faudra tenir compte de ce compromis précision/temps si vous voulez diminuer l'incertitude dans l'espoir d'obtenir une réponse plus précise.

Remarque

Le temps nécessaire au calcul de l'intégrale d'une fonction donnée ne dépend pas seulement du nombre de chiffres spécifiés dans le format d'affichage mais aussi, dans une certaine mesure, des bornes de l'intervalle d'intégration. Si le calcul d'une intégrale est trop long, la longueur de l'intervalle d'intégration (c'est-à-dire la distance entre les bornes) risque d'être trop grande par rapport à certaines caractéristiques de la fonction à intégrer. Cependant, dans la plupart des cas, vous n'avez pas à vous soucier des effets des bornes de l'intervalle d'intégration sur le temps de calcul. Nous reprendrons ces considérations plus loin, avec quelques exemples où les limites risquent d'accroître inutilement le temps de calcul et nous verrons comment traiter ces situations.

PRÉCISION DE LA FONCTION À INTÉGRER

La précision d'une fonction calculée par $\boxed{\int_x^y}$ dépend de celle de la fonction calculée par votre sous-programme. Cette précision, que vous indiquez dans votre format d'affichage, est soumise à trois critères principaux :

1. Précision des constantes empiriques de la fonction
2. Degré d'exactitude de la description d'une situation physique par la fonction
3. Étendue de l'erreur d'arrondi dans les calculs internes du calculateur.

Fonctions liées à des situations physiques

Les fonctions que nous avons intégrées dans le chapitre 9 et dans cette annexe — $\cos(\sin\theta)$, $\cos(\theta - \sin\theta)$, $\cos(4\theta - \sin\theta)$ et $(\sin x)/x$ — sont des exemples de **fonctions mathématiques pures.** Dans ce contexte, cela signifie que les fonctions ne contiennent pas de constantes empiriques et que ni les variables ni les bornes de l'intervalle d'intégration ne représentent en fait des quantités physiques. Pour ce type de fonctions, vous pouvez choisir le nombre de chiffres que vous voulez dans votre format d'affichage (neuf au maximum) pour obtenir la précision désirée dans le calcul de l'intégrale*. Le seul point à prendre en considération est le compromis précision/temps de calcul.

Mais il y a d'autres grandeurs dont il faut tenir compte dans l'intégration des fonctions relatives à des situations physiques. La question à se poser dans le cas de ces fonctions est la suivante : **La précision recherchée dans le calcul de l'intégrale est-elle justifiée par celle de la fonction ?** Par exemple, si la fonction contient des constantes empiriques dont la précision est limitée à trois chiffres significatifs, est-il sensé de spécifier une précision supérieure à trois chiffres dans le format d'affichage ?

Une autre considération importante — plus subtile et par conséquent plus facile à oublier — est que pratiquement chaque fonction relative à une situation physique a **une certaine imprécision intrinsèque** puisqu'elle ne constitue qu'un **modèle** mathémathique d'un processus ou d'un événement réel. Un modèle mathématique est lui-même une **approximation** qui ne tient pas compte des effets de facteurs connus ou inconnus qui ne sont pas suffisamment importants pour compromettre l'utilité des résultats.

---

* A condition que, malgré l'erreur d'arrondi, f(x) soit toujours calculée avec la précision correspondant au nombre de chiffres affichés.

La fonction de distribution normale est un exemple de modèle mathématique

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

qui s'est révélé utile pour fournir des informations concernant des organismes vivants, les dimensions de certains produits, les températures moyennes etc. Voici un autre modèle mathématique :

$$C = \frac{C_0}{\sqrt{\pi Dt}} \int_{\frac{x}{2\sqrt{Dt}}}^{\infty} e^{-Y^2/4Dt}\,dy$$

qui représente une solution particulière de l'équation de diffusion pour les semi-conducteurs. En général, ces descriptions mathématiques sont basées sur des considérations théoriques ou déduites de données expérimentales. Leur utilité pratique repose sur certaines hypothèses qui consistent, par exemple, à ignorer les effets des facteurs relativement négligeables. C'est ainsi que la précision des résultats obtenus en prenant la fonction de distribution normale comme modèle de distribution de certaines quantités dépend de la taille de la population étudiée. La précision des résultats obtenus par la solution de l'équation de diffusion ignore les effets des quanta. Enfin, la précision des résultats obtenus par l'équation $s = s_0 - 1/2\ gt^2$, qui donne la hauteur d'un corps qui tombe, ne tient pas compte de la variation de g avec l'altitude et l'accélération due à la gravité.

Ainsi, la précision des résultats obtenus par les descriptions mathématiques du monde physique est limitée. Si l'on se contente d'un résultat numérique à trois chiffres, on peut ignorer les effets de nombreux facteurs et hypothèses. D'autre part, si l'on pouvait inclure ces facteurs et hypothèses dans une description mathématique plus précise — qui ne resterait toujours qu'un modèle — ils étendraient leurs effets au-delà de la cinquième position décimale. Si vous calculez une intégrale dont la précision apparente dépasse celle du modèle qui décrit le comportement réel du processus ou de l'événement, vous n'avez aucune raison d'utiliser toute la précision apparente du résultat.

Erreur d'arrondi dans les calculs internes

Sur n'importe quel instrument de calcul — y compris votre HP-34C — les résultats sont « arrondis » à un nombre fini de chiffres (10 dans votre HP-34C). A cause de cette **erreur d'arrondi,** les résultats obtenus, principalement les résultats de l'évaluation d'une fonction contenant plusieurs opérations mathématiques, risquent de ne pas être précis aux 10 chiffres affichés. Notez que l'erreur d'arrondi existe dans n'importe quelle évaluation d'une fonction mathématique et non pas exclusivement dans celle d'une fonction intégrée par $\boxed{\int_y^x}$.

Si f(x) est une fonction relative à une situation physique, l'imprécision due à l'erreur d'arrondi est généralement négligeable par rapport à celle qui résulte des constantes empiriques etc... Si f(x) est ce que nous avons appelé une fonction mathématique pure, sa précision n'est limitée que par l'erreur d'arrondi. En général, il faudrait avoir recours à une analyse complexe pour connaître exactement le nombre de chiffres affectés par l'erreur d'arrondi. Dans la pratique, c'est l'expérience et non pas l'analyse qui permet de déterminer exactement les effets de l'erreur d'arrondi.

Dans certaines situations, l'erreur d'arrondi donne des résultats inattendus, en particulier si l'on compare les résultats des calculs d'intégrales équivalentes mathématiquement mais qui diffèrent par une transformation de variables. La description de ces situations — qui sont peu probables dans des applications classiques — ne constitue pas le propos de ce manuel.

INCERTITUDE ET FORMAT D'AFFICHAGE

A cause de l'erreur d'arrondi, le sous-programme que vous avez écrit pour calculer f(x) ne peut pas calculer exactement f(x), mais il calcule

$$\hat{f}(x) = f(x) \pm \delta_1(x)$$

$\delta_1(x)$ étant l'incertitude de f(x) causée par l'erreur d'arrondi.

Si f(x) concerne une situation physique, la fonction que vous voulez intégrer n'est pas f(x) mais

$$F(x) = f(x) \pm \delta_2(x)$$

$\delta_2(x)$ étant l'incertitude liée à $\hat{f}(x)$ et causée par l'approximation de la situation physique réelle.

Comme $f(x) = \hat{f}(x) \pm \delta_1(x)$, la fonction à intégrer est

$$F(x) = \hat{f}(x) \pm \delta_1(x) \pm \delta_2(x)$$

ou

$$F(x) = \hat{f}(x) \pm \delta(x)$$

$\delta(x)$ étant l'incertitude nette liée à la fonction f(x).

Donc, l'intégrale à calculer est

$$\begin{aligned}\int F(x)\,dx &= \int_a^b \left[\hat{f}(x) \pm \delta(x)\right] dx \\ &= \int_a^b \hat{f}(x)\,dx \pm \int_a^b \delta(x)\,dx \\ &= I \pm \Delta.\end{aligned}$$

I étant l'approximation de $\int_a^b F(x)\,d\,x$ et Δ l'incertitude liée à l'approximation. L'algorithme $\boxed{\int}$ place le nombre I dans le registre X et le nombre Δ dans le registre Y.

L'incertitude $\delta(x)$ de $\hat{f}(x)$, la fonction calculée par votre sous-programme, s'obtient de la manière suivante. Supposons que les valeurs de la fonction doivent être précises à trois chiffres, vous utilisez donc le format d'affichage SCI 2. Vous n'aurez alors à l'affichage que les chiffres précis dans la mantisse des valeurs de la fonction, soit : **1,23 −04.**

$$\Delta = \int_a^b \delta(x)\,dx$$
$$= \int_a^b [0.5 \times 10^{-n+m(x)}]\,dx$$

Pour calculer cette intégrale, l'algorithme utilise les échantillons de $\delta(x)$ d'une manière sensiblement analogue à celle qu'il utilise pour calculer l'approximation de l'intégrale de la fonction à l'aide des échantillons de $\hat{f}(x)$.

Notez que si une intégrale est approchée en format FIX, $m(x) = 0$ et l'incertitude obtenue dans l'approximation est égale à

$$\Delta = 0{,}5 \times 10^{-n}\,(b - a)$$

Comme Δ est proportionnel au facteur 10^{-n}, l'incertitude d'une approximation change d'un facteur plus ou moins égal à 10 pour chaque chiffre spécifié dans le format d'affichage. Ceci n'est généralement pas exact en format SCI ou ENG car le changement du nombre de chiffres spécifiés dans le format suppose que l'évaluation de la fonction soit effectuée avec différents échantillons de sorte que $\delta(x) \sim 10^{m(x)}$ aurait différentes valeurs.

En principe, vous n'êtes pas obligé de déterminer avec précision l'incertitude de la fonction, car cela suppose souvent des analyses très compliquées. S'il est plus facile d'estimer l'incertitude des valeurs d'une fonction comme une incertitude **relative,** il est généralement plus pratique d'utiliser les formats SCI ou ENG. D'autre part, s'il est plus facile d'évaluer l'incertitude des valeurs d'une fonction comme une incertitude **absolue,** il est préférable d'utiliser le format FIX. Ce format n'est pas recommandé (car il conduit à des résultats inattendus) pour les intégrations de fonctions ayant des valeurs **et** une incertitude extrêmement faibles dans tout l'intervalle d'intégration ou de fonctions dont la valeur et l'incertitude varient énormément dans l'intervalle d'intégration. De même, le format SCI n'est pas recommandé (car il donne également des résultats inattendus) si la valeur de la fonction devient beaucoup plus petite que son incertitude. Si les résultats du calcul d'une intégrale semblent étranges, il est préférable de changer de format.

Comme le format d'affichage arrondit le nombre contenu dans le registre X au nombre affiché, il en résulte que l'incertitude des valeurs de la fonction est égale à $\pm$ 0,005 $\times$ 10^{-4} = $\pm$ 0,5 $\times$ 10^{-6}. Donc, si vous choisissez le format d'affichage SCI n ou ENG n, où n est un entier, l'incertitude des valeurs de la fonction est égale à

$$\delta(x) = 0{,}5 \times 10^{-n} \times 10^{m}(x)$$
$$= 0{,}5 \times 10^{-n+m(x)}$$

Dans cette formule, n est le nombre de chiffres spécifiés dans le format d'affichage et m(x) l'exposant de la valeur de la fonction à x que l'on obtiendrait si la valeur était affichée en format SCI.

L'incertitude est proportionnelle au facteur $10^{m(x)}$ qui représente l'ordre de grandeur de la valeur de la fonction à x. Les formats d'affichage SCI ou ENG impliquent donc une incertitude **relative** à l'ordre de grandeur de la fonction.

De même, si la valeur d'une fonction est affichée en FIX n, l'arrondi de l'affichage implique que l'incertitude inhérente aux valeurs de la fonction est égale à

$$\delta(x) = 0{,}5 \times 10^{-n}$$

Comme cette incertitude est indépendante de l'ordre de grandeur de la fonction, le format FIX implique une incertitude **absolue.**

Chaque fois que l'algorithme $\int_y^x$ échantillonne la fonction à une valeur donnée de x, il examine également un échantillon de δ (x), l'incertitude de la valeur de la fonction à x. Pour effectuer ce calcul, il utilise le nombre de chiffres n actuellement spécifiés dans le format d'affichage et (si le format choisi est SCI ou ENG) la grandeur m(x) de la valeur de la fonction à x. Le nombre Δ, qui est l'incertitude de l'approximation de l'intégrale calculée, est l'intégrale de δ(x).

CALCUL DES INTÉGRALES AVEC UNE PRÉCISION MAXIMALE

En format SCI ou ENG, il est possible d'afficher des nombres avec une mantisse qui peut contenir jusqu'à sept chiffres. On obtient en général le même affichage pour SCI 8 ou SCI 9 que pour SCI 7. Cependant, l'incertitude de l'intégrale calculée est plus faible en format SCI 8 ou SCI 9 qu'en format SCI 7. La même chose est vraie pour les estimations obtenues en ENG.

En format SCI (ou ENG) 9*, vous pouvez calculer une intégrale avec le maximum de précision. Si le calculateur est en mode RUN, vous pouvez le faire **directement** en appuyant sur f SCI 9 ou **indirectement** en appuyant sur 9 STO f I h DSP I (en admettant que le format d'affichage soit déjà positionné sur SCI ou ENG). Si le calculateur est en mode PRGM, vous ne pouvez pas vous mettre **directement** en mode d'affichage [SCI] 8, [SCI] 9, [ENG] 8 ou [ENG] 9. Si vous tentez de le faire, le code de touche indiquera [SCI] 7 ou [ENG] 7 et les intégrales seront calculées avec une incertitude correspondant à un format d'affichage de sept chiffres. Pour calculer les intégrales avec le maximum de précision en mode PRGM vous devez donc introduire **indirectement** le format d'affichage en utilisant [DSP I].**

Pour voir comment cela fonctionne, mettez le commutateur PRGM-RUN sur PRGM et introduisez le programme suivant pour calculer l'intégrale Si (2) avec le maximum de précision.

Appuyez sur	Affichage	
[h] [LBL] [A]	001 – 25, 13, 11	Label du programme contenant [∫] dans la ligne programme.
9	002 – 9	Introduction de 9 dans X.
[STO] [f] [I]	003 – 23, 14, 23	Stockage de 9 dans I.
[h] [DSP I]	004 – 25 11	Format d'affichage sur neuf chiffres (Ce programme suppose que le format d'affichage ait été fixé manuellement à [SCI] avant l'exécution du programme).
[g] [R↓]	005 – 15 22	Descente dans la pile pour que le 9 introduit dans le registre X, ligne de programme 2, ne devienne pas la borne supérieure de l'intervalle d'intégration.
[f] [∫] [2]	006 – 14, 72, 2	Calcul de l'intégrale $_0\int^t$ (sinx)/x dx.
[h] [RTN]	007 – 25 12	
[h] [LBL] 2	008 – 25, 13, 2	Label du sous programe évaluant f(x) = (sin x)/x.
[f] [SIN]	009 – 14 7	
[x⇄y]	010 – 21	
[÷]	011 – 71	
[h] [RTN]	012 – 25 12	

* A condition que f(x) soit calculée avec une précision de 10 chiffres significatifs.

** Si le registre I contient un nombre négatif quand vous appuyez sur [h] [DSP I], les nombres seront **affichés** tels qu'ils apparaîtraient si le registre I était à 0. Cependant, l'algorithme [∫] prendra en compte les nombres négatifs pour déterminer l'incertitude d'une approximation. Le plus petit nombre pris en compte pour déterminer l'incertitude d'une approximation est – 6. Si le registre I contient un nombre inférieur à – 6, l'approximation sera effectuée comme si le registre I contenait – 6.

Remettez le commutateur PRGM-RUN sur RUN. Introduisez les bornes de l'intervalle d'intégration dans les registres X et Y, puis appuyez sur [A] pour exécuter le programme.

Appuyez sur	Affichage	
[f] [SCI] 3	**0,000 00**	Format d'affichage [SCI]. L'exécution du programme suivant (en appuyant sur [A]) va faire passer le nombre de chiffres spécifiés de 3 à 9. (L'affichage suppose qu'il ne reste pas de résultats de l'exemple précédent.)
0 [ENTER↑]	**0,000 00**	Introduction de la borne inférieure dans le registre Y.
2	**2,**	Introduction de la borne supérieure dans le registre X.
[g] [RAD]	**2,000 00**	Mode trigonométrique en radians (ceci est inutile si vous n'avez pas éteint le calculateur ni changé de mode trigonométrique depuis la dernière fois).
[A]	**1,605412 00**	Calcul de Si(2) avec le maximum de précision.
[x⇄y]	**6,000000 – 10**	Incertitude de l'approximation.
[x⇄y]	**1,605412 00**	Affichage de l'incertitude de l'approximation.
[h] [MANT]	**1605412977**	Les 10 chiffres de l'approximation.

Comme le chiffre le plus significatif de l'incertitude correspond à la dixième position décimale, l'incertitude indique que l'estimation est correcte à au moins neuf positions décimales. En fait, l'estimation correspond, par les neuf positions décimales, à la valeur donnée pour Si(2) dans les tables de fonctions mathématiques.

OBTENTION DE L'APPROXIMATION INSTANTANÉE D'UNE INTÉGRALE

La pression de [R/S] pendant un calcul d'intégrale arrête le calcul tout comme l'exécution d'un programme. Le calculateur s'arrête à la ligne sur laquelle il se trouve dans le sous-programme que vous avez écrit pour évaluer la fonction et affiche le résultat de l'exécution de la ligne précédente. Notez qu'après l'arrêt du calcul, l'approximation actuelle de l'intégrale n'est **pas** le nombre contenu dans le registre X ni celui d'un registre quelconque de la pile. Comme pour tout autre programme, la pression de [R/S] relance le calcul à partir de la ligne où il s'était arrêté.

Lorsque le calcul d'une intégrale vous paraît trop long, vous pouvez l'arrêter et afficher l'approximation actuelle. Vous pouvez en effet obtenir l'approximation actuelle, mais non pas son incertitude. L'algorithme [∫] met à jour l'approximation actuelle et la range dans le registre LAST X après avoir évalué la fonction à chaque nouvel échantillon. Pour obtenir l'approximation actuelle, il suffit donc d'arrêter le calculateur, d'exécuter le sous-programme de la fonction pas à pas jusqu'à ce que le calculateur ait fini d'évaluer la fonction et de mettre à jour l'approximation actuelle, enfin de rappeler le contenu du registre LAST X.

Notez que pendant que le calculateur met à jour l'approximation actuelle, l'affichage ne clignote pas comme d'habitude lorsque le calculateur exécute le sous-programme de la fonction. Vous pouvez donc éviter d'exécuter votre sous-programme pas à pas en arrêtant le calculateur au moment où l'écran est vide.

En résumé, pour obtenir l'approximation actuelle d'une intégrale, suivez la procédure ci-dessous :

1. Appuyez sur [R/S] pour arrêter le calculateur, de préférence lorsque l'écran est vide.

2. Quand le calculateur s'arrête en affichant un nombre, mettez le commutateur PRGM-RUN sur PRGM.

   a. Si l'affichage montre la ligne de programme contenant le label de votre sous-programme, mettez le commutateur PRGM-RUN à nouveau sur RUN et passez à l'étape 3.

   b. Si vous n'avez pas appuyé sur [R/S] au moment où l'écran était vide, une autre ligne de votre sous-programme sera affichée. Remettez le commutateur PRGM-RUN sur RUN et appuyez plusieurs fois sur [SST] jusqu'à ce que vous voyiez apparaître **25 12** à droite de l'écran (ou 000– à gauche) pendant que vous maintenez votre pression sur la touche [SST]; relâchez la touche et attendez que le calculateur s'arrête en affichant un nombre.

3. Appuyez sur [h] [LST x]. L'approximation actuelle va apparaître sur l'écran.

Si vous voulez continuer à calculer l'approximation finale, appuyez sur [CLx] [+] [R/S]. La pile se remplit alors à nouveau de la valeur de x et le calculateur repart.

Recalculons, par exemple, l'intégrale Si(2) et affichons l'approximation actuelle au bout d'une ou deux minutes.

Appuyez sur	Affichage	
[g] [R↓] [g] [R↓]	**2,000000 00**	Descente dans la pile jusqu'à ce que la borne supérieure apparaisse dans le registre X.
[A]	Clignotement	Départ du calcul de l'intégrale.

Au bout d'une ou deux minutes, arrêtez le calculateur et vérifiez l'approximation actuelle :

Appuyez sur	Affichage	
[R/S]	**6,771087 −01**	Arrêtez le calculateur en appuyant sur [R/S] quand l'écran est vide. (Bien entendu, le nombre affiché par **votre** calculateur dépend du moment où vous avez appuyé sur [R/S].)

Mettez le commutateur PRGM-RUN sur PRGM pour vérifier si le calculateur s'est arrêté au label de votre sous-programme.

Affichage	
001 – 25, 13, 2	Label 2.

Comme le calculateur s'est arrêté au label de votre sous-programme, vous pouvez rappeler l'estimation actuelle du registre LAST X après avoir remis le commutateur PRGM-RUN sur RUN.

Appuyez sur	Affichage	
[h] [LST x]	**1,605412 00**	Approximation actuelle de l'intégrale. (Ici aussi, le nombre affiché dépend du moment où vous avez appuyé sur [R/S].)

Ceci ne s'applique qu'au cas où vous n'auriez pas mis d'instruction [RTN] à la fin de votre sous-programme.

Pour continuer le calcul et obtenir l'approximation finale :

Appuyez sur	Affichage	
CLx	**6,771087 −01**	Retour de la valeur actuelle de x dans le registre X.
R/S	**1,605412 00**	Approximation finale de l'intégrale.

CAUSES POSSIBLES DE RÉSULTATS INCORRECTS

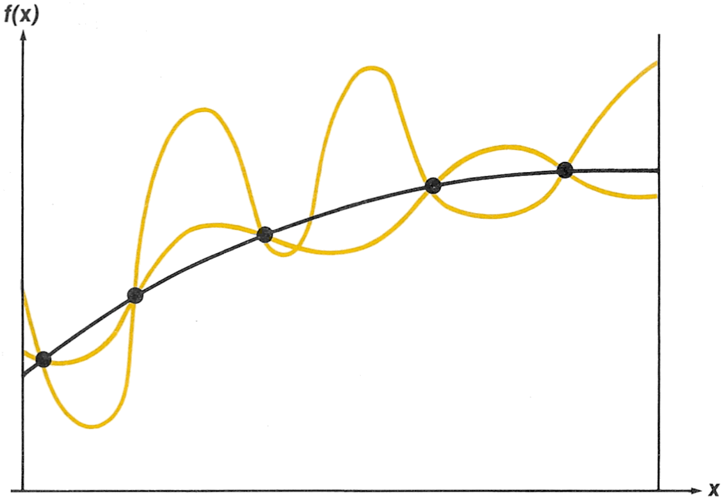
Bien que l'algorithme de $\int_y^x$ utilisé dans votre HP-34C soit l'un des meilleurs qui existent, il peut, dans certains cas, donner des résultats erronés. **Cette possibilité est néanmoins très faible.** L'algorithme de $\int_y^x$ fournit des résultats exacts avec pratiquement toutes les fonctions continues. Seules les fonctions dont le comportement est **extrêmement** erratique présentent un risque d'erreur en fin de calcul. Ces fonctions sont rares dans les problèmes relatifs à des situations physiques réelles ; on peut d'ailleurs généralement les reconnaître et les traiter directement.

Étudions de plus près le fonctionnement de l'algorithme de $\int_y^x$ pour voir comment il peut conduire à des résultats erronés. Ceci nous permettra de reconnaître les caractéristiques générales des fonctions recélant des risques d'erreurs. Enfin, nous verrons comment vérifier la précision d'une approximation, en cas de besoin.

Comme nous l'avons vu précédemment, l'algorithme de $\int_y^x$ échantillonne la fonction f(x) pour différentes valeurs de x dans l'intervalle d'intégration. En calculant une moyenne pondérée des valeurs de la fonction aux points échantillons, l'algorithme approche l'intégrale de f(x).

Malheureusement, comme l'algorithme ne sait rien sur f(x) si ce n'est les valeurs que prend la fonction aux points puis comme échantillons, il ne peut pas faire la différence entre f(x) et une fonction quelconque coïncidant avec f(x) aux points échantillons. Cette situation est décrite dans le schéma suivant qui représente (sur une partie de l'intervalle d'intégration) trois parmi les infiniment nombreuses fonctions dont les graphes comprennent un nombre fini très élevé de points échantillons.

Avec ce nombre de points, l'algorithme effectue le même calcul pour toutes les fonctions représentées. Les intégrales réelles des fonctions tracées en noir et en jaune sont pratiquement identiques, l'approximation sera donc relativement précise si f(x) est l'une de ces fonctions. Par contre, l'intégrale réelle de la fonction tracée en bleu est très différente de celle des autres, par conséquent l'approximation actuelle sera plutôt imprécise si cette fonction est f(x).

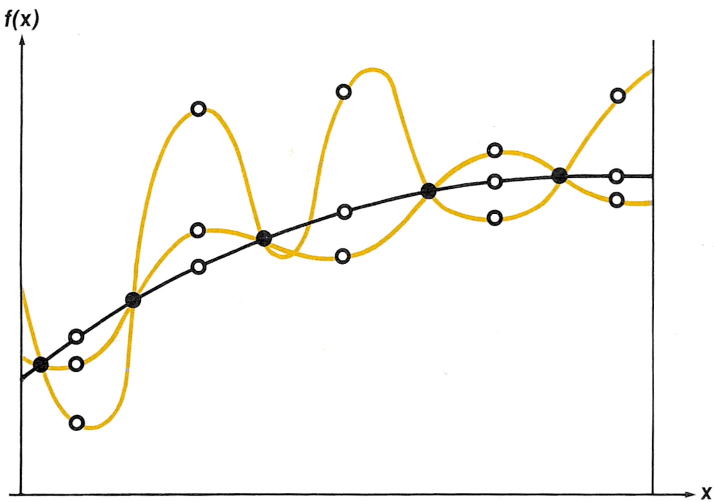


Supposons que la différence entre l'approximation calculée d'après ce nombre de points et les approximations précédentes soit inférieure à l'incertitude indiquée par le nombre de chiffres spécifiés dans le format d'affichage. Dans ce cas, l'algorithme de $\boxed{\int_y^x}$ s'arrête, affichant l'approximation actuelle comme étant la meilleure approximation de l'intégrale **compte tenu de l'incertitude que vous avez implicitement acceptée.** Ainsi, pour certaines fonctions, telle que la fonction tracée en bleu, le calculateur peut vous donner une approximation assez peu précise **car il n'échantillonne la fonction qu'à un nombre fini de points.** Ceci constitue le cas extrême du compromis mentionné plus haut entre la précision et le temps de calcul : comme vous ne voulez pas attendre un temps infini (pour échantillonner la fonction à un nombre de points infini), vous n'avez pas la garantie **absolue** d'obtenir une approximation dont la précision corresponde à l'incertitude.

En prenant la situation inverse, supposons que l'incertitude soit tellement faible (parce que le nombre de chiffres spécifié dans le format d'affichage est suffisamment grand) que l'approximation de l'intégrale calculée avec ce nombre de points échantillons ne soit plus assez précise. Dans ce cas, l'algorithme échantillonne f(x) pour des points supplémentaires. Cette situation est décrite dans le deuxième schéma qui représente les trois mêmes fonctions, tracées avec le premier ensemble de points échantillons.

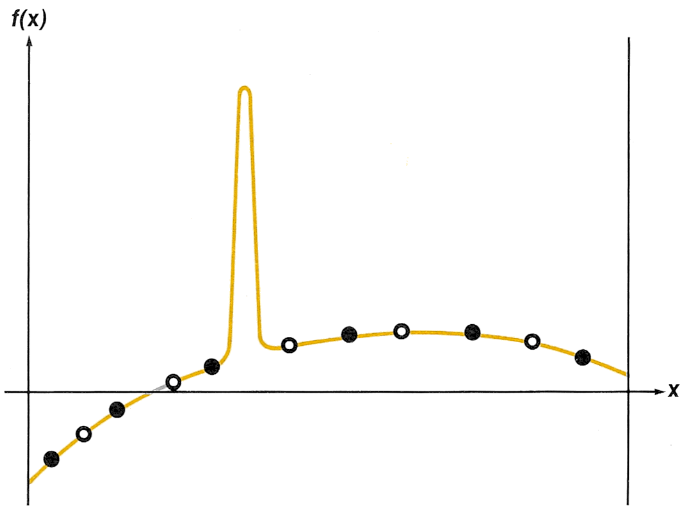
Bien que les trois fonctions représentées sur le schéma aient les mêmes valeurs que pour le premier ensemble de points, la fonction en bleu a des valeurs très différentes aux nouveaux points échantillons. Quand l'algorithme traite les nouvelles valeurs prises par la fonction, il trouve que la différence entre l'approximation actuelle et les précédentes est nettement supérieure à l'incertitude acceptable. Par conséquent, il continue à évaluer la fonction à un nombre de points de plus en plus grand jusqu'à ce que la différence entre approximations successives soit minime. L'approximation obtenue est donc précise puisqu'en n'acceptant qu'une incertitude relativement faible, vous avez consenti à attendre le temps nécessaire.

Néanmoins, dans la pratique, vous ne voulez pas attendre une éternité (vous n'auriez plus besoin des résultats alors!). En imposant cette contrainte à l'algorithme, vous devez admettre que la fonction ne sera pas évaluée à un nombre de points infini et que, par conséquent, l'algorithme risque de ne pas prendre en compte les « pics » éventuels de la fonction. Cette situation est décrite dans le troisième schéma qui représente une fonction continue sauf en un endroit.



Malgré la densité relative des points échantillons, le « pic » ne peut être décelé par aucun échantillon. Comme, au bout de plusieurs itérations successives, les approximations sont très voisines, l'algorithme s'arrête sur une approximation fausse puisqu'il n'a pas pu détecter le pic.

Ceci provient du fait que la fonction paraissait avoir un comportement très régulier par ailleurs, dans l'intervalle d'intégration. Sauf pour ce pic, elle est continue dans tout l'intervalle représenté sur le schéma. (En fait, si l'on considérait le graphe de ces fonctions dans tout l'intervalle d'intégration, on s'apercevrait peut-être qu'elles ne sont pas continues mais présentent des variations soudaines. Le schéma ne représente qu'une vue développée d'une petite partie de l'intervalle d'intégration donnant une apparence de continuité aux brusques variations caractéristiques). En évaluant la fonction avec un nombre de points suffisamment dense, l'algorithme peut connaître le comportement général de la fonction. Si le pic ne tranchait pas aussi nettement du reste de la forme de la fonction, l'algorithme réussirait à le détecter à l'occasion d'une itération quelconque ou bien détecterait des variations semblables. Dans ce cas, il augmenterait le nombre de points échantillons jusqu'à ce que des itérations successives fournissent des approximations qui tiennent compte des variations rapides, elles aussi **caractéristiques.**



Considérons, par exemple, l'approximation de

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

Comme nous évaluons cette intégrale numériquement, nous allons (à tort, nous le verrons plus loin) représenter la borne supérieure de l'intervalle d'intégration par 10^{99} qui est pratiquement le plus grand nombre que l'on puisse introduire dans le calculateur. Que va-t-il se passer ?

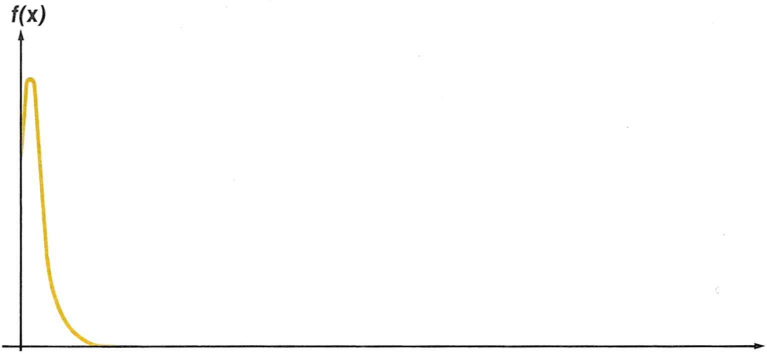
Mettez le commutateur PRGM-RUN sur PRGM et introduisez un sous-programme qui évalue la fonction $f(x) = x\ e^{-x}$.

Appuyez sur	**Affichage**
h LBL 1	**001 – 25, 13, 1**
CHS	**002 – 32**
g e^x	**003 – 15 1**
×	**004 – 61**
h RTN	**005 – 25 12**

Remettez le commutateur PRGM-RUN sur RUN, format d'affichage sur SCI 3, et introduisez les bornes de l'intervalle d'intégration dans les registres X et Y.

Appuyez sur	**Affichage**	
f SCI 3	**0,000 00**	Format d'affichage sur SCI 3. (L'affichage suppose qu'il ne reste pas de résultats de l'exemple précédent.)
0 ENTER↑	**0,000 00**	Introduction de la borne inférieure dans le registre Y.
EEX 99	**1, 99**	Introduction de la borne supérieure dans le registre X.
f $\int_y^x$ 1	**0,000 00**	Approximation de l'intégrale.

La réponse est franchement incorrecte puisque l'intégrale réelle de $f(x) = x\ e^{-x}$ de 0 à ∞ est exactement égale à 1. Cette erreur n'est **pas** due au fait que nous avons représenté ∞ par 10^{99} puisque l'intégrale réelle de cette fonction de 0 à 10^{99} est très voisine de 1. La raison de l'erreur est facile à comprendre si l'on considère le graphe de f(x) dans l'intervalle d'intégration.



Ce graphe accuse un pic très voisin de l'origine. (En fait, pour les besoins de l'illustration, la largeur du pic a été considérablement exagérée. A l'échelle réelle de l'intervalle d'intégration, ce pic ne pourrait pas se distinguer de l'axe vertical du graphe). Comme aucun point échantillon n'a pu lui faire découvrir le pic, l'algorithme a supposé que f(x) était constamment égale à zéro dans tout l'intervalle d'intégration. Même en augmentant le nombre de points en format SCI 9, aucun des points supplémentaires n'aurait pu faire découvrir ce pic dans la fonction, dans l'intervalle d'intégration considéré. Une meilleure solution sera proposée après une brève description de la nature générale des fonctions recélant des possibilités d'erreurs.

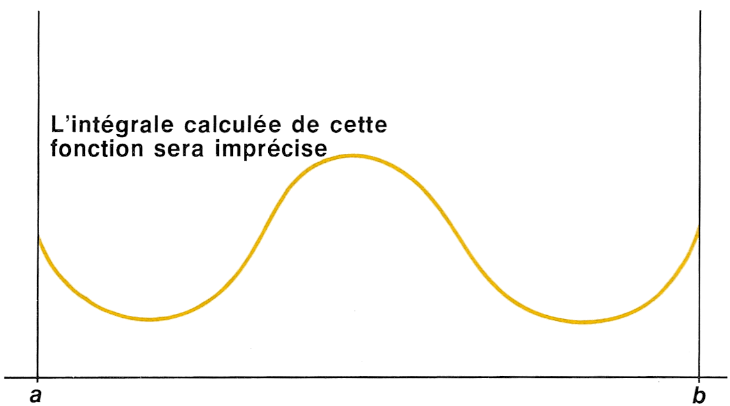
Nous avons vu comment l'algorithme de $\int_y^x$ pouvait donner des résultats erronés lorsque la fonction f(x) présentait quelque part une variation non caractéristique de son comportement général. Heureusement, ces fonctions sont tellement rares qu'il est peu probable que vous ayez à en intégrer une sans le savoir.
La meilleure façon de mettre en lumière les fonctions pouvant conduire à des résultats erronés, consiste à les décrire du point de vue mathématique dans une analyse complexe.* En termes plus simples, ces fonctions peuvent être caractérisées par le rythme de leurs variations et de celles

* Les approximations calculées par le HP-34C vont converger rapidement vers la réponse correcte à condition que l'intégrale f(x), considérée comme une fonction analytique de la variable complexe z, ne présente pas de singularité dans l'intervalle d'intégration ou dans le voisage de cet intervalle et que sa valeur moyenne dans cet intervalle ne soit pas nettement inférieure à la grandeur de la fonction près de cet intervalle.

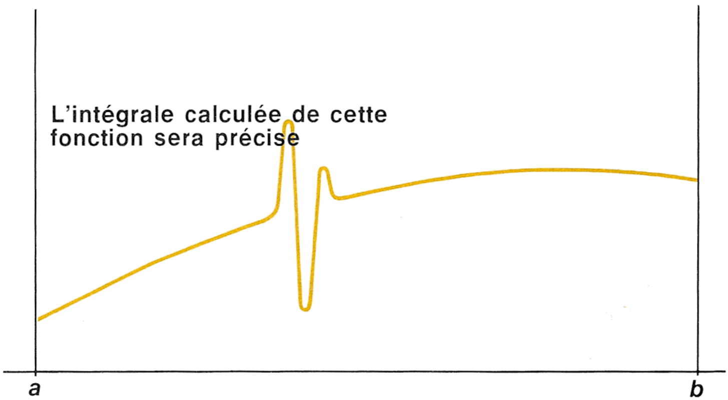
de leurs dérivées de degré inférieur dans l'intervalle d'intégration. En principe, l'algorithme de $\boxed{\int_y^x}$ s'arrêtera d'autant moins rapidement et l'approximation obtenue sera d'autant plus fiable que les variations seront plus rapides dans la fonction ou ses dérivées et que le degré de ces dérivées sera plus bas.

Notez que le rythme des variations de la fonction (ou de ses dérivées de degré inférieur) doit être déterminé par rapport à la largeur de l'intervalle d'intégration. Avec un nombre donné de points échantillons, une fonction f(x) ayant trois pics peut être mieux caractérisée par ses échantillons si les trois pics se répartissent sur pratiquement tout l'intervalle d'intégration que s'ils se concentrent sur une petite partie de l'intervalle. (Ces deux situations sont illustrées dans les schémas suivants). Si nous considérons les variations ou pics comme une sorte d'oscillation à l'intérieur de la fonction, le critère à prendre en considération est le rapport entre la période des oscillations et la largeur de l'intervalle d'intégration : l'algorithme s'arrêtera d'autant plus rapidement et l'approximation obtenue sera d'autant plus fiable que ce rapport sera élevé.

En principe, vous devez connaître suffisamment la fonction à intégrer pour savoir si elle présente des variations brusques dans l'intervalle d'intégration. Si vous ne connaissez pas la fonction et que vous la soupçonnez de poser des problèmes, vous pouvez rapidement tracer quelques points en évaluant la fonction à l'aide du sous-programme que vous avez écrit dans ce but.



Si, pour une raison quelconque, vous doutez de l'exactitude de l'approximation obtenue, il existe un moyen très simple pour la vérifier : divisez l'intervalle d'intégration en deux sous-intervalles ou plus, intégrez la fonction dans chaque sous-intervalle, puis additionnez les approximations obtenues. Votre fonction sera alors échantillonnée pour un nouvel ensemble de points, susceptible de mieux révéler les pics cachés. Si l'approximation initiale était exacte, elle sera égale à la somme des approximations calculées dans les sous-intervalles.



CAUSES DE PROLONGATION DU TEMPS DE CALCUL

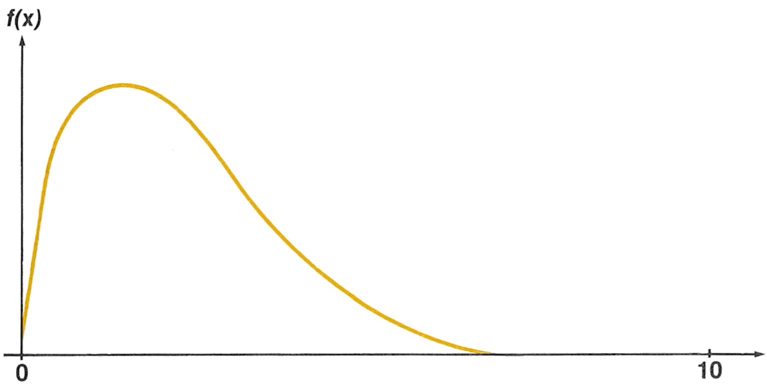
Dans l'exemple précédent, la réponse fournie par l'algorithme était fausse puisqu'il n'avait pas été capable de détecter la variation brusque survenue dans la fonction, car cette variation était trop courte par rapport à la largeur de l'intervalle d'intégration. En réduisant la largeur de l'intervalle on pourrait corriger l'erreur mais si l'intervalle était encore trop large, le temps perdu serait énorme.

Pour certaines intégrales telle que la précédente, le temps de calcul risque d'être inutilement prolongé par le fait que la largeur de l'intervalle d'intégration est trop grande pour certaines caractéristiques de la fonction à intégrer. Prenons une intégrale dont l'intervalle d'intégration est suffisamment large pour demander beaucoup de temps de calcul mais pas assez pour donner un résultat correct. Notons que, comme $f(x) = x$

e^{-x} se rapproche de zéro quand x se rapproche de ∞, l'effet sur l'intégrale de la fonction à des valeurs de x élevées est négligeable. Nous pouvons donc calculer l'intégrale en remplaçant ∞, borne supérieure de l'intervalle d'intégration, par un nombre inférieur à 10^{99}, mettons 10^3.

Appuyez sur	**Affichage**	
0 [ENTER↑]	**0,000 00**	Introduction de la borne inférieure dans le registre Y.
[EEX] 3	**1, 03**	Introduction de la borne supérieure dans le registre X.
[f] [∫ʸₓ] 1	**1,000 00**	Approximation de l'intégrale.
[x⇄y]	**1,824 – 04**	Incertitude de l'intégrale.

La réponse obtenue est la bonne mais le temps de calcul est très long. Ceci peut s'expliquer en comparant le graphe de la fonction dans l'intervalle d'intégration, à celui de la fonction pour des valeurs comprises entre x = 0 et x = 10.



Si l'on compare les deux graphes, on voit que la fonction n'est « intéressante » que pour des valeurs faibles de x. Lorsque x prend des valeurs plus élevées, la fonction n'est plus intéressante puisqu'elle décroît de manière continue et graduellement de façon très prévisible.

Nous avons vu que l'algorithme de [∫ʸₓ] échantillonnait la fonction en augmentant progressivement la densité des points de plus en plus denses jusqu'à ce que la différence entre approximations successives soit suffisamment faible. En d'autres termes, l'algorithme échantillonne la fonction avec un nombre croissant de points jusqu'à ce qu'il dispose d'informations suffisantes sur la fonction pour pouvoir fournir une approximation qui changerait très peu si l'on augmentait encore le nombre de points échantillons.

En posant l'intervalle d'intégration = (0,10) pour que l'algorithme n'ait à échantillonner la fonction qu'à des valeurs où elle est intéressante mais relativement continue, les points échantillons, après les premières itérations, n'apporteraient aucune nouvelle information sur le comportement de la fonction. Par conséquent, quelques itérations suffiraient pour réduire la différence entre approximations successives à une valeur où l'algorithme puisse s'arrêter et fournir une approximation dont la précision soit satisfaisante.

D'autre part, si l'intervalle d'intégration ressemblait davantage à celui du graphe précédent, la plupart des points échantillons se trouveraient dans une région où la pente de la fonction ne varierait pas sensiblement. Les quelques points échantillons correspondant à des valeurs faibles de x feraient ressortir un changement notable dans la fonction d'une itération à l'autre. Par conséquent, la fonction devrait être évaluée à des points supplémentaires avant que la différence entre approximations successives ne soit suffisamment faible.

Pour pouvoir approcher l'intégrale avec la même précision dans les grands et les petits intervalles d'intégration, il faut que la densité des points échantillons soit la même dans la région où la fonction est intéressante. Or, si l'intervalle est plus grand, le nombre total de points nécessaires pour obtenir la même densité est plus élevé. Les itérations doivent donc être plus nombreuses dans les grands intervalles pour atteindre la même précision que dans les petits, ce qui prolonge considérablement le temps de calcul de l'intégrale.

Comme le temps de calcul dépend du moment où les points échantillons atteignent une certaine densité dans la région où la fonction est intéressante, le calcul de l'intégrale d'une fonction sera d'autant plus long que l'intervalle d'intégration comprendra plus de régions où la fonction n'est pas intéressante. Heureusement, dans ces calculs, vous pouvez modifier le problème de manière à réduire considérablement le temps de calcul. Deux techniques sont utilisables : 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, le nombre de points nécessaires pour obtenir une approximation qui varie peu d'une itération à l'autre est élevé. Ce n'est pas le cas des régions où la pente de la fonction reste relativement constante, car même si l'on continuait à augmenter le nombre de points échantillons, on n'obtiendrait aucune nouvelle information capable de changer la différence entre approximations successives. Dans ces régions, on peut donc obtenir la même précision avec un nombre de points nettement inférieur en perdant moins de temps. Pour cela, on procède de la manière suivante :

1. Division de l'intervalle d'intégration en sous-intervalles dans lesquels la fonction est intéressante et en sous-intervalles où elle ne l'est pas.
2. Dans les sous-intervalles où la fonction est intéressante, calcul de l'intégrale dans le format d'affichage correspondant à la précision générale recherchée.
3. Dans les intervalles où la fonction n'est pas intéressante ou n'apporte que peu d'informations au calcul de l'intégrale, on calcule celle-ci avec une précision inférieure, c'est-à-dire en prenant un format d'affichage avec moins de chiffres.
4. Pour obtenir l'intégrale sur tout l'intervalle d'intégration, on ajoute les approximations et les incertitudes des intégrales calculées dans chaque sous-intervalle. Cette opération s'effectue facilement avec la touche [Σ+]

Avant de subdiviser l'intervalle d'intégration, vérifiez la possibilité d'un dépassement inférieur de capacité dans l'évaluation de la fonction autour de la borne supérieure ou inférieure d'intégration. Comme il est vain d'évaluer la fonction à des valeurs de x où se produit un dépassement inférieur de capacité, on peut, dans certains cas, réduire la borne supérieure d'intégration et par là le temps de calcul.

Rappelons qu'une fois que vous avez introduit le sous-programme d'évaluation de f(x), vous pouvez calculer f(x) pour n'importe quelle valeur de x en introduisant cette valeur dans le registre X et en appuyant sur [ENTER] [ENTER] [ENTER] [GSB] suivis du label du sous-programme.

En cas de dépassement inférieur de capacité à la borne supérieure de l'intervalle d'intégration, essayez des nombres plus petits jusqu'à ce que vous vous approchiez du point où le calculateur ne « déborde » plus.

Appuyez sur	**Affichage**	
[EEX] 3	**1, 03**	Introduction de la borne supérieure dans le registre X.
[ENTER↑] [ENTER↑] [ENTER↑]	**1,000 03**	Remplissage de la pile avec x.
[GSB] 1	**0,000 00**	Dépassement inférieur de capacité à la borne supérieure.

Rappelez-vous que lorsque le calcul d'une quantité quelconque donne un résultat inférieur à 10^{-99}, le résultat est remplacé par un zéro. C'est ce que l'on appelle dépassement inférieur de capacité.

300 ENTER↑ ENTER↑ ENTER↑	**3,000 02**	Essai d'une valeur plus faible de x.
GSB 1	**0,000 00**	A nouveau, dépassement de capacité.
200 ENTER↑ ENTER↑ ENTER↑	**2,000 02**	Essai d'une valeur plus faible de x.
GSB 1	**2,768 −85**	Le calculateur ne « déborde » pas à x = 200. Essai d'un nombre compris entre 200 et 250.
225 ENTER↑ ENTER↑ ENTER↑	**2,250 02**	
GSB 1	**4,324 −96**	Le calculateur est proche du dépassement inférieur de capacité.

Vous pouvez maintenant rechercher la plus faible valeur de x pour laquelle il y a dépassement inférieur de capacité en utilisant la touche SOLVE.

Appuyez sur	**Affichage**	
g R↓	**2,250 02**	Descente dans la pile jusqu'à ce que la dernière valeur essayée se trouve dans les registres X et Y.
f SOLVE 1	**2,280 02**	La valeur minimale de x à laquelle il y a dépassement inférieur de capacité est environ 228.

Nous savons désormais que nous allons nous limiter à des valeurs de x comprises entre 0 et 228. Comme la fonction n'est intéressante pour des valeurs de x inférieures à 10, divisons l'intervalle d'intégration en ce point. Le problème devient :

$$\int_0^{1\,000} xe^{-x}dx \approx \int_0^{228} xe^{-x}dx = \int_0^{10} xe^{-x}dx + \int_{10}^{228} xe^{-x}dx$$

Appuyez sur	**Affichage**	
0 ENTER↑	**0,000 00**	Introduction de la borne inférieure du premier sous-intervalle.

10	10,	Introduction de la borne supérieure du premier sous-intervalle.
f ∫ 1	9,995 −01	Calcul de l'intégrale entre 0 et 10 en SCI 3.
f CLEAR Σ	9,995 −01	Effacement des registres-mémoire.
f Σ+	1,000 00	Somme de l'approximation et de son incertitude dans les registres R_1 et R_3.
x⇄y	1,841 −04	Incertitude de l'approximation.
g R↓ g R↓	1,000 01	Descente dans la pile jusqu'à apparition de la borne supérieure du premier intervalle dans le registre X.
228	228,	Introduction de la borne supérieure du deuxième intervalle dans le registre X. Montée de la borne supérieure du premier intervalle dans le registre Y. Elle devient la borne inférieure du deuxième intervalle.
f SCI 0	2, 02	Format d'affichage SCI 0 pour un calcul rapide dans l'intervalle 10, 228. Si l'incertitude de l'approximation n'est pas suffisamment précise, l'approximation peut être reprise avec un format comportant un nombre plus important de chiffres.
f ∫ 1	5, −04	Calcul de l'intégrale entre 10 et 228 en format SCI 0.
f SCI 3	5,328 −04	Retour en format SCI 3.
x⇄y	7,568 −05	Vérification de l'incertitude de l'approximation. Comme elle est inférieure à celle du premier sous-intervalle, l'approximation obtenue en format SCI 0 est suffisamment précise.
x⇄y	5,328 −04	Retour de l'approximation et de son incertitude dans les registres X et Y, respectivement, avant sommation dans les registres mémoire.

f Σ+	**2,000 00**	Sommation de l'approximation et de son incertitude.
RCL f Σ+	**1,000 00**	Intégrale dans tout l'intervalle (0,228)
x⇄y	**2,598 −04**	Incertitude de l'intégrale.

Le calcul de l'intégrale dans les deux sous-intervalles n'a pris qu'une fraction du temps nécessaire au calcul de l'intégrale dans l'intervalle (0,228). Or, l'incertitude obtenue pour l'approximation totale est à peine supérieure à celle de l'approximation unique calculée sur tout l'intervalle.

Transformation de variables

Dans beaucoup de cas où la fonction change très lentement sur la plus grande partie d'un intervalle d'intégration très large, le temps de calcul de l'intégrale peut être réduit par une transformation de variables.

Reprenons l'intégrale

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

Soit $u = e^{-x}$

On a $x = -\ln u$

et $dx = -du/u$

En substituant ces nouvelles valeurs dans l'intégrale $\int_0^\infty xe^{-x}dx$, on obtient :

$$\int_0^\infty xe^{-x}dx = \int_{-0}^{e^{-\infty}} (-\ln u)\,(u)\,(du/u) = \int_1^0 \ln u\, du$$

Mettez le commutateur PRGM-RUN sur PRGM et introduisez un sous-programme évaluant la fonction f(u) = 1n u.

Appuyez sur	**Affichage**
h LBL 3	**001 – 25, 13, 3**
f LN	**002 – 14 1**
h RTN	**003 – 25 12**

Remettez le commutateur PRGM-RUN sur RUN et introduisez les bornes de l'intervalle d'intégration, puis appuyez sur [f] [∫] 3 pour calculer l'intégrale.

Appuyez sur	Affichage	
1 [ENTER↑]	**1,000 00**	Introduction de la borne inférieure.
0	**0,**	Introduction de la borne supérieure.
[f] [∫] 3	**9,998 – 01**	Approximation de l'intégrale équivalente.
[x⇄y]	**2,130 – 04**	Incertitude de l'approximation.

Considérant l'incertitude de cette approximation, elle correspond à la valeur calculée plus haut pour l'intégrale originale. Mais elle n'a demandé qu'une fraction du temps de calcul.

ANNEXE C. SERVICE APRÈS VENTE ET MAINTENANCE

VOTRE CALCULATEUR HEWLETT-PACKARD

Votre calculateur possède les caractéristiques fondamentales de tout produit construit par Hewlett-Packard : performances élevées, technologie d'avant-garde et, comme tous les produits de notre société, il apporte une contribution indiscutable. Tous nos calculateurs sont fabriqués suivant des normes très rigoureuses et les opérations de contrôle sont effectuées dans nos propres usines.

Lorsque vous achetez un calculateur Hewlett-Packard, vous traitez avec une société dont l'image de marque est une garantie de ses fabrications.

FONCTIONNEMENT SUR SECTEUR

Le calculateur vous est livré avec une batterie cadmium-nickel rechargeable. Au cas où la batterie serait à charger, vous pouvez alors utiliser le calculateur au moyen du chargeur-adaptateur (dans ce cas, batterie en place dans le calculateur).

ATTENTION

Si vous utilisez le calculateur sur le secteur, sans batterie, vous risquez de le détériorer.

Pour utiliser le chargeur, procédez de la manière suivante :

1. Il n'est pas nécessaire de mettre le calculateur sur OFF.

2. Introduisez le connecteur femelle du chargeur dans la prise arrière du calculateur.

3. Branchez la prise mâle du chargeur sur le secteur.

ATTENTION

Vous risquez de détériorer votre calculateur si vous utilisez un chargeur autre que le chargeur HP fourni avec votre calculateur.

RECHARGE DES BATTERIES

Les batteries se rechargent même pendant l'utilisation du calculateur sur le secteur au moyen du chargeur-adaptateur.

Pour recharger les batteries (batteries en place dans le calculateur), raccordez le calculateur au secteur au moyen du chargeur-adaptateur 110/220 V. Une batterie vide sera complètement rechargée en :

17 heures : calculateur en service (ON);
6 à 12 heures : calculateur hors service (OFF).

Un temps de charge moins long permettra de travailler moins longtemps. Que le calculateur soit en service (ON) ou non (OFF), il peut rester branché sur le secteur, sans aucun risque de surcharge de la batterie.

Remarque

Le chargeur de batterie chauffe légèrement quand il est branché sur le secteur.

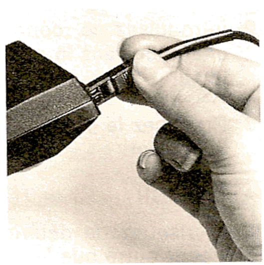
UTILISATION SUR BATTERIE

Pour utiliser le calculateur sur batterie uniquement, retirez seulement la prise femelle du chargeur de l'arrière du calculateur (même s'il n'est pas connecté au calculateur, le chargeur-adaptateur peut rester branché sur le secteur). La batterie du calculateur vous permet une utilisation autonome. Une batterie bien chargée permet 3 heures de calcul intensif. Si vous prenez la précaution d'éteindre votre calculateur quand vous n'en avez pas besoin, vous pourrez travailler tout une journée sur la batterie.

REMPLACEMENT DE LA BATTERIE

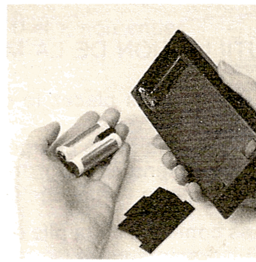
Pour remplacer la batterie, suivez les instructions suivantes :

1. Mettez le commutateur ON/OFF du calculateur sur la position OFF et débranchez le chargeur-adaptateur du calculateur.

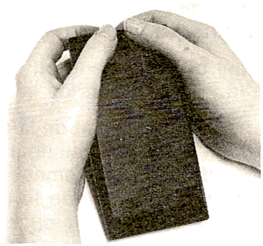


2. Faites glisser les verrous de la porte du logement de batterie jusqu'à ce que celle-ci s'entrouvre. Faites glisser la porte.

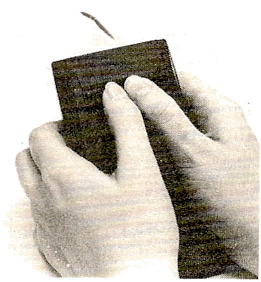
3. Quand la porte est enlevée, retournez le calculateur et imprimez-lui une légère secousse pour que la batterie tombe dans votre main.



4. Mettez en place la nouvelle batterie dans le calculateur. Celui-ci ne fonctionnera que si la batterie est bien à sa place.



5. Introduisez la porte du logement de la batterie et faites-la glisser pour la remettre en place.



6. Retournez le calculateur et remettez-le en marche pour vous assurer de la mise en place correcte de la batterie. Effectuez éventuellement l'ajustement nécessaire.

UTILISATION DE LA MÉMOIRE PERMANENTE

Lorsque vous placez le calculateur sur OFF, les informations suivantes sont conservées :
- — programmes chargés en mémoire
- — contenu des registres de stockage
- — état de l'affichage (FIX, SCI ou ENG et nombre de décimales).

Les contenus de la pile opérationnelle, du registre LAST X, l'unité d'angle et les pointeurs de sous-programmes ne sont pas conservés.

L'exécution du test automatique (STO ENTER↑) n'efface ni la mémoire programme, ni les registres, ni le mode d'affichage.

Quel que soit l'endroit où l'exécution a été arrêtée le pointeur retourne à la ligne 000 lors de la remise sous tension.

Le fonctionnement de la mémoire permanente demande que les batteries soient dans le calculateur. Si l'indicateur de baisse de puissance s'allume, éteindre immédiatement le calculateur et le connecter au secteur ou remplacer la batterie le plus tôt possible. Si les batteries sont complètement déchargées ou retirées du calculateur, toutes les informations contenues dans la mémoire permanente sont perdues. Si vous faites tomber le calculateur ou si l'alimentation de la mémoire permanente est interrompue, que le calculateur soit sur ON ou sur OFF, les programmes en mémoire, le contenu des registres et le mode d'affichage peuvent être perdus. Dans ce cas, lors de la remise sous tension le calculateur affichera Pr Error, appuyez sur une touche quelconque pour annuler cet affichage.

ATTENTION

La mise hors tension du HP-34C pendant l'exécution d'un programme ou d'une fonction peut annuler ou altérer tout ou partie de la mémoire permanente.

ANNEXE D. CONDITIONS D'ERREUR

En cas d'exécution d'opérations illicites — par exemple division par zéro — l'écran affiche Error suivi d'un chiffre. Pour effacer, appuyez sur une des touches. Les opérations illicites sont les suivantes :

Error 0 : opérations mathématiques

[÷]	lorsque $x = 0$
[y^x]	lorsque $y = 0$ et $x \leqslant 0$ ou $y < 0$ et x non entier
[$\sqrt{x}$]	lorsque $x < 0$
[$1/x$]	lorsque $x = 0$
[LOG]	lorsque $x \leqslant 0$
[LN]	lorsque $x \leqslant 0$
[SIN^{-1}]	lorsque $\lvert x \rvert > 1$
[COS^{-1}]	lorsque $\lvert x \rvert > 1$
[STO] [÷]	lorsque $x = 0$
[Δ%]	lorsque $y = 0$

Error 1 : dépassement de capacité dans un registre. Valeur absolue du contenu d'un registre supérieure à $9{,}999999999 \times 10^{99}$ (sauf pour [Σ+] et [Σ−]).

Error 2 : numéro de registre illicite
Registre appelé inexistant ou converti en mémoire programme.

Error 3 :

[$\bar{x}$]	lorsque $n = 0$
[s]	lorsque $n \leqslant 1$
[r]	lorsque $n \leqslant 1$
[$\hat{y}$]	lorsque $n \leqslant 1$
[L.R.]	lorsque $n \leqslant 1$

Remarque : le calculateur affiche aussi Error 3 dans le cas d'une division par zéro ou d'un calcul d'une racine carrée d'un nombre négatif dans l'une des formules suivantes :

$$Sx = \sqrt{\frac{M}{n(n-1)}} \qquad Sy = \sqrt{\frac{N}{n(n-1)}} \qquad r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M} \qquad B = \frac{M\Sigma y - P\Sigma x}{n \cdot M}$$

(A et B sont les valeurs calculées par [L.R.] où y = Ax + B).

$$\hat{y} = \frac{M\Sigma y + P(n.x - \Sigma x)}{n.M}$$

où

$M = n \, \Sigma \, x^2 - (\Sigma x)^2$
$N = n \, \Sigma \, y^2 - (\Sigma y)^2$
$P = n \, \Sigma \, xy - \Sigma \, y \, \Sigma \, x$

Error 4 : numéro de ligne ou label illicite.
Le numéro de ligne appelé n'est pas utilisé ou est inexistant (> 210), tentative pour charger plus de 210 lignes de programme ou label inexistant.

Error 5 :
Appels récurrents de la fonction [$\int_y^x$] ou [SOLVE]
Appel de [$\int_y^x$] par elle-même dans une sous-routine.
Appel de [SOLVE] par elle-même dans une sous-routine.

Error 6 :
[SOLVE] ne peut donner une racine avec les estimations données.

Error 7 :
Label illicite (4-9) avec [$\int$] ou [SOLVE], un nom d'indicateur illicite (4-9).

Error 8 :
Trop de niveau de sous-programmes.

L'autotest ([STO] [ENTER↑]) a détecté un problème. La mémoire programme, les contenus des registres et le format d'affichage ne sont pas effacés par l'autotest.

Pr Error
Mémoire permanente effacée à cause d'une panne d'alimentation.

ANNEXE E. PILE OPÉRATIONNELLE ET REGISTRE LAST X

Votre calculateur a été conçu pour fonctionner de la façon la plus naturelle. Comme vous avez pu le remarquer tout au long de ce manuel, il vous suffit d'effectuer les calculs comme vous les feriez à la main, une seule opération à la fois, presque sans vous préoccupez de ce qui se passe dans la pile opérationnelle. Dans certains cas, particulièrement en programmation, lorsque vous voulez effectuer une opération spéciale dans la pile, les explications et le tableau ci-dessous vous aideront.

FIN D'INTRODUCTION DE DONNÉES

La plupart des opérations, qu'elles soient exécutées comme instructions de programme ou à partir du clavier sont des fins d'introduction de données. Ceci signifie que le calculateur sait que tout chiffre introduit après ces opérations fait partie d'un nouveau nombre.

MOUVEMENTS DANS LA PILE OPÉRATIONNELLE

On peut considérer trois types d'opérations selon la façon dont elles affectent le contenu de la pile :
— les opérations qui interdisent les mouvements
— les opérations qui autorisent les mouvements
— les opérations neutres.

Opérations qui interdisent les mouvements

Il y en a quatre. Tout nombre, introduit après pression de l'une d'entre elles, est écrit dans le registre d'affichage X sans affecter le reste de la pile. Ces quatre opérations sont :
ENTER↑, CLx, Σ+ et Σ-

Opérations qui autorisent les mouvements

Ce sont la plupart des opérations disponibles sur le calculateur, y compris les fonctions mathématiques à un ou deux nombres telles que x^2 ou $\sqrt{x}$. Tout nombre introduit après pression d'une de ces touches déplace les contenus des registres de la pile opérationnelle d'un rang vers le haut. Le passage de mode PRGM en mode RUN fait partie de ces opérations.

Opérations neutres

Certaines opérations telles que et FIX sont neutres, c'est-à-dire qu'elles n'altèrent pas l'état des mouvements de la pile. Si ceux-ci ont été interdits par une pression de ENTER↑, appuyez sur f FIX n et introduisez un nombre ; le contenu de la pile ne sera pas déplacée. De même, si les mouvements ont été autorisés par l'exécution de x^2, appuyez sur f FIX n et introduisez un nombre ; le contenu de la pile se trouve décalé vers le haut.

Les opérations neutres sont les suivantes :

FIX	BST	CLEAR Σ
SCI	SST (en mode RUN, SST peut exécuter une instruction autorisant les mouvements de la pile)	CHS
ENG		MANT
DEG		R/S
RAD		PSE
GRD	MEM	
DSP I	CLEAR PREFIX	
GTO • nnn	CLEAR REG	

Opérations conservant x dans le registre LAST x

−	Σ+	SIN⁻¹	1/x
+	Σ−	x!	y^x
×	%	COS	e^x
÷	Δ%	COS⁻¹	10^x
→H.MS	ŷ	TAN	→R
→H	FRAC	TAN⁻¹	→P
ABS	INT	√x	
	LN	x^2	
→R	LOG		
→D	SIN		

INDEX

Carte de service

En cas de mauvais fonctionnement de votre calculateur, reportez-vous au Manuel d'utilisation. Si une révision s'avère nécessaire, veuillez nous retourner le calculateur avec chargeur, batterie et cette carte de service dûment remplie (voir les instructions d'expédition).

Quel est votre problème ?

☐ Absence d'affichage ☐ Affichage incorrect après l'autotest

☐ Chargeur de batterie ☐ Batterie ☐ Panne intermittente

Description du problème : ____________________

____

Sous garantie : ☐ Oui ☐ Non

Aucune garantie ne sera prise en considération sans présentation d'une preuve de la date de l'achat (facture ou carte de service remplie par le vendeur lors de l'achat).
Pour les machines hors garantie, le renvoi de cette carte avec un calculateur en réparation est considéré comme l'acceptation du coût de celle-ci.
Votre revendeur pourra vous donner toute information nécessaire sur les prix des réparations. Il pourra de même se charger des contacts avec notre société.

Carte de service Série E

En cas de mauvais fonctionnement de votre calculateur, reportez-vous au Manuel d'utilisation. Si une révision s'avère nécessaire, veuillez nous retourner le calculateur avec chargeur, batterie et cette carte de service dûment remplie (voir les instructions d'expédition).

Quel est votre problème?

☐ Absence d'affichage ☐ Affichage incorrect après l'autotest

☐ Chargeur de batterie ☐ Batterie ☐ Panne intermittente

Description du problème: ______________________________

____

Sous garantie: ☐ Oui ☐ Non

Aucune garantie ne sera prise en considération sans présentation d'une preuve de la date de l'achat (facture ou carte de service remplie par le vendeur lors de l'achat).
Pour les matériels hors garantie, la carte est considérée par HP comme une acceptation de la réparation et du coût de celle-ci.
Pour toute information complémentaire concernant les coûts des réparations, contactez le revendeur le plus proche.

Carte d'enregistrement HP-34C

Vous nous aiderez en nous retournant cette carte dûment remplie. Merci.

1) Pays de l'achat: ____________

2) Date de l'achat:
Mois ______ Année 19____

3) Lieu de l'achat: ____________

- ☐ 1 Grand magasin
- ☐ 2 Revendeur matériel de bureau
- ☐ 3 Magasin radio/hifi/photo
- ☐ 4 Association d'étudiants
- ☐ 5 Directement chez HP
- ☐ 6 Autre : ____________

4) Délai de livraison

- ☐ 1 Livraison immédiate
- ☐ 2 Moins de 2 semaines
- ☐ 3 Plus de 2 semaines

5) Type d'achat

- ☐ 1 Personnel
- ☐ 2 Par votre société
- ☐ 3 Cadeau

6) Comment avez-vous connu ce calculateur ?

- ☐ 1 Article de presse
- ☐ 2 Publicité
- ☐ 3 Envoi de documentation
- ☐ 4 Autre utilisateur
- ☐ 5 Mon professeur
- ☐ 6 J'ai déjà un calculateur HP
- ☐ 7 Démonstration dans un magasin
- ☐ 8 Visite d'un vendeur

7) Votre appréciation

de la documentation		des conseils
☐ 1	Excellente	☐ 11
☐ 2	Satisfaisante	☐ 12
☐ 3	Correcte	☐ 13
☐ 4	Mauvaise	☐ 14
☐ 5	Pas nécessaire	☐ 15

8) Votre expérience du calculateur

- ☐ 1 C'est mon premier calculateur
- ☐ 2 Calculatrice 4 fonctions

Calculateur scientifique

Non programmable		Programmable
☐ 30	HP	☐ 50
☐ 31	T.I.	☐ 51
☐ 32	Casio	☐ 52
☐ 33	Canon	☐ 53
☐ 41	Toshiba	☐ 61
☐ 40	Sinclair	☐ 60
☐ 37	Commodore	☐ 57

9) Importance des caractéristiques

	Indispensable	Utile	Sans int.
Solve	☐ 1	☐ 7	☐ 13
Intégrale	☐ 2	☐ 8	☐ 14
Mém. continue	☐ 3	☐ 9	☐ 15
Alloc. mém.	☐ 4	☐ 10	☐ 16
Insertion/suppr.	☐ 5	☐ 11	☐ 17
Adressage ind.	☐ 6	☐ 12	☐ 18

10) Principaux domaines d'applications dans l'ordre (1, 2, 3, etc.)

- ☐ 1 Physique
- ☐ 2 Chimie
- ☐ 3 Médecine/biologie
- ☐ 4 Mathématique
- ☐ 5 Statistiques
- ☐ 6 Logique
- ☐ 7 Electronique
- ☐ 8 Mécanique
- ☐ 9 Construction
- ☐ 10 Autre techn. : ____________
- ☐ 11 Finances/gestion
- ☐ 41 Navigation mer
- ☐ 42 Navigation air
- ☐ 43 Education
- ☐ 44 Jeux
- ☐ 45 Loisirs : ____________
- ☐ 46 Autre : ____________

11) Votre formation ou vos études actuelles

- ☐ 1 Apprentissage
- ☐ 2 Lycée/CET

Ecole professionnelle :

- ☐ 3 Technique
- ☐ 4 Commercial
- ☐ 5 Autre : ____________

Université :

- ☐ 6 Sciences naturelles
- ☐ 7 Médecine
- ☐ 8 Mathématique
- ☐ 9 Ingénierie
- ☐ 10 Economie
- ☐ 11 Sciences sociales
- ☐ 12 Autre : ____________

12) Votre groupe d'âge

- ☐ 1 Moins de 14 ans
- ☐ 2 15–19 ans
- ☐ 3 20–25 ans
- ☐ 4 26–35 ans
- ☐ 5 36–50 ans
- ☐ 6 Plus de 50 ans

13) Votre activité

- ☐ 1 Etudiant
- ☐ 11 Education/formation
- ☐ 12 Recherche et développement
- ☐ 13 Production
- ☐ 14 Organisation/planification
- ☐ 15 Ventes
- ☐ 16 Comptabilité/finances
- ☐ 17 Informatique
- ☐ 18 Gestion
- ☐ 19 Conseils techniques
- ☐ 20 Conseils gestion
- ☐ 21 Profession libérale

14) Votre position

- ☐ 1 Président de société/propriétaire
- ☐ 2 Directeur général
- ☐ 3 Directeur de département
- ☐ 4 Responsable de service
- ☐ 5 Spécialiste
- ☐ 6 Employé

Merci de nous retourner cette carte rapidement !

Nom ______________________________

Adresse ______________________________

Code postal ______________________________

Localité ______________________________

Pays ______________________________

Société ______________________________

Position ______________________________

Cette adresse ne doit en aucun cas être utilisée pour envoyer votre calculateur en réparation.
Consultez le chapitre «Maintenance» du manuel.

HEWLETT-PACKARD S.A.
P.O. Box
CH-1217 MEYRIN 2
Switzerland

Calculateurs HEWLETT PACKARD Série E

Renseignements d'achat

(à remplir par le vendeur)

Modèle HP ________ E

Numéro de série ____________________

Date de l'achat ____________________

Numéro de facture ____________________

Vendu par ____________________

(Cachet et signature du revendeur)

Renseignements client

Nom ____________________

Rue ____________________ N° ________

Code postal ____________________

Localité ____________________

Pays ____________________

Téléphone ____________________

HEWLETT PACKARD

00034-90008 KWP

Imprimé à Singapour