

Manuel d'utilisation



Un petit effort !

En prenant le temps de remplir cette carte réponse, vous nous aidez à mieux comprendre vos besoins. Lisez d'abord les questions avant de répondre. Merci d'avance.

Modèle HP-28S Date d'achat _____

Nom _____

Adresse _____

Code postal. Ville _____

Age _____ Tél. (_____) _____ Prof. ____ ou privé ____

1. Quel POSTE occupez-vous ? (Ne cochez qu'une seule case)

- | | | |
|---|---|--|
| 101 <input type="checkbox"/> Etudiant | 105 <input type="checkbox"/> Cadre sup. | 109 <input type="checkbox"/> Indépendant, prof. libérale |
| 102 <input type="checkbox"/> Educateur, chercheur | 106 <input type="checkbox"/> P.d.g. | 110 <input type="checkbox"/> Retraité |
| 103 <input type="checkbox"/> Cadre | 107 <input type="checkbox"/> Représentant | 111 <input type="checkbox"/> Autre _____ |
| 104 <input type="checkbox"/> Cadre moyen | 108 <input type="checkbox"/> Technicien | |

2. Quel est votre DOMAINE D'ACTIVITE ? (Ne cochez qu'une seule case)

- | | |
|---|---|
| 201 <input type="checkbox"/> Mécanique | 209 <input type="checkbox"/> Achats, planning, gestion des stocks |
| 202 <input type="checkbox"/> Génie civil | 210 <input type="checkbox"/> Comptabilité, audit |
| 203 <input type="checkbox"/> Electricité/électronique | 211 <input type="checkbox"/> Finance, analyse d'investissements |
| 204 <input type="checkbox"/> Chimie | 212 <input type="checkbox"/> Services administratifs/direction générale |
| 205 <input type="checkbox"/> Autre ingénierie _____ | 213 <input type="checkbox"/> Marketing |
| 206 <input type="checkbox"/> Topographie | 214 <input type="checkbox"/> Ventes |
| 207 <input type="checkbox"/> Informatique | 215 <input type="checkbox"/> Services après-vente, maintenance |
| 208 <input type="checkbox"/> Contrôle de qualité | 216 <input type="checkbox"/> Autre _____ |

3. Dans quel SECTEUR travaillez-vous ? (Ne cochez qu'une seule case)

- | | |
|--|---|
| 301 <input type="checkbox"/> Education | 310 <input type="checkbox"/> Chimie, raffinage |
| 302 <input type="checkbox"/> Banque, finance | 311 <input type="checkbox"/> Agriculture, élevage, expl. forestière |
| 303 <input type="checkbox"/> Assurance | 312 <input type="checkbox"/> Industrie et distribution agro-alimentaire |
| 304 <input type="checkbox"/> Immobilier | 313 <input type="checkbox"/> Fabrication de biens industriels |
| 305 <input type="checkbox"/> Affaires/conseils commerciaux | 314 <input type="checkbox"/> Fabrication de biens de consommation |
| 306 <input type="checkbox"/> Conseils techniques | 315 <input type="checkbox"/> Transports |
| 307 <input type="checkbox"/> Logiciel, services informatiques | 316 <input type="checkbox"/> Communication, distr. él., gaz, tél. |
| 308 <input type="checkbox"/> Construction, architecture | 317 <input type="checkbox"/> Services publics, armée |
| 309 <input type="checkbox"/> Mines, extraction pétrolière, prospection | 318 <input type="checkbox"/> Autre _____ |

4. Où avez-vous acheté votre calculateur HP ? (Ne cochez qu'une seule case)

- | | |
|--|--|
| 401 <input type="checkbox"/> Magasin de produits informatiques | 407 <input type="checkbox"/> Vente par correspondance |
| 402 <input type="checkbox"/> Magasin de fournitures de bureau | 408 <input type="checkbox"/> Papeterie |
| 403 <input type="checkbox"/> Librairie | 409 <input type="checkbox"/> Acheté par entreprise/école |
| 404 <input type="checkbox"/> Grand magasin | 410 <input type="checkbox"/> Directement chez HP |
| 406 <input type="checkbox"/> Sur catalogue | 411 <input type="checkbox"/> Autre _____ |

5. Ou avez-vous entendu parler de ce modèle la première fois ?

- | | |
|--|---|
| 501 <input type="checkbox"/> Utilisateur de calculateur HP | 505 <input type="checkbox"/> Publi-postage |
| 502 <input type="checkbox"/> Amis, collègues, professeur | 506 <input type="checkbox"/> Vendeur |
| 503 <input type="checkbox"/> Publicité écrite | 507 <input type="checkbox"/> Brochure dans un magasin |
| 504 <input type="checkbox"/> Articles de presse | 508 <input type="checkbox"/> Autre _____ |

Prière
d'affranchir

Hewlett-Packard France
Département Calculatrices
45, rue des 3 sœurs
Centre d'Affaires Paris Nord II
F-93420 Villepinte, France



HP-28S

Advanced Scientific Calculator

Manuel d'utilisation



Edition 1 février 1988
Numéro de référence 00028-90080

Avertissement

1. Les informations contenues dans ce document peuvent faire l'objet de modifications sans préavis.
2. En raison de la complexité des techniques informatiques, ce document est remis au lecteur dans le seul but de faciliter sa compréhension du produit dont il traite. HPF décline en conséquence toute responsabilité pour tout dommage pouvant résulter des informations contenues dans ce document.
3. HPF ne garantit ni la fiabilité ni les conséquences de l'utilisation de ses produits logiciels lorsqu'ils sont utilisés sur des produits dont il n'a pas assuré la fourniture.
4. Les informations contenues dans ce document sont originales. Elles ont été conçues et mises au point par Hewlett-Packard. L'acheteur s'interdit en conséquence, sauf accord préalable et écrit de HPF :
 - de les divulguer ou d'en faciliter la divulgation ;
 - de les copier ou de les reproduire en tout ou en partie par n'importe quel moyen et sous n'importe quelle forme ;
 - de les traduire dans toute autre langue.

© Hewlett-Packard Co. 1987. Tous droits réservés.

Pour des informations plus spécifiques sur la garantie qui couvre ce calculateur, consultez les pages 291 et 295.

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

Historique de l'impression

Edition 1

Février 1988

Mfg. No. 00028-90081

Bienvenue au HP-28S

Félicitations ! Le HP-28S vous aidera à résoudre des problèmes complexes, dont certains étaient inaccessibles aux calculateurs jusqu'à présent. Le HP-28S combine une grande puissance de calcul numérique à une nouvelle dimension : le *calcul symbolique*. Ce calculateur permet de formuler un problème de manière symbolique, d'obtenir une solution symbolique qui montre le comportement global du problème, et d'obtenir, à partir de cette solution symbolique, des valeurs numériques. Voici les principales caractéristiques du HP-28S.

- Manipulation algébrique. Développement, mise en facteur ou réarrangement des termes d'une expression et résolution symbolique d'une équation pour une variable.
- Calcul différentiel et intégral. Calcul des dérivées et des intégrales définies ou non-définies.
- Solutions numériques. Résolution d'une expression ou d'une équation pour toute variable. Il est également possible de résoudre un *système* d'équations linéaires. Vous pouvez utiliser des nombres complexes, des vecteurs et des matrices aussi aisément que des nombres réels.
- Traçage de courbes. Traçage des expressions, des équations et des données statistiques.
- Conversion d'unités. Conversions entre toute combinaison équivalente des 120 unités intégrées, à condition qu'elles soient de même nature. Vous pouvez aussi définir vos propres unités.
- Statistiques. Calculs statistiques sur une ou sur deux variables et calcul des probabilités.
- Bases de nombres binaires. Calcul avec nombres binaires, octaux et hexadécimaux, manipulation de bits.
- Saisie directe des formules algébriques, logique RPN (*notation polonaise inverse*) pour calculs interactifs.

Ce *Manuel d'utilisation* comprend trois parties. La première, « Principes de base », montre comment résoudre quelques problèmes simples. La deuxième partie, « Résumé des fonctions du calculateur » approfondit la description des caractéristiques du calculateur et vous aide à adapter ces exemples à vos problèmes spécifiques. La troisième partie, « Programmation » décrit les caractéristiques des programmes et en présente quelques exemples.

Le *Manuel de référence* décrit les commandes en détail. C'est un dictionnaire de menus, décrivant les concepts et les commandes de chacun des menus du HP-28S.

Nous vous conseillons de passer d'abord aux exemples de la première partie de ce manuel pour vous familiariser avec le fonctionnement pratique du calculateur, puis de passer à la deuxième partie pour approfondir votre compréhension et votre connaissance du calculateur. Si vous vous posez des questions sur une commande, consultez la rubrique qui lui est consacrée dans le *Manuel de référence*. Si c'est la programmation qui vous intéresse, passez à la troisième partie du présent manuel.

Ces manuels vous montrent l'utilisation du HP-28S pour les calculs mathématiques, mais ne constituent pas pour autant un manuel de mathématiques. Ils supposent que vous êtes déjà familiarisé avec les principes mathématiques évoqués ici.

Il n'est par ailleurs pas nécessaire de connaître tous les sujets mathématiques décrits pour se servir du calculateur. Il n'est pas utile de connaître le calcul différentiel et intégral, par exemple, pour effectuer des calculs statistiques.

Table des matières

15	Comment utiliser ce manuel
15	Le contenu
16	Si vous avez un problème précis

1ère partie : principes de base

1	18	Pour commencer
	18	Préliminaires
	18	Ouverture et fermeture du boîtier
	19	Compartiment des piles et liaison IR
	20	Mise sous et hors tension
	20	Effacement de la mémoire (réinitialisation)
	21	Réglage du contraste de l'affichage
	21	Calculs au clavier
	25	Tour d'horizon
	25	Caractéristiques principales et concepts de base
	31	Le catalogue de commandes

2	34	Arithmétique
	36	Saisie et affichage de nombres
	36	Choix du séparateur décimal
	37	Choix du format d'affichage des nombres
	39	Saisie de nombres
	40	Fonctions monadiques
	41	Fonctions diadiques
	41	Addition et soustraction
	41	Multiplication et division
	42	Puissances et racines
	43	Calcul de pourcentage

- 43 Echange des niveaux 1 et 2
- 44 Effacement d'objets dans la pile
- 45 Calculs en chaîne
- 47 Si vous exécutez une mauvaise fonction par erreur

3

- 48 **Utiliser des variables**
- 48 Introduction aux variables
- 49 Création d'une variable numérique
- 50 Rappel d'une variable numérique
- 50 Evaluation d'une variable numérique
- 51 Changement de la valeur d'une variable
- 52 Suppression d'une variable
- 52 Changement du nom d'une variable
- 54 Créer une variable de programme
- 56 Rappel d'une variable de programme
- 56 Evaluation d'une variable de programme
- 57 Noms placés entre guillemets ou non

4

- 58 **Répéter un calcul**
- 58 Création d'une expression
- 60 Création d'un répertoire
- 63 Utilisation du Solver pour répéter un calcul
- 66 Utilisation d'un autre jeu de valeurs
- 68 Utilisation d'une autre expression
- 71 Retour au répertoire HOME
- 72 Résumé

5

- 73 **Fonctions opérant sur les nombres réels**
- 73 Fonctions trigonométriques
- 73 Choix du mode d'angle
- 74 Utilisation de π
- 76 Conversion d'heures et d'angles
- 77 Fonctions logarithmiques, exponentielles et hyperboliques
- 78 Autres fonctions sur les nombres réels
- 79 Définition de nouvelles fonctions

6	82	Fonctions opérant sur les nombres complexes
	82	Utilisation de nombres complexes
	84	Utilisation de coordonnées polaires
	86	Fonction utilisateur pour l'addition de coordonnées polaires
7	89	Représentation graphique d'expressions
	91	Impression d'un tracé
	91	Modification de l'échelle du tracé
	93	Translation du tracé
	94	Redéfinir les limites de traçage
	97	Représentation graphique d'équations
8	98	Résolution d'équations
	98	Recherche du zéro d'une expression
	100	Recherche d'un minimum ou d'un maximum
	103	Calculs sur des flux financiers constants
9	107	Solutions symboliques
	107	Recherche des zéros d'une expression du second degré
	109	Isolation d'une variable
	110	Développement et regroupement
	112	Utilisation de FORM
10	117	Calcul différentiel et intégral
	117	Calcul de la dérivée d'une expression
	118	Dérivation pas-à-pas
	120	Dérivation complète
	120	Intégration d'une expression
	121	Intégration symbolique des polynômes
	122	Intégration numérique des expressions

11	124	Vecteurs et matrices
	124	Vecteurs
	124	Saisie d'un vecteur
	125	Multiplication et division d'un vecteur par un nombre
	125	Addition et soustraction de vecteurs
	126	Calcul du produit vectoriel
	126	Calcul du produit scalaire
	126	Matrices
	127	Saisie de matrice
	127	Visualisation d'une grande matrice
	128	Inversion d'une matrice
	128	Calcul du déterminant
	128	Multiplication de deux tableaux
	128	Multiplication de deux matrices
	129	Multiplication d'une matrice par un vecteur
	130	Solution d'un système d'équations linéaires

12	131	Statistiques
	132	Saisie de données
	133	Modification de données
	134	Statistiques sur une variable
	134	Calcul de la moyenne
	135	Calcul de l'écart-type
	135	Calcul de la variance
	135	Statistiques sur deux variables
	136	Spécification d'une paire de colonnes
	136	Calcul de corrélation
	136	Calcul de la covariance
	137	Régression linéaire
	137	Prévision

13	138	Arithmétique binaire
	138	Choix de la taille de mot
	139	Choix de la base
	139	Saisie d'entiers binaires
	140	Calcul avec des entiers binaires

14	141	Conversion d'unités
	141	Le catalogue UNITS
	143	Conversion d'unités
	144	Conversion de chaînes d'unités
	146	Vérification des unités
	147	Fonctions utilisateur pour conversion d'unités
 15	 149	 Impression
	149	Impression de l'affichage
	150	Impression d'un relevé continu
	151	Impression du niveau 1
	152	Impression de la pile
	152	Impression d'une variable

2ème partie : résumé des caractéristiques du calculateur

16	154	Objets
	155	Nombres réels
	155	Nombres complexes
	156	Entiers binaires
	156	Chaînes
	157	Tableaux
	158	Listes
	159	Noms
	160	Programmes
	161	Objets algébriques
	161	Expressions
	162	Equations
	163	Constantes symboliques
 17	 164	 Opérations, commandes et fonctions

18	166	La ligne de commande
	166	Le menu du curseur
	168	Touches de saisie
	169	Délimiteurs et séparateurs d'objets
	169	Modes de saisie
	171	Exceptions
	171	Sélection manuelle des modes de saisie
	172	Indication des modes par le curseur
	173	Exécution de la ligne de commande
	173	Modification d'objets existants
	174	Rappel des lignes de commande
	175	La ligne de commandes en tant que chaîne

19	176	La pile opérationnelle
	176	Revue des concepts concernant la pile
	177	Affichage du contenu de la pile
	177	Manipulation de la pile
	179	Variables locales
	179	Rappel des derniers arguments
	180	Rappel de la pile
	181	La pile opérationnelle en tant que liste

20	182	Mémoire
	182	Mémoire utilisateur
	182	Variables globales
	183	Répertoires
	187	Caractéristiques de récupération
	188	Manque de mémoire
	190	Optimisation des performances

21	192	Menus
	193	Menus de commandes
	194	Menus des opérations
	194	Menus des variables
	195	Menus personnalisés

22	196	Catalogue des commandes
	197	Recherche d'une commande
	197	Vérification de la syntaxe des commandes
23	198	Evaluation
	199	Données
	199	Noms
	200	Evaluation de noms locaux
	200	Evaluation de noms globaux
	201	Procédure
	201	Evaluation de programmes
	202	Evaluation d'expressions algébriques
	203	Evaluation de fonctions
24	205	Modes
	205	Modes généraux
	207	Modes de saisie et d'affichage
	210	Modes de récupération après erreur
	211	Exceptions mathématiques
	212	Modes d'impression
25	215	Opérations du système
	216	Impression de l'affichage
	216	Contrôle du contraste
	216	Opérations d'effacement
	216	Attention
	217	Arrêt du système
	217	Réinitialisation de la mémoire

218	Opérations de test
218	Test répétitif
219	Test du clavier

3ème partie : programmation

26	222	Structures de programmes
	222	Structure de variable locale
	223	Structures conditionnelles
	226	IF ... THEN ... ELSE ... END
	226	IFTE (fonction If-Then-Else-End)
	227	IF ... THEN ... END
	227	IFT (commande If-Then-End)
	227	Détection d'erreur
	228	Structures de boucles finies
	228	START ... NEXT
	229	FOR compteur ... NEXT
	230	...incrément STEP
	231	Structures de boucles indéfinies
	231	DO ... UNTIL ... END
	232	WHILE ... REPEAT ... END
	233	Structures de programme emboîtées
27	234	Programmes interactifs
	234	Demande de saisie
	235	Demande de choix
	235	Exemple plus complexe
28	240	Exemples de programmes
	241	Fonctions de boîtes
	241	SBOITE (Surface d'une boîte)
	244	SBOITE sans variables locales
	245	RBOITE (rapport de la surface et du volume d'une boîte)
	246	Suite de Fibonacci
	247	FIB1 (Nombres de Fibonacci, récursion)
	248	FIB2 (Nombres de Fibonacci, par boucle)
	249	Comparaison de FIB1 et FIB2
	250	Exécution pas-à-pas

253	Développement et mise en facteur complets
253	MULTI (exécution multiple)
255	EXCO (développement et regroupement complets)
257	Affichage d'un entier binaire
257	COMPLET (ajoute des espaces devant un nombre)
258	PRESERVE (stocke et rappelle l'état précédent)
259	BAFFICHE (affichage binaire)
262	Sommations
263	SOMME (matrice de sommation)
265	ΣLIT (extrait un élément de ΣCOV)
266	ΣX2 (somme des carrés de x)
266	ΣY2 (somme des carrés de y)
267	ΣXY (somme des produits de x et y)
270	Médiane de données statistiques
270	TRI (trie une liste)
272	LMED (médiane d'une liste)
273	MEDIAN (médiane des données statistiques)
275	Changement de répertoires
276	PREC (répertoires parents)
277	SUIV (aller à un sous-répertoire)

Annexes & index

A	282	Assistance, batteries et service
	282	Réponse à quelques questions souvent posées
	286	Piles
	289	Entretien
	289	Environnement
	289	Déterminer si votre calculateur a besoin d'être réparé
	291	Garantie
	293	Si le calculateur a besoin d'être réparé
	296	Les menus et leurs jeux de libellés
	317	Index des touches
	323	Index des sujets

Comment utiliser ce manuel

Nous vous recommandons de lire ce manuel en entier en exécutant chaque exemple cité. Si vous n'en avez pas le temps, nous vous conseillons l'approche suivante :

1. Lisez les cinq premiers chapitres de la première partie pour vous familiariser avec le calculateur.
2. Deux des annexes du *Manuel de référence* comparent le HP-28S avec d'autres types de calculateurs :
 - Si vous connaissez d'autres calculateurs Hewlett-Packard utilisant la « notation polonaise inverse » ($(3 \text{ [ENTER] } 2 \text{ [x] } = 5)$), lisez l'annexe B du Manuel de référence en page 357.
 - Si vous êtes un habitué des calculateurs utilisant la notation algébrique classique ($(3 \text{ [x] } 2 \text{ [=] } 5)$), lisez l'annexe C du *Manuel de référence* en page 363.
3. Si vous êtes plus particulièrement intéressé par un sujet traité plus loin dans la première partie, passez directement à cet endroit et essayez les exemples proposés.

Le contenu

La première partie, « Principes de base » vous indique comment résoudre quelques problèmes simples. Ce sera l'occasion d'apprendre le fonctionnement du HP-28S, les types d'objet et les menus.

La deuxième partie, « Résumé des fonctions du calculateur », va plus loin. Elle présente des options et des détails de l'utilisation qui n'ont pas été couverts dans la première partie.

La troisième partie, « Programmation », décrit les caractéristiques de programmation du HP-28S. Le dernier chapitre, « Exemples de programmes », contient de courts programmes illustrant les techniques de programmation.

Si vous avez un problème précis

Au cours de votre travail, il se peut que vous vous posiez des questions sur les exemples utilisés ou sur les caractéristiques utilisées dans certains des exemples. Ce manuel, ainsi que le *Manuel de référence*, contient des informations supplémentaires.

- Si vous vous heurtez à des problèmes, voyez « Réponses à des questions fréquemment posées » en page 282.
- Consultez l'« Index des touches » en page 327.
- Les commandes de chaque menu sont brièvement décrites à l'annexe D, « Les menus et leurs jeux de libellés », en page 306.
- Pour trouver des informations plus détaillées sur un menu, consultez le *Manuel de référence*. Tous les menus, ainsi que d'autres sujets, sont classés par ordre alphabétique. Le contenu du dictionnaire se trouve sur le dos de couverture du *Manuel de référence*.
- Pour trouver des informations sur une commande, consultez « Index des opérations », en fin de *Manuel de référence*. Vous y trouverez une référence à une page du dictionnaire (un menu, en général) et une référence à une commande précise.

1ère partie

Principes de base

Page	18	1 : Pour commencer
	34	2 : Arithmétique
	48	3 : Utilisation de variables
	58	4 : Calculs répétitifs
	73	5 : Fonctions utilisant des nombres réels
	82	6 : Fonctions utilisant des nombres complexes
	89	7 : Représentation graphique d'expressions
	98	8 : Résolution d'équations
	107	9 : Solutions symboliques
	117	10 : Calcul différentiel et intégral
	124	11 : Vecteurs et matrices
	131	12 : Statistiques
	138	13 : Arithmétique binaire
	141	14 : Conversion d'unités
	149	15 : Impression

Pour commencer

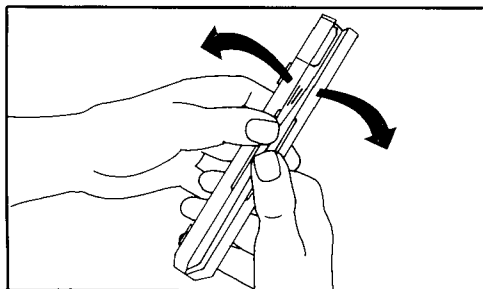
Ce chapitre effectue un survol rapide du HP-28S. Vous y trouverez d'abord les caractéristiques extérieures du calculateur. Ensuite, le clavier est illustré pour mettre en évidence ses caractéristiques principales et celles de l'affichage. Enfin, un exemple très simple montre le fonctionnement du calculateur, et les concepts de base sur lesquels il se fonde.

Préliminaires

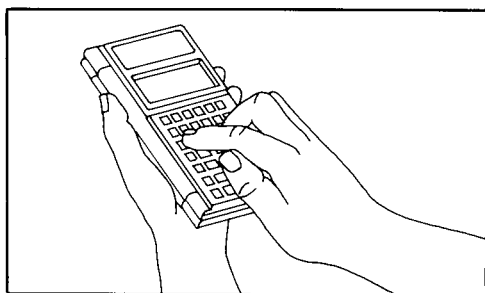
Cette section décrit les caractéristiques de base du calculateur.

Ouverture et fermeture du boîtier

Le calculateur forme son propre boîtier, qui s'ouvre et se referme comme un livre ou un étui à cigarettes, avec les pouces.

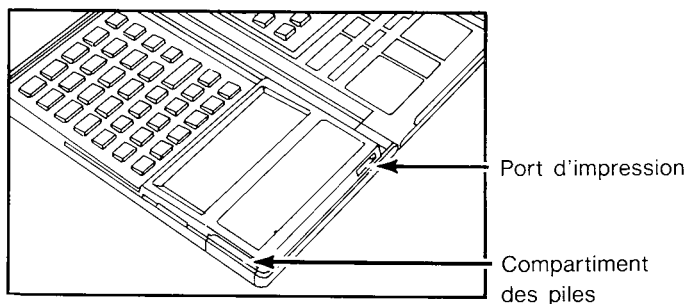


La partie gauche du calculateur—clavier alphabétique—peut être repliée sur le dos du clavier de droite. Ceci est utile lorsque vous utilisez votre calculateur loin de votre bureau ou d'une surface plane, ou pour économiser de l'espace.



Compartiment des piles et liaison IR

Lorsque le calculateur est ouvert, notez l'emplacement du compartiment des piles et de l'émetteur de rayons infra-rouges qui sert à communiquer avec l'imprimante.



Le HP-28S est alimenté par trois piles alcalines au *cadmium nickel*, fournies avec le calculateur. Si elles ne sont pas encore installées, suivez les instructions à partir de la page 213.

Lorsque vous utilisez le HP-28S avec une imprimante, le calculateur lui envoie ses données par des signaux infra-rouges émis par le port d'impression et reçus par l'imprimante ; le fonctionnement de l'imprimante est décrit en détail au chapitre 15.

Mise sous et hors tension

Appuyez sur la touche **[ON]** pour mettre le calculateur sous tension. Le HP-28S possède une *mémoire permanente* et toutes les données présentes dans le calculateur, y compris le contenu de l'affichage, sont conservées lorsque vous éteignez le calculateur.

Lorsque le HP-28S est sous tension, **[ON]** joue le rôle de touche ATTN (*attention*), rappelé par l'inscription qui figure juste en-dessous. La touche **[ON]** efface le texte que vous avez saisi et arrête le déroulement des programmes.

Appuyez sur les touches **[■]** et **[OFF]** pour éteindre le calculateur. Appuyez sur la touche « Shift » (que l'on nomme aussi *touche préfixe*), puis sur la touche **[ON]** ; vous remarquerez que le mot OFF est imprimé juste au-dessus de cette touche.

Si le calculateur est laissé inactif pendant plus de dix minutes, il s'éteint automatiquement pour économiser les piles. Appuyez sur **[ON]** pour le réactiver.

Effacement de la mémoire (réinitialisation)

Vous pouvez remettre le calculateur dans son état initial en le ré-initialisant. *Toutes les informations introduites dans le calculateur sont alors perdues.* Tous les choix de modes que vous aviez faits (choix du séparateur décimal, mode d'angles, etc.) sont effacés et le calculateur reprend les choix *par défaut*.

Pour réinitialiser la mémoire :

1. Appuyez et maintenez la pression sur **[ON]**.
2. Simultanément, appuyez et maintenez la pression sur la touche **[INS]** (à l'extrémité supérieure gauche du clavier de droite).
3. En même temps, appuyez et relâchez **[▶]** (à l'extrémité supérieure droite du clavier de droite).
4. Relâchez **[INS]**.
5. Relâchez **[ON]**.

Le calculateur déclenche son avertisseur et affiche **Memory Lost** (mémoire perdue). Ce message disparaît dès que vous appuyez sur une touche quelconque.

Si vous entamez la procédure, mais changez ensuite d'avis, *continuez à appuyer sur la touche* **[ON]** tout en appuyant sur **[DEL]** (juste à côté de **[INS]**), et ensuite relâchez **[ON]**. La touche **[DEL]** annule la séquence de ré-initialisation.

Réglage du contraste de l'affichage

Il est possible de modifier le contraste de l'affichage pour l'adapter à un angle de vision ou à une intensité lumineuse différents.

Pour régler le contraste :

1. Appuyez et maintenez la pression sur **[ON]**.
2. Appuyez sur **[+]** une ou plusieurs fois pour augmenter le contraste ou sur **[-]** pour le diminuer.
3. Relâchez **[ON]**.

Calculs au clavier

Essayez le petit calcul suivant :

$$(15 + 23) \times \sin 30^\circ$$

Les étapes sont identiques à celles suivies pour un calcul sur papier : calculez d'abord $15 + 23$, ce qui donne un résultat intermédiaire ; calculez ensuite le sinus de 30° , ce qui produit un deuxième résultat intermédiaire ; combinez enfin les deux résultats pour obtenir la réponse.

Si, en introduisant un nombre, vous faites une erreur, vous pouvez :

- Appuyer sur **[◀]** pour effacer le dernier chiffre saisi.
- Appuyer sur **[ON]** pour effacer *tous* les chiffres saisis.

Commencez avec une feuille blanche.

■ CLEAR

4:	
3:	
2:	
1:	

L'affichage montre la *pile*, qui constitue votre zone de travail. La pile est vide pour l'instant.

Appuyez sur [1] [5] pour placer 15 dans la *ligne de commande*.

15

3:	
2:	
1:	
15	

Remarquez que la pile se décale vers le haut pour laisser la place nécessaire à la ligne de commande. Seuls trois niveaux de la pile restent affichés.

Placez 15 dans la pile.

ENTER

4:	
3:	
2:	
1:	15

Le nombre présent dans la ligne de commande va au *niveau 1 de la pile*, comme indiqué par 1 : à gauche. Remarquez que la ligne de commande disparaît et que quatre niveaux de la pile sont de nouveau affichés.

Appuyez sur [2] [3] pour placer 23 dans la ligne de commande.

23

3:	
2:	
1:	15
23	

Placez 23 au niveau 1.

ENTER

4:	
3:	
2:	15
1:	23

Le nombre 15, qui se trouvait au niveau 1, est placé dans le niveau 2.

Ajoutez 15 et 23.

+

```
4:
3:
2:
1: 38
```

Les nombres 15 et 23 sont retirés de la pile, et leur somme, 38, est renvoyée au niveau 1. Laissez ce résultat intermédiaire dans la pile pendant que vous calculez le deuxième résultat intermédiaire.

Pour calculer le sinus de 30° vous allez utiliser le menu TRIG.

TRIG

```
3:
2:
1: 38
SIN ASIN COS ACOS TAN ATAN
```

La ligne inférieure de l'affichage montre les six commandes disponibles. Ces six *libellés* (de **SIN** à **ATAN**) définissent les six *touches de menu* (les touches gris clair se trouvant juste en-dessous de l'affichage).

Appuyez sur **3** **0** pour écrire 30 en ligne de commande.

30

```
2:
1: 38
30
SIN ASIN COS ACOS TAN ATAN
```

Placez 30 au niveau 1.

ENTER

```
3:
2: 38
1: 30
SIN ASIN COS ACOS TAN ATAN
```

Le résultat précédent, $15 + 23 = 38$, est « soulevé » et placé au niveau 2.

Calculez à présent le sinus de 30° .

SIN

```
3:
2: 38
1: .5
SIN ASIN COS ACOS TAN ATAN
```

Le nombre 30 est retiré du niveau 1, et son sinus (0,5) est renvoyé au niveau 1. Le résultat précédent, 38, reste au niveau 2.

Calculez maintenant $38 \times 0,5$.



Les nombres 38 et 0,5 sont retirés des registres 1 et 2 et leur produit, 19, est renvoyé au niveau 1.

Ce qui termine le calcul de

$$(15 + 23) \times \sin 30^\circ = 19.$$

En résumé, voici la procédure générale de calcul :

1. Frappez un nombre dans la *ligne de commande*.
2. Appuyez sur **ENTER** pour placer ce nombre dans la *pile*.
3. Appuyez sur la touche qui exécute la commande (si cette commande ne figure pas sur le clavier, choisissez le *menu* qui la contient et appuyez sur la *touche de menu* qui se trouve sous le libellé approprié).

L'exemple précédent a montré que tous les calculs se passent dans la *pile*. C'est la raison pour laquelle vous appuyez sur **ENTER**. En pratique, *vous n'avez besoin d'appuyer sur ENTER que pour séparer deux nombres saisis consécutivement*—dans l'exemple, pour séparer 15 et 23. Essayez de refaire l'exemple en omettant les deuxième et troisième pressions sur **ENTER**.

Le style de calcul illustré ci-dessus, qui vous fait saisir les nombres dans la pile opérationnelle avant d'exécuter une opération mathématique, est nommé *notation polonaise inverse*, RPN (*Reverse Polish Notation*), ou logique de pile. Presque toutes les commandes du HP-28S, pas seulement les calculs, utilisent cette logique. Ce système utilise deux règles simples :

- Les *arguments* d'une fonction, doivent être présents dans la pile lorsque la fonction est exécutée.
- Les résultats d'une fonction sont renvoyés dans la pile, où ils peuvent devenir les arguments d'une autre fonction.

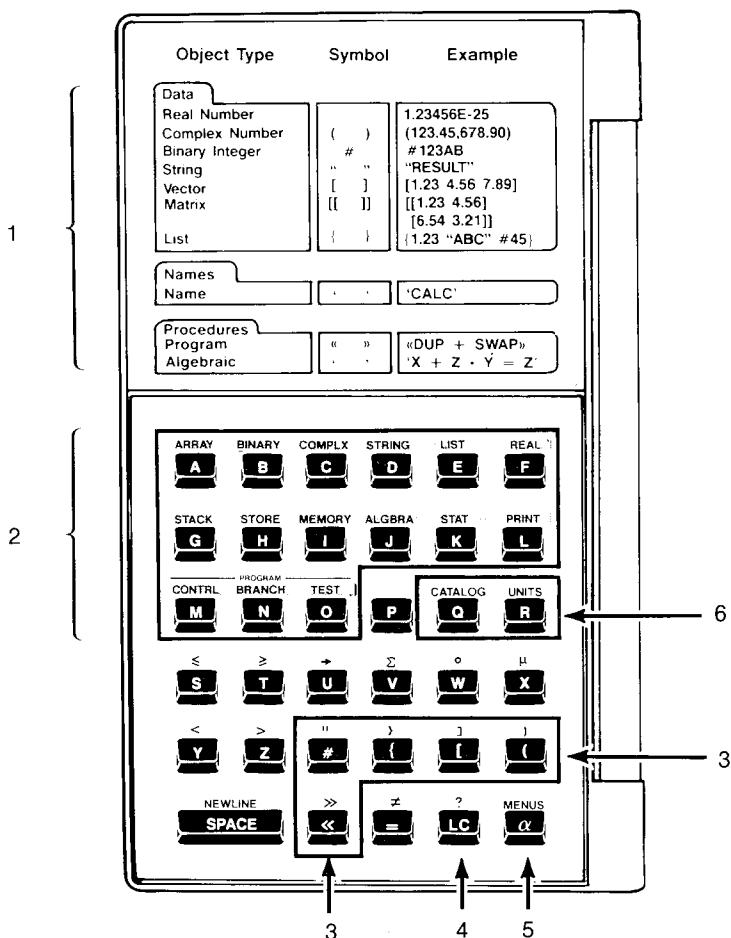
Vous pouvez aussi calculer en saisissant une expression sous forme algébrique, telle qu'elle pourrait apparaître imprimée dans un livre. Au chapitre suivant, vous exécuterez le calcul précédent en utilisant une expression algébrique.

Tour d'horizon

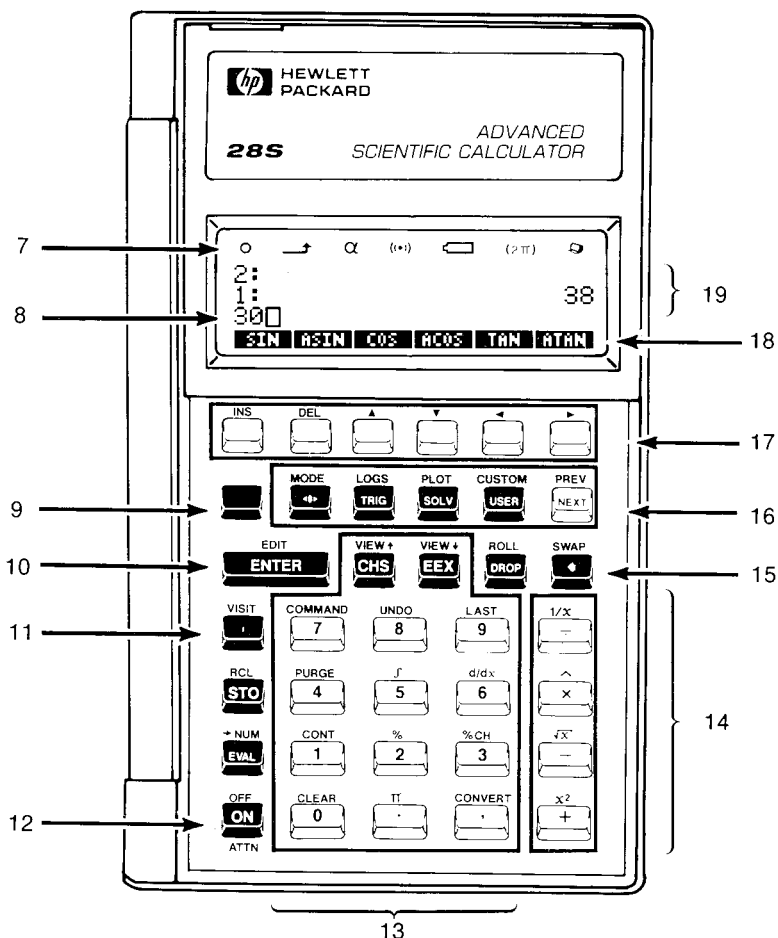
Cette section présente certaines caractéristiques principales du calculateur, y compris un catalogue de commandes qui les énumère et les décrit.

Caractéristiques principales et concepts de base

Les illustrations des pages 26 et 27 montrent le clavier et l'affichage du calculateur, en indiquant leurs parties principales. Les numéros, dans les descriptions qui vont suivre, correspondent à ceux des illustrations.




1. Types d'objets et de formats
2. Choix des menus (avec touche préfixe Shift)
3. Délimiteurs d'objets
4. Minuscules
5. Saisie alphabétique
6. Listes de commandes et d'unités (avec touche préfixe Shift)







- | | |
|--|---|
| 7. Témoins d'état | 13. Saisie numérique |
| 8. Ligne de commande | 14. Arithmétique |
| 9. Touche préfixe (Shift) | 15. Espace arrière |
| 10. Saisie en ligne de commande | 16. Choix des menus, ligne de menu suivante |
| 11. Délimiteur des objets symboliques | 17. Touches de menu |
| 12. Marche/arrêt ; effacement de la ligne de commande ; arrêt de programme | 18. Libellés de menu |
| | 19. Niveaux de la pile opérationnelle |





1. Types d'objets et formats. Ce tableau indique les délimiteurs des 10 types fondamentaux d'objets. Un « objet » est l'un quelconque des éléments avec lesquels vous travaillez sur votre calculateur.

- Nombres réels (*Real Number*), tels que 5 ou $-4,3 \times 10^{15}$.
- Nombres complexes (*Complex Number*), couple de nombres réels représentant un nombre complexe $x + iy$ ou un point sur un plan.
- Nombres entiers binaires (*Binary Integer*), entiers sans signe composés uniquement de zéros et de uns.
- Chaînes (*String*), séquences arbitraires de caractères.
- Vecteurs (*Vector*), tableaux à une seule dimension utilisés en algèbre linéaire.
- Matrices (*Matrix*), tableaux à deux dimensions utilisés en algèbre linéaire.
- Listes (*List*), séquences arbitraires d'objets.
- Noms (*Name*), qui permettent de nommer et de stocker d'autres objets et d'effectuer des calculs symboliques.
- Programmes (*Program*), qui vous permettent de créer vos propres commandes.
- Objets algébriques (*Algebraic*), expressions mathématiques et équations.

2. Choix de menus (avec la touche Shift). Utilisez les touches de choix de menus pour assigner des commandes aux touches de menu. Appuyez par exemple sur  **ARRAY** pour sélectionner le menu **ARRAY** (*tableaux*).

3. Délimiteurs d'objets. Ces symboles indentifient les différents types d'objets (voir au point 15, « Types d'objets et formats », ci-dessus). , par exemple, identifie les entiers binaires, tandis que  et  indentifient les programmes.





Les nombres réels n'ont pas besoin de délimiteurs. Les objets symboliques (noms et expressions algébriques) nécessitent le délimiteur  placé sur le clavier de droite (voir le point 11).

4. Minuscules (Lower-case). Appuyez sur  pour frapper des lettres minuscules. Ce mode sera en cours jusqu'à ce que vous appuyez à nouveau sur , sur  pour traiter la ligne de commande ou sur  pour effacer la ligne de commande.

5. Mode de saisie. La ligne de commande a trois modes de saisie, convenant chacun à un certain type d'objet. Les modes changent automatiquement lorsque vous frappez des objets ; vous pouvez cependant parfois désirer contrôler le changement ; la touche α vous permet de choisir le mode de saisie que vous désirez.

6. Liste des commandes et des unités. ■ $\boxed{\text{CATALOG}}$ produit une liste de toutes les commandes du HP-28S et de leurs arguments (voir page 31). ■ $\boxed{\text{UNITS}}$ produit une liste des unités reconnues par la fonction de conversion (page 141).

7. Témoins d'état. Ces petits symboles, qui apparaissent sur la partie supérieure de l'affichage, indiquent l'état du calculateur :

Témoin d'état	Signification
	Programme suspendu.
	Vous avez appuyé sur ■ (« Shift »).
α	Mode de saisie alpha.
$((\bullet))$	Occupé, non disponible pour la saisie.
	Piles faibles.
(2π)	Mode radians.
	Envoi de données vers l'imprimante.

8. Ligne de commande. Le texte que vous frappez s'inscrit en ligne de commande.

9. Shift. Appuyez sur la touche préfixe, ■, pour exécuter une commande imprimée en couleur au-dessus d'une touche.

10. Saisie de la ligne de commande. Appuyez sur $\boxed{\text{ENTER}}$ pour traiter le texte se trouvant en ligne de commande.

11. Délimiteur pour les objets symboliques. Les délimiteurs identifient les types d'objet ; les objets symboliques sont des noms et des expressions algébriques. Pour saisir un objet symbolique, appuyez sur $\boxed{'}$ avant et, si nécessaire, après l'objet.

Les nombres réels ne nécessitent pas de délimiteurs. Les délimiteurs des autres types d'objets se trouvent dans la partie inférieure du clavier de gauche (voyez les points 1 et 3).

12. Mise sous tension, effacement de la ligne de commande, arrêt du programme. Pour mettre le calculateur sous tension, appuyez sur **[ON]** ; pour l'éteindre, appuyez sur **[OFF]** (OFF est imprimé juste au-dessus de **[ON]**). « Appuyez sur **[OFF]** » veut dire appuyez sur la touche « shift », **[]**, puis sur **[ON]**.)

Lorsque le calculateur est sous tension, **[ON]** se comporte en tant que touche ATTN (*attention*) pour effacer le texte se trouvant en ligne de commande ou arrêter le déroulement d'un programme.

13. Saisie de nombres. Utilisez les touches numériques **[0]** à **[9]**, **[CHS]** (changer de signe) et **[EEX]** (entrer l'exposant). Selon le choix de *séparateur décimal* effectué, (page 36), utilisez **[.]** ou **[/]** pour séparer partie entière et partie fractionnaire. La saisie de nombres est traitée en page 39.

14. Arithmétique. Les fonctions arithmétiques sont décrites dans les sections « Fonctions monadiques », en page 40, et « Fonctions diadiques », en page 41.

15. Espace arrière. Appuyez sur **[←]** pour effacer le dernier caractère frappé.

16. Choix de menu, jeu de libellés suivant. Utilisez les touches de sélection du menu pour assigner des commandes aux touches grises de menu. Appuyez sur **[TRIG]**, par exemple, pour choisir le menu TRIG.

Lorsqu'aucune touche de menu n'est visible, le *menu du curseur* est activé. Les opérations du menu du curseur (**[INS]** à **[▶]**) sont imprimées en blanc au-dessus des touches de menu. Lorsque les touches de menu sont visibles, appuyez sur **[↔]** pour sélectionner le menu du curseur. Pour rétablir le menu précédent, appuyez à nouveau sur **[↔]**.

Un menu peut comporter plusieurs lignes de libellés, avec un maximum de six commandes par ligne. Appuyez sur **[NEXT]** pour afficher la ligne suivante du menu en cours. Appuyez sur **[PREV]** pour ré-afficher la ligne de libellés précédente.

Le clavier de gauche comporte d'autres touches de choix de menu. Vous trouverez une liste alphabétique de tous les menus, y compris une description des commandes de chaque menu, à l'annexe D, « Les menus et leurs jeux de libellés ».

17. Touches de menu. Les touches de menu sont définies par les libellés de menu. Si aucun libellé n'est visible, ces touches effectuent normalement les opérations du menu du curseur, indiquées en blanc au-dessus d'elles.

18. Libellés de menu. Les libellés de menu définissent les fonctions affectées aux touches de menu.

19. Niveaux de la pile. La pile indique les objets avec lesquels vous êtes en train de travailler. Chaque niveau de pile numéroté (niveau 1, niveau 2 etc.) contient *un* objet.

Le catalogue de commandes

Le HP-28S contient un catalogue de commandes, classées par ordre alphabétique. Le catalogue indique la syntaxe de chaque commande et les arguments qui lui sont nécessaires. Voyez l'« Index des opérations », en fin du *Manuel de référence*.

Lancez le catalogue.

■ CATALOG

HEURT			
NEXT	PREV	USE	FETCH QUIT

La première commande est ABORT.

Le listage du catalogue interrompt le fonctionnement normal du calculateur. Les touches **NEXT** et **PREV** affichent les commandes suivantes et précédentes. La touche **USE** affiche les arguments nécessaires à la commande en cours. Les touches **FETCH** et **QUIT** mettent fin au catalogue, rétablissant le fonctionnement normal du catalogue.

Essayez les touches **NEXT** et **PREV** pour passer d'une commande à l'autre. Vous pouvez maintenir une de ces touches enfoncée pour faire défiler le catalogue.

Il est également possible de passer à la première commande du catalogue commençant par une lettre donnée, par exemple « T ».

T

```
TAN
NEXT PREV      USE FETCH QUIT
```

La première commande est TAN. Si vous appuyez sur un des symboles (autre qu'une lettre) du clavier de gauche, la catalogue passe à la première commande commençant par ce symbole. Essayez « Σ ».

Σ

```
 $\Sigma$ +
NEXT PREV      USE FETCH QUIT
```

La première commande commençant par un *sigma* est Σ + . Si vous appuyez sur une touche du clavier de gauche et qu'aucune commande ne commence par ce symbole, le catalogue passe automatiquement à +, la première commande non-alphabétique.

#

```
+
NEXT PREV      USE FETCH QUIT
```

Vérifiez la syntaxe de +.

USE

```
USAGE: +
2: Real Number
1: Real Number
NEXT PREV      QUIT
```

Ceci vous montre que vous pouvez ajouter deux nombres entiers. Affichez la combinaison suivante.

NEXT

```
USAGE: +
2: Real Number
1: Complex Number
NEXT PREV      QUIT
```

Ceci montre qu'il est possible d'ajouter des nombres réels et complexes. Passez à la combinaison suivante.

NEXT

```
USAGE: +
2: Complex Number
1: Real Number
NEXT PREV      QUIT
```

Ceci indique que les nombres réels et complexes peuvent être placés dans n'importe quel ordre.

Consultez les 14 combinaisons restantes. Voici la dernière :

```
USAGE: +
2: Any Object
1: List
NEXT PREV [ ] [ ] [ ] QUIT
```

Lorsque vous avez passé en revue les combinaisons, revenez au catalogue principal.

QUIT

```
+
NEXT PREV [ ] [ ] [ ] USE FETCH QUIT
```

Vous pouvez maintenant passer à une autre commande du catalogue et revenir au fonctionnement normal du calculateur.

QUIT

```
3:
2:
1: 19
SIN ASIN COS ACOS TAN ATAN
```

Vous pouvez aussi sortir du catalogue en appuyant sur **FETCH**, ce qui écrit dans la ligne de commande le nom de la commande en cours.

2

Arithmétique

Il y a deux manières de faire des opérations arithmétiques sur le HP-28S : avec la pile, comme au chapitre précédent, ou bien en saisissant une *expression* qui représente le calcul. Dans le chapitre précédent, vous avez calculé :

$$(15 + 23) \times \sin 30^\circ$$

Voici comment vous pourriez effectuer ce même calcul en utilisant une expression :

Effacez la pile et passez au menu TRIG.

 CLEAR  TRIG







Commencez l'expression.





Le curseur change de forme, indiquant le passage en *mode algébrique*. Vous verrez les effets de ce mode lorsque vous saisirez l'expression.

Frappez la première partie de l'expression.

 15  +  23 



En mode de saisie algébrique, la touche $\boxed{+}$ écrit le caractère + en ligne de commande au lieu d'exécuter l'opération.

Continuez l'expression.

$\boxed{\times}$ $\boxed{\text{SIN}}$



2:
1:
'(15+23)*SIN(
SIN ASIN COS ACOS TAN ATAN

En mode de saisie algébrique, la touche $\boxed{\times}$ écrit le signe * en ligne de commande, et la touche $\boxed{\text{SIN}}$ y écrit SIN(.

Complétez cette expression et placez-la dans la pile opérationnelle.

30 $\boxed{\text{ENTER}}$



3:
2:
1: '(15+23)*SIN(30)'
SIN ASIN COS ACOS TAN ATAN

Le calculateur ajoute la parenthèse droite,), et le dernier délimiteur '.

Évaluez à présent cette expression :

$\boxed{\text{EVAL}}$



3:
2:
1: 19
SIN ASIN COS ACOS TAN ATAN

L'expression est retirée de la pile et le résultat, 19, est renvoyé au niveau 1.

Ceci termine le calcul de :

$$(15 + 23) \times \sin 30^\circ = 19.$$

Pour effectuer un calcul qui est déjà rédigé sous forme d'expression, tel qu'il apparaîtrait dans un livre, il est plus facile de saisir l'expression et de l'évaluer. Par contre, pour observer les résultats intermédiaires du calcul ou pour effectuer un calcul pas à pas, il est plus facile d'utiliser la pile opérationnelle. Les résultats seront identiques.

La relation entre les calculs effectués dans la pile et les expressions est mise en évidence au chapitre 4, « Calculs répétitifs ». Dans ce chapitre, vous effectuerez des calculs dans la pile, en utilisant des noms au lieu de nombres, de façon à créer une expression.

Saisie et affichage de nombres

Certains modes modifient la manière dont les nombres sont affichés. Voici comment. Placez le nombre $\frac{2}{3}$ dans la pile.

Placez le chiffre 2 dans la pile.

2

3:		
2:		19
1:		2
SIN ASIN COS ACOS TAN ATAN		

Divisez par 3.

3

3:		
2:		19
1:	.66666666666667	7
SIN ASIN COS ACOS TAN ATAN		

Le résultat renvoyé au niveau 1, est une approximation décimale de $\frac{2}{3}$, affichée conformément au format numérique en cours (séparateur décimal et mode d'affichage des nombres). La section suivante décrit les différents formats disponibles.

Choix du séparateur décimal

Les Etats-Unis utilisent le point comme séparateur décimal. La plupart des autres pays du monde utilisent la *virgule* comme séparateur décimal.

Il est possible de choisir le virgule comme séparateur décimal de la façon suivante :

Passez au menu MODE.

3:		
2:		19
1:	.66666666666667	7
STO → / FIX SCI ENG DEG → RAD		

La première ligne de libellés du menu MODE apparaît. Affichez la deuxième ligne de ce menu.

NEXT

```

3:
2:
1: .666666666667
CMD= UNDO= LAST= ML= RDX= PRMO

```

Choisissez la virgule comme séparateur décimal.

RDX,

```

3:
2:
1: ,666666666667
CMD= UNDO= LAST= ML= RDX= PRMO

```

Les séparateurs sont maintenant des virgules au lieu de points, et un petit carré apparaît dans le libellé **RDX** pour indiquer que le mode « RDX, » est activé.

Choix du format d'affichage des nombres

Vous pouvez choisir le nombre de décimales affichées.

Revenez à la première ligne de libellés du menu MODE.

NEXT

```

3:
2:
1: ,666666666667
STD= FIX= SCI= ENG= DEG= RAD

```

Vous pouvez passer de la dernière ligne de libellés d'un menu à la première en appuyant sur **NEXT**. Le menu MODE ne comporte que deux lignes ; la touche **NEXT** vous fera par conséquent passer de l'une à l'autre.

Le menu vous présente quatre choix : STD (standard), FIX (fixe), SCI (scientifique) et ENG (*engineering*, ingénieur). Le libellé STD comporte un petit carré, indiquant que c'est le choix en cours. Dans ce mode, le nombre de décimales dépend du type de valeur affiché.

Pour un nombre entier, aucune décimale n'est affichée ; dans le cas de l'exemple ci-dessus, le nombre maximal de 12 décimales est affiché.

Les autres formats d'affichage affichent un nombre de décimales donné—de 0 à 11—sans égard au type de valeur affichée. Dans la suite de ce manuel, nous représenterons les autres formats d'affichage avec deux décimales. Seul l'affichage des nombres est arrondi ; de manière interne, les nombres sont inchangés.

Affichez $\frac{2}{3}$, arrondi à deux décimales.

2 **FIX**

3:	
2:	19,00
1:	0,67
STD FIX SCI ENG DEG RAD	

Affichez $\frac{2}{3}$ sous forme de *mantisse* et d'*exposant*, la mantisse étant arrondie à deux décimales.

2 **SCI**

3:	
2:	1,90E1
1:	6,67E-1
STD FIX SCI ENG DEG RAD	

La valeur du nombre est le produit de la mantisse et de 10 élevé à la puissance de l'exposant. La mantisse est toujours comprise entre 1 et 9 9999999999.

Affichez $\frac{2}{3}$ sous forme de mantisse et d'exposant, la mantisse étant arrondie à deux décimales et l'exposant étant un multiple de 3.

2 **ENG**

3:	
2:	19,0E0
1:	667,E-3
STD FIX SCI ENG DEG RAD	

Revenez au mode d'affichage standard.

STD

3:	
2:	19
1:	,666666666667
STD FIX SCI ENG DEG RAD	

Saisie de nombres


Vous pouvez saisir les nombres sous forme de mantisse et d'exposant, la valeur du nombre étant constituée du produit de la mantisse et de 10 élevé à la puissance de l'exposant. La mantisse, ou l'exposant, ou les deux, peuvent être négatifs.

Saisissez par exemple le nombre $-4,2 \times 10^{-12}$.

Commencez par les chiffres de la mantisse.

4,2


2:	19
1:	,6666666666667
4,2	
STD FIX SCI ENG DEG RAD	

Si vous faites une erreur, appuyez sur  pour l'effacer et saisissez les chiffres corrects.

Rendez ensuite la mantisse négative.





2:	19
1:	,6666666666667
-4,2	
STD FIX SCI ENG DEG RAD	

« CHS » signifie *changer de signe*—si vous appuyez à nouveau sur , la mantisse sera à nouveau positive.

Passez à l'exposant.



2:	19
1:	,6666666666667
-4,2E	
STD FIX SCI ENG DEG RAD	

« EEX » signifie *entrer l'exposant*. Le E en ligne de commande marque l'exposant de 10. Si vous appuyez sur  par erreur, vous pouvez effacer le E en appuyant sur , comme vous le feriez pour effacer un chiffre incorrect.

Frappez les chiffres de l'exposant.

12

2:	19
1:	,6666666666667
-4,2E12	
STD FIX SCI ENG DEG RAD	

Rendez l'exposant négatif.

[CHS]

```
2: 19
1: ,6666666666667
-4,2E-12
STD FIN SCI ENG DEG RAD
```

Placez le nombre dans la pile.

[ENTER]

```
3: 19
2: ,6666666666667
1: -4,2E-12
STD FIN SCI ENG DEG RAD
```

N'oubliez pas d'utiliser **[CHS]** pour frapper les nombres négatifs. Si ce manuel demande de frapper -4 **[x]**, vous devrez appuyer sur **[4]** **[CHS]** **[x]**.

Fonctions monadiques

Les fonctions qui s'exercent sur un seul nombre—par exemple créer l'opposé d'un nombre ou en extraire la racine carrée—sont nommées fonctions *monadiques*. Elles agissent toutes sur le nombre placé au niveau 1. Il y en a quatre sur le clavier :

- **[CHS]** donne l'opposé du nombre.
- **[1/x]** donne l'inverse du nombre.
- **[√]** donne la racine carrée du nombre.
- **[x²]** donne le carré du nombre.

Si vous frappez un nombre, il n'est pas nécessaire d'appuyer sur **[ENTER]** avant d'exécuter une fonction monadique. La pression sur la touche de fonction exécutera automatiquement **ENTER** à votre place. Vous calculez par exemple $\frac{1}{8}$ de la façon suivante :

8 **[1/x]**

```
3: ,6666666666667
2: -4,2E-12
1: ,125
STD FIN SCI ENG DEG RAD
```

Fonctions diadiques

Les fonctions qui agissent sur deux nombres—telles qu'une addition—sont nommées fonctions *diadiques*. Elles agissent toutes sur les nombres des niveaux 1 et 2.

Lorsque vous divisez 2 par 3, vous placez les deux nombres dans la pile avant d'appuyer sur la touche $\boxed{\div}$. En pratique il n'est pas nécessaire d'appuyer sur $\boxed{\text{ENTER}}$ pour placer le second nombre dans la pile, parce que la fonction le fait pour vous.

Addition et soustraction

Calculez $36 + 17$.

36 $\boxed{\text{ENTER}}$
17 $\boxed{+}$

3:	-4,2E-12
2:	,125
1:	53
STD= FIX SCI ENG DEG= RAD	

Le résultat est 53.

Dans le cas de l'addition, l'ordre des nombres n'a pas d'importance. Il est par contre essentiel dans celui de la soustraction. Calculez maintenant $91 - 27$.

91 $\boxed{\text{ENTER}}$
27 $\boxed{-}$

3:	,125
2:	53
1:	64
STD= FIX SCI ENG DEG= RAD	

Le résultat est 64.

Multiplication et division

Calculez 13×6 .

13 $\boxed{\text{ENTER}}$
6 $\boxed{\times}$

3:	53
2:	64
1:	78
STD= FIX SCI ENG DEG= RAD	

Le résultat est 78.

Dans le cas de la multiplication, l'ordre des nombres n'a pas d'importance ; ce n'est pas le cas pour la division. Calculez maintenant $182/14$.

182
14

3:	64
2:	78
1:	13
STD FIX SCI ENG DEG RAD	

Le résultat 13.

Puissances et racines

L'ordre des nombres est important pour l'élevation à une puissance et pour les calculs de racine. Calculez 5^3 .

5
3

3:	78
2:	13
1:	125
STD FIX SCI ENG DEG RAD	

Le résultat : 125.

Pour calculer $\sqrt[4]{2401}$, placez d'abord 2 401 sur la pile.

2401

3:	13
2:	125
1:	2401
STD FIX SCI ENG DEG RAD	

Elevez maintenant 2 401 à la puissance $\frac{1}{4}$.

4

3:	13
2:	125
1:	7
STD FIX SCI ENG DEG RAD	

Le résultat est 7.

Calcul de pourcentage

Calculez 40 % de 85.

85

40

3:	125
2:	7
1:	34
STD= FIX SCI ENG DEG= RAD	

Le résultat est 34.

Pour le calcul de pourcentage l'ordre des nombres n'est pas important. Il l'est cependant pour la fonction « différence en pourcentage ».

Calculons le pourcentage de changement de 60 à 75.

60

75

3:	7
2:	34
1:	25
STD= FIX SCI ENG DEG= RAD	

Le résultat est +25, ce qui signifie que 75 est 25 % plus élevé que 60.

Echange des niveaux 1 et 2

Pour toutes les fonctions où l'ordre des nombres est important, la soustraction, la division, l'élévation à une puissance, l'extraction de racine et différence en pourcentage—la touche permet d'échanger les contenus des niveaux 1 et 2. Si le nombre 25 est présent au niveau 1 de la pile et si vous désiriez calculer $30 - 25$.

Frappez 30.

30

2:	34
1:	25
30	
STD= FIX SCI ENG DEG= RAD	

Echangez les valeurs.

3:	34
2:	30
1:	25
STD= FIX SCI ENG DEG= RAD	

Remarquez que a effectué un ENTER à votre place.

Soustrayez 25 de 30.



3:	7
2:	34
1:	5
STO= FIN= SCI= ENG= DEG= RAD=	

Ce qui donne 5.

Effacement d'objets dans la pile

Au fil des exemples des pages précédentes, nous avons accumulé un certain nombre de valeurs dans la pile opérationnelle, qui grandit au fur et à mesure de la saisie de nouveaux objets. Les objets saisis restent dans la pile jusqu'à ce que vous les utilisiez dans une opération ou que vous les effaciez.

Vous pouvez les effacer un par un ou tous ensemble.

Effacez le nombre du niveau 1.

DROP

3:	125
2:	7
1:	34
STO= FIN= SCI= ENG= DEG= RAD=	

Les objets qui se trouvaient aux niveaux supérieurs descendent chacun d'un niveau.

Effacez maintenant tous les objets de la pile.

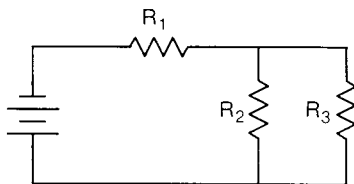
CLEAR

3:	
2:	
1:	
STO= FIN= SCI= ENG= DEG= RAD=	

Prenez l'habitude d'effacer la pile lorsque vous entamez un nouveau problème. Lorsque vous y travaillerez, vous aurez ainsi la certitude que tous les objets de la pile sont pertinents et ne sont pas des restes des opérations précédentes.

Calculs en chaîne

Lorsque vous exécutez des calculs complexes, la pile se comporte comme une zone de stockage temporaire pour conserver les résultats intermédiaires. Ce stockage se produit automatiquement ; supposons par exemple que vous désiriez calculer la résistance totale du circuit électrique suivant :



La formule pour la résistance totale de ce circuit est la suivante :

$$R_{\text{totale}} = R_1 + \frac{1}{\frac{1}{R_2} + \frac{1}{R_3}}$$

Si R_1 , R_2 et R_3 ont respectivement comme résistance 8, 6 et 3 ohms, calculez :

$$R_{\text{totale}} = 8 + \frac{1}{\frac{1}{6} + \frac{1}{3}}$$

Calculez comme suit :

Placez 8 dans la pile opérationnelle.

8



Vous laisserez 8 dans la pile jusqu'à ce qu'il soit temps de l'ajouter au reste du calcul.

Placez $\frac{1}{6}$ dans la pile.

6  

```
3:
2:
1:      ,1666666666667
STD=  FIX  SCI  ENG  DEG=  RAD
```

Puis, placez $\frac{1}{3}$ dans la pile.

3  

```
3:
2:      ,1666666666667
1:      ,3333333333333
STD=  FIX  SCI  ENG  DEG=  RAD
```

Ajoutez les inverses de 6 et de 3.



```
3:
2:
1:      ,500
STD=  FIX  SCI  ENG  DEG=  RAD
```

Obtenez l'inverse de la somme.

```
3:
2:
1:
STD=  FIX  SCI  ENG  DEG=  RAD
```

Achevez le calcul de la R_{totale} .



```
3:
2:
1:      10
STD=  FIX  SCI  ENG  DEG=  RAD
```

Le résultat est 10 ohms.

Si vous exécutez une fonction par erreur

Le HP-28S comporte des sauvegardes qui vous aident à faire marche arrière lorsque une fonction a été exécutée par erreur. Les étapes suivantes renversent les effets d'une fonction, qu'elle soit monadique ou diadique.

1. Appuyez sur ■ **UNDO** pour récupérer le contenu précédent de la pile.
2. Si un nombre se trouvait en ligne de commande au moment où l'erreur s'est produite, appuyez sur ■ **COMMAND** pour récupérer le contenu précédent de la ligne de commande.
3. Continuez votre calcul.

Utilisation des variables

Vous pouvez stocker un nombre (ou tout autre objet) en créant une variable. Une variable est la combinaison d'un nom et d'un autre objet. Le nom identifie la variable, et l'autre objet constitue son contenu. Vous pouvez alors utiliser le nom de la variable pour faire référence à son contenu.

Les variables sont stockées en *mémoire utilisateur*, qui est une partie de la mémoire du calculateur distincte de la pile opérationnelle. Cette dernière est destinée au stockage temporaire d'objets avec lesquels vous êtes en train de travailler, par exemple des nombres que vous utilisez dans un calcul. La mémoire utilisateur, elle, est destinée au stockage à long terme des variables, par exemple des nombres que vous utilisez fréquemment ou répétitivement.

Ce chapitre présente l'utilisation des variables numériques, mais les variables sont utiles pour toutes sortes d'objets. Par exemple, un programme n'a pas de nom propre ; vous pouvez le nommer en le stockant dans une variable et vous pouvez ensuite l'exécuter en mentionnant uniquement le nom de la variable.

Les étapes de la création, du rappel, de l'évaluation, de la modification, du changement de nom ou de la suppression sont identiques pour *toutes* les variables, quel que soit leur contenu. Cette uniformité rend le HP-28S à la fois facile à utiliser et très puissant.

Introduction aux variables

Les variables les plus simples sont les variables numériques. Cette section vous montre comment créer, rappeler et évaluer une variable numérique.

Création d'une variable numérique

Supposons que, dans vos calculs, vous utilisiez fréquemment la valeur 133, représentant un volume. Créez une variable nommée « VOL » (*volume*), comme ceci :

Effacez la pile et choisissez le menu USER.

3:	
2:	
1:	

Le menu USER montre vos variables. Il est vide parce que vous n'avez pas encore créé de variables.

Placez le nombre au niveau 1.


133 

3:	
2:	
1:	133

Placez le nom 'VOL' au niveau 1.

 VOL 

3:	
2:	133
1:	'VOL'

Notez que le deuxième délimiteur, ' , a été ajouté pour vous. Le nombre 133 passe au niveau 2. Dans la pratique il n'est pas nécessaire d'appuyer sur  , mais nous l'avons inclus ici pour clarifier l'explication.

Créez maintenant la variable VOL.



3:	
2:	
1:	
	VOL

Le nombre et le nom sont retirés de la pile pour créer la variable VOL contenant la valeur de 133 ; la variable figure maintenant dans le menu USER.

Rappel d'une variable numérique

Une fois la variable VOL créée, renvoyez sa valeur dans la pile.
Placez le nom VOL dans la pile, en utilisant le menu USER.

 **VOL** **ENTER**

3:					
2:					
1:					'VOL'
VOL					

Rappelez le contenu de VOL.

 **RCL**

3:					
2:					
1:					133
VOL					

Voici le nombre que vous avez stocké dans VOL.

Si vous avez l'habitude d'utiliser un calculateur qui comporte des registres de stockage, le rappel de valeurs est pour vous un processus familier. Sur le HP-28S, on ne rappelle pas fréquemment les variables, le plus souvent on les *évalue*.

Evaluation d'une variable numérique

Pour les variables numériques, « évaluer » équivaut à « rappeler »—évaluer une variable numérique renvoie dans la pile le nombre stocké. Vous verrez que l'évaluation est plus simple que le rappel. Plus loin dans ce chapitre, vous créerez une variable de programme et vous verrez qu'évaluation et rappel peuvent avoir des effets très différents.

Renvoyez la valeur de VOL à la pile par évaluation.

 **VOL**

3:					
2:					133
1:					133
VOL					

Vous pouvez aussi évaluer VOL en frappant son nom *sans guillemets*.

VOL **ENTER**

3:					133
2:					133
1:					133
VOL					

Changement de la valeur d'une variable

La procédure de changement de la valeur d'une variable est identique à celle de création. La nouvelle valeur remplace simplement l'ancienne.

Changez maintenant la valeur de la variable VOL de 133 à 151.

Ecrivez la nouvelle valeur en ligne de commande.

151

2:		133
1:		133
151		
VOL		

Notez que le curseur apparaît désormais sous forme de « boîte vide ». Il changera à l'étape suivante.

Ecrivez le nom de la variable en ligne de commande.

☐ VOL

2:		133
1:		133
151	VOL	
VOL		

Le curseur a changé lorsque vous avez appuyé sur ☐ pour indiquer le nouveau mode de *saisie*.

Au début, la ligne de commande était en *mode d'exécution immédiate*, qui convient aux calculs effectués au clavier. Lorsque vous avez appuyé sur ☐, qui indique le début d'un nom ou d'une expression, la ligne de commande est passée en *mode de saisie algébrique*, qui convient à la saisie de noms et d'expressions :

- La pression sur une touche de fonction telle que ☐ provoque l'apparition du caractère + plutôt que l'exécution de la commande.
- La pression sur une touche du menu USER écrit le nom de la variable et ne provoque pas l'évaluation de cette variable.

Stockez maintenant cette nouvelle valeur dans la variable.

☐ STO

3:		133
2:		133
1:		133
VOL		

Vérifiez la nouvelle valeur.

☐ VOL

3:	133
2:	133
1:	151
VOL	

Suppression d'une variable : ☐ PURGE

Lorsque vous avez fini d'utiliser la variable VOL, supprimez-la de la mémoire utilisateur.

Ecrivez le nom de la variable en ligne de commande.

☐ VOL

2:	133
1:	151
VOL	
VOL	

Le guillemet ☐ est nécessaire pour éviter l'évaluation de la variable.

Purgez la variable VOL de la mémoire utilisateur.

☐ PURGE

3:	133
2:	133
1:	151

Le libellé de VOL disparaît du menu USER.

Changement du nom d'une variable

Il est possible de changer le nom d'une variable en créant une *nouvelle* variable de même valeur et en purgeant la variable d'origine.

Dans cette section vous passerez d'abord en revue les étapes nécessaires pour renommer une variable, puis vous écrirez un programme qui contient ces mêmes étapes, pour finalement stocker le programme dans une variable qui permet de l'exécuter en utilisant son nom.

Pour préparer le travail, créez une variable—pour pouvoir la renommer ensuite. Créez A, de valeur 10.

Placez 10 dans la pile.

10 **ENTER**

3:		133
2:		151
1:		10

Créez la variable A.

A **STO**

3:		133
2:		133
1:		151
	A	

Notez que A apparaît dans le menu USER.

Supposons que vous désiriez renommer A en B. Placez l'ancien nom dans la pile.

A **ENTER**

3:		133
2:		151
1:		'A'
	A	

Placez le nouveau nom dans la pile.

B **ENTER**

3:		151
2:		'A'
1:		'B'
	A	

Ceci achève la préparation ; la variable existe, l'ancien nom est sur la pile, le nouveau nom aussi. *Ancien* et *nouveau* noms sont les *arguments* du programme, et le programme supposera qu'ils se trouvent dans la pile dans cet ordre. Les étapes qui suivent se trouvent dans le programme.

Ces étapes utilisent trois commandes couramment utilisées pour manipuler la pile : OVER, ROT et SWAP. Vous verrez leur fonctionnement au moment de l'exécution.

Copiez l'ancien nom au niveau 1. Utilisez pour cela la commande OVER du menu STACK.

STACK **OVER**

3:		'A'
2:		'B'
1:		'A'
	OVER OVER DROPE ROT LIST	

Rappelez le contenu de la variable.

RCL

3:	'A'
2:	'B'
1:	10
[DUP] [OVER] [DUP2] [DROPE] [ROT] [LIST]	

Placez le nouveau nom au niveau 1, en utilisant la commande ROT (*rotation*).

ROT

3:	'B'
2:	10
1:	'A'
[DUP] [OVER] [DUP2] [DROPE] [ROT] [LIST]	

Purgez l'ancienne variable (faites-le avant de créer la nouvelle variable, vous éviterez de faire une copie supplémentaire de la valeur).

PURGE

3:	151
2:	'B'
1:	10
[DUP] [OVER] [DUP2] [DROPE] [ROT] [LIST]	

Placez le contenu dans le nouveau nom et dans le bon ordre.

SWAP

3:	151
2:	10
1:	'B'
[DUP] [OVER] [DUP2] [DROPE] [ROT] [LIST]	

Créez la nouvelle variable.

STO

3:	133
2:	133
1:	151
[DUP] [OVER] [DUP2] [DROPE] [ROT] [LIST]	

Vous pouvez maintenant créer un programme qui automatise ces étapes.


Créer une variable de programme

Frappez d'abord le programme, puis stockez-le dans une variable.

Commencez le programme par le délimiteur de programme :


«

2:	133
1:	151
«	
<div> <div>DUF</div> <div>OVER</div> <div>DUF2</div> <div>DROP2</div> <div>ROT</div> <div>LIST</div> </div>	

Notez que le curseur a changé de forme, et que le témoin ***α*** apparaît, deux indications que nous sommes maintenant en *mode alpha*. La pression sur une touche correspondant à une opération programmable quelconque écrit le nom de cette opération en ligne de commande. Seules les opérations non-programmables, telles que l'appui sur  pour effacer un caractère, sont exécutées.


Frappez maintenant les étapes que nous avons exécutées précédemment.

OVER




RCL

ROT



PURGE



SWAP

STO

ENTER

2:	151
1:	« OVER RCL ROT PURGE SWAP STO »
<div> <div>DUF</div> <div>OVER</div> <div>DUF2</div> <div>DROP2</div> <div>ROT</div> <div>LIST</div> </div>	

Ici aussi, le délimiteur de clôture, » , a été ajouté pour vous.

Stockez le programme dans la variable RENOMMER.

' RENOMMER

STO

3:	133
2:	133
1:	151
<div> <div>DUF</div> <div>OVER</div> <div>DUF2</div> <div>DROP2</div> <div>ROT</div> <div>LIST</div> </div>	

Vérifiez le menu USER.

USER

3:	133
2:	133
1:	151
<div> <div>RENO</div> <div>E</div> <div></div> <div></div> <div></div> <div></div> </div>	

RENOMmer, sous forme abrégée, apparaît dans le menu USER.

Vous êtes prêt à exécuter RENOMMER. Vous le ferez d'abord d'une manière détournée, en utilisant RCL, puis de façon normale, en utilisant le menu USER. La différence entre ces méthodes accentue la distinction entre le rappel et l'évaluation d'une variable de programme.

Rappel d'une variable de programme

Pour cet exemple, renommez la variable B en C.

Placez l'ancien et le nouveau nom sur la pile.

☐ **B**
☐ C

3:		151
2:		'B'
1:		'C'
RENO B		

Rappelez le programme RENOMMER.

☐ **RENO**

2:		'C'
1:	« OVER RCL ROT PURGE SWAP STO »	
RENO B		

Pour n'importe quelle variable, RCL renvoie simplement le contenu de la variable dans la pile opérationnelle.

Evaluation d'une variable de programme

Pour exécuter un programme dans la pile, il est nécessaire de l'évaluer explicitement.

3:		133
2:		133
1:		151
C RENO		

Le menu USER montre que B a été renommée en C.

Il n'était pas nécessaire de rappeler le programme dans la pile pour l'exécuter, mais cela nous a permis de montrer comment RCL fonctionne et comment EVAL provoque l'exécution de programmes. Vous verrez ensuite comment exécuter plus facilement un programme.

Cette fois vous renommerez C en D. Placez l'ancien nom et le nouveau dans la pile.

☐ **C**
☐ D

3:		151
2:		'C'
1:		'D'
C RENO		

Renommez C en D.

RENO

3:	133
2:	133
1:	151
D	RENO

Le menu USER montre que C a été renommé et est devenu D. Vous avez exécuté le programme en appuyant simplement sur une touche du menu USER.

Noms placés entre guillemets ou non

Dans les exemples qui précèdent, vous avez utilisé des noms de variables des deux manières. Les guillemets `"` sont importants : ils distinguent le *nom* d'une variable de son *contenu*. Voici un résumé des raisons pour lesquelles certains noms sont placés entre guillemets et d'autres non.

- Les noms entre guillemets représentent le nom lui-même. Les guillemets empêchent le nom d'être évalué ; le nom est par conséquent envoyé vers la pile et peut devenir l'argument d'une commande. Dans ce chapitre vous avez utilisé des noms entre guillemets comme arguments pour STO, RCL, PURGE et pour le programme RENOMMER.
- Les noms sans guillemets sont utilisés pour évaluer la variable qui porte ce nom. Les noms sans guillemet ne passent pas dans la pile opérationnelle : l'objet stocké dans la variable est traité selon son type. Les variables numériques sont renvoyées à la pile et les programmes sont exécutés. Vous verrez ce qui se passe pour les autres types de variables plus loin dans ce manuel.

Si vous frappez un nom sans guillemet qui n'est pas associé à une variable, le texte entre guillemets est envoyé dans la pile opérationnelle.

Calculs répétitifs

Dans ce chapitre vous créez une expression contenant des variables numériques, puis vous utiliserez une caractéristique majeure de ce calculateur : l'algorithme de résolution d'équations Solver.

Au chapitre 2 vous avez effectué un calcul en frappant une expression qui contenait des nombres et en l'évaluant. Dans ce chapitre vous créez une expression en calculant dans la pile, et en utilisant des noms comme arguments symboliques. Vous utiliserez l'algorithme de résolution d'équations pour assigner des valeurs aux variables et pour évaluer l'expression. Chaque fois que vous évaluez l'expression, le calcul est effectué avec les valeurs en cours des variables. Si vous modifiez la valeur de l'une des variables, vous pouvez simplement ré-évaluer l'expression.

Au chapitre 3 vous avez créé des variables numériques et une variable de programme. Dans ce chapitre vous créez des variables d'expressions et des variables de nom (souvenez-vous, tout objet peut être stocké dans une variable). Vous étudierez en outre les répertoires, qui sont des ensembles de variables.

Création d'une expression


Nous allons répéter le calcul des résistances effectué au chapitre 2, mais cette fois nous utiliserons des noms comme arguments, plutôt que des nombres.

$$R_{\text{totale}} = R_1 + \frac{1}{\frac{1}{R_2} + \frac{1}{R_3}}$$

Effacez la pile et sélectionnez le menu du curseur.

■ CLEAR

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

Si un menu est affiché, appuyez sur  pour sélectionner le menu du curseur.

Placez le nom 'R1' dans la pile opérationnelle.

▢ R1 ENTER

```
4:
3:
2:
1: 'R1'
```

Une fois encore, le délimiteur de « fermeture », ' , est ajouté pour vous. Vous laisserez R1 dans la pile jusqu'à ce qu'il soit temps de l'ajouter au reste du calcul.

Placez l'inverse de R2 dans la pile.

▢ R2 ENTER ■ 1/x

```
4:
3:
2: 'R1'
1: 'INV(R2)'
```

Placez l'inverse de R3 dans la pile.

▢ R3 ENTER ■ 1/x

```
4:
3: 'R1'
2: 'INV(R2) '
1: 'INV(R3) '
```

Ajoutez les inverses de R2 et de R3.

▢ +

```
4:
3: 'R1'
2: 'INV(R2)+INV(R3) '
1:
```

Déterminez l'inverse de la somme.

■ 1/x

```
3: 'R1'
2: 'INV(INV(R2)+INV(R3) '
1: ' )'
```

Ajoutez R1 et son inverse.

+

```
3:
2:
1: 'R1+INV(INV(R2)+INV(
  R3))'
```

L'expression qui résulte représente la R_{totale} .

Vous auriez pu frapper cette expression directement en prenant soin d'ajouter toutes les parenthèses. Chaque expression étant équivalente à un calcul effectué dans la pile, vous pouvez choisir la méthode qui vous convient le mieux.

Création d'un répertoire

Un répertoire est un ensemble de variables. Pour l'instant vous travaillez dans le répertoire HOME qui existe même après une ré-initialisation du calculateur. Vous allez créer un autre répertoire, dans HOME, qui contiendra lui-même des sous-repertoires.

Voici quelques-uns des concepts que vous utiliserez.

- Un seul répertoire peut être le *répertoire en cours* ; seules ses variables apparaissent dans le menu USER.
- Si un répertoire A contient un autre répertoire B, A est nommé *répertoire-parent* du répertoire B, et B est un *sous-répertoire* de A.
- Si vous commencez au niveau du répertoire en cours et que vous trouvez le répertoire-parent, puis le répertoire-parent suivant et ainsi de suite, vous arriverez toujours au répertoire HOME. Cette succession des répertoires se nomme le *chemin du répertoire en cours*.

Vous pouvez vérifier le chemin en cours en exécutant la commande PATH.

Sélectionnez le menu MEMORY.

MEMORY

```
2:
1: 'R1+INV(INV(R2)+INV(
  R3))'
MEM MENU ORDER PATH HOME EDIT
```

Vérifiez le chemin de répertoire en cours.

PATH

```
3:
2: 'R1+INV(INV(R2))+INV...
1: { HOME }
MEM MENU ORDER PATH HOME CROIR
```

La liste que renvoie la commande PATH commence toujours par HOME et se termine par le répertoire en cours. HOME est le point de départ de tous les chemins et, puisque vous n'avez pas encore créé d'autre répertoire, il est aussi le répertoire en cours.

Pour regrouper vos problèmes de génie électrique, créez un sous-répertoire nommé GE.

☐ GE **CROIR**

```
3:
2: 'R1+INV(INV(R2))+INV...
1: { HOME }
MEM MENU ORDER PATH HOME CROIR
```

Passez à ce nouveau répertoire.

GE

```
3:
2: 'R1+INV(INV(R2))+INV...
1: { HOME }
MEM MENU ORDER PATH HOME CROIR
```

Vérifiez à nouveau le chemin en cours.

PATH

```
3: 'R1+INV(INV(R2))+INV...
2: { HOME }
1: { HOME GE }
MEM MENU ORDER PATH HOME CROIR
```

Le répertoire en cours est maintenant GE. Affichez le menu USER pour constater l'un des effets du passage au répertoire GE.

```
3: 'R1+INV(INV(R2))+INV...
2: { HOME }
1: { HOME GE }
MEM MENU ORDER PATH HOME CROIR
```

Notez que le programme RENOMMER (que vous avez créé au chapitre dernier) n'apparaît pas. Seules les variables du *répertoire en cours* apparaissent dans le menu USER. RENOMMER se trouve dans le répertoire HOME.

Vous pouvez toujours exécuter RENOMMER en frappant son nom. Vous pouvez accéder à toute variable présente dans le chemin du répertoire en cours (HOME GE, dans notre exemple) en frappant simplement son nom.

Ceci est l'un des avantages des répertoires : si vous placez des programmes d'utilisation courante tels que RENOMMER dans le répertoire HOME, vous pouvez toujours les exécuter sans qu'ils viennent encombrer le menu USER.

Vous pouvez maintenant travailler dans le nouveau répertoire GE. Éliminez les deux chemins de la pile.

DROP **DROP**

```

2:
1: 'R1+INV(INV(R2)+INV(
  R3))'

```

Stockez l'expression dans une variable nommée EQ1 (équation numéro 1). Vous verrez plus tard pourquoi elle est nommée ainsi.

' EQ1 **STO**

```

3:
2:
1:
EQ1

```

La variable EQ1 apparaît dans le menu USER.

Supposons que vous allez utiliser cette expression pour une série de problèmes que vous désirez tester indépendamment les uns des autres. Pour ce faire, vous pouvez placer les valeurs correspondant à chaque problème dans un sous-répertoire consacré à ce problème.

Créez un sous-répertoire SP1 (série-parallèle 1) pour le premier problème.

' SP1 **ENTER**
CRDIR **ENTER**

```

3:
2:
1:
SP1 EQ1

```

Le nom de ce nouveau sous-répertoire apparaît dans le menu USER. Appuyez sur la touche de menu pour passer à SP1.

SP1

```

3:
2:
1:

```

Le menu USER est à nouveau vide, puisque le répertoire en cours, SP1, est vide.

Vérifiez le répertoire en cours.

PATH

3:					
2:					
1:	{	HOME	GE	SP1	}

Vous pouvez trouver n'importe quelle variable des répertoires HOME ou GE en frappant simplement leurs noms, parce que ces répertoires se trouvent sur le chemin du répertoire en cours (HOME GE SP1), mais le menu USER ne montre que les variables du répertoire en cours, SP1.

Vous êtes maintenant prêt à utiliser l'algorithme de résolution d'équations *Solver* avec l'expression EQ1.

Utilisation du Solver pour répéter un calcul

Il y a trois étapes de base pour utiliser l'algorithme de résolution d'équations avec une expression.

1. Stockez l'expression (ou son nom) dans une variable nommée EQ (équation). Le Solver nécessite la présence d'une variable portant ce nom.
2. Utilisez le Solver pour donner des valeurs aux variables.
3. Utilisez le Solver pour évaluer l'expression.

Répétez ensuite les points 2 et 3 pour les autres valeurs.

Voici les étapes requises pour cet exemple.

1ère étape : stocker le nom EQ1 dans une variable EQ.

Ceci pourrait surprendre. Pourquoi stocker un *nom* dans une variable ? Pourquoi ne pas stocker l'*expression* dans EQ ? La raison est que le nom EQ1 est plus court et plus facile à retenir que l'expression complète. Plus tard, il sera aussi plus facile de passer d'une équation à l'autre si leurs noms sont simples et courts.

Placez le nom EQ1 sur la pile.

3:					
2:	{	HOME	GE	SP1	}
1:					'EQ1'

Si vous avez oublié le premier délimiteur, ['], vous avez maintenant l'expression elle-même dans la pile ; dans ce cas appuyez sur **DROP** pour éliminer l'expression et recommencer.
 Passez au menu SOLVE.

SOLV

```
3:
2:      { HOME GE SP1 }
1:      'EQ1'
STEQ REEQ SOLVR ISOL QUAD SHOW
```

Employez STEQ (*Store Equation*) pour stocker le nom EQ1 dans la variable EQ.

STEQ

```
3:
2:
1:      { HOME GE SP1 }
STEQ REEQ SOLVR ISOL QUAD SHOW
```

2ème étape : assigner les valeurs aux variables.

Affichez le menu Solver.

SOLVR

```
3:
2:
1:      { HOME GE SP1 }
R1 R2 R3 EXPR=
```

Les variables de l'équation en cours apparaissent dans le menu Solver (si l'équation contient plus de six variables, la pression sur **NEXT** affiche les lignes de libellés suivantes).

Ce menu a un aspect différent du menu USER parce que son fonctionnement est différent : le menu Solver *stocke des valeurs* dans les variables plutôt que de les évaluer.

Vous pouvez maintenant assigner des valeurs aux variables R1, R2 et R3. Stockez en premier lieu le nombre 8 dans la variable R1.

8 **R1**

```
R1: 8
2:
1:      { HOME GE SP1 }
R1 R2 R3 EXPR=
```

La pression sur **R1** équivaut à placer 'R1' dans la pile et à appuyer sur **STO**. Notez que la première ligne de l'affichage montre le nom de la variable et sa valeur.

Stockez le nombre 6 dans R2.

6

R2: 6				
2:	{ HOME GE SP1 }			
1:				
R1	R2	R3	EXPR=	

Stockez le nombre 3 dans R3.

3

R3: 3				
2:	{ HOME GE SP1 }			
1:				
R1	R2	R3	EXPR=	

3ème étape : évaluez l'expression.

Le libellé de menu signifie « cette expression égale » ; la pression sur cette touche provoque l'évaluation de l'expression.

EXPR=10				
2:	{ HOME GE SP1 }			
1:				10
R1	R2	R3	EXPR=	

La valeur (10) est renvoyée en niveau 1 et apparaît en caractères inversés sur la première ligne de l'affichage.

4ème étape : répétez les étapes 2 et 3 pour les autres valeurs. Que se passe-t-il par exemple si R3 vaut 12 ?

Stockez 12 dans R3.

12

R3: 12				
2:	{ HOME GE SP1 }			
1:				10
R1	R2	R3	EXPR=	

Évaluez l'expression pour les valeurs en cours des variables.

EXPR=12				
2:				10
1:				12
R1	R2	R3	EXPR=	

La nouvelle valeur (12) est renvoyée au niveau 1, et apparaît en caractères inversés sur la première ligne de l'affichage.

Utilisation d'un autre jeu de valeurs

Supposons que vous ayez à résoudre un problème similaire, avec des valeurs R1, R2 et R3 différentes, pour revenir ensuite aux valeurs assignées pour l'instant. Vous pourriez ré-entrer les valeurs chaque fois que vous changez de problème, mais nous allons vous montrer une procédure plus simple. Elle comporte trois étapes :

1. Créer un nouveau répertoire pour les nouvelles valeurs.
2. Définir la même expression pour EQ.
3. Utiliser le Solver pour assigner des valeurs et évaluer l'expression.

Ce processus vous montre un autre avantage des répertoires : *dans un répertoire, une seule variable peut exister sous un nom particulier ; mais il n'y a pas de limites au nombre de répertoires qui peuvent contenir des variables portant le même nom.*

1ère étape : créer un nouveau répertoire.

Puisque le nouveau répertoire est une autre version de SP1, appelez-le SP2 et créez-le sous le même répertoire-parent (GE). Voici le premier « branchement » dans la structure de répertoire—deux sous-répertoires, SP1 et SP2 sous le même répertoire-parent GE.

Pour pouvoir créer un sous-répertoire de GE, il faut que ce dernier soit le répertoire en cours (tout répertoire créé maintenant le serait sous SP1).

Passez au répertoire GE.

GE

No Current Equation	
2:	10
1:	12
STEC REEC SOLVE ISOL QUAD SHOW	

Le calculateur émet une tonalité, affiche No Current Equation et active le menu SOLVE : il n'y a pas d'équation « EQ » dans le répertoire GE.

Créez le répertoire SP2.

☐ SP2

☒ MEMORY

3:	{ HOME GE SP1 }
2:	10
1:	12
MEM MENU ORDER PATH HOME CROIR	

Passez au répertoire SP2.

SP2

```

3:      { HOME GE SP1 }
2:                                     10
1:                                     12
MEM MENU ORDER PATH HOME CROIR

```

Vérifiez le chemin de répertoire en cours.

```

3:                                     10
2:                                     12
1:      { HOME GE SP2 }
MEM MENU ORDER PATH HOME CROIR

```

Notez que HOME et GE se trouvent dans le chemin du répertoire en cours, comme lorsque SP1 était le répertoire en cours, mais SP1 n'apparaît plus. Vous pouvez toujours accéder aisément aux variables de HOME (RENOMMER par exemple) et de GE (EQ1), mais pas à celles de SP1 (EQ, R1, R2 et R3). Vous pouvez maintenant créer de nouvelles variables.

2ème étape : définissez la même expression en tant que EQ.

Comme plus haut, utilisez STEQ pour stocker le nom EQ1 dans la variable EQ.

```

3:                                     10
2:                                     12
1:      { HOME GE SP2 }
STEQ RCEQ SOLVR ISOL QUAD SHOW

```

3ème étape : utilisez le Solver comme plus haut pour assigner des valeurs et évaluer l'expression. Supposons que les valeurs du nouveau problème soient les suivantes :

$$R1 = 11, \quad R2 = 21, \quad R3 = 7$$

Passez au menu Solver.

```

3:                                     10
2:                                     12
1:      { HOME GE SP2 }
R1  R2  R3  EXPR=  [ ]  [ ]

```

Assignez les valeurs.

11
 21
 7

```

3: 7
2:                                     12
1:      { HOME GE SP2 }
R1  R2  R3  EXPR=  [ ]  [ ]

```

Évaluez l'expression.

EXPR=

```
EXPR=16,25
2: { HOME GE SP2 }
1: 16,25
R1 R2 R3 EXPR=
```

Pour revenir au problème précédent, vous devez exécuter GE (pour passer au répertoire GE), SP1 (pour passer au répertoire SP1) et appuyer sur **SOLV** **SOLVR** (pour activer le menu Solver) ; toutes les valeurs des variables seront les mêmes que lorsque vous avez quitté SP1.

Utilisation d'une autre expression

Vous avez maintenant deux ensembles de valeurs à utiliser avec l'expression EQ1 ; essayez à présent de créer une deuxième expression, EQ2, qui soit capable d'utiliser les deux ensembles de valeurs. Deux étapes pour ce faire :

1. Passez au répertoire GE, créez la nouvelle expression et stockez-la dans la variable EQ2.
2. Passez au répertoire SP1 ou SP2, changez la valeur de EQ de « EQ1 » en « EQ2 », puis utilisez le Solver pour évaluer l'expression.

Étape 1 : passez au répertoire GE, créez la nouvelle expression et stockez-la dans une variable EQ2.

Passez au répertoire GE.

GE **ENTER**

```
No Current Equation
2: { HOME GE SP2 }
1: 16,25
STER RCEX SOLVR ISOL QUAD SHOW
```

Créez la nouvelle expression. Dans cet exemple, EQ2 sera une copie corrigée de l'expression EQ1.

Renvoyez l'expression stockée dans EQ1 à la pile opérationnelle.

USER EDIT

```
2: 16,25
1: 'R1+INV(INV(R2))+INV(
  R3))'
```

Renvoyez l'expression en ligne de commande.

EDIT

```
1: 'R1+INV(INV(R2))+INV(
  R3))'
  'R1+INV(INV(R2))+INV(
  R3))'
```

L'expression du niveau 1 apparaît en caractères inverses pour vous prévenir qu'elle sera remplacée par le contenu de la ligne de commande. Le témoin alpha **α** apparaît, indiquant que le mode de saisie alpha est en cours.

Modifiez maintenant l'expression pour refléter la formule :

$$R_{\text{totale}} = R_1 + \frac{1}{\frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_3}}$$

Placez le curseur sur la dernière rangée de la ligne de commande (les mouvements sont indiqués sur le *menu du curseur*—les libellés imprimés en blanc juste au-dessus des touches de menu).

▼

```
1: 'R1+INV(INV(R2))+INV(
  R3))'
  'R1+INV(INV(R2))+INV(
  R3))'
```

Le menu du curseur est actif lorsque la ligne de commande existe et qu'aucun autre menu n'est affiché. Vous pouvez activer ou désactiver le menu du curseur en appuyant sur la touche **↔**. Le fait d'appuyer sur **EDIT** met automatiquement le menu du curseur en action.

Placez le curseur juste après R3.

→ → →

```
1: 'R1+INV(INV(R2))+INV(
  R3))'
  'R1+INV(INV(R2))+INV(
  R3))'
```

Choisissez le mode d'insertion.

INS

```
1: 'R1+INV(INV(R2)+INV(
R3))'
'R1+INV(INV(R2)+INV(
R3)♦'
```

La forme du curseur change ; la flèche indique que le texte sera inséré à gauche du caractère qui se trouve à la position du curseur. Une seconde pression sur **INS** vous remettra en mode de remplacement.

Frappez le deuxième terme de R3.

+ **■** **1/x** **(** R3

```
1: 'R1+INV(INV(R2)+INV(
R3))'
'R1+INV(INV(R2)+INV(
R3) + INV (R3♦'
```

Remplacez l'expression du niveau 1 par l'expression corrigée de la ligne de commande.

ENTER

```
2: 16,25
1: 'R1+INV(INV(R2)+INV(
R3)+INV(R3))'
SP2 SP1 EQ1
```

Stockez la nouvelle expression dans une variable EQ2.

' EQ2 **STO**

```
3: 12
2: { HOME GE SP2 }
1: 16,25
EQ2 SP2 SP1 EQ1
```

Etape 2 : passez au répertoire SP1 ou SP2, changez la définition de EQ (de EQ1 en EQ2) et utilisez le Solver pour évaluer l'expression.

Pour cet exemple, utilisez les valeurs de SP1 avec la nouvelle expression.

Passez au répertoire SP1.

SP1

```
3: 12
2: { HOME GE SP2 }
1: 16,25
R3 R2 R1 EQ
```

Changez la définition de EQ, de EQ1 en EQ2.

☐ EQ2

```
3: 12
2: ( HOME GE SP2 )
1: 16,25
STEQ REEQ SOLVR ISOL QUAD SHOW
```

Evaluez l'expression EQ2 avec les valeurs de SP1.

```
EXPR=11
2: 16,25
1: 11
R1 R2 R3 EXPR=
```

Pour évaluer EQ2 avec les valeurs de SP2, vous devriez exécuter GE (pour passer au répertoire GE), puis répéter l'étape 2 ci-dessus en substituant SP2 à SP1.

Retour au répertoire HOME...

En supposant que vous en ayez fini avec vos calculs d'électricité, vous pouvez revenir au répertoire HOME. Puisque c'est un répertoire intégré dans la machine, son nom se trouve dans le menu MEMORY.

Passez au répertoire HOME.

```
3: ( HOME GE SP2 )
2: 16,25
1: 11
MEM MENU ORDER PATH HOME CROIS
```

Vérifiez le menu USER.

```
3: ( HOME GE SP2 )
2: 16,25
1: 11
GE 0 REND
```

Le libellé de menu est le seul témoin de ce que vous avez créé dans ce chapitre—EQ1, EQ2, les sous-répertoires SP1 et SP2 et toutes les variables qu'ils contiennent. Voici un avantage majeur de la structure en répertoires : *du point de vue du répertoire-parent, un répertoire complet, ses variables et ses propres sous-répertoires, apparaît simplement sous la forme d'un nom.*

Résumé

Voici les grandes lignes de la stratégie que vous avez suivie dans ce chapitre.

- Création d'un répertoire pour chaque ensemble de problèmes.
- Stockage des expressions nécessaires pour chaque problème dans des variables.
- Création d'un sous-répertoire pour les valeurs spécifiques de chaque problème.
- Utilisation du Solver avec diverses combinaisons d'expressions et de valeurs.

5

Fonctions opérant sur les nombres réels

Ce chapitre introduit les menus TRIG, LOGS et REAL. Le menu TRIG contient des fonctions trigonométriques et des commandes qui se rapportent aux mesures d'angle. Le menu LOGS contient des fonctions logarithmiques, exponentielles et hyperboliques. Le menu REAL contient des commandes supplémentaires pour les nombres réels.

Toutes les commandes de ces menus sont brièvement décrites dans l'annexe D, « Les menus et leurs jeux de libellés ». Pour une description complète des commandes, lisez les sections « TRIG », « LOGS » ou « REAL » dans le *Manuel de référence*.

Fonctions trigonométriques

Cette section vous montre comment choisir le mode d'angle adéquat, comment effectuer des calculs avec π et comment convertir les mesures d'angle.

Choix du mode d'angle

Le calculateur est capable d'interpréter des angles en degrés ($1/360$ d'un cercle) ou en radians ($1/2\pi$ d'un cercle). Le choix par défaut est le « mode degrés ». Pour les exemples qui vont suivre, passez en « mode radians ».

Effacez la pile et choisissez le menu MODE.

☒ CLEAR ☐ MODE

3:
2:
1:
STD FTX SCI ENG DEG RAD

Les deux libellés se trouvant à l'extrême droite, **DEC** (*degrés*) et **RAD** (*radians*), représentent vos choix. Remarquez que le libellé **DEC** porte un petit carré, ce qui indique que le mode d'angle en cours est « degrés ».

Choisissez le mode radians.

RAD



Le témoin radians (2π) apparaît et les menus de libellés changent (la plupart des illustrations dans ce manuel n'indiquent pas les témoins d'état. Pour trouver le témoin (2π), voyez l'illustration en page 27).

Affichez la première ligne du menu TRIG.

TRIG



Ces fonctions sont monadiques et agissent sur le nombre du niveau 1. Pour les nombres réels, le mode d'angle affecte la façon dont SIN (*sinus*), COS (*cosinus*) et TAN (*tangente*) interprètent leurs arguments et dont ASIN (*arc sine, sinus hyperbolique inverse*), ACOS (*arc cosine, cosinus hyperbolique inverse*) et ATAN (*arc tangent, tangente hyperbolique inverse*) expriment leurs valeurs.

Vous utiliserez la fonction SIN au cours de la discussion de π qui suit.

Utilisation de π

Le nombre transcendant π ne peut pas être représenté exactement sous forme décimale. En général, l'arrondi à 12 chiffres du calculateur (3,14159265359) produit des résultats exacts à la 12ème décimale, ce qui est suffisant pour la plupart des applications.

Le HP-28S possède aussi une constante symbolique, π , qui représente π exactement. En mode radians, les fonctions SIN, COS et TAN reconnaissent la constante symbolique π et donnent un résultat exact. Les fonctions SIN et COS reconnaissent aussi $\pi/2$.

Pour les autres fonctions, la constante symbolique π produit une expression contenant π . Si vous forcez un résultat sous forme de nombre réel, le calculateur effectue une approximation.

Pour démontrer la différence entre 3,14159265359 et π , calculez le sinus de chacun d'eux.

Placez ' π ' au niveau 1.

■ π ENTER

```
3:
2:
1:                                     'π'
SIN ASIN COS ACOS TAN ATAN
```

Bien que cet objet semble être un nom, c'est en fait une expression comportant un seul terme, la constante symbolique π .

Forcez un résultat sous forme de nombre réel en utilisant \rightarrow NUM (to number)

■ \rightarrow NUM

```
3:
2:
1:                                     3,14159265359
SIN ASIN COS ACOS TAN ATAN
```

L'approximation (12 chiffres) de π (3,14159265359) est renvoyée au niveau 1.

Calculez le sinus de l'approximation de π .

SIN

```
3:
2:
1:                                     -2,06761537357E-13
SIN ASIN COS ACOS TAN ATAN
```

Le résultat ($-2,06761537357 \times 10^{-13}$) ne vaut pas exactement zéro parce que l'argument (3,14159265359) n'est pas exactement égal à π .

Calculez maintenant le sinus de π .

■ π SIN

```
3:
2:
1:                                     -2,06761537357E-13
SIN ASIN COS ACOS TAN ATAN
```

La fonction SIN reconnaît la constante symbolique π et renvoie le résultat correct (0).

Conversion d'heures et d'angles

Le menu TRIG contient des commandes qui vous permettent de convertir des valeurs entre les systèmes d'unités degrés-minutes-secondes et degrés décimaux, ainsi qu'entre les degrés et les radians. Ces commandes se situent dans la troisième ligne du menu TRIG.

Affichez brièvement la deuxième ligne du menu TRIG.

NEXT

3:	
2:	-2,06761537357E-13
1:	0
P→R R→P R→C C→R ARG	

Ces commandes manipulent les nombres complexes et sont reproduites dans le menu COMPLEX. Les nombres complexes sont traités au chapitre suivant.

Affichez la troisième ligne du menu TRIG.

NEXT

3:	
2:	-2,06761537357E-13
1:	0
→HMS HMS→ HMS+ HMS- D→R R→D	

Les quatre commandes HMS (*heures-minutes-secondes*) vous permettent de manipuler des nombres dont la partie fractionnaire représente des minutes et des secondes ; c'est-à-dire des nombres qui suivent le format spécial nommé HMS décrit ci-dessous :

$h.MMSSs$

où h représente des heures (ou degrés), MM représente des minutes, SS représente des secondes et s représente des fractions de seconde. MM et SS représentent chacun deux chiffres ; h et s représentent chacun un nombre quelconque de chiffres.

Les commandes →HMS (*décimal-en-HMS*) et HMS→ (*HMS-en-décimal*) convertissent des valeurs entre les formats décimaux et HMS. Les commandes HMS+ (*HMS plus*) et HMS- (*HMS moins*) ajoutent des nombres au format HMS et donnent le résultat dans ce même format.

Convertissez, par exemple, $141^{\circ} 26' 15''$ en degrés décimaux.

Entrez le nombre en format HMS.

141,2615

```
3: -2,06761537357E-13
2: 0
1: 141,2615
[HMS] [HMS+] [HMS-] [D→R] [R→D]
```

Convertissez le nombre du format HMS en degrés décimaux.

[HMS→]

```
3: -2,06761537357E-13
2: 0
1: 141,4375
[HMS] [HMS+] [HMS-] [D→R] [R→D]
```

Les deux autres fonctions de cette ligne de menu, D→R (*degrés-en-radians*) et R→D (*radians-en-degrés*) permettent de convertir un nombre réel représentant un angle, de degrés en radians, et réciproquement.

Convertissez le nombre du niveau 1 de degrés en radians.

[D→R]

```
3: -2,06761537357E-13
2: 0
1: 2,46855006079
[HMS] [HMS+] [HMS-] [D→R] [R→D]
```

Vous venez donc de calculer :

$$141^{\circ} 26' 15'' = 141,4375^{\circ} = 2,46855006079 \text{ radians}$$

Fonctions logarithmiques, exponentielles et hyperboliques

Le menu LOGS contient des fonctions logarithmiques et exponentielles, népériennes et en base 10, ainsi que des fonctions hyperboliques. La section « LOGS » du Manuel de référence décrit ces fonctions en détails.

Affichez la première ligne du menu LOGS.

■ [LOGS]

```
3: -2,06761537357E-13
2: 0
1: 2,46855006079
[LOG] [ALOG] [LN] [EXP] [LNPI] [EXPM]
```

Les fonctions LOG (*logarithme en base 10*) et ALOG (*antilogarithme en base 10*) calculent les logarithmes et exponentielles en base 10. Les fonctions LN (*logarithme népérien*) et EXP (*exponentielle népérienne*) calculent les logarithmes et exponentielles en base e (e est un nombre transcendant approximativement égal à 2,71828182846).

Pour un argument x , la fonction LNP1 (*ln plus 1*) calcule $\ln(x + 1)$, et la fonction EXPM (*exp moins 1*) calcule $(\exp x) - 1$. Pour des arguments proches de 0, ces fonctions donnent une plus grande précision que la séquence correspondante de fonctions (vous trouverez un exemple utilisant LNP1 en page 128, sous le titre « Flux financiers constants »).

Affichez la deuxième ligne du menu LOGS.

NEXT

```
3: -2,06761537357E-13
2: 0
1: 2,46855006079
SINH ASINH COSH ACOSH TANH ATANH
```

Les fonctions de cette ligne du menu sont les fonctions hyperboliques et leurs inverses : SINH (*sinus hyperbolique*) et ASINH (*sinus hyperbolique inverse*), COSH (*cosinus hyperbolique*) et ACOSH (*cosinus hyperbolique inverse*), et TANH (*tangente hyperbolique*) et ATANH (*tangente hyperbolique inverse*). Ces fonctions sont dérivées de e^x , la fonction exponentielle. Ce sont toutes des fonctions monadiques opérant sur le nombre présent dans le niveau 1 de la pile.

Autres fonctions sur les nombres réels

Le menu REAL contient des fonctions opérant essentiellement sur des nombres réels.

Sélectionnez le menu REAL.

REAL

```
3: -2,06761537357E-13
2: 0
1: 2,46855006079
NEG FACT RAND RDZ MAXR MINR
```

La fonction NEG (*negate*, opposé) donne $-x$ pour un argument x . La fonction FACT (*factorielle*) donne $n!$ pour un entier positif n ou la fonction gamma $\Gamma(x + 1)$ pour un argument non entier x . La commande RAND (*random number*, nombre aléatoire) donne un nombre aléatoire calculé à partir d'une racine spécifiée par RDZ (*randomize*, racine de nombre aléatoire).

Les fonctions MAXR (*réel maximal*) et MINR (*réel minimal*) donnent des constantes symboliques pour le plus grand et le plus petit nombre réel positif représentable sur le HP-28S (pour forcer un résultat numérique avec une constante symbolique, voyez « Utilisation de π » en page 74.)

Cette section présente l'utilisation de la fonction NEG. Par commodité, vous pouvez exécuter NEG en appuyant sur **[CHS]** (*changement de signe*) si la ligne de commande n'est pas affichée. Pour entrer la commande NEG sur la ligne de commande, par exemple lorsque vous entrez un programme, appuyez sur **[NEG]** ou sur **[N]** **[E]** **[G]**.

Changez maintenant deux fois le signe du nombre dans le niveau 1 en appuyant sur **[CHS]** puis sur **[NEG]**.

Changez le signe du nombre dans le niveau 1.

[CHS]

3:	-2,06761537357E-13	0
2:		0
1:	-2,46855006079	0
NEG FACT RAND RDE MAXR MINR		

Changez à nouveau le signe du nombre.

[NEG]

3:	-2,06761537357E-13	0
2:		0
1:	2,46855006079	0
NEG FACT RAND RDE MAXR MINR		

Définition de nouvelles fonctions

Vous pouvez créer des variables de programme opérant de la même façon que les fonctions intégrées ; vous pouvez même les utiliser dans des expressions. Ces variables de programme, appelées *fonctions utilisateurs* doivent satisfaire aux conditions suivantes :

- Elles doivent spécifier explicitement leurs arguments.
- Elles doivent donner un seul résultat.

Vous pouvez, par exemple, définir la fonction de cotangente COT de la façon suivante : $\cot x = 1/\tan x$.

Commencez le programme.

◀

```
2: 0
1: 2,46855006079
* → x
NEG FACT RAND RDE MARK MINR
```

Spécifiez l'argument.

◀ → LC x LC

```
2: 0
1: 2,46855006079
* → x
NEG FACT RAND RDE MARK MINR
```

La flèche vers la droite indique que le nom suivant représente une *variable locale*, variable restreinte à la fonction ou au programme dans laquelle elle est définie.

Il est judicieux de suivre quelques règles pour distinguer les variables locales des variables ordinaires, dites « globales ». Dans ce manuel, nous utiliserons des minuscules pour les variables locales et des majuscules pour les autres. (La touche **LC** alterne entre les frappes en minuscules et en majuscules.)

Définissez la fonction.

' ◀ 1/x (TAN (LC x

```
2: 0
1: 2,46855006079
* → x' INV (TAN(x)
NEG FACT RAND RDE MARK MINR
```

Entrez le programme.

ENTER

```
2: 2,46855006079
1: * → x 'INV(TAN(x))'
»
NEG FACT RAND RDE MARK MINR
```

Le calculateur ajoute les parenthèses droites et les délimiteurs pour vous.

Ce programme prend un argument dans la pile (syntaxe RPN) ou dans une expression (syntaxe algébrique) et l'appelle x ; puis évalue l'expression $1/\tan x$ en utilisant la définition locale de x .

Stockez le programme dans une variable nommée COT.

['] COT **[STO]**

```
3: -2,06761537357E-13
2: 0
1: 2,46855006079
NEG FCT RAND RDZ MAXR MINR
```

Vous pouvez désormais utiliser COT dans une expression suivant la syntaxe RPN ou algébrique, de la même façon que les fonctions trigonométriques intégrées.

Calculez $\cot 45^\circ$ en syntaxe RPN.

[MODE] **[DEG]**
45 **[USER]** **[COT]**

```
3: 0
2: 2,46855006079
1: 1
COT GE 0 REMO
```

Calculez $\cot -45^\circ$ en syntaxe algébrique.

['] **[COT]** **[(-]** -45 **[ENTER]**

```
3: 2,46855006079
2: 1
1: 'COT(-45)'
COT GE 0 REMO
```

Évaluez l'expression.

[EVAL]

```
3: 2,46855006079
2: 1
1: -1
COT GE 0 REMO
```

6

Fonctions opérant sur les nombres complexes

Le HP-28S accepte un type d'objet qui représente les nombres complexes. Le nombre complexe $z = 3 + 4i$, par exemple, est représenté par l'objet $(3, 4)$. Chaque nombre complexe étant un objet unique, vous pouvez effectuer des calculs sur les nombres complexes aussi facilement que sur les nombres réels.

Les deux nombres réels composant un nombre complexe peuvent représenter les coordonnées d'un point dans un plan. Le HP-28S utilise par exemple les nombres complexes pour représenter les coordonnées de traçage. La deuxième section de ce chapitre décrit deux systèmes de coordonnées, rectangulaire et polaire, et indique comment convertir les coordonnées d'un point d'un système dans l'autre.

Utilisation des nombres complexes

La plupart des fonctions opérant sur les nombres réels peuvent également opérer sur les nombres complexes. Vous pouvez, par exemple, effectuer des calculs arithmétiques sur des nombres complexes de la même façon que sur des nombres réels : placez-les dans la pile opératoire et exécutez la fonction. Effectuez les calculs suivants :

$$((9 + 2i) + (-4 + 3i)) \times (6 + i)$$

Effacez la pile et entrez $9 + 2i$.

■ CLEAR
[] 9 [SPACE] 2 [ENTER]

3:	
2:	
1:	(9.2)
COT	GE 0 RENO

Ajoutez $-4 + 3i$. (Appuyez sur $\boxed{4}$ $\boxed{\text{CHS}}$ pour entrer -4 .)

$\boxed{(-}$ $\boxed{-4}$ $\boxed{\text{SPACE}}$ $\boxed{3}$ $\boxed{+}$

```
3:
2:
1: (5.5)
COT GE D RENO
```

Multipliez par $6 + i$.

$\boxed{(-}$ $\boxed{6}$ $\boxed{\text{SPACE}}$ $\boxed{1}$ $\boxed{\times}$

```
3:
2:
1: (25.35)
COT GE D RENO
```

Un argument réel peut parfois produire un résultat complexe.

Calculez $\sqrt{-4}$.

$\boxed{-4}$ $\boxed{\sqrt{x}}$

```
3:
2: (25.35)
1: (0.2)
COT GE D RENO
```

Calculez $\arcsin 2$.

$\boxed{2}$ $\boxed{\text{TRIG}}$ $\boxed{\text{ASIN}}$

```
2: (0.2)
1: (1.57079632679,
-1.31695789692)
SIN ASIN COS ACOS TAN ATAN
```

Les fonctions spécifiques aux nombres complexes se trouvent dans le menu COMPLEX.

Sélectionnez le menu COMPLEX.

$\boxed{\blacksquare}$ $\boxed{\text{COMPLX}}$

```
2: (0.2)
1: (1.57079632679,
-1.31695789692)
R→C C→R RE IM CONJ SIGN
```

Toutes les commandes du menu COMPLEX sont brièvement décrites dans l'annexe D, « Les menus et leurs jeux de libellés ». La section « COMPLEX » dans le Manuel de référence décrit ces commandes en détail.

- $\text{R} \rightarrow \text{C}$ (*réel-en-complexe*) convertit deux nombres réels x et y en un nombre complexe (x, y) .
- $\text{C} \rightarrow \text{R}$ (*complexe-en-réel*) convertit un nombre complexe (x, y) en deux nombres réels x et y .
- RE (*partie réelle*) donne x pour un argument complexe (x, y) .

- IM (*partie imaginaire*) donne y pour un argument complexe (x, y) .
- CONJ (*conjugué*) donne $(x, -y)$ pour un argument complexe (x, y) .
- SIGN donne $(x/\sqrt{x^2 + y^2}, y/\sqrt{x^2 + y^2})$ pour un argument complexe (x, y) .

Affichez la ligne suivante du menu COMPLEX.

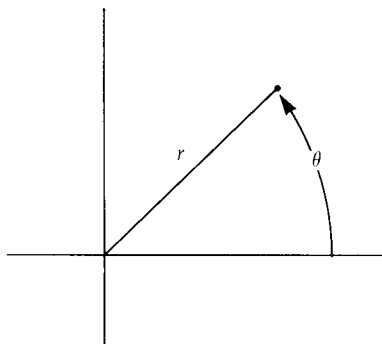
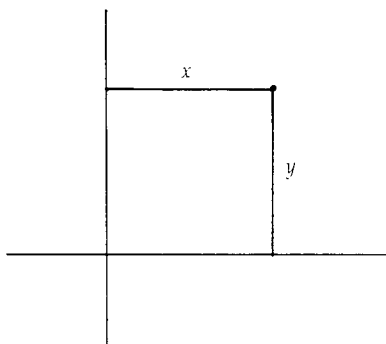
NEXT

2:	(0.2)
1:	(1,57079632679.
	-1,31695789692)
R→P	P→R
ABS	NEG
ARG	

Utilisation des coordonnées polaires

Les fonctions suivantes (exceptée NEG) permettent de manipuler des nombres complexes représentant des coordonnées polaires.

La position d'un point sur un plan est généralement décrite selon l'un des deux systèmes de coordonnées suivants : rectangulaire (x, y) et polaire (r, θ) . La figure suivante illustre les deux systèmes.



- **R→P** (*rectangulaire-en-polaire*) convertit un nombre complexe denotation rectangulaire (x, y) en notation polaire (r, θ) .
- **P→R** (*polaire-en-rectangulaire*) convertit un nombre complexe denotation polaire (r, θ) en notation rectangulaire (x, y) .
- **ABS** (*valeur absolue*) donne r pour un argument complexe (x, y) .
- **NEG** donne $(-x, -y)$ pour un argument complexe (x, y) .
- **ARG** donne θ pour un argument complexe (x, y) .

Remarquez que seul **P→R** interprète un nombre complexe comme jeu de coordonnées polaires ; toutes les autres fonctions, arithmétiques, trigonométriques, logarithmiques, hyperboliques etc., interprètent les nombres complexes comme des jeux de coordonnées rectangulaires. Souvenez-vous de cette importante règle : *tout nombre complexe représentant des coordonnées polaires doit être converti en coordonnées rectangulaires avant d'être utilisé dans des calculs.*

Supposons qu'au cours d'une croisière vous parcouriez 2 milles au cap 36° puis 3 milles au cap 65° . Quels sont la distance et le cap résultant, avec deux décimales ?

Sélectionnez le mode Degrés et l'affichage FIX 2.

MODE DEG 2 FIX

```
3: (25.00,35.00)
2: (0.00,2.00)
1: (1.57,-1.32)
STD FIX SCI ENG DEG RAD
```

Entrez la première distance et le premier cap.

2 SPACE 36

```
2: (0.00,2.00)
1: (1.57,-1.32)
(2 36
STD FIX SCI ENG DEG RAD
```

Convertissez ces valeurs en coordonnées rectangulaires.

COMPLX NEXT P→R

```
3: (0.00,2.00)
2: (1.57,-1.32)
1: (1.62,1.18)
R→P P→R ABS NEG ARG
```

Entrez la deuxième distance et le deuxième cap.

3 SPACE 65

```
2: (1.57,-1.32)
1: (1.62,1.18)
(3 65
R→P P→R ABS NEG ARG
```

Convertissez ces valeurs en coordonnées rectangulaires.

P→R

3:	(1,57,-1,32)
2:	(1,62,1,18)
1:	(1,27,2,72)
R→P P→R ABS NEG ARG	

Ajoutez les coordonnées rectangulaires.

+

3:	(0,00,2,00)
2:	(1,57,-1,32)
1:	(2,89,3,89)
R→P P→R ABS NEG ARG	

Convertissez la somme en coordonnées polaires.

R→P

3:	(0,00,2,00)
2:	(1,57,-1,32)
1:	(4,85,53,46)
R→P P→R ABS NEG ARG	

La distance totale est 4,85 milles et le cap résultant 53,46 °.

Fonction utilisateur pour l'addition de coordonnées polaires

Voici un simple programme SOMP (*somme polaire*) pour automatiser le processus que vous avez suivi manuellement dans la section précédente.

Commencez le programme.

«

2:	(1,57,-1,32)
1:	(4,85,53,46)
«	
R→P P→R ABS NEG ARG	

Spécifiez les arguments. (Séparez les arguments par un espace.)

→ LC x SPACE y

2:	(1,57,-1,32)
1:	(4,85,53,46)
«	→ x y
R→P P→R ABS NEG ARG	

La flèche vers la droite indique que les noms qui suivent sont des *variables locales*, variables restreintes à la fonction dans laquelle elles sont définies.

Définissez la fonction.

`['] R→P ([P→R ([x [] +`
`P→R ([y [ENTER]`

2:	(4,85.53,46)
1:	« + x y 'R→P(P→R(x)+ P→R(y))' »
R→P P→R ABS NEG ARG	

Le calculateur ajoute les parenthèses droites et les délimiteurs pour vous.

Ce programme prend deux arguments dans la pile (syntaxe RPN) ou dans l'expression (syntaxe algébrique) et les appelle x et y , puis calcule les coordonnées polaires de la somme des coordonnées rectangulaires de x et y .

Stockez le programme dans une variable nommée SOMP.

`['] SOMP [STO]`

3:	(0,00.2,00)
2:	(1,57.-1,32)
1:	(4,85.53,46)
R→P P→R ABS NEG ARG	

Utilisez maintenant SOMP pour répéter le calcul précédent, une fois en syntaxe RPN et une autre fois en syntaxe algébrique.

Entrez la première distance et le premier cap.

`[] 2 [SPACE] 36 [ENTER]`

3:	(1,57.-1,32)
2:	(4,85.53,46)
1:	(2,00.36,00)
R→P P→R ABS NEG ARG	

Entrez la deuxième distance et le deuxième cap.

`[] 3 [SPACE] 65`

2:	(4,85.53,46)
1:	(2,00.36,00)
(3 65	
R→P P→R ABS NEG ARG	

Exécutez SOMP.

`[USER] [SOMP]`

3:	(1,57.-1,32)
2:	(4,85.53,46)
1:	(4,85.53,46)
SOMP	COT GE 0 REND

Le résultat est identique au précédent.

Essayez maintenant la syntaxe algébrique.

2:	(4,85.53,46)
1:	'SOMP((2.36).(3.65))
SOMP COT GE 0 REND	

Les parenthèses extérieures et la virgule centrale isolent les arguments de SOMP ; les autres parenthèses et virgules font partie de la syntaxe des nombres complexes. N'oubliez pas que vous devez placer *deux* jeux de parenthèses lorsque vous utilisez un nombre complexe comme argument d'une expression qui suit la syntaxe algébrique.

Evaluez l'expression.

3:	(4,85.53,46)
2:	(4,85.53,46)
1:	(4,85.53,46)
SOMP COT GE 0 REND	

Représentation graphique d'expressions

Ce chapitre explique comment tracer des graphiques sur l'affichage du HP-28S. La représentation graphique donne une compréhension visuelle de la façon dont une expression se comporte, et facilite également la recherche d'une première estimation des racines, des minimums et des maximums d'une expression. Dans le chapitre suivant, nous vous indiquerons comment utiliser la fonction de résolution d'équations pour obtenir des valeurs exactes.

Dans le présent chapitre, vous allez apprendre à utiliser certaines des commandes du menu PLOT. L'annexe C, « Les menus et leurs jeux de libellés », décrit brièvement ces commandes, et la section « PLOT » du Manuel de référence en donne une description plus détaillée.

Au cours du premier exemple, vous allez tracer $\sin x$ en mode Radians, mais vous devez tout d'abord effectuer quelques préparatifs de façon à vous assurer que l'affichage correspondra aux illustrations.

Le traçage utilise une variable nommée PPAR pour stocker la liste des paramètres. Éliminez toute variable PPAR existante pour vous assurer que le tracé suivant utilisera les paramètres de traçage par défaut.

Effacez la pile et sélectionnez le menu PLOT.



Affichez la deuxième ligne du menu PLOT.



Éliminez toute variable PPAR existante.

☐ PPAR ☐ PURGE

```

3:
2:
1:
PPAR RES AXES CENTR SW SH

```

Sélectionnez le mode d'angle en Radians et le mode d'affichage numérique STD.

☐ MODE ☐ RAD ☐ STD

```

3:
2:
1:
STD FIX SCI ENG DEG RAD

```

Entrez maintenant l'expression.

☐ TRIG ☐ SIN X ☐ ENTER

```

3:
2:
1: 'SIN(X)'
SIN ASIN COS ACOS TAN ATAN

```

Stockez l'expression comme *équation en cours*—variable normale avec le nom spécial EQ. (Cette convention de nom est identique à celle utilisée au chapitre 4).

☐ PLOT ☐ STEQ

```

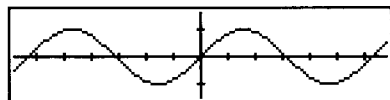
3:
2:
1:
STEQ RCEQ PMIN PMAX INDEF DRAW

```

La pression sur ☐ STEQ est équivalente à celle de ☐ EQ ☐ STO.

Tracez l'expression.

☐ DRAW



Attendez la disparition du témoin (●), indiquant que le traçage est terminé.

L'axe horizontal est celui de la variable indépendante (x dans cet exemple) et l'axe vertical est celui de la variable dépendante (valeur de l'expression $\sin x$). Les graduations sur les deux axes définissent des intervalles de longueur 1.

Impression d'un tracé

Si vous disposez d'une imprimante HP 82240A, vous pouvez imprimer une copie du tracé que vous venez de créer.

1. Préparez l'imprimante conformément aux instructions données dans son manuel.
2. Appuyez sur **[ON]** et maintenez cette touche enfoncée.
3. Appuyez sur **[L]** (touche au-dessus de laquelle est écrit « PRINT »).
4. Relâchez **[ON]**.

Cette séquence de touches est l'équivalent de la commande PRLCD (*imprimer LCD*, présente sur la première ligne du menu PRINT). Vous pouvez utiliser cette séquence pour imprimer une copie de l'affichage presque à tout moment, sans perturber le fonctionnement du calculateur.

Si vous rédigez un programme permettant de tracer une expression et d'imprimer le résultat, utilisez la séquence de commandes suivante :

...CLLCD DRAW PRLCD...

Reprenez l'exemple précédent et restaurez l'affichage normal de la pile.

[ON]



The screenshot shows the HP-41C calculator's LCD display. The top three lines show the stack with values 3, 2, and 1. The bottom line shows a menu with options: STEQ, RCEQ, PMIN, PMAK, INDEF, and DRAW. The DRAW option is currently selected.

Modification de l'échelle du tracé

En général, représenter graphiquement une expression ne donne pas des résultats aussi propres et clairs du premier coup. Si vous tracez une expression que vous ne connaissez pas, vous aurez peut-être besoin d'ajuster la zone de traçage, définie par les paramètres de traçage, afin de montrer les caractéristiques propres à l'expression.

Si vous savez d'avance quelle zone vous voulez tracer, vous pouvez modifier directement les paramètres de traçage dans PPAR (variable décrite en détail dans la section « PLOT » du Manuel de référence). Mais le plus souvent, vous devrez procéder par tâtonnement pour trouver la zone de traçage appropriée. Ce manuel vous indique comment utiliser les commandes PLOT pour « recentrer » le tracé.

A titre de deuxième exemple, tracez l'expression $x^3 - x^2 - x + 3$.

Placez l'expression au niveau 1.

$\boxed{X} \boxed{-} \boxed{X} \boxed{^} \boxed{3} \boxed{-} \boxed{X} \boxed{^} \boxed{2} \boxed{-} \boxed{X} \boxed{+} \boxed{3}$
 $\boxed{\text{ENTER}}$

```
3:
2:
1:      'X^3-X^2-X+3'
[STEQ] [RCEQ] [PMIN] [PMAK] [INDEP] [DRAW]
```

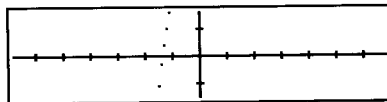
Stockez l'expression comme équation en cours.

$\boxed{\text{STEQ}}$

```
3:
2:
1:
[STEQ] [RCEQ] [PMIN] [PMAK] [INDEP] [DRAW]
```

Tracez l'expression.

$\boxed{\text{DRAW}}$



L'axe horizontal est celui des x et l'axe vertical, celui de l'expression $x^3 - x^2 - x + 3$.

Ce tracé met en évidence un *zéro* de l'expression, c'est-à-dire une valeur X pour laquelle l'expression est nulle. Ce zéro se situe là où le graphe de l'expression coupe l'axe des X . Au chapitre suivant, nous utiliserons l'algorithme de résolution d'équations pour trouver une valeur numérique précise de x pour ce zéro.

Pour voir une plus grande partie de ce graphe, réduisez l'échelle verticale et retracez la courbe.

Rétablissez l'affichage normal de la pile.

$\boxed{\text{ON}}$

```
3:
2:
1:
[STEQ] [RCEQ] [PMIN] [PMAK] [INDEP] [DRAW]
```

Réduisez de moitié l'échelle verticale en utilisant l'opération ***H** (*multiplication hauteur*) de la ligne suivante du menu.

NEXT 2 ***H**

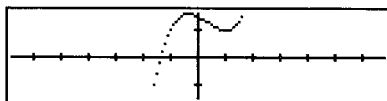
```

3:
2:
1:
PPAR RES AXES CENTR HW *H

```

Tracez à nouveau la courbe avec les nouveaux paramètres.

PREV **DRAW**



Les graduations de l'axe horizontal définissent toujours des intervalles de longueur 1, mais celles de l'axe vertical définissent maintenant des intervalles de longueur 2.

Vous allez maintenant translater le tracé, de manière à placer la partie intéressante de la courbe au centre de l'affichage.

Translation du tracé

Après chaque traçage, le calculateur laisse un réticule au centre de l'affichage. (Vous ne pouvez pas voir ce réticule lorsque les axes se trouvent eux-mêmes au centre, car il est superposé aux axes.) Le réticule vous permet de *numériser* un point quelconque de l'affichage, c'est-à-dire d'envoyer les coordonnées de ce point dans la pile. Vous allez maintenant numériser le point que vous voulez faire figurer au centre du tracé suivant, puis utiliser ce point pour ajuster les paramètres de traçage.

Placez le réticule à la position indiquée ci-dessous.

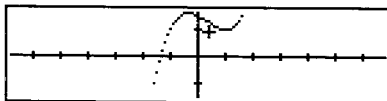
☐ (quatre fois)

☐ (neuf fois)



Numérisez le point.

INS



Retourner à l'affichage de la pile.

ON

```
3:
2:
1:      (,4.1,8)
STEQ RCEQ PMIN PMAW INDEF DRAW
```

Les coordonnées du point numérisé, représentées par un nombre complexe, se trouvent au niveau 1.

Redéfinissez le centre du tracé, en utilisant **CENTR** dans la ligne suivante du menu.

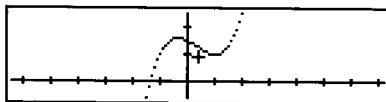
NEXT **CENTR**

```
3:
2:
1:
PPAR RES AWES CENTR #W #H
```

Les coordonnées sont prises dans la pile et servent à ajuster les paramètres de traçage. Contrairement à **#H**, **CENTR** ne modifie pas l'échelle.

Effectuez un nouveau tracé.

PREV **DRAW**

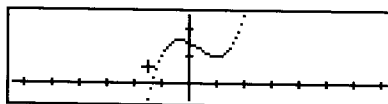


Effectuez maintenant un zoom sur une partie intéressante du tracé. Vous pourriez à nouveau utiliser **#H** avec un facteur d'échelle *fractionnaire*. Un facteur d'échelle de 0,5, par exemple, ramènerait l'échelle verticale à sa valeur initiale. Il existe néanmoins une façon plus souple d'agrandir (effectuer un « zoom ») un tracé.

Redéfinir les limites de traçage

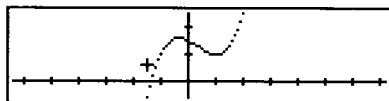
Vous allez, cette fois, numériser deux points, le premier pour l'angle inférieur gauche de la nouvelle zone de traçage et le second pour l'angle supérieur droit, afin d'agrandir la partie du tracé présentant les maximums et minimums locaux.

Placez le réticule sur l'angle inférieur gauche désiré.

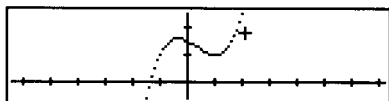


Numérisez le point.

INS

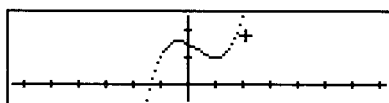


Placez le réticule sur l'angle supérieur droit désiré.



Numérisez le point.

INS



Retournez à l'affichage de la pile.

ON

```
3:
2:      (-1,5.1,2)
1:      (2,1.3,6)
STEC RCEG PMIN PMAX INDEF DRAW
```

Les coordonnées de l'angle inférieur gauche, représenté par un nombre complexe, se trouvent au niveau 2. Les coordonnées de l'angle supérieur droit se trouvent au niveau 1. Vos coordonnées peuvent différer légèrement de celles indiquées dans l'illustration.

Redéfinissez l'angle supérieur droit du tracé avec **PMAX** (traçage du maximum).

PMAX

```
3:
2:
1:      (-1,5.1,2)
STEC RCEG PMIN PMAX INDEF DRAW
```

Les coordonnées sont prises dans la pile et servent à ajuster les paramètres de traçage.

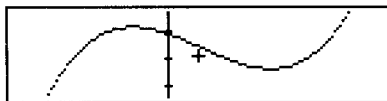
Redéfinissez l'angle inférieur droit du tracé avec **PMIN** (traçage du minimum).

PMIN

```
3:
2:
1:
STEC RCEG PMIN PMAX INDEF DRAW
```

Effectuez un autre tracé.

DRAW



Du fait que vous avez modifié la hauteur et la largeur du tracé, l'échelle verticale et l'échelle horizontale sont toutes les deux modifiées.

Le tracé montre deux *extrêmes* dans le graphe de l'expression, un maximum local et un minimum local. Dans le chapitre suivant, vous utiliserez l'algorithme de résolution d'équations pour trouver une valeur précise pour le minimum. Afin d'éviter de répéter toutes ces étapes, nécessaires pour générer les paramètres de traçage en cours, stockez la valeur en cours de PPAR dans une variable portant un nom différent. Pour créer à nouveau ce tracé, au chapitre suivant, il vous suffira de rendre à PPAR la valeur qu'elle possède actuellement.

Retournez à l'affichage de la pile.

ON

```
3:
2:
1:
-----
STEC RCEC FMIN FMAX INDEF DRAW
```

Placez les contenus en cours de PPAR dans la pile.

NEXT **PPAR**

```
1: { (-1,5,1,2)
   {2,1.9,6} X 1 (0.0)
   }
-----
PPAR RES AXES CENTR %W %H
```

Si vous désirez des informations complémentaires sur les paramètres et sur le traçage, consultez la section « PLOT » du *Manuel de référence*.

Créez une variable PPAR1 contenant les paramètres de traçage en cours.

' **PPAR** 1 **STO**

```
3:
2:
1:
-----
PPAR RES AXES CENTR %W %H
```

Vous êtes maintenant prêt à utiliser l'algorithme de résolution d'équations pour calculer les valeurs numériques précises de x pour le zéro, ainsi que le minimum local de l'expression.

Représentation graphique d'équations

Les exemples de ce chapitre étaient tous constitués d'expressions. Néanmoins, les mêmes règles et techniques s'appliquent au traçage d'équations. Lorsque la variable EQ contient une équation, DRAW trace *chaque côté* de l'équation comme une expression. Vous pouvez trouver une racine de l'équation en cherchant à quel endroit les deux courbes se coupent, car c'est à cet endroit que l'équation est nulle.

Résolution d'équations

Ce chapitre indique comment trouver un zéro et un minimum pour l'expression que vous avez représentée graphiquement au chapitre précédent. Vous aurez besoin de certains des résultats de ce chapitre ; suivez donc pas à pas les étapes du chapitre 7 si vous ne l'avez pas encore fait.

La section « SOLVE » du Manuel de référence décrit en détail l'algorithme de résolution d'équation.

Recherche du zéro d'une expression

L'exemple suivant suppose que l'expression $x^3 - x^2 - x + 3$ est toujours l'équation en cours et que vous avez créé la variable PPAR1, comme décrit au chapitre précédent. Vous allez tracer à nouveau l'expression, numériser une approximation pour un zéro de l'expression et utiliser l'algorithme de résolution d'équation pour trouver une valeur plus précise de x pour ce zéro.

Avant de commencer ces exemples, effacez la pile, sélectionnez le mode Radians et l'affichage FIX 2.

☐ CLEAR
☐ MODE ☒ RAD
 2 ☒ FIX

3:					
2:					
1:					
STD	FIX	SCI	ENG	DEG	RAD

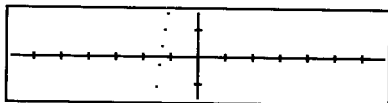
Éliminez toute variable PPAR existante de façon à vous assurer que le tracé suivant utilisera les paramètres par défaut.

☐ PLOT ☐ NEXT
☐ PPAR ☐ PURGE

3:					
2:					
1:					
PPAR	RES	ANES	CENTR	%W	%H

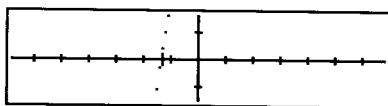
Tracez maintenant l'expression.

PREV **DRAW**



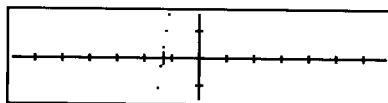
Ce tracé identifie un *zéro* de l'expression, valeur de x pour laquelle l'expression est nulle. Le zéro se trouve à l'endroit où la courbe coupe l'axe horizontal.

Placez le réticule à l'endroit approximatif de l'intersection de la courbe et de l'axe horizontal. Utilisez **▲**, **▼**, **◀** et **▶** pour déplacer le réticule.



Numérisez ce point.

INS



Vous allez maintenant utiliser ce point pour trouver la valeur exacte de x correspondant à ce zéro de l'expression. (Dans cet exemple, l'expression possède plusieurs racines, l'estimation permet d'identifier celle que vous voulez calculer.)

Retournez à l'affichage de la pile.

ON

```

3:
2:
1:          (-1,30.0,00)
STEQ RCEZ PMIN PMAX INDEF DRAW

```

Les coordonnées du point numérisé, représentées par un nombre complexe, se trouvent au niveau 1. (Vos coordonnées peuvent différer légèrement de celles indiquées dans l'illustration.)

Sélectionnez le menu de l'algorithme de résolution d'équations.

SOLV **SOLVR**

```

3:
2:
1:          (-1,30.0,00)
X  EXPR=

```

L'algorithme de résolution d'équations montre toutes les variables de l'équation en cours (X uniquement dans cet exemple).

Stockez l'estimation numérisée dans la variable X.

X: (-1,300,000)
2:
1:
X EXP=

Bien que l'estimation numérisée comporte deux coordonnées, l'algorithme de résolution d'équations n'utilise que la première comme estimation.

Calculez maintenant X.

X: -1,36
Sign Reversal
1: -1,36
X EXP=

Le message `Sign Reversal` indique que l'algorithme de résolution d'équations a trouvé une solution approximative, correcte sur 12 chiffres. Si l'algorithme avait trouvé une solution exacte, il aurait affiché le message `Zero`. Ces messages *qualitatifs* sont présentés dans la section « SOLVE » du Manuel de référence.

Retournez maintenant à l'affichage de la pile.

3:
2:
1: -1,36
X EXP=

Recherche d'un minimum ou d'un maximum

Pour trouver le zéro d'une expression, l'algorithme de résolution d'équations échantillonne la courbe, à partir de l'estimation, et essaie de trouver des points de plus en plus proches de l'axe x . Si votre estimation est proche d'un *minimum local positif* ou *négatif*, il n'y a pas de point voisin qui soit plus proche de l'axe x . Dans ce cas, l'algorithme trouve le minimum (ou maximum) plutôt qu'un zéro. (L'algorithme ne s'égare généralement pas vers de tels minimums ou maximums, sauf si votre estimation l'y force.)

Regardez le graphique que vous avez créé au chapitre précédent, page 96. Il indique que l'expression possède un minimum local positif et un maximum local positif. L'algorithme de résolution d'équations peut trouver le minimum car, dans la région considérée, c'est le point le *plus proche* de l'axe x , il ne trouve pas le maximum car, dans la région considérée, c'est le point le *plus éloigné* de l'axe x .

Au cours de cette section, vous allez d'abord tracer l'expression en utilisant les paramètres de traçage stockés dans la variable PPAR1, puis vous allez numériser trois points pour estimer le minimum et enfin vous allez utiliser l'algorithme de résolution d'équations pour trouver une valeur plus précise du minimum.

Placez dans la pile la liste stockée dans PPAR1.

USER **PPAR1**

```
1: { (-1,50,1,20)
    (2,10,3,60) X 1,00
    (0,00,0,00) }
N PPAR PPAR1 EQ SOMP COT
```

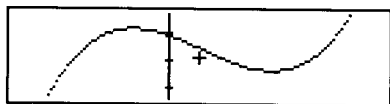
Affectez les valeurs de PPAR1 à la variable PPAR.

' **PPAR** **STO**

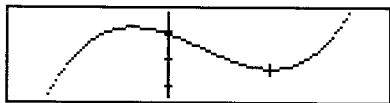
```
3:
2:
1: -1,36
N PPAR PPAR1 EQ SOMP COT
```

Tracez l'expression.

PLOT **DRAW**



Placez le réticule le plus près possible du minimum.

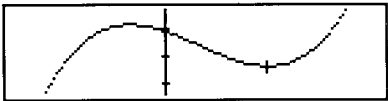


Numérisez le point.

INS



Placez le réticule immédiatement à gauche du minimum.



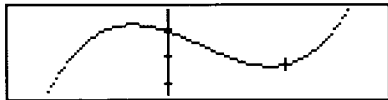
Numérisez le point.

INS



Placez le réticule immédiatement à droite du minimum et numérisez le point.

INS



Retournez à l'affichage de la pile.

ON

3:	(0,99.1,97)
2:	(0,86.2,05)
1:	(1,15.2,05)
STEC REEC FMIN FMAX INDER DRAW	

Les trois points se trouvent dans les niveaux 1, 2 et 3 (vos points peuvent différer légèrement de ceux de l'illustration.)

Combinez maintenant les trois points en une liste pour vous permettre de les manipuler comme un objet unique. Cette application illustre l'utilisation typique des listes : la combinaison de plusieurs objets en un seul.

LIST 3 →LIST

1:	((0,99.1,97)
	(0,86.2,05)
	(1,15.2,05))
→LIST LIST2 PUT GET PUT1 GET1	

Sélectionnez le menu de l'algorithme de résolution d'équations.

SOLV SOLVR

1:	((0,99.1,97)
	(0,86.2,05)
	(1,15.2,05))
X	EXPRE

Le menu de l'algorithme de résolution d'équations contient toutes les variables de l'équation en cours (X uniquement dans cet exemple).

Stockez la liste de points dans la variable X.

X

```
X: ( (0,99,1,97) (0,8..
      (2,10,3,60) X 1,00
      (0,00,0,00) )
X  EXPR=
```

La liste de points de la pile est stockée dans la variable X comme estimations initiales.

Résolvez l'équation pour X.

X

```
X: 1,00
Extremum
1: 1,00
X  EXPR=
```

Le message **Extremum** indique que l'algorithme de résolution d'équations a trouvé un minimum ou un maximum de l'expression.

Retournez à l'affichage normal de la pile.

ON

```
3: -1,36
2: ( (-1,50,1,20) (2,1..
1: 1,00
X  EXPR=
```

Calculez la valeur extrême.

EXPR=

```
EXPR=2,00
2: 1,00
1: 2,00
X  EXPR=
```

La valeur minimum est 2.

Calculs sur des flux financiers constants

Cette section montre comment utiliser l'algorithme de résolution d'équations pour effectuer des calculs sur des flux financiers constants (FCST) avec un nombre de périodes n , un taux d'intérêt périodique $i\%$, un paiement pmt , une valeur actuelle va et une valeur future vf . La formule de FSCT est la suivante :

$$(1 - vapu) \times pmt \times (100/i) + va = -vf \times vapu$$

où




$$\begin{aligned} \text{vapu (valeur actuelle d'un paiement unique)} &= (1 + i/100)^{-n} \\ &= \exp(-n \times \ln(1 + i/100)). \end{aligned}$$

Cette formule suppose que les paiements sont effectués à la fin de chaque période.

Voici les étapes générales du calcul :

1. Entrez l'expression de *vapu* et stockez-la dans la variable VAPU.
2. Entrez l'équation et stockez-la dans la variable FCST.
3. Définissez FSCT comme équation en cours.
4. Utilisez l'algorithme de résolution d'équations pour calculer l'une quelconque des cinq variables *n*, *i*, *pmt*, *va* ou *vf*, en fonction de valeurs spécifiques des quatre autres.

Avant de commencer, effacez la pile et sélectionnez le mode d'affichage FIX 2.

 CLEAR
 MODE 2  FIX

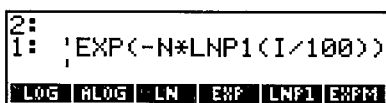


3:
2:
1:
STO FIX SCI ENG DEG RAD

Etape 1. Entrez l'expression pour *vapu* et stockez-la dans la variable VAPU.

Entrez l'expression pour *vapu*.




 LOGS  EXP  - N  x  LN P1
 ÷ 100  ENTER



2:
1: EXP(-N*LN P1(I/100))
LOG RLOG LN EXP LN P1 EXP P1

Cette expression bénéficie de la plus grande précision de LN P1 pour calculer $\ln(1 + i/100)$.

Créez la variable VAPU et vérifiez le menu personnalisé USER.

 VAPU  STO
 USER



3:
2:
1:
VAPU N PPAR PPAR1 EQ SOMP

Etape 2. Entrez l'équation et stockez-la dans la variable FCST.

Entrez l'équation de FCST.

(1 - VAPU) x PMT x
 100 ÷ I + VA = - VF x VAPU
 ENTER

2:
 1: '(1-VAPU)*PMT*100/I+
 VA=-VF*VAPU'
 VAPU N PPAR PPAR1 EQ SOMP

Créez la variable FCST.

FCST STO

3:
 2:
 1:
 FCST VAPU N PPAR PPAR1 EQ

Le menu USER contient le nouveau libellé pour FCST.

Etape 3. Définissez FCST comme équation en cours.

Entrez le nom FCST.

FCST

2:
 1:
 'FCST'
 FCST VAPU N PPAR PPAR1 EQ

Stockez le nom FCST dans la variable EQ.

SOLV STEQ

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

Etape 4. Utilisez l'algorithme de résolution d'équations pour calculer l'une quelconque des cinq variables *n*, *i*, *pmt*, *va* ou *vf* en fonction des quatre autres.

Sélectionnez le menu de l'algorithme de résolution d'équations.

SOLVR

3:
 2:
 1:
 N I PMT VA VF LEFT=

Toutes les variables de FCST et de VAPU sont présentes dans le menu (les variables de VAPU y apparaissent car l'équation en cours, FCST, contient VAPU.)

Avec les valeurs $N = 30 \times 12$; $I = 11,5/12$; $PMT = -630$ et $VF = 0$, calculez VA (la variable PMT a une valeur négative car elle représente de l'argent versé ; alors que l'argent reçu correspond à des valeurs positives.)

Affectez tout d'abord la valeur de N.

30 12

N: 360,00					
2:					
1:					
<input type="button" value="N"/>	<input type="button" value="I"/>	<input type="button" value="PMT"/>	<input type="button" value="VA"/>	<input type="button" value="VF"/>	<input 185="" 205"="" 379="" 94="" data-label="Text" type="button" value="LEFT=</input></td></tr> </table> </div> <div data-bbox="/> <p>Affectez la valeur de I.</p>

11,5 12

I: 0,96					
2:					
1:					
<input type="button" value="N"/>	<input type="button" value="I"/>	<input type="button" value="PMT"/>	<input type="button" value="VA"/>	<input type="button" value="VF"/>	<input 295="" 316"="" 423="" 94="" data-label="Text" type="button" value="LEFT=</input></td></tr> </table> </div> <div data-bbox="/> <p>Affectez la valeur de PMT.</p>

-630

PMT: -630,00					
2:					
1:					
<input type="button" value="N"/>	<input type="button" value="I"/>	<input type="button" value="PMT"/>	<input type="button" value="VA"/>	<input type="button" value="VF"/>	<input 400="" 405="" 426"="" 94="" data-label="Text" type="button" value="LEFT=</input></td></tr> </table> </div> <div data-bbox="/> <p>Affectez la valeur de VF.</p>

0

VF: 0,00					
2:					
1:					
<input type="button" value="N"/>	<input type="button" value="I"/>	<input type="button" value="PMT"/>	<input type="button" value="VA"/>	<input type="button" value="VF"/>	<input 394="" 516="" 537"="" 94="" data-label="Text" type="button" value="LEFT=</input></td></tr> </table> </div> <div data-bbox="/> <p>Calculez maintenant VA.</p>

VA: 63617,64					
Zero					
1:	63617,64				
<input type="button" value="N"/>	<input type="button" value="I"/>	<input type="button" value="PMT"/>	<input type="button" value="VA"/>	<input type="button" value="VF"/>	<input 626="" 669"="" 878="" 91="" data-label="Text" type="button" value="LEFT=</input></td></tr> </table> </div> <div data-bbox="/> <p>Le message Zero indique que la valeur donnée est une solution exacte de l'équation en cours.</p>

Solutions symboliques

Ce chapitre décrit deux méthodes pour trouver des solutions symboliques. Il y a une méthode simple qui permet de résoudre les équations à deux inconnues en calculant l'expression linéaire qui représente ses deux zéros. Une autre méthode permet d'obtenir une solution symbolique pour une variable dans des équations plus générales.

Chaque méthode est valable pour des expressions et des équations. Le *zéro* d'une expression $f(x)$ est équivalent à la *racine* de l'équation $f(x) = 0$, et la *racine* de l'équation $f(x) = g(x)$ est équivalente au *zéro* de l'expression $f(x) - g(x)$.

Recherche des zéros d'une expression du second degré






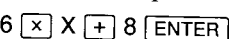
Vous pouvez trouver les deux zéros d'une expression du second degré sans tracer de courbe ou fournir des estimations. L'exemple suivant résout $x^2 - 6x + 8$.

Avant de commencer l'exemple, effacez la pile et sélectionnez le mode d'affichage STD.



Placez l'expression dans la pile.

  2  6  X  8 



Placez le nom X dans la pile, pour indiquer la variable à rechercher.

X **ENTER**

```
3:
2:      'X^2-6*X+8'
1:      'X'
STD=  FIX  SCI  ENG  DEG  RAD=
```

Calculez les zéros en utilisant QUAD (*quadratique*) dans le menu SOLVE.

SOLVE **QUAD**

```
3:
2:
1:      '(6+s1*2)/2'
STEP=  RCEQ  SOLVR  ISOL  QUAD  SHOW=
```

Cette expression représente les deux solutions de l'expression quadratique. La variable s1 représente un signe arbitraire, +1 ou -1, et chaque valeur de s1 correspond à un zéro de l'expression.

Stockez l'expression comme équation en cours.

STEQ

```
3:
2:
1:
STEP=  RCEQ  SOLVR  ISOL  QUAD  SHOW=
```

Affichez le menu de l'algorithme de résolution d'équations.

SOLVR

```
3:
2:
1:
S1  EXPR=  [ ]  [ ]  [ ]  [ ]
```

s1 est la seule variable dans l'équation en cours.

Définissez tout d'abord s1 comme un signe positif.

1 **S1**

```
S1: 1
3:
2:
1:
S1  EXPR=  [ ]  [ ]  [ ]  [ ]
```

Renvoyez l'une des solutions dans le niveau 1.

EXPR=

```
EXPR=4
3:
2:
1:      4
S1  EXPR=  [ ]  [ ]  [ ]  [ ]
```

Définissez maintenant s1 comme un signe négatif.

-1 S1

s1: -1				
2:				
1:	4			
S1	EXPR=			

Renvoyez l'autre solution dans le niveau 1.

EXPR=

EXPR=2				
2:				
1:	2			
S1	EXPR=			

Les deux racines de $x^2 - 6x + 8$ sont $x = 4$ et $x = 2$.

Isolation d'une variable

Le HP-28S peut isoler une seule occurrence d'une variable dans une équation et renvoyer une expression représentant la solution symbolique de l'équation. En d'autres termes, si x est la variable pour laquelle l'équation est résolue, et si a , b et c sont les autres variables de l'équation, le fait d'isoler x donne une expression en fonction de a , b et c , telle que l'équation soit résolue lorsque x prend la valeur de l'expression.

Pour le premier exemple, isolez x dans l'équation

$$a(x + 3) - b = c.$$

Cet exemple est simple parce qu'il n'y a qu'une seule occurrence de x . Les exemples suivants montrent comment manipuler une équation pour n'obtenir qu'une seule occurrence de x .

Effacez la pile.

■ CLEAR

3:				
2:				
1:				
S1	EXPR=			

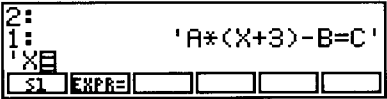
Placez l'équation dans la pile.

A x (X + 3 ■) - B = C
ENTER

3:				
2:				
1:	'A*(X+3)-B=C'			
S1	EXPR=			

Spécifiez la variable à isoler.

☐ X



Isolez x , à l'aide de la fonction ISOL (*isoler*) du menu SOLVE.



L'expression donnée représente une solution symbolique de l'équation pour x —c'est-à-dire l'équation

$$a(x + 3) - b = c$$

est résolue lorsque $x = (c + b)/a - 3$.

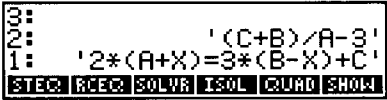
Développement et regroupement

Si x apparaît plusieurs fois, vous devez manipuler l'équation de façon à éliminer toutes les occurrences de x , sauf une. L'exemple suivant montre comment isoler x dans l'équation

$$2(a + x) = 3(b - x) + c.$$

La stratégie dans ce cas sera de développer l'équation, de soustraire des deux côtés de l'équation le terme x présent dans l'un des membres, de regrouper l'équation pour annuler le terme x d'un côté et de placer un seul terme x de l'autre côté, et ensuite l'isoler.

Placez l'équation dans la pile.



Sélectionnez le menu ALGBRA.



Dans cet exemple, vous utiliserez **EXPAN** (*expand*, développer) et **COLCT** (*collect*, regrouper en facteur) pour manipuler l'équation. Dans l'exemple suivant, vous utiliserez **FORM** (*former une expression algébrique*) pour manipuler une équation. Toutes les commandes du menu ALGEBRA sont brièvement décrites dans l'annexe D, « Les menus et leurs jeux de libellés ». La section « ALGEBRA » du Manuel de référence décrit ce menu plus en détail. De plus, FORM, un éditeur algébrique puissant, possède sa propre section « ALGEBRA (FORM) » dans le Manuel de référence.

Développez les deux côtés de l'équation.

EXPAN

```
3:
2:      '(C+B)/A-3'
1:      '2*A+2*X=3*B-3*X+C'
COLCT EXPAN SIZE FORM DSUB ENSUB
```

Pour soustraire des deux côtés de l'équation le terme en x ($2x$) présent du côté gauche, placez tout d'abord ce dernier dans la pile.

$\boxed{2} \times \boxed{X}$ **ENTER**

```
3:      '(C+B)/A-3'
2:      '2*A+2*X=3*B-3*X+C'
1:      '2*X'
COLCT EXPAN SIZE FORM DSUB ENSUB
```

Soustrayez ensuite $2x$ des deux côtés.

$\boxed{-}$

```
2:      '(C+B)/A-3'
1:      '2*A+2*X-2*X=3*B-3*X'
      '+C-2*X'
COLCT EXPAN SIZE FORM DSUB ENSUB
```

Regroupez l'équation.

COLCT

```
3:
2:      '(C+B)/A-3'
1:      '2*A=3*B+C-5*X'
COLCT EXPAN SIZE FORM DSUB ENSUB
```

Chaque côté est regroupé de façon indépendante et les termes x s'annulent du côté gauche.

Vous pouvez maintenant isoler x dans l'équation. Spécifiez la variable à isoler.

\boxed{X}

```
2:      '(C+B)/A-3'
1:      '2*A=3*B+C-5*X'
      'X'
COLCT EXPAN SIZE FORM DSUB ENSUB
```

Isolez x . La commande ISOL apparaît sur la deuxième ligne du menu ALGEBRA, ainsi que dans le menu SOLVE.

NEXT **ISOL**

```
3:
2:      '(C+B)/A-3'
1:      '(3*B+C-2*A)/5'
TAYLR ISOL QUAD SHOW DSGET EXGET
```

L'expression obtenue représente une solution symbolique de l'équation pour x —c'est-à-dire que l'équation

$$2(a + x) = 3(b - x) + c$$

est résolue lorsque $x = (3b + c - 2a)/5$.

Utilisation de FORM

Si l'équation contient plusieurs occurrences de x , si l'une d'elles a un coefficient symbolique, la commande COLCT ne combinera pas les coefficients. L'exemple suivant isole x pour l'équation

$$a(x + b) + 2x = c,$$

dans laquelle x apparaît plusieurs fois et a le coefficient symbolique a . La stratégie dans ce cas sera de développer l'équation, d'utiliser FORM pour regrouper les coefficients de x et ensuite d'isoler x .

Placez l'équation dans la pile.

A **x** **(** **x** **+** **B** **)** **+** **2** **x** **=**
C **ENTER**

```
3:
2:      '(C+B)/A-3'
1:      '(3*B+C-2*A)/5'
TAYLR ISOL QUAD SHOW DSGET EXGET
```

Développez l'équation.

NEXT **EXPAN**

```
3:
2:      '(C+B)/A-3'
1:      '(3*B+C-2*A)/5'
COLCT EXPAN SIZE FORM DSGET EXGET
```

Utilisez FORM pour regrouper les coefficients de x .

FORM

```
((((1)*X)+(A*B)))+(2*X))=
C)
COLCT EXPAN LEVEL EXGET (←) (→)
```

Le fonctionnement normal du calculateur est interrompu lorsque FORM est actif. L'affichage FORM présente l'équation avec toutes ses sous-expressions placées entre parenthèses. Vous utiliserez FORM pour manipuler des sous-expressions dans l'équation.

Le but est de combiner $(A * X)$ et $(2 * X)$ en un seul terme $((A + 2) * X)$. Les trois étapes nécessaires sont présentées ci-dessous comme vous le feriez sur papier. La forme de l'équation en cours est :

$$(ax + ab) + 2x = c$$

La première étape consiste à permuter ax et ab pour obtenir :

$$(ab + ax) + 2x = c$$

La deuxième associe ax et $2x$, donnant :

$$ab + (ax + 2x) = c$$

La troisième fusionne ax et $2x$, ce qui donne :

$$ab + (a + 2) x = c$$

Etape 1. Permutez ax et ab .

Placez le curseur (le ou les caractère(s) en vidéo inverse) sur le +.



La position du curseur détermine la sous-expression sur laquelle vous agirez. Dans l'exemple, vous voulez permuter les arguments de la sous-expression $((A * X) + (A * B))$.

Affichez la première ligne des fonctions de manipulation pour +.

NEXT

$$(((A * X) + (A * B)) + (2 * X)) =$$

C)

-() ↔ ←M →M ←A →A

Les manipulations qui apparaissent lorsque vous appuyez sur NEXT agissent sur la fonction ou la variable indiquée par le curseur ; dans le cas présent elles agissent sur +.

Permutez les arguments par rapport au signe + en utilisant la fonction ↔ (permutation).

↔

$$(((A * B) + (A * X)) + (2 * X)) =$$

C)

-() ↔ ←M →M ←A →A

Retournez au menu FORM principal.

ENTER

$$(((A * B) + (A * X)) + (2 * X)) =$$

C)

COLCT EXPAN LEVEL EXGET [+]

Etape 2. Associez ax et $2x$.

Placez le curseur sur le deuxième +.

[+] [+] [+] [+]

$$(((A * B) + (A * X)) + (2 * X)) =$$

C)

COLCT EXPAN LEVEL EXGET [+]

Vous allez maintenant agir sur la sous-expression

$$(((A * B) + (A * X)) + (2 * X))$$

pour associer les termes $(A * X)$ et $(2 * X)$ en une seule sous-expression.

Affichez la première ligne de manipulations pour +.

NEXT

$$(((A * B) + (A * X)) + (2 * X)) =$$

C)

-() ↔ ←M →M ←A →A

Ce sont les mêmes manipulations que précédemment, puisque cette sous-expression est à nouveau est une addition.

Associez les termes $(A * X)$ et $(2 * X)$ dans la sous-expression $((A * X) + (2 * X))$, en utilisant \rightarrow (associer à droite).

\rightarrow

$$(((A * B) + ((A * X) + (2 * X))) = C)$$

$\leftarrow 0$ \leftrightarrow $\leftarrow M$ $M \rightarrow$ $\leftarrow A$ $A \rightarrow$

Retournez au menu FORM principal.

ENTER

$$(((A * B) + ((A * X) + (2 * X))) = C)$$

COLCT EXPAN LEVEL ERGET \leftarrow \rightarrow

Etape 3. Fusionnez ax et $2x$.

Placez le curseur sur le deuxième +.

\leftarrow \leftarrow \leftarrow \leftarrow

$$((A * B) + ((A * X) + (2 * X))) = C$$

COLCT EXPAN LEVEL ERGET \leftarrow \rightarrow

Vous voulez maintenant manipuler la sous-expression $((A * X) + (2 * X))$ pour combiner les coefficients de X .

Affichez la première ligne de manipulations pour +.

NEXT

$$((A * B) + ((A * X) + (2 * X))) = C$$

$\leftarrow 0$ \leftrightarrow $\leftarrow M$ $M \rightarrow$ $\leftarrow A$ $A \rightarrow$

Combinez les coefficient de X avec \rightarrow (mettre en facteur à droite).

\rightarrow

$$((A * B) + ((A + 2) * X)) = C$$

$1/0$ \leftrightarrow $\leftarrow 0$ $0 \rightarrow$ $\leftarrow A$ $A \rightarrow$

Ceci combine $(A * X)$ et $(2 * X)$ en un seul terme $((A + 2) * X)$.

Sortez de FORM et renvoyez l'équation modifiée dans la pile.

ON

3:	'(C+B)/A-3'
2:	'(3*B+C-2*A)/5'
1:	'A*B+(A+2)*X=C'

COLCT EXPAN SIZE FORM ODSUB EXSUB

Maintenant que x n'apparaît qu'une seule fois dans l'équation, vous pouvez l'isoler.

Spécifiez la variable à isoler.

☐ X

```
2:      '(3*B+C-2*A)/5'
1:      'A*B+(A+2)*X=C'
'X'
COLT EXPAN SIZE FORM DEVS8 EXVS8
```

Isolez x .

```
3:      '(C+B)/A-3'
2:      '(3*B+C-2*A)/5'
1:      '(C-A*B)/(A+2)'
TAYLR ISOL QUAD SHOW DESET EXDET
```

L'expression renvoyée représente une solution symbolique de l'équation pour x —c'est-à-dire que l'équation

$$a(x + b) + 2x = c$$

est résolue lorsque $x = (c - ab)/(a + 2)$.

Calcul différentiel et intégral

Vous pouvez dériver de façon symbolique toute expression pour laquelle une dérivée plausible existe. Les intégrations sont plus restreintes : vous pouvez calculer une *intégrale numérique définie* pour une expression quelconque mais vous ne pouvez calculer d'*intégrale symbolique exacte* que pour les polynômes.

Ce chapitre contient des exemples simples de calcul de dérivée et d'intégrale indéfinie ou définie. La section « Calcul différentiel et intégral » du Manuel de référence traite ces sujets plus en détail.

Calcul de la dérivée d'une expression

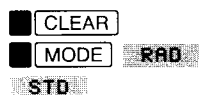
Vous pouvez calculer la dérivée d'une expression pas-à-pas, en observant comment le calculateur applique les règles de la dérivation, ou vous pouvez la calculer complètement, en un seul coup. Les résultats sont identiques. Au cours de cette section, vous calculerez deux fois la dérivée d'une même expression, en utilisant les deux méthodes.

Dérivation pas-à-pas

Supposons que vous vouliez trouver :

$$\frac{d}{dx} \tan(x^2 + 1)$$

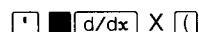
Avant de commencer l'exemple, effacez la pile, sélectionnez le mode Radians et l'affichage numérique STD.



Supprimez la variable X (si elle existe).



Commencez maintenant l'expression pour la dérivée à partir de la variable de dérivation.



Entrez maintenant l'expression à dériver.



Cette expression représente la dérivée, par rapport à x , de $\tan(x^2 + 1)$.

Évaluez l'expression.



Le résultat reflète l'application de la règle de dérivation :

$$\frac{d}{dx} \tan(x^2 + 1) = \frac{d}{d(x^2 + 1)} \tan(x^2 + 1) \times \frac{d}{dx} (x^2 + 1)$$

La dérivée de la fonction tangente a été évaluée. Vous allez maintenant évaluer la dérivée de $x^2 + 1$.

Évaluez une nouvelle fois l'expression.

Eval

```
2:
1: '(1+SQ(TAN(X^2+1))) *
   dX(X^2)'
SIN ASIN COS ACOS TAN ATAN
```

Le résultat reflète la dérivée de la somme :

$$\frac{d}{dx} (x^2 + 1) = \frac{d}{dx} x^2 + \frac{d}{dx} 1$$

La dérivée de 1 est 0, le terme disparaît donc. Évaluez maintenant la dérivée de x^2 .

Évaluez une troisième fois l'expression.

Eval

```
2:
1: '(1+SQ(TAN(X^2+1))) *
   (dX(X)*2*X^(2-1))'
SIN ASIN COS ACOS TAN ATAN
```

Le résultat reflète à nouveau la règle de la dérivation :

$$\frac{d}{dx} x^2 = \frac{d}{dx} (x)^2 \times \frac{d}{dx} x$$

La dérivée de x^2 a été évaluée. Évaluez enfin la dérivée de x elle-même.

Évaluez une quatrième fois l'expression.

Eval

```
2:
1: '(1+SQ(TAN(X^2+1))) *
   (2*X)'
SIN ASIN COS ACOS TAN ATAN
```

Voici la dérivée complètement évaluée.

Dérivation complète

Pour dériver une expression en une seule étape, effectuez la dérivation comme une opération dans la pile opérationnelle :

$$\frac{d}{dx} \tan (x^2 + 1)$$

Placez l'expression à dériver dans la pile.

CLEAR
 TAN X ^ 2 + 1 ENTER

```
3:
2:
1: 'TAN(X^2+1)'
```

SIN ASIN COS ACOS TAN ATAN

Spécifiez la variable de dérivation.

X ENTER

```
3:
2: 'TAN(X^2+1)'
```

1: 'X'

SIN ASIN COS ACOS TAN ATAN

Calculez la dérivée.

d/dx

```
2:
1: '(1+SQ(TAN(X^2+1))) * (2*X)'
```

SIN ASIN COS ACOS TAN ATAN

La dérivée complète apparaît au niveau 1.

Intégration d'une expression

Le HP-28S calcule l'intégrale indéfinie d'une expression par *intégration symbolique*, ce qui donne comme résultat une expression. Cette méthode donne un résultat exact uniquement pour les expressions polynomiales (pour les autres expressions, le HP-28S intègre une approximation de l'expression, obtenue à l'aide d'une série de Taylor. La section « Calcul différentiel et intégral » du Manuel de référence présente cette méthode d'une façon plus détaillée). Le premier exemple ci-après illustre l'intégration symbolique.

Au contraire, les intégrales définies sont calculées par *intégration numérique*, qui donne un résultat numérique. Cette méthode s'applique à n'importe quelle expression ayant un « comportement normal » au sens mathématique. L'intégration numérique est illustrée dans le deuxième exemple ci-après.

Intégration symbolique des polynômes

Dans cet exemple, vous allez intégrer symboliquement le polynôme

$$8x^3 + 9x^2 + 2x + 5.$$

Effacez la pile.

■ CLEAR

```
3:
2:
1:
SIN ASIN COS ACOS TAN ATAN
```

Placez le polynôme dans la pile.

8 [X] X [^] 3 [+] 9 [X] X [^] 2
[+] 2 [X] X [+] 5 [ENTER]

```
3:
2:
1: '8*X^3+9*X^2+2*X+5'
SIN ASIN COS ACOS TAN ATAN
```

Spécifiez la variable d'intégration.

[X] X [ENTER]

```
3:
2: '8*X^3+9*X^2+2*X+5'
1: 'X'
SIN ASIN COS ACOS TAN ATAN
```

Spécifiez le degré du polynôme.

3 [ENTER]

```
3: '8*X^3+9*X^2+2*X+5'
2: 'X'
1: '3'
SIN ASIN COS ACOS TAN ATAN
```

Intégrez le polynôme.

■ [∫]

```
2:
1: '5*X+X^2+3*X^3+2*X^4'
SIN ASIN COS ACOS TAN ATAN
```


Attendez que le témoin ((•)) disparaisse, indiquant que l'intégration est terminée. L'intégrale est renvoyée dans le niveau 1.

Intégration numérique des expressions

Dans cet exemple, vous allez calculer la valeur numérique de l'intégrale

$$\exp(x^3 + 2x^2 - x + 4) dx$$

Effacez la pile.

 CLEAR





```
3:
2:
1:
SIM ASIN COS ACOS TAN ATAN
```

Placez l'expression dans la pile.

  LOGS  EXP X  ^ 3  + 2
 x  ^ 2  - X  + 4  ENTER

```
3:
2:
1: 'EXP(X^3+2*X^2-X+4)'  
LOG ALOG LN EXP LNPL EXPM
```

Entrez le variable et les limites de l'intégration. Vous devez entrer ces valeurs comme des objets dans une liste (ceci est une utilisation typique des listes—combiner plusieurs objets de manière à pouvoir les traiter comme un objet unique).

 { X  SPACE 0  SPACE 1  ENTER

```
3:
2: 'EXP(X^3+2*X^2-X+4)'  
1: { X 0 1 }  
LOG ALOG LN EXP LNPL EXPM
```

X est la variable de l'intégration, 0 et 1 en sont les limites.

Entrez ensuite la précision souhaitée.

Si l'expression contient des constantes dérivées de données empiriques, utilisez la même précision que les constantes. Par exemple, si les constantes ont une précision de trois chiffres, spécifiez une précision de 0,001.

Dans cet exemple, vous allez effectuer l'intégration d'une expression qui ne comporte pas de constantes empiriques ; vous pouvez donc spécifier une précision de 12 chiffres. Toutefois, plus la précision choisie est grande, plus le processus itératif de l'intégration numérique est long. Nous vous recommandons, par conséquent, de spécifier une précision de 0,00001 pour cet exemple.

1E-5 ENTER

3:	'EXP(X^3+2*X^2-X+4)'
2:	(X 0 1)
1:	.00001
LOG ALOG LN EXP LNFI EXPM	

Calculez l'intégrale.



3:	
2:	103.117678153
1:	1.03086911923E-3
LOG ALOG LN EXP LNFI EXPM	

L'estimation de l'intégrale est renvoyée dans le niveau 2, tandis que le limite d'erreur se trouve dans le niveau 1.

La valeur de l'intégrale est $103,118 \pm 0,001$. Remarquez que la limite d'erreur obtenue est approximativement le produit de l'intégrale estimée et de la précision choisie.

11

Vecteurs et matrices

Le HP-28S vous permet de manipuler deux types de tableaux : les *vecteurs*, qui sont des tableaux uni-dimensionnels, et les *matrices*, qui sont des tableaux bi-dimensionnels. Vous pouvez introduire les vecteurs et les matrices sous forme d'objets individuels, appelés *objets en tableau*, et les utiliser dans des calculs aussi facilement que des nombres.

Ce chapitre traite des calculs de base utilisant des tableaux réels—vecteurs ou matrices dont les éléments sont des nombres réels. Vous pouvez aussi effectuer des calculs avec des tableaux dont les éléments sont des nombres complexes.

L'annexe D, «Les menus et leurs jeux de libellés », décrit brièvement toutes les commandes du menu ARRAY. La section « ARRAY » du Manuel de référence en donne une description plus complète.

Vecteurs

Cette section présente l'arithmétique des vecteurs, le produit vectoriel et le produit scalaire.

Saisie d'un vecteur

Avant de commencer ces exemples, effacez la pile et sélectionnez le mode d'affichage STD.



Entrez le vecteur [2 3 4]. Vous pouvez séparer les éléments du vecteur par des espaces ou des points (si vous utilisez la notation avec virgule décimale). Si vous utilisez la notation américaine — point décimal — vous pouvez utiliser une virgule pour séparer les éléments. Pour éviter toute confusion nous utiliserons la touche espace dans ce manuel.)

[2 [SPACE] 3 [SPACE] 4 [ENTER]

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

Multiplication et division d'un vecteur par un nombre

Multipliez le vecteur par 15.

15 [x].

```
3:
2:
1: [ 30 45 60 ]
STD= FIX SCI ENG DEG RAD=
```

Pour la multiplication, l'ordre des arguments est quelconque, de la même façon que pour les multiplications de nombres. Par contre, pour les divisions, le vecteur doit se trouver au niveau 2 et le nombre au niveau 1.

Divisez le vecteur par 5.

5 [÷]

```
3:
2:
1: [ 6 9 12 ]
STD= FIX SCI ENG DEG RAD=
```

Addition et soustraction de vecteurs

Vous pouvez ajouter et soustraire des vecteurs, de la même façon que vous additionnez et soustrayez des nombres, à condition que ces vecteurs aient le même nombre d'éléments. Pour la soustraction, l'ordre des arguments est important, tout comme il est important dans une simple soustraction.

Pour cet exemple, soustrayez le vecteur [-10 20 30] de celui se trouvant au niveau 1.

[-10 [SPACE] 20 [SPACE] 30 [-

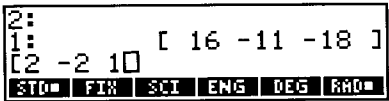
```
3:
2:
1: [ 16 -11 -18 ]
STD= FIX SCI ENG DEG RAD=
```

Calcul du produit vectoriel

Calculez le produit vectoriel du vecteur se trouvant au niveau 1 et du vecteur $[2 \ -2 \ 1]$ (le produit vectoriel est seulement défini pour des vecteurs à deux ou trois éléments).

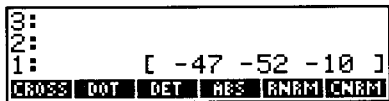
Entrez le vecteur.

[2] [SPACE] - 2 [SPACE] 1



Calculez le produit vectoriel avec la fonction CROSS de la troisième ligne du menu ARRAY.

■ [ARRAY] [NEXT] [NEXT] **CROSS**

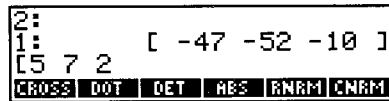


Calcul du produit scalaire

Calculez le produit scalaire du vecteur présent au niveau 1 et du vecteur $[5 \ 7 \ 2]$ (les deux vecteurs doivent avoir le même nombre d'éléments).

Entrez le vecteur.

[5] [SPACE] 7 [SPACE] 2



Calculez le produit scalaire.

DOT



Matrices

Cette section décrit l'inversion des matrices et le calcul des déterminants. Ces deux calculs sont limités aux matrices carrées, c'est-à-dire à celles qui ont le même nombre de colonnes et de lignes.

Les calculs effectués sur les vecteurs s'appliquent aussi aux matrices (à l'exception des produits scalaires et vectoriels). Vous pouvez multiplier ou diviser une matrice par un nombre, et vous pouvez additionner ou soustraire des matrices (à condition qu'elles aient les mêmes dimensions).

Saisie de matrice

Entrez la matrice suivante :

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 1 & 2 & 4 \end{bmatrix}$$

Commencez la matrice.

[]

```
2:
1:                                     -619
[ ]
CROSS DOT DET ABS RNRM CNRM
```

Entrez chaque ligne de la matrice comme un vecteur différent.

[] 1 [SPACE] 2 [SPACE] 3

[] 1 [SPACE] 3 [SPACE] 3

[] 1 [SPACE] 2 [SPACE] 4 [ENTER]

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

Visualisation d'une grande matrice

Lorsqu'une matrice possède beaucoup d'éléments — ou beaucoup d'éléments non entiers — il se peut que vous ne puissiez la visualiser entièrement. Pour visualiser l'ensemble d'une matrice de grandes dimensions, utilisez **[EDIT]** (si la matrice se trouve au niveau 1) ou **[VISIT]** pour renvoyer la matrice dans la ligne de commande. Vous pouvez alors utiliser le menu des touches de curseur pour afficher une partie quelconque de la matrice. La section « Corriger un objet », au chapitre 18, présente cette procédure en détail.

Inversion d'une matrice

La matrice présente au niveau 1 étant carrée, vous pouvez en calculer l'inverse.



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

CROSS	DOT	DET	ABS	RNORM	CNORM
1	1	1	1	1	1

Calcul du déterminant

La matrice présente au niveau 1 étant carrée, vous pouvez en calculer le déterminant.

[illegible]

Multiplication de deux tableaux

Vous pouvez utiliser la fonction `×` pour multiplier deux matrices ou une matrice et un vecteur (utilisez `CROSS` ou `DOT` pour multiplier deux vecteurs, comme décrit précédemment).

Multiplication de deux matrices

L'ordre des arguments est important lors de la multiplication de matrices. Le nombre de *colonnes* de la matrice du niveau 2 doit être égal au nombre de *lignes* de la matrice du niveau 1. Vous pouvez, par exemple, calculer le produit matriciel suivant.

$$\begin{bmatrix} 2 & 2 \\ 4 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 2 & 2 & 1 & 4 \\ 3 & 4 & 2 & 1 \end{bmatrix}$$

Pour calculer ce produit matriciel :

Entrez la première matrice.

[1] [1] 2 [SPACE] 2
[1] 4 [SPACE] 1
[1] 2 [SPACE] 3 [ENTER]

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

Entrez la deuxième matrice.

[1] [1] 2 [SPACE] 2 [SPACE] 1 [SPACE] 4
[1] 3 [SPACE] 4 [SPACE] 2 [SPACE] 1

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

Multipliez les matrices.

[x]

```
1: [[ 10 12 6 10 ]  
    [ 11 12 6 17 ]  
    [ 13 16 8 11 ]]  
CROSS DOT DET ABS RNRM CNRM
```

Multiplication d'une matrice par un vecteur

L'ordre des arguments est important lors de la multiplication d'une matrice par un vecteur. La matrice doit se trouver au niveau 2 et le vecteur au niveau 1. Le nombre de *colonnes* dans la matrice doit être égal au nombre d'*éléments* du vecteur.

Dans l'exemple suivant, multipliez la matrice présente au niveau 1 par le vecteur [3 1 1 2].

Entrez le vecteur.

[1] 3 [SPACE] 1 [SPACE] 1 [SPACE] 2

```
1: [[ 10 12 6 10 ]  
    [ 11 12 6 17 ]  
    [ 3 1 1 2 ]  
CROSS DOT DET ABS RNRM CNRM
```

Multipliez la matrice par le vecteur.

[x]

```
3: -619  
2: 1  
1: [ 68 85 85 ]  
CROSS DOT DET ABS RNRM CNRM
```

Solution d'un système d'équations linéaires

Pour résoudre un système de n équations linéaires avec n variables, utilisez un vecteur *constant* à n éléments, une matrice *coefficient* $n \times n$ *coefficient* et la division (\div). Le vecteur constant contient les valeurs constantes des équations. La matrice coefficient contient les coefficients des variables.

L'exemple suivant montre comment résoudre un système de trois équations linéaires indépendantes à trois variables. Supposons que les équations sont :

$$\begin{aligned} 3x + y + 2z &= 13 \\ x + y - 8z &= -1 \\ -x + 2y + 5z &= 13 \end{aligned}$$

Entrez le vecteur constant.

[1] 13 [SPACE] -1 [SPACE] 13 [ENTER]

```
3: [ 68 85 85 ]
2: [ 13 -1 13 ]
1: [ 13 -1 13 ]
CROSS DOT DET ABS RNRM CNRM
```

Entrez la matrice coefficient.

[1] [1] 3 [SPACE] 1 [SPACE] 2
[1] 1 [SPACE] 1 [SPACE] -8
[1] -1 [SPACE] 2 [SPACE] 5

```
2: [ 68 85 85 ]
1: [ 13 -1 13 ]
[[3 1 2[1 1 -8[-1 2 5]
CROSS DOT DET ABS RNRM CNRM
```

Résolvez le système d'équations.

\div

```
3: [ 68 85 85 ]
2: [ 13 -1 13 ]
1: [ 2 5 1 ]
CROSS DOT DET ABS RNRM CNRM
```

Les valeurs dans le vecteur solution sont les racines des équations :

$$x = 2, \quad y = 5, \quad z = 1$$

Pour résoudre des systèmes d'équations sous ou sur-déterminés, ou des systèmes quasi-singuliers, consultez la section « ARRAY » du *Manuel de référence*.

Statistiques

Ce chapitre décrit comment saisir des données statistiques et effectuer des calculs statistiques avec des observations sur une ou deux variables avec les commandes du menu STAT. L'annexe D, « Les menus et leurs jeux de libellés », décrit brièvement l'ensemble des commandes du menu STAT. La section « STAT » du Manuel de référence les décrit en détail.

Le tableau suivant liste le pourcentage d'augmentation de l'indice des prix à la consommation (IPC), de l'indice des prix à la production (IPP) et du taux de chômage (TC), sur une période de cinq ans aux Etats-Unis. Saisissez ces données et tirez-en des statistiques.

Données de l'exemple statistique

Année	IPC	IPP	TC
1975	9,1	9,2	8,5
1976	5,8	4,6	7,7
1977	6,5	6,1	7,0
1978	7,6	7,8	6,0
1979	11,5	19,3	5,8

Saisie des données

Les données statistiques sont stockées dans une matrice statistique nommée Σ DAT, qui est une matrice ordinaire portant un nom particulier. Chaque ligne de la matrice contient une observation, qui, dans cet exemple, comprend les valeurs de IPC, IPP et TC pour une année.

Avant de commencer, effacez la pile et sélectionnez le mode d'affichage numérique FIX 2.

CLEAR
MODE 2 **FIX**



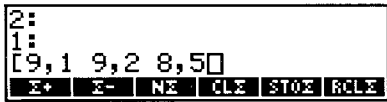
Effacez toute donnée statistique précédente avec la fonction CL Σ du menu STAT (toute Σ DAT existante est supprimée).

STAT **CL Σ**



Entrez l'observation pour 1975.

[] 9,1 **SPACE** 9,2 **SPACE** 8,5



Stockez cette observation dans Σ DAT.

Σ +



Le calculateur crée automatiquement une nouvelle matrice nommée Σ DAT. L'observation pour 1975 se trouve dans la première ligne de Σ DAT.

Entrez l'observation pour 1976.

[] 5,8 **SPACE** 4,6 **SPACE** 7,7

Σ +



L'observation pour 1976 est ajoutée à Σ DAT pour former la deuxième ligne de cette matrice.

Entrez l'observation pour 1977.

[] 6,5 [SPACE] 6,1 [SPACE] 7 [Σ+]

```
3:
2:
1:
[Σ+  Σ-  NE  CLE  STOE  RCLΣ]
```

L'observation pour 1977 est ajoutée à ΣDAT pour former la troisième ligne de la matrice.

Modification de données

Si vous faites une erreur en saisissant des données et que vous vous en apercevez avant d'appuyer sur [Σ+], vous pouvez simplement corriger la ligne de commande. Mais supposons que vous croyez avoir fait une erreur en entrant l'observation pour 1976. Vous pouvez renvoyer cette observation dans la pile, la corriger et la renvoyer dans ΣDAT.

Retirez l'observation pour 1977 (dernière ligne de ΣDAT) et renvoyez-la dans la pile.

[Σ-]

```
3:
2:
1: [ 6,50 6,10 7,00 ]
[Σ+  Σ-  NE  CLE  STOE  RCLΣ]
```

Retirez l'observation pour 1976 (dernière ligne de ΣDAT) et renvoyez-la dans la pile.

[Σ-]

```
3:
2: [ 6,50 6,10 7,00 ]
1: [ 5,80 4,60 7,70 ]
[Σ+  Σ-  NE  CLE  STOE  RCLΣ]
```

Si vous constatez que vous avez fait une erreur dans cette observation, appuyez sur [EDIT] pour placer cette observation dans la ligne de commande, corrigez la donnée erronée, puis appuyez sur [ENTER] pour placer l'observation dans la pile (consultez « Modification d'objets existants » au chapitre 18).

Rentrez l'observation corrigée pour 1976 dans ΣDAT.

[Σ+]

```
3:
2:
1: [ 6,50 6,10 7,00 ]
[Σ+  Σ-  NE  CLE  STOE  RCLΣ]
```

Rentrez l'observation pour 1977 dans Σ DAT.

$\Sigma+$

```
3:
2:
1:
Σ+ Σ- NE CLΣ STOS RCLΣ
```

Entrez maintenant les autres observations (pour 1978 et 1979) et vérifiez que vous avez correctement entré les cinq observations.

[7,6] [SPACE] 7,8 [SPACE] 6 $\Sigma+$

[11,5] [SPACE] 19,3 [SPACE] 5,8

$\Sigma+$

NE

```
3:
2:
1:
Σ+ Σ- NE CLΣ STOS RCLΣ 5,00
```

Statistiques sur une variable

Dans cette section, vous allez calculer la moyenne, l'écart-type et la variance de IPC, IPP et TC. Les données de IPC sont contenues dans la première colonne de Σ DAT, celles de IPP se trouvent dans la deuxième colonne et celles de TC, dans la troisième.

Affichez la deuxième ligne du menu STAT.

[NEXT]

```
3:
2:
1:
TOT MEAN SDIV VAR MAXE MINE 5,00
```

Voici les commandes de moyenne, écart-type et variance.

Calcul de la moyenne

Calculez la moyenne.

Σ MEAN

```
3:
2:
1:
TOT MEAN SDIV VAR MAXE MINE 5,00
[ 8,10 9,40 7,00 ]
```

La moyenne d'IPC est 8,1 ; celle d'IPP est 9,4 et celle de TC est 7.

Calcul de l'écart-type

Calculez l'écart-type.

SDEV

3:	5,00			
2:	[8,10	9,40	7,00]
1:	[2,27	5,80	1,14]
TOT MEAN SDEV VAR MAXE MIN				

L'écart-type de l'échantillon pour IPC est de 2,27 ; pour IPP de 5,8 et pour TC de 1,14.

Calcul de la variance

Calculez la variance.

VAR

3:	[8,10	9,40	7,00]
2:	[2,27	5,80	1,14]
1:	[5,17	33,64	1,30]
TOT MEAN SDEV VAR MAXE MIN				

La variance de l'échantillon pour IPC est de 5,17 ; pour IPP de 33,64 et pour TC de 1,3.

Statistiques sur deux variables

Dans cette section vous calculerez le coefficient de corrélation et la covariance de IPC et IPP, puis vous utiliserez une régression linéaire pour prédire des valeurs de IPP à partir de celles de IPC.

Affichez la troisième ligne du menu STAT.

NEXT

3:	[8,10	9,40	7,00]
2:	[2,27	5,80	1,14]
1:	[5,17	33,64	1,30]
CORR COV LR PREDV				

Voici les commandes permettant d'effectuer les calculs de corrélation, covariance, régression linéaire et prévision.

Spécification d'une paire de colonnes

Avant d'effectuer des statistiques sur deux variables, vous devez spécifier les colonnes de la matrice de statistiques Σ DAT qui contiennent les données dépendantes et indépendantes. Dans l'exemple proposé, IPC (colonne 1) doit être la variable indépendante et IPP (colonne 2) doit être la variable dépendante.

Spécifiez les colonnes 1 et 2 comme données indépendantes et dépendantes respectivement.

1.2 COLZ
[ENTER]

3:	[8,10	9,40	7,00]
2:	[2,27	5,80	1,14]
1:	[5,17	33,64	1,30]
COLZ CORR COV LR PREDV					

Les nombres 1 et 2 sont stockés dans la liste nommée Σ PAR, une liste ordinaire portant un nom spécial. Les commandes qui effectuent des calculs statistiques sur deux variables font référence à Σ PAR.

Si vous ne spécifiez pas les colonnes des données indépendantes et dépendantes, le calculateur utilise par défaut les colonnes 1 et 2. Pour l'exemple en cours, il n'était donc pas nécessaire de spécifier les colonnes. Vous devez néanmoins utiliser COLZ si vos données indépendantes et dépendantes ne sont pas dans les colonnes 1 et 2.

Calcul de corrélation

Calculez la corrélation.

CORR

3:	[2,27	5,80	1,14]
2:	[5,17	33,64	1,30]
1:				0,96	
COLZ CORR COV LR PREDV					

La corrélation de IPC et IPP est de 0,96.

Calcul de la covariance

Calculez la covariance de l'échantillon.

COV

3:	[5,17	33,64	1,30]
2:				0,96	
1:				12,65	
COLZ CORR COV LR PREDV					

La covariance de l'échantillon de IPC et IPP est de 12,65.

Régression linéaire

Calculez la droite qui s'ajuste le mieux aux données de IPC et IPP.

LR

3:	12,65			
2:	-10,43			
1:	2,45			
COLS	CORR	COV	LR	PREDV

L'ordonnée à l'origine de la droite est $-10,43$ et sa pente est $2,45$.
L'ordonnée à l'origine et la pente sont en outre stockées dans la liste ΣPAR .

Prévision

Supposons que vous vouliez effectuer une prévision des valeurs de IPP lorsque IPC prend les valeurs 6 et 7. La valeur prévue peut être calculée à partir de la pente et de l'ordonnée à l'origine stockée dans ΣPAR .

Calculez la valeur prévue de IPP lorsque IPC prend la valeur 6.

6 PREDV

3:	-10,43			
2:	2,45			
1:	4,26			
COLS	CORR	COV	LR	PREDV

La valeur prévue est $4,26$.

Calculez la valeur prévue de IPP lorsque IPC prend la valeur 7.

7 PREDV

3:	2,45			
2:	4,26			
1:	6,71			
COLS	CORR	COV	LR	PREDV

La valeur prévue est $6,71$.

13

Arithmétique binaire

Ce chapitre indique comment effectuer des opérations arithmétiques sur des entiers binaires. Chaque entier binaire est composé de 1 à 64 bits et représente un nombre binaire sans signe. Pour faciliter la saisie des nombres binaires et la lecture des résultats, vous pouvez choisir la base décimale, hexadécimale, octale ou binaire. Mais votre choix n'affecte pas la représentation interne des entiers binaires, et les commandes opèrent sur les entiers binaires bit par bit.

L'annexe D, « Les menus et leurs jeux de libellés », décrit brièvement les commandes du menu BINARY. La section « BINARY » du Manuel de référence les présente en plus grand détail.

Choix de la taille de mot

La taille de mot en cours affecte la longueur des entiers binaires résultant des commandes et l'affichage des entiers binaires dans la pile. La taille peut varier de 1 à 64 bits, la taille par défaut étant 64 bits. Supposons que vous choisissiez une taille de mot de 16 bits.

Avant de commencer l'exemple, effacez la pile et affichez le menu BINARY.

3:
2:
1:
DEC
HEX
OCT
BIN
STWS
ROWS

Spécifiez une taille de mot de 16 bits.

16 **STWS**

3:	
2:	
1:	
DEC	HEX OCT BIN STWS RCWS

Si vous entrez maintenant un mot de plus de 16 bits, seuls les 16 bits les plus significatifs sont affichés.

Choix de la base

La base en cours affecte la valeur des entiers binaires saisis et l'affichage des entiers binaires dans la pile. Vous pouvez choisir parmi les bases décimale, hexadécimale, octale ou binaire, la base par défaut étant la base décimale.

Choisissez la base hexadécimale.

HEX

3:	
2:	
1:	
DEC	HEX OCT BIN STWS RCWS

Le libellé pour **HEX** contient maintenant un petit carré, indiquant que HEX est la base en cours.

Saisie d'entiers binaires

Entrez l'adresse $24FF_{16}$.

24FF **ENTER**

3:	
2:	
1:	# 24FFh
DEC	HEX OCT BIN STWS RCWS

Le « h » minuscule est l'indicateur de base HEX. Lorsque vous entrez un nombre, il n'est pas nécessaire de spécifier la base, sauf si le nombre n'est pas exprimé dans la base en cours.

Regardez comment cet entier binaire est représenté dans d'autres bases. Il n'est pas nécessaire de modifier l'entier binaire, seulement le mode en cours.

Passez en base DEC.

DEC

```
3:
2:
1:      # 9471d
DEC=  HEX  OCT  BIN  STWS  RCWS
```

Passez en base OCT.

OCT

```
3:
2:
1:      # 22377o
DEC  HEX  OCT=  BIN  STWS  RCWS
```

Passez en base BIN.

BIN

```
3:
2:
1:      # 1001001111111b
DEC  HEX  OCT  BIN=  STWS  RCWS
```

Retournez en base HEX.

HEX

```
3:
2:
1:      # 24FFh
DEC  HEX=  OCT  BIN  STWS  RCWS
```

Calcul avec des entiers binaires

Calculez quelle est l'adresse inférieure de $1F0_{16}$ à l'adresse précédente.

1F0

```
3:
2:
1:      # 230Fh
DEC  HEX=  OCT  BIN  STWS  RCWS
```

La différence est placée au niveau 1, comme pour les autres nombres.

Vous pouvez utiliser conjointement des entiers binaires et des nombres réels décimaux dans vos calculs. Les entiers réels entrés sans le délimiteur # sont interprétés en base 10, même lorsque la base binaire est active.

Calculez, par exemple, quelle est l'adresse inférieure de 27_{10} à l'adresse précédente.

27

```
3:
2:
1:      # 22F7h
DEC  HEX=  OCT  BIN  STWS  RCWS
```

14

Conversion d'unités

Ce chapitre vous montre comment opérer les conversions d'unités — comment convertir les valeurs numériques d'un système de mesure en unités d'un autre système. La section « UNITS » du Manuel de référence donne des informations plus détaillées sur les conversions..

Le catalogue UNITS

Le catalogue UNITS liste en ordre alphabétique toutes les unités incorporées dans le HP-28S. Il vous permet de vérifier l'orthographe et la définition des unités.

Effacez tout d'abord la pile et sélectionnez le mode d'affichage numérique STD.



3:
2:
1:
STD FIN SCI ENG DEG RAD



Lancez le catalogue UNITS.





a are
100
m^2
NEXT PREV . FETCH QUIT

La première unité, « are », est abrégée « a », sa valeur est de 100 m².

Les touches  et  du menu (pas les touches permanentes du clavier) vous permettent de parcourir le catalogue.

Vous pouvez aller directement à la première unité commençant par une lettre donnée en appuyant sur la touche de cette lettre.

[S]

```
s second
1
S
NEXT PREV      FETCH QUIT
```

La rubrique de la commande « second » indique que l'abréviation correcte est « s » et que la valeur est 1 seconde. L'unité « second » est définie en fonction d'elle-même car c'est une unité fondamentale.

Veillez à utiliser les abréviations exactement comme elles apparaissent dans le catalogue UNITS. Le HP-28S reconnaît, par exemple, le « s » (minuscule) comme abréviation de seconde, mais pas le « S » (majuscule).

Consultez la rubrique de « day » (jour).

[D]

```
d day
86400
S
NEXT PREV      FETCH QUIT
```

Cette rubrique indique que l'abréviation correcte est « d » et que sa valeur est 86 400 secondes.

Consultez ensuite la rubrique de l'unité « foot » (pied).

[F]

```
F farad
1
A^2*s^4/kg*m^2
NEXT PREV      FETCH QUIT
```

Le catalogue liste d'abord la rubrique de « farad ». Avancez de sept rubriques dans le catalogue.

NEXT NEXT NEXT NEXT
NEXT NEXT NEXT

```
ft int'l foot
.3048
M
NEXT PREV      FETCH QUIT
```

Le catalogue montre maintenant la rubrique « international foot » (pied international) ; l'unité suivante est le « survey foot » utilisé comme mesure topographique de l'autre côté de l'Atlantique.

Vous pouvez écrire l'abréviation du « international foot » dans la ligne de commande.

FETCH

```
2:
1:
ft
STD FIN SCI ENG DEG RAD
```

Le calculateur retourne à l'affichage normal et la ligne de commande contient l'abréviation de l'unité.

Les exemples de ce chapitre vous montrent comment saisir directement des unités, mais vous pouvez aussi utiliser **UNIT** et **FETCH**, si vous préférez.

Effacez la ligne de commande.

ON

```
3:
2:
1:
STO= FIX SCI ENG DEG RAD=
```

Conversion d'unités

Convertissez tout d'abord 15 °C en degrés Fahrenheit.

Placez la valeur numérique dans la pile.

15 **ENTER**

```
3:
2:
1: 15
STO= FIX SCI ENG DEG RAD=
```

Entrez l'abréviation des « degrés Celsius ».

° C **ENTER**

```
3:
2:
1: °C 15
STO= FIX SCI ENG DEG RAD=
```

L'abréviation d'unité est convertie en un nom.

Entrez l'abréviation des « degrés Fahrenheit ».

° F **ENTER**

```
3:
2:
1: °F 15
STO= FIX SCI ENG DEG RAD=
```


L'abréviation d'unité est convertie en un nom.

Convertissez la valeur numérique de l'ancienne unité dans la nouvelle.

CONVERT

```
3:
2:
1: °F 59
STO= FIX SCI ENG DEG RAD=
```

Le résultat indique que 15 °C est équivalent à 59 °F.

Convertissez maintenant 40 pouces en millimètres. Cette fois-ci, laissez  **CONVERT** exécuter automatiquement ENTER pour vous.

Effacez la pile et entrez la valeur numérique.

 **CLEAR**
40 **ENTER**



3:
2:
1: 40
STD= FIX SCI ENG DEG RAD=

Entrez l'abréviation des pouces, (inches).

LC in **ENTER**



3:
2: 40
1: 'in'
STD= FIX SCI ENG DEG RAD=

Entrez l'abréviation des « millimètres » et effectuez la conversion.

L'unité « millimètre » ne se trouve pas dans le catalogue UNITS, car c'est une unité *préfixée*—unité « m » (mètre) préfixée par « m » (millième). De la même façon, « km » est l'abréviation préfixée correspondant au kilomètre et « ms » est celle de la milliseconde. Une liste complète des préfixes est donnée dans la section « UNITS » du *Manuel de référence*.

LC mm  **CONVERT**



3:
2: 1016
1: 'mm'
STD= FIX SCI ENG DEG RAD=

Le résultat indique que 40 pouces est équivalent à 1 016 millimètres.

Conversion des chaînes d'unités

Les chaînes sont en fait pour le HP-28S des objets qui contiennent des caractères. Vous pouvez utiliser des *chaînes d'unités* pour définir des unités plus complexes que celles utilisées jusqu'à présent.

Une chaîne d'unité peut représenter une unité élevée à une puissance, par exemple, « m^2 », ou le produit de deux unités, par exemple « m*kg », ou une combinaison de puissances et de produits.

Une chaîne d'unité peut aussi représenter un quotient, par exemple « m/s ». Le symbole / ne peut apparaître plus d'une fois. Groupez toutes les unités directes devant le symbole / et toutes les unités inverses après le /. Par exemple, « mètre par seconde par seconde » est représenté par « m/s^2 ».

Convertissez maintenant 1 kilomètre par heure en mètres par seconde.

Effacez la pile et entrez la valeur numérique.

■ CLEAR

1 ENTER

```

3:
2:
1: 1
STD= FIX SCI ENG DEG RAD=
  
```

Entrez l'abréviation pour « kilomètre par heure ».

LC " km ÷ h ENTER

```

3:
2:
1: "km / h" 1
STD= FIX SCI ENG DEG RAD=
  
```

Entrez l'abréviation pour « mètres par seconde ».

Il n'y a pas d'unité intégrée pour « mètres par seconde », vous devez donc utiliser une chaîne d'unité.

■ " LC m ÷ s

```

2:
1: "km / h" 1
"m / s"
STD= FIX SCI ENG DEG RAD=
  
```

Le calculateur active le mode de saisie alpha (indiqué par la forme du curseur) lorsque vous commencez à entrer la chaîne. En mode alpha, toutes les commandes sont écrites dans la ligne de commande, il est donc utile d'appuyer sur ENTER pour terminer la chaîne.

ENTER

```

3:
2: "km / h" 1
1: "m / s"
STD= FIX SCI ENG DEG RAD=
  
```

Convertissez la valeur de l'ancienne unité dans la nouvelle.

■ CONVERT

```

3:
2: ,277777777778
1: "m / s"
STD= FIX SCI ENG DEG RAD=
  
```

Le résultat indique que 1 kilomètre par heure est équivalent à 0,27777777778 mètres par seconde.

Convertissez maintenant 10 pieds cubes en gallons.

Effacez la pile et entrez la valeur numérique.

■ **CLEAR**
10 **ENTER**

```
3:
2:
1: 10
STD= FIX SCI ENG DEG RAD=
```

Entrez la chaîne d'unité pour « pieds cubes ».

■ " **LC** ft ■ ^ 3 **ENTER**

```
3:
2:
1: "ft ^ 3" 10
STD= FIX SCI ENG DEG RAD=
```

Entrez l'abréviation des « gallons US » et effectuez la conversion.

LC gal ■ **CONVERT**

```
3:
2: 74,8051948052
1: 'gal'
STD= FIX SCI ENG DEG RAD=
```

Le résultat indique que 10 pieds cubes est équivalent à 74,8051948052 gallons.

Vérification des unités

L'utilisation d'unités incorrectes peut donner des résultats aberrants ou des erreurs **Inconsistent Units** (unités incompatibles). La solution, dans les deux cas, consiste à consulter le catalogue **UNITS** ou la section « **UNITS** » du Manuel de référence.

Des résultats numériques aberrants peuvent se produire lorsque vous utilisez une unité de dimension correcte, mais de valeur numérique incorrecte. Si vous voulez, par exemple, convertir un acre (unité de mesure de surface américaine valant 40 ares en pieds carrés ; le résultat est plus grand que 43 560. Ceci se produit parce qu'il existe deux unités de pieds, le « ft » (pied international) et le « ftUS » (unité de topographie américaine). La conversion d'un acre en « ftUS » donne exactement 43 560.

Un message **Inconsistent Units** survient lorsque vous utilisez une unité avec des dimensions incorrectes, par exemple, si vous utilisez « g » (gramme) comme unité de force. L'unité correcte est le « gf » (gramme-force).

Fonctions utilisateur pour conversion d'unités

Si vous effectuez fréquemment certaines conversions d'unités, vous pouvez écrire des fonctions utilisateurs pour ces conversions. Au cours de cette section, vous allez écrire les fonctions utilisateurs O→G et G→O pour convertir des onces en grammes et réciproquement. Vous pouvez les utiliser en notation RPN ou algébrique car ce sont des fonctions utilisateur.

Souvenez-vous que les fonctions utilisateur doivent satisfaire à deux conditions :

- Elles doivent spécifier leurs arguments de façon explicite.
- Elles doivent donner un seul résultat.

Ecrivez tout d'abord O→G.

Commencez le programme et spécifiez l'argument.

⌈ ⌋ ⌈ → ⌋ LC x

```
2: 74,8051948052
1: 'gal'
⌈ → ⌋ x ⌈
```

STD= FIX SCI ENG DEG RAD=

La touche vers la droite indique que le nom suivant est une *variable locale*, limitée à ce programme.

Définissez la conversion.

⌈ x ⌈ ' oz ⌈ ' ' g ⌈ ⌈ CONVERT
DROP ENTER

```
2: 'gal'
1: ⌈ → ⌋ x ⌈ x ' oz' 'g'
CONVERT DROP ⌈ ⌈
```

STD= FIX SCI ENG DEG RAD=

Le calculateur ajoute les délimiteurs de clôture pour vous.

Ce programme prend un argument dans la pile (syntaxe RPN) ou dans l'expression (syntaxe algébrique) et l'appelle x ; convertit x d'onces en grammes et élimine l'unité gramme de la pile.

Stockez le programme dans une variable nommée O→G.

' O ⌈ → ⌋ G STO

```
3:
2: 74,8051948052
1: 'gal'
```

STD= FIX SCI ENG DEG RAD=

Ecrivez maintenant $G \rightarrow O$.

Commencez le programme et spécifiez l'argument.

\leftarrow \blacksquare \rightarrow LC x

```

2:      74,8051948052
1:      'gal'
 $\leftarrow$   $\rightarrow$  x  $\blacksquare$ 
STO= FIX SCI ENG DEG RAD=
  
```

Définissez la conversion.

\leftarrow x ' g ' ' oz ' \blacksquare CONVERT
DROP ENTER

```

2:      'gal'
1:       $\leftarrow$   $\rightarrow$  x  $\leftarrow$  x 'g' 'oz'
      CONVERT DROP  $\leftarrow$   $\leftarrow$ 
STO= FIX SCI ENG DEG RAD=
  
```

Ce programme prend un argument dans la pile (syntaxe RPN) ou dans l'expression (syntaxe algébrique) et l'appelle x ; convertit x de grammes en onces et élimine l'unité onces de la pile.

Stockez le programme dans une variable nommée $G \rightarrow O$.

' G \blacksquare \rightarrow O STO

```

3:
2:      74,8051948052
1:      'gal'
STO= FIX SCI ENG DEG RAD=
  
```

Pour tester les conversions, calculez le nombre de grammes dans une once et convertissez le résultat en once. Le résultat doit être 1.

Convertissez 1 once en grammes.

1 USER \blacksquare $G \rightarrow O$

```

3:      74,8051948052
2:      'gal'
1:      28,349523125
G $\rightarrow$ O O $\rightarrow$ G  $\Sigma$ FAR  $\Sigma$ ORT VA VF
  
```

Il y a environ 28 grammes dans une once. Convertissez maintenant ce résultat en once.

\blacksquare $G \rightarrow O$

```

3:      74,8051948052
2:      'gal'
1:      1
G $\rightarrow$ O O $\rightarrow$ G  $\Sigma$ FAR  $\Sigma$ ORT VA VF
  
```

Les résultat est 1, ce qui est correct.

Impression

Ce chapitre décrit quelques commandes de base pour l'utilisation du HP-28S avec une imprimante HP 82240A. Le manuel d'utilisation de l'imprimante présente le positionnement relatif de l'imprimante et du HP-28S, et la mise en marche de l'imprimante.

L'annexe D, « Les menus et leurs jeux de libellés », décrit brièvement les commandes du menu PRINT. La section « PRINT » du Manuel de référence donne des informations plus détaillées.

Impression de l'affichage

Vous pouvez imprimer une image de l'affichage de la façon suivante.

1. Appuyez sur **[ON]** et maintenez la touche enfoncée.
2. Appuyez sur **[L]** (la touche au-dessus de laquelle est imprimé « PRINT »).
3. Relâchez **[ON]**.

Cette séquence de touches est l'équivalent sur le clavier de la commande PRLCD (*print LCD*, imprimer l'écran), située dans la première ligne du menu PRINT. Vous pouvez utiliser ces séquences pour imprimer l'affichage presque à tout moment, sans perturber le fonctionnement du calculateur.

Si vous voulez qu'un programme imprime l'affichage, utilisez simplement la commande PRLCD du menu PRINT.

Effacez la pile et effacez le menu PRINT.



- PR1 (*print 1*) imprime l'objet au niveau 1.
- PRST (*print stack*) imprime tous les objets de la pile.
- PRVAR (*print variable*) imprime le nom et le contenu d'une variable.
- PRLCD (*print LCD*) imprime l'affichage.
- CR (*carriage right*) imprime une ligne blanche.
- TRAC (*trace on/off*) active ou désactive le mode d'impression Auto.

Impression d'un relevé continu

Pour imprimer un relevé continu de vos calculs, activez le mode d'impression Auto.

TRAC

```
3:
2:
1:
PR1 PRST PRVAR PRLCD CR TRAC=
```

Un carré apparaît dans le libellé TRAC pour indiquer que le mode d'impression Auto est actif.

Additionnez maintenant deux nombres pour voir ce qui se passe. Placez tout d'abord 44 dans la pile.

44 ENTER

```
44 ENTER
1: 44
```

Le calculateur imprime l'entrée et le résultat du niveau 1.

Ajoutez maintenant 72.

72 +

```
72 +
1: 116
```

Le calculateur imprime à nouveau l'entrée et le niveau 1.

Désactivez le mode d'impression Auto.

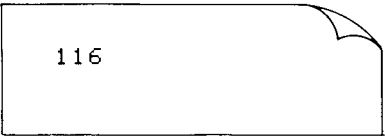
TRAC

```
3:
2:
1: 116
PR1 PRST PRVAR PRLCD CR TRAC=
```

Impression du niveau 1

Au lieu d'imprimer tous les résultats avec le mode d'impression Auto, vous pouvez imprimer des résultats de façon sélective en utilisant PR1.

PR1



Le résultat reste au niveau 1, inchangé.

Vous pouvez imprimer un message en plaçant une chaîne au niveau 1. Pour imprimer le message « OK », par exemple, placez la chaîne dans la pile.

■ " OK ENTER



Imprimez maintenant le message.

PR1

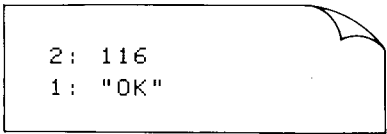


Seul le contenu de la chaîne est imprimé, pas les guillemets.

Impression de la pile

Vous pouvez imprimer tous les objets de la pile en utilisant PRST.

 **PRST**



2: 116
1: "OK"

Le contenu de la pile est inchangé.

Impression d'une variable

Vous pouvez imprimer le nom et le contenu d'une variable sans la rappeler dans la pile. Stockez, par exemple, la chaîne « OK » dans une variable nommée « A », puis imprimez la variable A.

Créez la variable A avec la valeur « OK ».

 A **STO**



3:
2:
1: 116
PR1 PRST PRVAR PRLCD CR TRAC

Imprimez le nom et la valeur de la variable.

 A **PRVAR**



A
"OK"

Le calculateur élimine de la pile nom de la variable.

2ème partie

Résumé des caractéristiques du calculateur

Page 154	16 : Objets
164	17 : Opérations, commandes et fonctions
166	18 : Ligne de commande
176	19 : Pile opérationnelle
182	20 : Mémoire
192	21 : Menus
196	22 : Catalogue des commandes
198	23 : Evaluation
205	24 : Modes
215	25 : Opérations au niveau du système

Objets

La première partie de ce manuel contient des exemples des 10 types d'objet du HP-28S. Les objets sont les entités de base dans le calculateur, celles qui servent à formuler des problèmes et à trouver des solutions.

La plupart des objets ont pour but de faciliter votre travail en vous fournissant des types de données spécifiques. Imaginez si vous deviez représenter des tableaux par des nombres réels, en essayant de vous souvenir de la place de chacun, et si vous deviez écrire des programmes pour effectuer des calculs sur ces tableaux. Il est bien plus simple d'utiliser le HP-28S pour entrer les valeurs sous forme d'un objet de type tableau, que vous pouvez ensuite manipuler comme une entité unique, et sur lequel vous pouvez effectuer des calculs avec les fonctions arithmétiques normales.

L'idée d'utiliser différents types d'objet est née en outre des besoins de la programmation et du calcul symbolique : objets programmes et objets symboliques (noms et objets algébriques). Ces objets ne sont pas simplement des données car ils peuvent être évalués de façon à donner un résultat. (L'évaluation des objets est présentée au chapitre 23).

L'utilisation de multiples types de données, des opérations symboliques et de la programmation avec des objets, permet de minimiser le nombre de règles dont vous devez vous souvenir. Les objets sont tous entrés dans la ligne de commande, placés dans la pile ou stockés dans des variables, de la même façon, quel que soit le type de l'objet.

Le présent chapitre résume les informations importantes concernant chaque type d'objet et suggère des utilisations possibles.

Nombres réels

Les nombres réels représentent des nombres supérieurs à -10^{500} et inférieurs à 10^{500} . Ils sont stockés de façon interne sous forme d'une *mantisse* entre 1 et 9,999999999999, un signe (positif ou négatif) pour la mantisse, un *exposant* entre 0 et 499 et un signe pour l'exposant.

Format Heures-minutes-secondes. Vous pouvez utiliser les commandes HMS+ et HMS- pour ajouter et soustraire des nombres exprimés sous la forme d'heures, minutes et secondes (ou degrés, minutes et secondes). Pour tout autre calcul que l'addition et la soustraction, vous devez tout d'abord utiliser HMS→ pour convertir les nombres du format HMS au format décimal. (Consultez « TRIG » dans le Manuel de référence.)

Nombres complexes

Les nombres complexes sont des paires ordonnées de nombres réels qui représentent la *partie réelle* et la *partie imaginaire* d'un nombre complexe, ou les coordonnées d'un point sur un plan.

Coordonnées rectangulaires et polaires. Aux chapitres 7 et 8 vous avez utilisé des nombres complexes pour effectuer des traçage et des numérisations. Chaque nombre complexe représentait des *coordonnées rectangulaires*, c'est-à-dire l'abscisse et l'ordonnée du point sur des axes perpendiculaires.

Le chapitre 6 décrit les *coordonnées polaires*, module et argument du vecteur, et présente les commandes R→P et P→R permettant d'effectuer des conversions entre les coordonnées polaires et rectangulaires. Vous pouvez utiliser les coordonnées polaires pour entrer des coordonnées et afficher des résultats, mais vous devez les convertir en coordonnées rectangulaires avant tout calcul.

Objets algébriques. Lorsque vous entrez un nombre complexe dans un objet algébrique, vous devez utiliser deux jeux de parenthèses, comme dans l'expression 'SIN((0.1))'. La paire de parenthèses extérieures est requise par la fonction SIN(), et la paire intérieure représente les délimiteurs des nombres complexes.

Entiers binaires

Les entiers binaires représentent une séquence de bits. La longueur de la séquence, entre 1 et 64 bits, dépend de la taille de mot en cours. La base d'affichage des entiers binaires détermine la façon dont ils apparaissent, mais n'a aucun effet sur leur représentation interne.

Entiers de grande dimension. L'utilisation des entiers binaires avec affichage en base décimale permet d'exprimer de façon exacte un entier positif de 19 chiffres, c'est-à-dire 7 chiffres de plus que pour les nombres réels.

Exemple de programmation. Les programmes de la section « Affichage d'un entier binaire », en page 257, permettent d'afficher un entier binaire dans chacune des quatre bases.

Enregistrement de l'environnement. La commande RCLF (*recall flags*, enregistrement des indicateurs d'état) donne un entier binaire représentant l'état des 64 indicateurs utilisateur ; la commande STOF (*store flags*, stockage des indicateurs) définit les indicateurs utilisateur conformément à un entier binaire utilisé comme argument (voir le programme « PRESERVE »).

Chaînes

Une chaîne se compose d'une séquence de caractères. La première partie a illustré les utilisations suivantes des chaînes.

- Au chapitre 14, « Conversion d'unités », vous avez utilisé des chaînes pour représenter une combinaison de produits d'unité et d'élévation d'unité à une puissance.
- Au chapitre 15, « Impression », vous avez entré un message sous forme d'une chaîne pour l'imprimer. Vous pouvez aussi afficher des messages avec la commande DISP décrite au chapitre 27, « Programmes interactifs ».

Une chaîne représente le plus souvent du texte, mais chaque caractère peut aussi représenter une valeur numérique comprise entre 0 et 255. Les commandes CHR (*character*, caractère) et NUM (*numéral*) effectuent des conversions entre les caractères et leurs valeurs numériques.

Caractères non présents sur le clavier. Vous pouvez entrer une valeur numérique et appuyer sur CHR pour afficher certains caractères qui ne sont pas présents sur le clavier du HP-28S. Vous pouvez en outre imprimer certains caractères qui ne peuvent pas être imprimés. La section « STRING » du Manuel de référence contient une liste de tous les caractères.

Chaînes graphiques. La commande LCD→ (*LCD dans chaîne*) donne une chaîne graphique représentant l'image affichée ; la commande →LCD (*chaîne dans LCD*) affiche une image représentée par une chaîne graphique utilisée comme argument. Consultez le Manuel de référence pour plus de détails sur ces commandes.

Manipulations des chaînes. Les programmes de la section « Affichage d'un entier binaire », en page 257, illustrent la conversion d'un objet sous forme de chaîne, le compte du nombre de caractères et la fusion de deux chaînes.

Tableaux

Les tableaux ayant une seule dimension sont appelés *vecteurs*, et ceux ayant deux dimensions sont appelés *matrices*. Ils peuvent contenir des nombres réels ou complexes. Le chapitre 11, « Vecteurs et matrices », illustre les calculs de base sur les tableaux. La première partie du manuel décrit en outre les utilisations suivantes des tableaux.

- Le chapitre 11 illustre la solution d'un système linéaire de n équations à n inconnues en utilisant un vecteur constant de n éléments et une matrice coefficient $n \times n$. Consultez la section « ARRAY » du Manuel de référence pour plus de détail.
- Le chapitre 12, « Statistiques », illustre l'utilisation de la matrice ΣDAT pour stocker les données de statistiques.

En syntaxe algébrique. Si un tableau est stocké dans une variable, vous pouvez faire référence aux éléments du tableau en utilisant le nom de la variable comme fonction. Vous pouvez, par exemple, représenter la somme des troisième et cinquième éléments d'un vecteur V sous la forme ' $V(3)+V(5)$ '.

Manipulations de tableau. Les programmes des sections « Sommations », en page 262, et « Médiane de données statistiques », en page 270, illustrent la diversité des manipulations de tableau.

Listes

Les listes sont des séquences d'objets ; elles représentent la méthode la plus commune pour combiner plusieurs objets en un seul. La première partie du manuel a présenté les utilisations suivantes des listes.

- Au chapitre 4, « Calculs répétitifs », la commande PATH donne une liste des répertoires de noms, du répertoire HOME au répertoire en cours.
- Au chapitre 7, « Traçage », la variable de liste PPAR contient les paramètres utilisés par DRAW.
- Au chapitre 8, « Résolution d'équations », vous avez utilisé comme estimation une liste contenant trois points numérisés.
- Au chapitre 10, « Calcul différentiel et intégral », vous avez spécifié la variable d'intégration ainsi que les limites d'intégration supérieure et inférieure en les combinant dans une liste.
- Au chapitre 12, « Statistiques », vous avez utilisé la variable de liste ΣPAR pour stocker les paramètres des statistiques sur deux variables.

En syntaxe algébrique. Si une liste est stockée dans une variable, vous pouvez faire référence à des éléments de la liste en utilisant le nom de la variable comme une fonction. Vous pouvez, par exemple, représenter la somme des troisième et cinquième éléments d'une liste L sous la forme ' $L(3)+L(5)$ '.

Listes et pile opérationnelle. Le programme MEDIAN, page 273, montre comment placer les éléments d'une liste dans la pile et combiner des objets de la pile dans une liste.

Tri d'une liste. Le programme SORT, page 270, montre comment trier les éléments d'une liste.

Extraction des éléments d'une liste. Le programme LMED, page 272, montre comment extraire des éléments d'une liste.

Noms

Les noms sont des séquences de caractères permettant de nommer d'autres objets. Les noms peuvent avoir un maximum de 127 caractères, bien que des considérations d'ordre pratique suggèrent de limiter la longueur des noms à cinq ou six caractères.

Les caractères disponibles sur le clavier sont les lettres, les chiffres et $\pi \rightarrow \mu$. Le premier caractère ne peut pas être un chiffre. Les caractères suivants ne peuvent pas être utilisés dans des noms.

- Les caractères séparant des objets : délimiteurs (# [] " ' { } () « »), espace, point ou virgule.
- Symboles d'opérateurs algébriques (+ - * / ^ √ = < > ≤ ≥ ≠ ÷ ∫)

Le calculateur détermine si le nom est local ou global lors du traitement de la ligne de calcul : si le nom est utilisé par une structure de programme pour créer une variable locale, le nom est local à cette structure de programme ; sinon, le nom est global.

Noms locaux. Au cours de la première partie, vous avez rédigé des fonctions utilisateur qui créent des variables locales. Tout au long du manuel, nous avons utilisé des lettres minuscules pour les noms locaux de façon à les différencier des noms globaux. Il est important de se souvenir que c'est la commande \rightarrow qui a rendu les noms locaux, et non pas l'utilisation de minuscules. Si vous nommez une variable locale e ou i , la définition locale remplace la définition intégrée.

Noms globaux. Tous les autres noms dans la première partie du manuel étaient de type global. Par exemple :

- Les noms des variables globales (variables numériques utilisées pour le traçage ou l'algorithme de résolution d'équations ; toutes les variables du menu USER).
- Les noms des répertoires.
- Les noms utilisés de façon symbolique, sans référence à des valeurs spécifiques (arithmétique symbolique, solutions symboliques et calcul différentiel et intégral).

Les noms des commandes, y compris e , i et π , ne peuvent pas être utilisés comme noms globaux.

Les noms suivants sont réservés pour des utilisations spécifiques :

- EQ, équation en cours utilisée par les commandes de l'algorithme de résolution d'équations et de PLOT.
- ΣPAR, liste de paramètres pour les commandes statistiques.
- PPAR, liste de paramètres pour les commandes de traçage.
- ΣDAT, tableau statistique en cours.
- s1, s2 etc., créés par ISOL et QUAD pour représenter des signes arbitraires obtenus dans des solutions symboliques.
- n1, n2 etc., sont créés par ISOL pour représenter des entiers arbitraires obtenus dans des solutions symboliques.
- Les noms commençant par « der » font référence aux dérivées définies par l'utilisateur.

Vous pouvez utiliser ces noms pour vos propres besoins, mais certaines commandes les utilisent comme argument implicite.

Programmes

Les programmes sont des séquences d'objets et de commandes. Chaque programme est essentiellement une ligne de commande transformée en un objet ; entourez la ligne de commande de délimiteurs de programme pour indiquer que vous voulez en préserver le contenu pour une exécution ultérieure.

Les menus PROGRAM BRANCH, PROGRAM CONTROL et PROGRAM TEST contiennent des commandes spéciales. Ces menus sont présentés dans la section « Programmes » du Manuel de référence.

Vous avez écrit cinq programmes au cours de la 1^{ère} partie du manuel :

- Au chapitre 3, vous avez écrit un programme pour renommer des variables et vous l'avez stocké dans la variable RENOMMER.
- Au chapitre 5, vous avez écrit un programme pour calculer une cotangente, et vous l'avez stocké dans la variable COT.
- Au chapitre 6, vous avez écrit un programme permettant d'ajouter des coordonnées polaires, et vous l'avez stocké dans la variable PSUM.
- Au chapitre 14, vous avez écrit des programmes permettant d'effectuer des conversions entre les onces et les grammes, et vous les avez stockés dans les variables O→G et G→O.

Fonctions utilisateurs. Les programmes COT, PSUM, O→G et G→O sont des fonctions utilisateur—they commencent par la commande → et un ou plusieurs noms, qui définissent une ou plusieurs variables, suivie par une expression ou un programme. Lorsque la fonction utilisateur est stockée dans une variable, vous pouvez utiliser le nom de cette dernière dans la syntaxe algébrique de la même façon qu'une fonction intégrée.

Structures de programmes. La commande → suivie de noms et d'une expression ou d'un programme est appelée une *structure de variable locale*, une des structures de programme. Il existe en outre des structures de programme pour le branchement (telles que IF ... THEN ... ELSE ... END) et les boucles (telles que DO ... UNTIL ... END). Consultez les descriptions au chapitre 26, « Structures de programmes », et les exemples au chapitre 28.

Programmes sans nom. Il n'est pas toujours nécessaires de stocker les programmes dans des variables pour qu'ils puissent être utilisés (voir la section « Développement et regroupement complets », en page 253, ainsi que « Affichage d'un entier binaire », en page 257).

Objets algébriques

Les objets algébriques regroupent une ou plusieurs fonctions ainsi que leurs arguments qui peuvent être des nombres, des noms ou des sous-expressions. Les objets algébriques sont écrits et affichés conformément à la syntaxe algébrique, similaire à la notation mathématique écrite. Ils sont de deux types : les expressions et les équations.

Expressions

Au cours de la première partie du manuel, vous avez utilisé des expressions de trois façons différentes : sous forme de données, d'expressions et d'équations implicites.

Expressions en tant que données. Lorsque vous effectuez des calculs avec des expressions, tel que l'addition de deux expressions, l'élévation d'une expression au carré, ou le calcul de la dérivée d'une expression, le résultat est une nouvelle expression. Dans ces cas, les expressions sont interprétées comme des données à manipuler, indépendantes de toute valeur affectées aux variables.

Expressions en tant que fonctions. Au chapitre 4, vous avez créé l'expression RTOT, vous avez affecté des valeurs aux variables, et vous avez utilisé l'algorithme de résolution d'équations pour évaluer RTOT de façon à calculer le résultat. Dans ce cas, l'expression a été interprétée en tant que fonction, qui utilise les données entrées pour calculer un résultat.

Expressions en tant qu'équations implicites. Au chapitre 8, vous avez utilisé l'algorithme de résolution d'équations pour chercher le zéro *numérique* d'une expression—c'est-à-dire la valeur numérique de la variable indépendante pour laquelle l'expression prend la valeur 0. Au chapitre 9, vous avez utilisé QUAD pour trouver le zéro *symbolique*—c'est-à-dire une expression qui, substituée à la variable indépendante, donne à l'expression originale la valeur 0.

Dans les deux cas, l'expression $f(x)$ est interprétée comme l'équation $f(x) = 0$, car le zéro de l'expression est la *racine* de l'équation.

Equations

Les équations se composent de deux expressions reliées par un signe égal (=). Les mathématiques utilisent le signe égal de deux façons :

- Pour indiquer une proposition, telle que « $x^2 = 4$ » ou « $x^2 + y^2 = 1$ ». Dans ce cas, l'équation n'est valide que pour certaines valeurs des variables.
- Pour indiquer une identité ou une définition, telle que « $\sin 2x = 2 \sin x \cos x$ » ou « $y = 3x^2 + 2x + 5$ ». Dans ce cas, l'équation est valide quelles que soient les valeurs des variables.

Dans le HP-28S, les équations ne sont utilisées que pour les propositions ; pour créer une définition telle que « $y = 3x^2 + 2x + 5$ », l'expression ' $3*X^2+2*X+5$ ' est *stockée dans une variable* nommée Y.

Dans la section « Calculs sur des flux financiers constants » en page 103, FCST et VAPU sont exprimés mathématiquement sous forme d'équations. L'équation FCST, qui n'est valide que pour certaines valeurs de ses variables, est entrée sous forme d'équation ; mais VAPU, dont la valeur est définie quelles que soient les valeurs de ses variables, est créée en tant que variable.

Equations en tant que données. Lorsque vous effectuez des calculs avec des équations, tels que l'addition de deux équations, l'élévation d'une équation au carré, ou le calcul de la dérivée d'une équation, le résultat est une autre équation. Chaque membre de l'équation est traité indépendamment—chaque membre d'une expression est traitée en tant que donnée. L'équation conserve sa nature de proposition, selon laquelle elle n'est valide que pour certaines valeurs de ses variables.

Résolution d'équations. Lorsque vous résolvez une équation de façon numérique, ainsi que vous l'avez fait dans la section « Calculs sur des flux financiers constants », vous trouvez la valeur de la variable indépendante qui satisfait l'égalité. De la même façon, lorsque vous résolvez une équation de façon symbolique, comme vous l'avez fait dans la section « Isolation d'une variable » en page 109, vous trouvez une expression qui, substituée à la variable indépendante, satisfait l'équation.

Constantes symboliques

Les objets algébriques peuvent contenir les constantes symboliques suivantes. Ces constantes ressemblent à des noms mais sont en fait des fonctions.

- MINR (*minimum réel*) représente le plus petit réel positif. Sa valeur numérique est 1,00000000000E-499.
- MAXR (*maximum réel*) représente le plus grand réel positif. Sa valeur numérique est 9,99999999999E499.
- e représente la base des logarithmes népériens. Sa valeur numérique dans le HP-28S est 2,71828182846.
- π représente le rapport de la circonférence d'un cercle sur son diamètre. Sa valeur numérique dans le HP-28S est 3,14159265359.
- i représente le nombre imaginaire $\sqrt{-1}$. Sa valeur numérique est (0, 1).

Dans les modes Constantes numériques et Résultat numérique, l'évaluation des constantes symboliques donne leurs valeurs numériques ; sinon, l'évaluation donne leurs formes symboliques (ces modes sont décrits au chapitre 24).

17

Opérations, commandes et fonctions

Chaque procédure intégrée au HP-28S peut être classée dans l'une des catégories suivantes : opération, commande, fonction ou fonction analytique.

- Une *opération* est une procédure quelconque intégrée au calculateur.
- Une *commande* est une opération programmable.
- Une *fonction* est une commande autorisées dans les calculs algébriques.
- Une *fonction analytique* est une fonction pour laquelle le HP-28S fournit une dérivée et un inverse.

Le tableau suivant donne des exemples de chaque type.

Opérations

Opérations Non programmables	Commandes		
	Comman- des RPN	Fonctions	
		Non- analytiques	Analytiques
<div>INS</div> <div>NEXT</div> <div>EDIT</div> <div>VIEW↑</div> <div>ENTER</div> <div>EEX</div> <div>COMMAND</div> <div>UNDO</div> <div>CONT</div> <div>ON</div>	<div>SWAP</div> <div>DROP</div> <div>LAST</div> <div>RCL</div> <div>PURGE</div> <div>∫</div> <div>STO</div> <div>EVAL</div> <div>CLEAR</div> <div>CONVERT</div>	<div>ABS</div> <div>∂</div> <div>IP</div> <div>MAX</div> <div>OR</div> <div>%CH</div> <div>R→D</div> <div>R→P</div> <div>XPON</div> <div>≠</div>	<div>ASIN</div> <div>EXP</div> <div>INV</div> <div>LN</div> <div>NEG</div> <div>SIN</div> <div>SINH</div> <div>SQ</div> <div>+</div> <div>=</div>



Les procédures intégrées sont généralement caractérisées par la catégorie la plus haute à laquelle elles appartiennent. SWAP et IP, par exemple, sont toutes deux des commandes, mais SWAP est uniquement une commande alors que IP est aussi une fonction, on appelle donc IP une fonction.

L'index des opérations à la fin du Manuel de référence identifie la catégorie à laquelle appartient chaque procédure. Voici quelques commentaires concernant chaque catégorie :

- La plupart des opérations non-programmable ne peuvent être exécutées qu'en appuyant sur une touche. Certaines d'entre elles ont néanmoins des équivalents programmable : l'opération **TRIG**, par exemple, (sélection du menu TRIG) peut être utilisée dans un programme en exécutant **21 MENU**, et l'opération **RAD** (sélection du mode d'angle Radians) peut être utilisée en exécutant **60 FS**.
- La plupart des commandes RPN effectuent des manipulations de la pile ou modifient la mémoire utilisateur au lieu d'effectuer des calculs sur des valeurs numériques.
- La plupart des fonctions non analytiques sont des calculs mathématiques sans inverses—c'est-à-dire qu'elles renvoient certaines caractéristiques des arguments, mais que ces derniers ne peuvent pas être reconstitués à partir du résultat. Parmi ces fonctions se trouvent la partie entière, la valeur absolue et le signe.
- En mathématiques, une fonction de variables complexes est *analytique* si elle peut être exprimée sous forme d'une série de puissances en tout point de son domaine ; dans ce cas, elle a un inverse et une dérivée. Le HP-28S possèdent peu d'exceptions à cette règle. Il n'y a, par exemple, pas de dérivée pour la commande **Σ**, bien qu'il soit possible d'en calculer une ; une dérivée est fournie pour la fonction **ABS**, bien qu'elle soit non analytique au point $0 + 0i$.

Toutes les procédures intégrées sont disponibles par une touche, soit sur le clavier soit dans un menu. Lorsque vous appuyez sur une touche, le résultat exacte dépend du type de procédure et du *mode de saisie* (voir chapitre suivant).

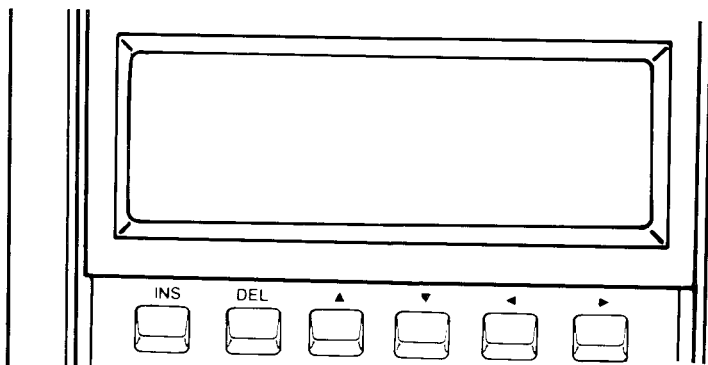
La ligne de commande

La ligne de commande contient un nombre quelconque de caractères représentant des objets sous forme textuelle. Elle apparaît en bas de l'affichage (juste au-dessus des libellés de menu, si présents) lorsque vous commencez à saisir un objet, ou lorsque vous utilisez  **EDIT** ou  **VISIT** pour modifier le contenu d'un objet existant.

La ligne de commande peut contenir plusieurs lignes de texte. Si vous entrez plus de 23 caractères sur une ligne, les caractères défilent vers la gauche. Trois points (...) apparaissent à l'extrémité gauche de l'affichage pour indiquer que des caractères sont masqués. Si vous essayez de déplacer le curseur au-delà de l'extrémité gauche de l'affichage, les caractères masqués réapparaissent, et les caractères les plus à droite défilent en dehors de l'écran. Trois points apparaissent alors à l'extrémité droite de l'affichage. Lorsque la ligne de commande contient plusieurs lignes de texte, toutes les lignes défilent simultanément vers la gauche ou la droite.

Le menu du curseur







Le menu du curseur est un menu spécial contenant des opérations de mise au point. Ce menu est actif dès que la ligne de commande est présente et qu'aucun libellé de menu n'est visible. Le menu du curseur contient des touches préfixées et non préfixées. Ces dernières sont libellées en blanc au-dessus des touches correspondantes du clavier, comme indiqué sur l'illustration suivante.



Si vous appuyez sur une touche du menu du curseur et maintenez cette touche enfoncée (excepté la touche **INS**), l'opération est répétée jusqu'à ce que vous relâchiez la touche.











Touche	Description
INS	Alterne entre les modes Remplacement et Insertion. En mode Remplacement, les caractères frappés remplacent les caractères présents. En mode Insertion, les nouveaux caractères sont insérés entre les caractères existants.
DEL	Supprime le caractère à la position du curseur.
▲	Déplace le curseur d'une ligne vers le haut.
▼	Déplace le curseur d'une ligne vers le bas.
◀	Déplace le curseur d'un caractère vers la gauche.
▶	Déplace le curseur d'un caractère vers la droite.

Les touches préfixées du menu du curseur (excepté **INS**) sont équivalentes à des répétitions des opérations non préfixées.

Touche	Description
 INS	Supprime tous les caractères à gauche du curseur.
 DEL	Supprime le caractère à la position du curseur et tous les suivants.
	Place le curseur au début de la ligne.
	Place le curseur à la fin de la ligne.
	Place le curseur à l'extrémité gauche de la ligne.
	Place le curseur à l'extrémité droite de la ligne.

Touches de saisie

Les touches suivantes servent lors de l'entrée d'objets dans la ligne de commande.

Touche	Description
 ↔	Menu du curseur actif/inactif. Cette touche alterne entre l'activation et la désactivation du menu du curseur.
 CHS	Changement de signe. Lorsque le curseur se trouve sur un nombre, cette touche en change le signe. Lorsque le curseur ne se trouve pas sur un nombre, cette touche écrit un signe moins. (Si la ligne de commande est présente, cette touche exécute la commande NEG.)
 EEX	Entrée d'exposant. Lorsque le curseur est positionné sur un nombre sans exposant : écrit le caractère E derrière le nombre. Lorsque le curseur est positionné sur un nombre avec un exposant : positionne le curseur derrière E. Si le curseur n'est pas positionné sur un nombre : écrit 1E.
	Espace arrière. Supprime le caractère à gauche du curseur et déplace le curseur (ainsi que les caractères qui le suivent) d'un espace vers la gauche. Si vous appuyez sur  et maintenez la touche enfoncée, l'action est répétée jusqu'à ce que vous relâchiez la touche.
 LC	Lettres minuscules. Alterne entre les majuscules et les minuscules. Lorsque la ligne de commande est créée, le mode Majuscules est actif ; les touches  à  placent les lettres A à Z à l'affichage. En mode Minuscules, les touches  à  placent les lettres a à z à l'affichage.

- **MENUS** **Verrouillage menu.** Alterne entre les modes Menu verrouillé et Menu déverrouillé. Lorsque le mode Menu verrouillé est actif, les positions préfixées et non préfixées des touches des trois premières lignes du clavier de gauche (lettres **A** à **R**) sont échangées. Il n'est pas nécessaire d'appuyer sur ■ avant les touches **ARRAY** à **UNITS**, mais il le faut avant les lettres A à R.
- ON** **Attention.** Annule la ligne de commande.

Délimiteurs et séparateurs d'objets

Pour entrer plusieurs objets ou commandes dans une même ligne de commande, vous devez les séparer par un des symboles suivants :

- Un délimiteur d'objet : `< > [] { } # " ' « »`.
- Un espace ou une nouvelle ligne. La séquence ■ **NEWLINE** place un caractère « nouvelle ligne » dans la ligne de commande à la position du curseur. Les caractères de nouvelle ligne sont équivalents à des espaces lors de l'exécution de la ligne de commande.
- Un point (ou une virgule si vous avez choisi le point comme séparateur décimal).

Modes de saisie

Le HP-28S possède trois modes pour faciliter la saisie : *Immédiat*, *Algébrique* et *Alpha*. Ces modes facilitent la saisie de différents types d'objets. Souvenez-vous des distinctions effectuées au chapitre précédent, « Opérations, commandes et fonctions » :

- Les *opérations* ne sont pas programmables.
- Les *commandes* peuvent être utilisées dans les programmes mais pas dans les calculs algébriques.
- Les *fonctions* (analytiques et non analytiques) et les *noms* peuvent être utilisés dans les programmes et dans les calculs algébriques.

Le calculateur reconnaît ces distinctions lorsque vous entrez des objets dans la ligne de commande. La pression sur une touche d'opération (telle que **[ENTER]**) provoque toujours l'exécution de l'opération. Le mode de saisie en cours affecte principalement ce qui se produit lorsque vous appuyez sur une touche de commande (telle que **[STO]**), une touche de fonction (telle que **[+]**), ou une touche de menu USER.

Mode de saisie immédiate. Ce mode permet d'entrer des nombres, des listes et des tableaux. En mode de saisie immédiate :

- La pression sur une touche de commande exécute la ligne commande, puis la commande.
- La pression sur une touche de fonction exécute la ligne de commande, puis la fonction.
- La pression sur une touche de menu USER exécute la ligne de commande, puis évalue le nom correspondant.

Mode de saisie algébrique. Ce mode permet d'entrer des noms et des expressions algébriques. Si vous appuyez sur **[]** au début de la ligne de commande, le calculateur active automatiquement le mode de saisie algébrique. Dans ce mode :

- La pression sur une touche de commande exécute la ligne de commande, puis exécute la commande.
- La pression sur une touche de fonction exécute la ligne de commande, puis exécute la fonction. Si l'argument de la fonction doit être entre parenthèses, la parenthèse gauche est incluse.
- La pression sur une touche de menu USER écrit le nom correspondant dans la ligne de commande.

Mode de saisie alpha. Ce mode permet de saisir des chaînes et des programmes. La pression sur **[]** ou **[]** active automatiquement le mode de saisie alpha et affiche le témoin ***α***. Dans ce mode :

- La pression sur une touche de commande écrit le nom de la commande dans la ligne de commande.
- La pression sur une touche de fonction écrit le nom de la fonction dans la ligne de commande.
- La pression sur une touche de menu USER écrit le nom correspondant dans la ligne de commande.

Si le curseur se trouve à la fin de la ligne de commande, ou si le mode Insertion est actif, des espaces séparent les commandes successives.

Exceptions

Pour vous permettre de choisir un mode lorsque le mode de saisie immédiate ou algébrique est actif, les touches de commande suivantes s'exécutent sans perturber la ligne de commande.

- **STD**, **DEC** et **RAD** du menu MODE.
- **DEC**, **HEX**, **OCT** et **BIN** du menu BINARY.

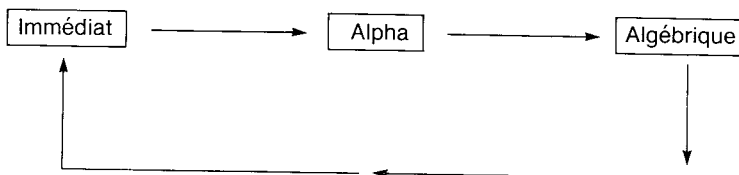
Les commandes ci-dessus n'ayant de raison d'être que dans un programme, la pression sur l'une de ces touches écrit dans tous les cas le nom de la commande dans la ligne de commande.

- **HALT** du menu PROGRAM CONTROL.
- Toutes les touches du menu PROGRAM BRANCH.

Pour éviter les pertes accidentelles de variables, la touche **CLUSR** (menu MEMORY) écrit toujours CLUSR dans la ligne de commande. Vous devez alors appuyer sur **ENTER** pour exécuter la commande.

Sélection manuelle des modes de saisie

Le calculateur alterne automatiquement entre les modes de saisie immédiate et algébrique chaque fois que vous appuyez sur **[]** pour commencer ou terminer un nom ou une expression algébrique. Il passe en outre au mode de saisie alpha lorsque vous appuyez sur **[]** ou **[α]**. Vous pouvez sélectionner manuellement le mode de saisie en appuyant sur la touche **[α]**. Cette touche vous permet d'alterner les modes de saisie comme illustré ci-dessous.









Sélection manuelle des modes de saisie

Vous pouvez par conséquent alterner entre les trois modes en appuyant une ou deux fois sur α . Voici quelques exemples de l'utilisation de la touche α .

- Supposons que vous vouliez écrire un programme que vous n'exécuterez qu'une fois ou deux. Appuyez sur α pour sélectionner le mode de saisie alpha ; entrez le programme sans délimiteur ; appuyez sur ENTER pour exécuter le programme ; appuyez sur $\blacksquare \text{COMMAND}$ pour renvoyer le programme dans la ligne de commande ; appuyez sur ENTER pour exécuter à nouveau le programme.
- Supposons que vous vouliez éliminer simultanément plusieurs variables. Appuyez sur $\{$ pour commencer une liste ; appuyez sur α pour sélectionner le mode de saisie alpha ; appuyez sur les touches de menu USER correspondant aux variables à supprimer ; appuyez sur ENTER pour placer la liste dans la pile ; appuyez $\blacksquare \text{PURGE}$.
- Supposons que vous rédigez un programme et que vous vouliez utiliser le caractère \rightarrow dans un nom. Le mode de saisie alpha étant actif, la pression sur $\blacksquare \rightarrow$ écrit la commande « \rightarrow » séparée par des espaces. Appuyez sur α pour sélectionner le mode de saisie algébrique ; appuyez sur $\blacksquare \rightarrow$; appuyez sur $\alpha \alpha$ pour retourner au mode de saisie alpha.

Indication des modes par le curseur

L'apparence du curseur identifie le mode de saisie en cours ainsi que le mode Remplacement ou Insertion. Le tableau suivant identifie les six combinaisons possibles de modes de saisie et de mode Insertion ou Remplacement.

	Insertion	Remplacement
Mode de saisie immédiate		
Mode de saisie algébrique		
Mode de saisi alpha		

Exécution de la ligne de commande




Lorsque vous appuyez sur **ENTER** (ou sur une touche qui exécute la fonction ENTER dans le mode de saisie en cours), le calcul exécute la procédure ci-dessous :

1. Il allume le témoin d'activité ((●)).
2. Si UNDO est actif, il sauvegarde une copie de la pile en cours.
3. Il cherche les délimiteurs et les séparateurs d'objet dans la chaîne présente dans la ligne de commande, puis découpe cette chaîne en sous-chaînes.
4. Il identifie le type d'objet de chaque sous-chaîne en fonction des règles de syntaxe.
5. Si COMMAND est actif, il sauvegarde dans la pile une copie de la ligne de commande.
6. Il exécute la ligne de commande.
7. Il éteint le témoin d'activité ((●)).

Si une sous-chaîne ne peut pas être identifiée à l'étape 4, les étapes 5 et 6 ne sont pas effectuées. Au lieu de cela, le calculateur affiche **Syntax Error** et met le texte incorrect en vidéo inverse, suivi du curseur. Si l'erreur provient d'une syntaxe incomplète, le curseur est positionné à la fin de la ligne.

Modification d'objets existants

Vous pouvez renvoyer un objet existant dans la ligne de commande, le visualiser ou le modifier, puis remplacer l'objet original par la nouvelle version, si vous le désirez.

Touche	Description
 EDIT	Modification niveau 1. Renvoie dans la ligne de commande l'objet présent au niveau 1.
<i>n</i>  VISIT	Modification niveau <i>n</i>. Renvoie dans la ligne de commande l'objet présent au niveau <i>n</i> .
'nom'  VISIT	Modification d'un variable. Renvoie dans la ligne de commande le contenu de la variable spécifiée.

Le menu du curseur et le mode de saisie alpha sont activés. L'objet original, s'il est visible, est mis en valeur pour vous rappeler que vous modifiez cet objet et que l'original existe toujours.

Avoir visualisé ou modifié un objet, vous pouvez :

- Appuyer sur **[ON]** pour annuler la modification, effacer la ligne de commande et laisser l'objet original inchangé.
- Appuyer sur **[ENTER]** (ou sur une touche qui exécute l'opération ENTER) pour remplacer l'objet original.

Si le menu du curseur est toujours actif lorsque vous finissez la modification, le calculateur restaure le menu précédent.

Rappel des lignes de commande

Le HP-28S sauvegarde le contenu des quatre dernières lignes de commande exécutées. La séquence **[COMMAND]** permet d'afficher successivement les lignes de commandes précédentes en commençant par la plus récente ; chaque ligne de commande rappelée remplace la ligne en cours. Si vous appuyez plus de quatre fois sur **[COMMAND]**, la séquence recommence à la ligne de commande la plus récente.

L'utilisation de **[COMMAND]** est illustrée dans les sections « Si vous exécutez une fonction par erreur », en page 47, et « Sélection manuelle des modes de saisie », en page 171.

Vous pouvez désactiver cette caractéristique en appuyant sur **[CMD]** dans le menu MODE. Le carré disparaît du libellé, indiquant que les lignes de commandes ne seront pas sauvegardées. Appuyez à nouveau sur **[CMD]** pour réactiver cette caractéristique.

La ligne de commande en tant que chaîne

Le texte que vous entrez sur la ligne de commande est équivalent au contenu d'une chaîne, c'est-à-dire à une séquence de caractères. Vous pouvez exécuter une ligne de commande par programme en entrant le texte dans une chaîne et en exécutant STR→ (*string-to-objets*, chaîne en objets). Cette technique est particulièrement utile pour stocker des programmes sous forme texte, qui est plus compacte que la forme objet. En outre, tout nom local existant lors de l'exécution de STR→ sera reconnu dans la ligne de commande.

La pile opérationnelle

Le présent chapitre reprend les concepts étudiés concernant la pile et décrit les commandes permettant de manipuler des objets dans la pile. Il décrit en outre brièvement l'utilisation des variables locales dans le but de simplifier les manipulations de la pile.

Revue des concepts concernant la pile

La pile est une séquence de *niveaux* numérotés, contenant chacun un objet. Les objets que vous entrez dans la ligne de commande sont placés dans la pile lorsque vous exécutez ENTER. Le premier objet dans la ligne de commande est le premier objet placé dans la pile. Chaque objet est placé au niveau 1 et décale les autres objets d'un niveau vers le haut. La pile peut s'agrandir indéfiniment (dans les limites de la mémoire) ; il n'est donc pas nécessaire de vous inquiéter du nombre d'objets nécessaires pour un calcul avant de commencer la saisie.

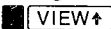

De façon générale, une commande retire de la pile des objets appelés *arguments* et les remplace par des objets appelés *resultats*. La fonction $+$, par exemple, remplace les deux arguments présents aux niveaux 1 et 2 par leur somme dans le niveau 1.

Les arguments doivent être présents dans la pile avant l'exécution de la commande. Cette logique, dans laquelle la commande suit les arguments, est appelée *logique de pile* ou *logique polonaise inverse*, en l'honneur du mathématicien polonais Jan Łukasiewicz (1878-1956).



Les résultats de la commande peuvent être utilisés comme arguments de la commande suivante. Si vous n'êtes pas encore prêt à utiliser les résultats, vous pouvez simplement les laisser dans la pile pour une utilisation ultérieure.

Les objets sortent de la pile par le niveau 1 et les objets restants dans la pile descendent tous d'un niveau. Il est préférable de stocker un objet dans une variable lorsque vous n'en avez pas besoin dans la pile de façon à faciliter la gestion des objets dont vous avez réellement besoin dans la pile. De la même façon, il est conseillé d'effacer l'ensemble de la pile avant de commencer un nouveau problème de façon à vous assurer que tous les objets présents concernent bien le problème en cours.

Affichage du contenu de la pile

Vous ne voyez normalement que les quelques premiers objets de la pile. Si l'objet présent au niveau 1 est de grande dimension, vous n'en voyez que le début. Les opérations  et  vous permettent de voir la première partie de tout objet présent dans la pile, et le contenu entier du niveau 1.

Ces opérations déplacent la « fenêtre » dans laquelle vous observez la pile. La taille de cette fenêtre peut varier entre 1 et 4 lignes, selon la présence d'un menu, de la ligne de commande ou des deux.

Touche	Description
	Déplace la fenêtre vers le haut (vers les niveaux de plus grand numéro).
	Déplace la fenêtre vers le bas (vers les niveaux de plus petit numéro).

La visualisation de la pile n'a aucun effet sur les contenus de la pile, de la ligne de commande, ou sur l'action des commandes.

Manipulation de la pile

Au cours de la première partie du manuel, vous avez utilisé des commandes simples pour manipuler la pile : CLEAR (pour effacer la pile), DROP (pour rappeler un objet au niveau 1) et SWAP (pour échanger les objets des niveaux 1 et 2). Le présent chapitre décrit brièvement toutes les commandes permettant de déplacer, copier et rappeler des objets de la pile ; consultez la section « STACK » du Manuel de référence pour plus de détail.

Déplacements d'objets dans la pile. Les commandes suivantes réarrangent les objets dans la pile ; le nombre d'objets ne change pas. Les commandes précédées de « n » demandent un nombre réel comme argument.

Commande	Description
SWAP	Echange les objets des niveaux 1 et 2.
ROT	Place l'objet du niveau 3 au niveau 1.
n ROLL	Place l'objet du niveau n au niveau 1.
n ROLLD	Place l'objet du niveau 1 au niveau n .

Les commandes ROT (*rotation, permutation*), ROLL et ROLLD (*roll down, défilement vers le bas*) décrivent le mouvement des objets d'un bloc. ROT place le contenu du niveau 3 au niveau 1 en effectuant une permutation d'un bloc de trois objets ; ROLL et ROLLD font défiler des blocs de n objets.

Copie d'objets de la pile. Les commandes suivantes renvoient une copie de un ou plusieurs objets de la pile. La copie d'un seul objet renvoie l'objet copié au niveau 1 et pousse vers le haut le reste de la pile (y compris l'objet original). Lorsque vous copiez plusieurs objets, ils sont copiés en bloc de manière similaire. Les commandes précédées de « n » nécessitent un nombre réel comme argument.

Commande	Description
DUP	Copie l'objet du niveau 1. (Si la ligne de commande n'est pas présente, vous pouvez exécuter DUP en appuyant sur <code>[ENTER]</code> .)
OVER	Copie l'objet du niveau 2.
n PICK	Copie l'objet du niveau n .
DUP2	Copie les objets des niveaux 1 et 2.
n DUPN	Copie les objets des niveaux 1 à n .

Elimination des objets de la pile. Les commandes suivantes éliminent un ou plusieurs objets de la pile. Les objets restants descendent dans les niveaux ainsi libérés. Les commandes précédées par « n » nécessitent un nombre réel comme argument.

Commande	Description
DROP	Elimine l'objet du niveau 1.
DROP2	Elimine les objets des niveaux 1 et 2.
n DROPN	Elimine les objets des niveaux 1 à n .
CLEAR	Elimine tous les objets de la pile.

Variables locales

Au cours de la première partie du manuel, vous avez rédigé quelques fonctions d'utilisateur, c'est-à-dire des programmes qui définissent des variables locales et les utilisent dans une expression ou un programme unique. Vous pouvez utiliser les fonctions d'utilisateur dans les expressions algébriques, de la même façon que les fonctions intégrées.

L'utilisation des variables locales réduit les manipulations de la pile. Lorsque vous créez des variables locales, leurs valeurs sont retirées de la pile. Vous pouvez ensuite y faire référence par leur nom au lieu de les chercher dans la pile.

Les variables locales ont d'autres applications que dans les fonctions d'utilisateur. Presque tous les exemples de programmation du chapitre 28 utilisent des variables locales. Consultez plus particulièrement « Fonctions BOX » en page 241, « MULTI (exécution multiple) » en page 253, « PRESERVE (sauvegarde et rappel d'un état précédent) » en page 258, et « SORT (tri d'une liste) » en page 270.

Rappel des derniers arguments

Le HP-28S sauvegarde les arguments de la dernière commande exécutée ; un, deux ou trois objets, selon la commande (si une commande n'utilise pas d'argument, les arguments précédents sont conservés). La commande LAST renvoie les arguments sauvegardés, dans les niveaux de la pile qu'ils occupaient à l'origine.

Si vous devez utiliser les mêmes arguments pour plusieurs commandes successives, la commande LAST renvoie une copie des arguments pour la commande suivante. Si les commandes ne demandent pas exactement les mêmes arguments, ou si les commandes ne se suivent pas immédiatement, il est plus facile d'utiliser les variables locales.

Vous pouvez désactiver LAST (c'est-à-dire la sauvegarde des arguments) en appuyant sur **LAST** dans le menu MODE. Le carré disparaît du libellé, indiquant que les arguments ne seront pas sauvegardés. Ceci n'est pas recommandé en règle générale car le calculateur utilise les arguments sauvegardés pour revenir à l'étape précédente lorsqu'une erreur a lieu. Néanmoins, si un programme ou une commande ne peut pas être exécuté par manque de mémoire, vous pouvez essayer de l'exécuter après avoir désactivé LAST. Veillez à réactiver LAST en appuyant à nouveau sur **LAST**.

Rappel de la pile

Chaque fois que vous appuyez sur **ENTER** (ou sur une touche qui exécute ENTER) le HP-28S sauvegarde tout d'abord une copie de la pile puis effectue l'action spécifiée. Si le résultat ne vous convient pas, vous pouvez rappeler la pile sauvegardée en appuyant sur **UNDO**. Remarquez que l'opération UNDO n'affecte que la pile, elle n'annule pas les modifications apportées aux indicateurs utilisateur ou aux variables. L'utilisation de **UNDO** est illustrée dans la section « Si vous exécutez une fonction par erreur » en page 47.

Vous pouvez désactiver cette caractéristique en appuyant sur **UNDO** dans le menu MODE. Le carré disparaît du libellé de menu, indiquant que la pile ne sera pas sauvegardée. Appuyez à nouveau sur **UNDO** pour réactiver cette caractéristique.

La pile opérationnelle en tant que liste

Le contenu de la pile est équivalent au contenu d'une liste, c'est-à-dire à une séquence d'objets. Vous pouvez placer tous les objets de la pile dans une liste en exécutant `DEPTH →LIST`. La commande `DEPTH` renvoie le nombre d'objets dans la pile et la commande `→LIST` (*stack to list*, pile dans une liste) combine le nombre d'objets spécifiés en une liste.

Le plus souvent, vous « ouvrirez » une liste dans la pile avec la commande `LIST→` (*list to stack*, liste dans la pile). Une fois les manipulations d'éléments effectuées dans la pile, vous pouvez les stocker à nouveau dans une liste avec la commande `→LIST`. Ces commandes sont illustrées dans la section « `MEDIAN` (médiane de données statistiques) » en page 273.

Mémoire

Le HP-28S utilise sa mémoire pour différents types d'informations, y compris la ligne de commande, la pile, la mémoire utilisateur, les caractéristiques de récupération après erreur et le système d'exploitation. La ligne de commande et la pile sont décrites aux chapitres 18 et 19. Le présent chapitre présente la mémoire utilisateur, y compris les répertoires ; il présente les effets du manque d'espace-mémoire sur la récupération après erreur et sur le système d'exploitation.

Mémoire utilisateur

La mémoire utilisateur peut contenir des variables et des répertoires permettant d'organiser ces dernières.

Variables globales

Une variable est la combinaison d'un nom et d'un autre objet. Le nom est affecté à la variable et l'autre objet représente la valeur ou le contenu de la variable.

Les variables *globales* sont celles stockées dans la mémoire utilisateur. Il existe en outre des variables *locales*, qui sont créées par des structures de programme et qui n'existent que pendant l'exécution de ces dernières. Les variables locales servent principalement d'alternative aux manipulations de la pile et sont décrites au chapitre 19, « La pile opérationnelle ».

Le contenu d'une variable peut être un objet de type quelconque. Au cours de la première partie de ce manuel, vous avez créé des variables contenant des valeurs numériques, des programmes, des expressions algébriques, des listes et des tableaux. Vous avez même créé des variables contenant le nom d'une autre variable.

Vous avez déjà utilisé les commandes suivantes pour créer, rappeler et éliminer des variables. Ces commandes traitent toutes les variables de la même façon, quel que soit leur contenu.

Commande	Description
STO	Crée une variable avec le contenu et le nom spécifié.
RCL	Rappelle le contenu de la variable spécifiée.
PURGE	Élimine une ou plusieurs variables spécifiées.

Répertoires

Au cours du chapitre 4, « Calculs répétitifs », vous avez utilisé l'algorithme de résolution d'équations pour calculer la résistance totale de circuits série-parallèle. Voici un résumé des concepts étudiés.

Il y a deux raisons principales pour créer des répertoires.

- *Grouper* les variables concernant une application ou un sujet particulier. Vous avez créé le répertoire GE pour vos problèmes de génie électrique, de telle façon que, lorsque vous travaillez sur ces problèmes, vous puissiez aisément isoler les variables utiles. De façon similaire, lorsque vous travaillez sur un autre type de problème, les variables de génie électrique restent accessibles mais cachées dans le répertoire GE.
- *Séparer* des jeux de variables utilisant des noms identiques. Vous avez créé les répertoires SP1 et SP2 (*série-parallèle-1* et *séries-parallèle-2*) dans le répertoire GE de façon à stocker séparément des valeurs différentes pour les variables R1, R2 et R3. Vous pouvez passer d'un jeu de valeurs à l'autre en changeant simplement de répertoire.

Création d'un répertoire. Pour créer un répertoire, vous devez entrer le nom et exécuter CRDIR (*create directory*, créer répertoire). Le nom du répertoire apparaît dans le menu USER. Le nouveau répertoire est appelé un *sous-répertoire* et celui qui le contient est nommé *répertoire parent*.

Répertoire en cours. A l'origine, le seul répertoire existant est le répertoire intégré HOME. Lorsque vous avez créé d'autres répertoires, vous pouvez en définir un comme *répertoire en cours* à un moment donné. Le répertoire en cours est celui dont les variables apparaissent dans le menu USER.

Pour choisir le répertoire en cours, il vous suffit d'en évaluer le nom. Si vous venez par exemple de créer un répertoire, vous pouvez le définir comme répertoire en cours en appuyant sur la touche appropriée du menu USER.

Presque toutes les commandes sur des variables utilisent uniquement celles du répertoire en cours. Vous ne pouvez modifier une variable que si elle se trouve dans le répertoire en cours.

Les commandes suivantes du menu MEMORY agissent sur le répertoire en cours.

Commande	Description
VARS	Donne une liste des noms de toutes les variables et de tous les répertoires dans le répertoire en cours.
ORDER	Réarrange les variables et répertoires dans le répertoire en cours conformément à la liste spécifiée.
CLUSR	Elimine toutes les variables et vide tous les répertoires dans le répertoire en cours.

Chemin en cours. Vous pouvez déterminer où vous vous trouvez dans la structure de répertoire en exécutant la commande PATH. Cette commande renvoie une liste spécifiant la suite de répertoires entre le répertoire HOME et le répertoire en cours.

Dans certains cas, le calculateur effectue sa recherche non seulement dans le répertoire en cours mais aussi dans le chemin complet de ce dernier. La recherche commence dans le répertoire en cours ; si la variable ne s'y trouve pas, la recherche continue dans le répertoire parent, et ainsi de suite jusqu'au répertoire HOME.

Ceci se produit lors de l'évaluation de noms ; sinon, vous ne pourriez jamais retourner au répertoire parent. L'évaluation de nom a lieu lorsque vous entrez un nom sans guillemets, que vous effectuez un traçage, que vous utilisez l'algorithme de résolution d'équations, que vous évaluez des expressions algébriques dans la pile etc.

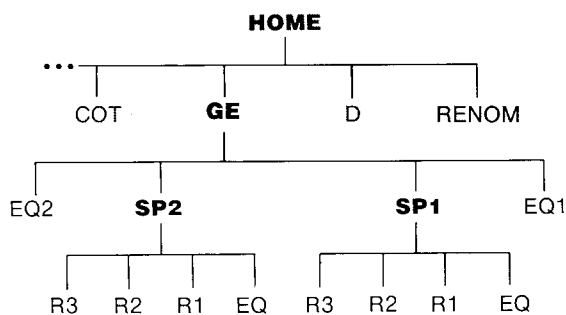
Les autres commandes qui effectuent des recherches dans le chemin en cours sont RCL et PRVAR (*print variables*, impression de variables). Remarquez qu'aucune des actions effectuant des recherches dans le chemin en cours ne peut modifier les variables.

Le répertoire HOME se trouvant toujours à la racine du chemin, le calculateur peut toujours y accéder. Nous vous recommandons de limiter le contenu du répertoire HOME aux sous-répertoires et aux variables qui doivent être disponibles à tout moment.

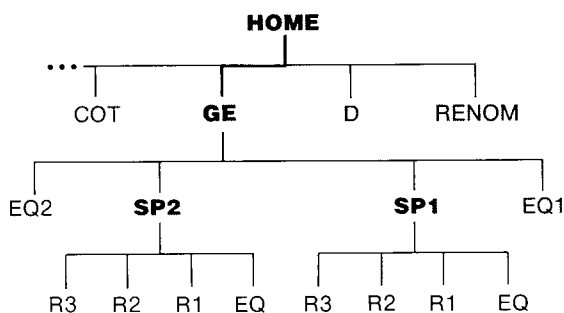
Structure de répertoire. Le diagramme ci-dessous illustre la structure de répertoire créée au chapitre 4. Dans le premier diagramme, HOME est le répertoire en cours ; dans le deuxième diagramme, GE ; et dans le troisième, SP2. Les diagrammes utilisent les symboles suivants.

Symboles utilisés dans les diagrammes de répertoire

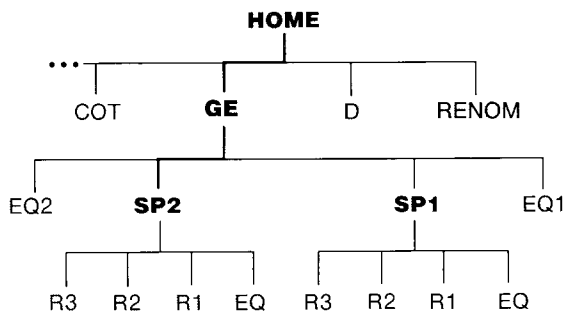
- nom** Nom d'un répertoire.
- nom** Nom du répertoire en cours.
- nom Un nom dans le répertoire en cours. Ces noms apparaissent dans le menu USER. Les variables correspondantes peuvent être modifiées.
- Le chemin en cours.
- nom Un nom dans le chemin en cours. Ces noms ne peuvent être trouvés que par évaluation, par RCL et par PRVAR. Les variables correspondantes ne peuvent pas être modifiées.



Le répertoire en cours est HOME



Le répertoire en cours est GE



Le répertoire en cours est SP2.

Elimination d'un répertoire. Vous pouvez éliminer un répertoire *vide* de la même façon qu'une variable : allez au répertoire qui contient le répertoire à supprimer, placez le nom du répertoire à supprimer dans la pile et exécutez PURGE.

Si le répertoire à éliminer contient des variables ou des sous-répertoires, vous devez éliminer ces derniers avant d'éliminer le répertoire. Voici une procédure générale :

1. Passez au répertoire à supprimer.
2. Exécutez CLUSR pour effacer le contenu du répertoire.
3. Passez au répertoire parent.
4. Supprimez le répertoire.

Si le calculateur donne un message d'erreur `Non-Empty Directory` à l'étape 2, le répertoire contient un sous-répertoire qui n'est pas vide. Dans ce cas, vous devez exécuter les étapes 1, 2 et 3 pour effacer le contenu du sous-répertoire. Vous pouvez alors continuer en exécutant les étapes 2 à 4 pour éliminer le répertoire.

Déplacement dans la structure de répertoire. Le chapitre 28 contient des programmes permettant de vous déplacer dans la structure des répertoires (aller au répertoire parent ou à un sous-répertoire). Voir « Changement de répertoires » en page 275.

Caractéristiques de récupération

Le HP-28S sauvegarde des copies des lignes de commande, des arguments et de la pile. Ces copies vous permettent de revenir en arrière après une erreur. Vous pouvez alors refaire un calcul sans recommencer par le début. Les copies des lignes de commande et des arguments sont aussi très utiles pour les calculs répétitifs.

Ces copies peuvent utiliser une quantité importante de mémoire. Chacune de ces caractéristiques de récupération peut être activée ou désactivée à l'aide du menu MODE.

Il est généralement souhaitable de laisser ces caractéristiques actives. S'il vous reste très peu de mémoire disponible et si des objets de grande dimensions ont été sauvegardés par les caractéristiques de récupération, vous pouvez désactiver et réactiver chacune de ces caractéristiques pour libérer de la mémoire.

Manque de mémoire

Le HP-28S contient 32 Koctets de mémoire utilisateur, dont 400 octets sont réservés pour le système. Presque toutes les opérations du HP-28S utilisent de la mémoire, même l'interprétation de la ligne de commande. L'espace mémoire utilisé par certaines commandes d'algèbre (COLCT, EXPAN, TAYLR) augmente rapidement avec la complexité des arguments. Essayez de conserver à tout moment quelques milliers d'octets de mémoire libres pour le fonctionnement du système.


L'instruction MEM du menu MEMORY permet de savoir à tout moment la quantité de mémoire disponible.

Le système d'exploitation du HP-28S partageant l'utilisation de la mémoire avec les objets de l'utilisateur, vous pouvez remplir la mémoire au point d'entraver ou de rendre impossible le fonctionnement normal du calculateur. Le HP-28S fournit une série de messages et de réponses en cas de manque de mémoire. Ces messages sont listés ci-dessous par ordre de sévérité.

Insufficient Memory. Il n'y a pas assez de mémoire pour exécuter une commande. Si LAST est actif, les arguments d'origine sont remis dans la pile. Si LAST est désactivé, les arguments sont perdus.

No Room for UNDO. Il n'y a pas assez de mémoire pour sauvegarder une copie de la pile. La caractéristique UNDO est automatiquement désactivée ; pour la réactiver, appuyez sur **UNDO** dans le menu MODE.

No Room to ENTER. Il n'y a pas assez de mémoire pour traiter la ligne de commande ; le calculateur efface la ligne de commande et affiche le message No Room to ENTER. Une copie de la ligne de commande est sauvegardée dans la pile de commande, si celle-ci est activée.

Si vous essayez de modifier un objet existant avec EDIT ou VISIT et si une copie de la ligne de commande est conservée dans la pile de commandes, éliminez l'original de l'objet, appuyez sur  **COMMAND** pour rappeler la ligne de commande contenant l'objet modifié et appuyez sur **ENTER** pour entrer la nouvelle version.

Low Memory! Il reste moins de 128 octets de mémoire libre. Ce message clignote lors de chaque pression de touche jusqu'à ce qu'il y ait plus de mémoire disponible. Effacez les objets inutiles avant de continuer vos calculs.

No Room To Show Stack. Il est parfois possible que le HP-28S ait suffisamment de mémoire pour terminer les opérations en cours, mais pas assez pour l'affichage normal de la pile. Les lignes de l'affichage qui devraient normalement contenir les objets de la pile, n'indiquent que le type de ces objets, par exemple, Real Number, Algebraic etc.

La quantité de mémoire nécessaire pour afficher un objet de la pile dépend du type d'objet ; les expressions algébriques nécessitent généralement le plus d'espace. Effacez un ou plusieurs objets de la mémoire, ou stockez un objet de la pile dans une variable de façon qu'il ne soit pas affiché.

Out of Memory. Dans certains cas extrêmes, le calculateur n'a plus assez de mémoire pour aucune opération, vous devez alors effacer une partie de la mémoire avant de continuer. Le calculateur active une procédure Out of Memory spéciale, qui crée l'affichage :

Out of Memory					
Purge?					
Command Stack					
YES	NO				

Le calculateur vous demande successivement d'effacer :

1. La pile COMMAND (si active).
2. La pile UNDO (si active).
3. Les arguments LAST (si actifs).
4. Le menu personnalisé (s'il y en a un).
5. La pile.
6. Chaque variable du répertoire HOME.

Appuyez sur la touche de menu **YES** pour chaque article à éliminer ; appuyez sur **NO** pour les articles à conserver.

Lorsque vous avez appuyé une fois sur **YES**, vous pouvez essayer de terminer la procédure Out of Memory en appuyant sur **ON**. S'il y a suffisamment de mémoire, le calculateur retourne à l'affichage normal de la pile ; sinon, il déclenche son avertisseur sonore et continue la procédure. Après une première passe dans tous les choix, la procédure Out of Memory essaie de retourner au fonctionnement normal. S'il n'y a toujours pas assez de mémoire, la procédure recommence au début.

Si vous appuyez sur **YES** pour un répertoire vide, le calculateur l'élimine. Si le répertoire contient des variables, le calculateur affiche ces dernières.

Optimisation des performances

Le calculateur effectue de temps en temps des tâches d'organisation pour optimiser l'utilisation de la mémoire. Vous ne remarquerez normalement que de courtes pauses lors du traçage ; néanmoins, lorsque la mémoire est presque remplie et que la pile contient des centaines d'objets, le calculateur peut répondre lentement à des opérations simples telles que la sélection d'un menu.

Cette section contient des informations permettant d'optimiser la vitesse (en réduisant les tâches d'organisation nécessaires) et la quantité de mémoire disponible (en augmentant l'efficacité des tâches d'organisation).

Pour optimiser la vitesse :

- Ne placez plus de quelques centaines d'objets dans la pile.
- Ne laissez pas de liste de grande taille (plusieurs centaines d'objets) dans la pile ; stockez-les en mémoire utilisateur.

Pour optimiser la mémoire disponible :



La procédure suivante efface la pile, les données de récupération d'erreur (COMMAND, UNDO, LAST), le menu personnalisé en cours (CUSTOM) et tout programme interrompu.

1. Effacez de la mémoire utilisateur les variables et répertoire inutiles.
2. Stockez en mémoire utilisateur tous les objets de la pile que vous voulez conserver.
3. Effectuez un Arrêt du système ()

Le répertoire en cours est désormais HOME.

Pour minimiser l'utilisation de mémoire pour les calculs

matriciels : stockez les tableaux dans des variables et utilisez des noms pour y faire référence ; évitez de les utiliser dans la pile. Voici une stratégie utile pour atteindre ce but.

1. Déterminez à l'avance le nombre de variables dont vous aurez besoin, y compris pour les résultats intermédiaires.
2. Créez des *petits* tableaux du type correct (réel ou complexe, vecteur ou matrice), stockez-les dans des variables et utilisez ensuite RDM pour ajuster les dimensions.
3. Effectuez les calculs en utilisant les commandes d'arithmétique de stockage du menu STORE.
4. Pour agir sur des éléments individuels, utilisez GET, GETI, PUT, PUTI avec le *nom de variable*, ou utilisez la syntaxe algébrique telle que 'A(5.6)' EVAL et 'B(3)' STO ; ne renvoyez pas le tableau entier dans la pile.

Menus

Chaque opération, commande et fonction du HP-28S est disponible sur le clavier ou dans un menu. Lorsque vous sélectionnez un menu, six *libellés de menu* apparaissent sur la ligne inférieure de l'affichage. Ces libellés constituent une *ligne de menu*, qui indique les définitions en cours des six *touches de menu* en haut du clavier. (Le menu du curseur est une exception ; ses définitions sont imprimées en blanc au-dessus des touches de menu). En plus des touches qui sélectionnent des menus spécifiques (■ [ARRAY], [TRIG]), ces touches contrôlent le fonctionnement du menu :

Touche

Description



Activation/désactivation du menu du curseur. Lorsque le menu du curseur n'est pas actif, cette touche le sélectionne ; lorsqu'il est actif, cette touche sélectionne le menu précédent.



Dernier menu personnalisé. Affiche le dernier menu personnalisé créé par la commande MENU.



Ligne suivante du menu. Affiche la ligne suivante des libellés de menu. Si la dernière ligne est affichée, cette touche affiche la première ligne.



Ligne précédente du menu. Affiche la ligne précédente du menu. Si la première ligne est affichée, cette touche affiche la dernière ligne.



Verrouillage du menu. Active ou désactive le verrouillage de menu. Lorsque le menu est verrouillé, le calculateur échange les positions préfixées et non-préfixées des trois premières lignes de touches du clavier de gauche (lettres [A] à [R]). Lorsque le verrouillage est activé, la pression sur la touche [A] sélectionne le menu ARRAY et la pression sur ■ [A] écrit la lettre A.

Menus de commandes

Les menus suivants contiennent les touches des opérations intégrées, la plupart desquelles sont des commandes programmables. L'annexe D, « Les menus et leurs jeux de libellés », décrit brièvement les commandes de chaque menu. Le Manuel de référence présente ces menus en ordre alphabétique et les décrit en détail.

L'action des touches de ces menus dépend du mode de saisie ; voir page 169.

Menu	Description
ALGEBRA	Commandes d'algèbre.
ARRAY	Commandes de vecteurs et matrices.
BINARY	Arithmétique sur des entiers binaires, conversion de base, manipulations de bits.
COMPLEX	Commandes de nombres complexes.
LIST	Commandes de listes.
LOGS	Fonctions logarithmiques, exponentielles et hyperboliques.
MEMORY	Mémoire utilisateur et répertoires.
MODE	Modes d'affichage, d'angle et de récupération après erreur.
PLOT	Commandes de traçage.
PRINT	Commandes d'impression.
PROGRAM BRANCH	Structures de branchement de programme.
PROGRAM CONTROL	Contrôle de programme, arrêt et opérations pas-à-pas.
PROGRAM TEST	Indicateurs et tests logiques.
REAL	Commandes de nombres réels.
SOLVE	Commandes de résolution numérique et symbolique, l'algorithme de résolution d'équations.

Menu	Description
STACK	Manipulation de la pile.
STAT	Commandes statistiques et de probabilité.
STORE	Arithmétique sur des valeurs stockées.
STRING	Chaînes de caractères.
TRIG	Fonctions trigonométriques, conversions d'angles et de coordonnées.

Menus des opérations

Les menus suivants contiennent les opérations non-programmables.

Menu	Description
Curseur	Modification de la ligne de commande. Décrit au chapitre 18.
CATALOG	Catalogue des commandes, y compris le sous-menu USAGE. Décrit au chapitre 22.
UNITS	Unités disponibles pour les conversions. Décrit au chapitre 14.

Menus des variables

Menu	Description
L'algorithme de résolution d'équations	Stockes des valeurs et résoud l'équation en cours. L'apparence particulière (lettres sombre sur fond clair) indentifie l'action particulière des touches.
USER	Affiche les variables et les sous-répertoires du répertoire en cours. L'action des touches dépend du mode de saisie en cours ; voir page 169.


Menus personnalisés

La commande MENU permet de créer un menu personnalisé à partir d'une liste de noms et de commandes. Le menu personnalisé peut être similaire au menu de l'algorithme de résolution d'équations menu ou au menu USER.


- Si le premier élément de la liste est la commande STO, suivie d'une séquence de noms, MENU crée un menu *de saisie personnalisée*. Ce menu ressemble à celui de l'algorithme de résolution d'équations et fonctionne de la même façon : la pression sur une touche du menu prend une valeur dans la pile et la stocke dans la variable correspondante. Consultez l'exemple au chapitre 27, « Programmes interactifs ».
- Si la liste contient une séquence de noms et de commandes (et si le premier élément est différent de STO), MENU crée un *menu utilisateur personnalisé*. Ce menu agit de façon similaire au menu USER et à un menu de commande. Consultez l'exemple de la section « Changement de répertoire » en page 275.

Catalogue des commandes

Au cours du chapitre 1, vous avez utilisé le catalogue des commandes pour vérifier l'orthographe de certaines commandes et pour déterminer les diverses combinaisons d'arguments pour la fonction +. Le présent chapitre passe en revue les opérations disponibles dans le catalogue, y compris le menu USAGE qui indique les combinaisons valides d'arguments.

La séquence  **CATALOG** affiche le menu CATALOG et la commande ABORT, qui est la première commande du catalogue.

Touche	Description
NEXT	Passe à la commande suivante du catalogue. Si vous maintenez cette touche enfoncée, le calculateur fait défiler le catalogue jusqu'à ce que vous relâchiez la touche.
PREV	Passe à la commande précédente du catalogue. Si vous maintenez cette touche enfoncée, le calculateur fait défiler le catalogue jusqu'à ce que vous relâchiez la touche.
USE	Active l'affichage du menu USAGE (voir ci-après) indiquant les arguments dans la pile utilisés par la commande.
FETCH	Sort du catalogue et écrit sur la ligne de commande le nom de la commande en cours.
QUIT	Sort du catalogue en laissant la ligne de commande inchangée.

Vous pouvez simultanément sortir du catalogue et effacer la ligne de commande en cours en appuyant sur .

Recherche d'une commande

Vous pouvez utiliser les touches du clavier de gauche pour aller à un endroit spécifique du catalogue.

- La pression sur une touche alphabétique du clavier de gauche vous place sur la première commande du catalogue dont le nom commence par la touche par cette lettre. S'il n'y a pas de commande commençant par cette lettre, le calculateur affiche la dernière commande commençant par la lettre précédente.
- La pression sur une touche non alphabétique (telle que Σ) vous place sur la première commande dont le nom commence par ce caractère. S'il n'y a pas de commande commençant par ce caractère, le calculateur affiche la commande +, la première commande commençant par un caractère non alphabétique.
- La pression sur $\boxed{\text{MENUS}}$ vous place sur $\rightarrow\text{STR}$, la dernière commande du catalogue.

Vérification de la syntaxe des commandes

Vous pouvez déterminer quels sont les types d'argument corrects pour la commande affichée dans le catalogue. La pression sur $\boxed{\text{USE}}$ active un deuxième niveau du catalogue, appelé menu USAGE, qui montre toutes les combinaisons d'arguments possibles pour la commande. Si la commande accepte plusieurs combinaisons d'arguments, les touches de menu suivantes apparaissent. (Si la commande accepte uniquement une seule combinaison d'arguments, les libellés $\boxed{\text{NEXT}}$ et $\boxed{\text{PREV}}$ ne sont pas affichés.)

Touche	Description
$\boxed{\text{NEXT}}$	Affichage la combinaison d'arguments suivante.
$\boxed{\text{PREV}}$	Affiche la combinaison d'arguments précédente.
$\boxed{\text{QUIT}}$	Retourne au catalogue précédent et affiche la commande en cours. Vous pouvez alors vous déplacer dans le catalogue ou sortir du catalogue en appuyant à nouveau sur $\boxed{\text{QUIT}}$.

Vous pouvez sortir de USAGE et du catalogue principal, et effacer la ligne de commande en cours en appuyant sur $\boxed{\text{ON}}$.

Evaluation

Toutes les opérations du calculateur, des simples calculs au clavier jusqu'aux programmes complexes, utilisent l'évaluation. Par exemple :

- Lorsque vous entrez des objets dans la ligne de commande et que vous appuyez sur **ENTER**, le calculateur traduit la ligne de commande en un programme, puis évalue ce dernier.
- Lorsque vous appuyez sur une touche du menu USER en mode de saisie immédiate, le calculateur évalue le nom correspondant.
- Lorsque vous calculez une dérivée pas-à-pas, vous appuyez sur **EVAL** pour évaluer l'expression au niveau 1.
- Lorsque vous utilisez l'algorithme de résolution d'équations pour calculer une solution numérique, la procédure stockée dans la variable EQ est évaluée de façon répétitive.

Il est plus facile de comprendre les opérations du calculateur en termes d'*évaluation différée* et d'*évaluation directe*. L'entrée d'un nom ou d'une expression algébrique, entre guillemets, indique que vous voulez différer l'évaluation de l'objet ; l'objet est simplement placé dans la pile.

L'évaluation différée est la base de la programmation sur tout les calculateurs ou ordinateurs, sinon tout programme serait exécuté dès sa rédaction. Le HP-28S élargit ce concept de façon uniforme pour permettre d'exécuter des opérations symboliques ; vous pouvez ainsi utiliser des noms et des expressions algébriques comme données dans des calculs symboliques. Vous choisissez quand vous voulez évaluer une expression. Vous pouvez en calculer la dérivée, en calculer une solution symbolique, substituer des valeurs à ses variables, etc. Vous pouvez aussi, bien entendu, en calculer la valeur numérique.

Le présent chapitre décrit ce qui se passe lorsque vous évaluez les différents types d'objets. De façon générale, considérez les *catégories d'objets* suivantes :

- *Données*. Cette catégorie contient les nombres réels, complexes et binaires, les chaînes, les tableaux et les listes. La « valeur » d'une donnée est le contenu de l'objet.
- *Noms*. Cette catégorie contient les noms globaux et locaux. La « valeur » d'un nom est généralement le contenu d'une variable.
- *Procédure*. Cette catégorie contient les expressions algébriques et les programmes. La « valeur » d'une procédure est le résultat du processus qu'elle définit.

Ces catégories définissent ce qui se produit lorsque vous évaluez un objet : renvoie de l'objet, du contenu d'une variable, ou du résultat d'un processus. Les sections suivantes décrivent ceci en détail.

Données

Les données constituent la catégorie d'objet la plus simple. L'évaluation des données donne la valeur de l'objet.

Remarquez que les listes sont des données générales ; elles peuvent contenir des objets d'un type quelconque. Considérez une liste de noms : chaque nom est « protégé » contre l'évaluation par la liste et ils ne peuvent pas être individuellement évalués tant qu'ils sont dans la liste.

Noms

De façon générale, la « valeur » d'un nom est le contenu d'une variable. L'évaluation des noms locaux est simple et est décrite en premier, suivie de l'évaluation des noms globaux.

Evaluation de noms locaux

Comme décrite au chapitre 19, l'utilisation des variables locales simplifie les manipulations de la pile. Le but des variables locales est (1) d'éliminer de la pile le contenu des variables et (2) de fournir une copie du contenu de la variable lorsque vous en avez besoin.

L'évaluation d'un nom local renvoie donc toujours dans la pile le contenu de la variable locale correspondante.

Evaluation de noms globaux

De façon générale, l'évaluation d'un nom global provoque l'évaluation du contenu de la variable globale correspondante. En d'autres termes, l'évaluation d'un nom global a le même effet que l'évaluation du terme qu'il représente.

Il y a deux exceptions à la règle générale :

- S'il n'y a aucune variable avec le nom spécifié, le calculateur renvoie le nom dans la pile. Un nom non défini utilisé comme variable est appelé *variable formelle*.
- Si le contenu de la variable spécifiée est une expression algébrique, l'expression n'est *pas évaluée*. Le calculateur évite d'évaluer ces objets de façon à vous permettre de continuer des calculs symboliques. Si vous voulez évaluer l'expression algébrique, placez l'expression au niveau 1 et exécutez la commande EVAL (pour évaluer une expression algébrique de façon répétitive jusqu'à ce qu'elle donne un résultat numérique, exécutez →NUM).

Si la variable contient une donnée, l'évaluation du nom de la variable est équivalent au rappel de son contenu. Néanmoins, l'évaluation du nom d'une variable peut entraîner une longue chaîne d'évaluations. Si une variable contient, par exemple, le nom d'une deuxième variable contenant elle-même celui d'une troisième variable, alors l'évaluation de la première variable provoque l'évaluation de la deuxième variable, de la troisième et du contenu de celle-ci.



Ne créez pas de variable dont la valeur est son propre nom, telle qu'une variable X contenant le nom 'X'. L'évaluation d'une telle variable crée une boucle sans fin. Pour arrêter une telle boucle, vous devez effectuer un arrêt du système (☐ON ☐▲), ce qui efface en outre la pile.

De même, ne créez pas de variables qui se fassent réciproquement référence (définition circulaire). L'évaluation d'une variable incluse dans une définition circulaire crée aussi une boucle sans fin.

Procédure

De façon générale, la « valeur » d'une procédure est le résultat du processus qu'elle définit. Les programmes sont les objets les plus courants dans la catégorie procédure, ils sont par conséquent décrits en premier, suivis des expressions algébriques.

Evaluation de programmes

Un programme est une séquence d'objets et de commandes. Ce manuel utilise les termes « évaluation d'un programme » et « exécution d'un programme » de façon interchangeable. L'évaluation d'un programme traite généralement son contenu de façon séquentiel, plaçant chaque objet dans la pile et exécutant chaque commande. Souvenez-vous en outre des points suivants :

- Les noms sans guillemets sont évalués alors que les noms entre guillemets sont envoyés dans la pile. La mise entre guillemets des noms permet d'en différer l'évaluation comme indiqué en page 57.
- Les structures de programme sont exécutées conformément à leurs propres règles. Au cours de la première partie du manuel, vous avez rédigé plusieurs fonctions utilisateur, contenant une *structure de variable locale*. Les structures de programmes sont décrites au chapitre 26.

Les règles d'évaluation des noms et des programmes sont à l'origine d'un des concepts de base pour la programmation du HP-28S. Au cours de cette présentation, le terme « programme » signifie un programme stocké dans une variable et l'expression « nom de programme » représente le nom de la variable contenant le programme.

Le concept de base est nommé *programmation structurée*. Cela signifie qu'une tâche complexe est découpée en modules et qu'un programme est rédigé pour chaque module. Le programme principal peut être relativement simple et reflète uniquement la logique globale du calcul. Il exécute les modules en incluant simplement leurs noms sans guillemets. Si un module doit être exécuté plusieurs fois, vous pouvez simplement spécifier le nom sans guillemets du module aux endroits appropriés. D'autres programmes principaux peuvent utiliser les mêmes modules.

La programmation structurée est présentée dans les sections « Développement et mise en facteur complète » en page 253, « Affichage d'un entier binaire » en page 257 et « Médiane de données statistiques » en page 270.

Evaluation d'expression algébriques

Chaque expression algébrique est équivalente à un programme contenant uniquement des noms et des fonctions sans guillemets. L'évaluation d'une expression algébrique donne le même résultat que l'évaluation du programme correspondant : les noms sans guillemets sont évalués et les fonctions sont exécutées. Ce sujet est présenté en détail dans la section « Evaluation d'objets algébriques » du Manuel de référence.

Le résultat de l'évaluation d'un nom dépend de l'existence ou non d'un variable portant ce nom, comme décrit dans la section « Evaluation de noms globaux » ci-avant. Par exemples :

- Si un nom fait référence à une fonction utilisateur, vous pouvez utiliser le nom de la fonction utilisateur comme une fonction intégré. L'évaluation de l'expression algébrique provoque l'exécution de la fonction utilisateur. Les arguments de la fonction utilisateur, placés entre parenthèses après le nom de la fonction, font parti de l'expression algébrique.

- Si un nom fait référence à un programme qui ne prend pas d'arguments dans la pile et qui donne un résultat unique, vous pouvez utiliser le nom du programme pour faire référence (indirectement) au résultat. L'évaluation de l'expression algébrique provoque l'exécution du programme, de telle façon que le nom programme se trouve remplacé par le résultat. Consultez les exemples de la section « Sommutations » en page 262.
- Si un nom fait référence à une autre expression algébrique, l'évaluation de la première ne provoque *pas* l'évaluation de la deuxième. La deuxième expression algébrique remplace simplement le nom de la première.

La fonction « = » est un cas particulier parmi les fonctions, car elle permet de différencier les équations des expressions. Selon le mode de résultat en cours (symbolique ou numérique), l'exécution de = donne soit une équation soit un résultat numérique.

- En mode de résultat symbolique, l'évaluation d'une équation donne une nouvelle équation. La nouvelle expression de gauche est le résultat de l'évaluation de l'expression de gauche initiale, et la nouvelle expression de droite est le résultat de l'évaluation de l'expression de droite initiale.
- En mode de résultat numérique, l'évaluation d'une équation donne la différence numérique entre l'expression de gauche initiale (évaluée numériquement) et l'expression de droite initiale (évaluée numériquement).

La section suivante décrit le mode résultat en plus grand détail.

Evaluation de fonctions

Lors de l'évaluation d'une fonction, son action dépend du mode de résultat en cours, qui peut être symbolique ou numérique. Ces modes sont en outre décrits dans le chapitre suivant, « Modes ».

Mode de résultat symbolique. Ce mode est celui utilisé par défaut ; les fonctions acceptent des arguments symboliques et donnent des résultats symboliques. L'action des fonctions en mode de résultat symbolique est évidente lorsque vous calculez avec des noms et des expressions pour créer des expressions plus importantes.

Mode de résultat numérique. Cette alternative sert plus particulièrement pour les traçages et pour l'algorithme de résolution d'équations. Il permet d'obtenir des résultats numériques à partir des fonctions. Dans ce mode, les fonctions évaluent de façon répétitive les arguments symboliques, acceptant uniquement des arguments numériques et donnant des résultats numériques.

Vous pouvez forcer l'évaluation d'un objet jusqu'à ce qu'il donne un résultat numérique en exécutant \rightarrow NUM (*to number*, vers nombre) ; c'est ce que vous avez fait au chapitre 5 pour obtenir la valeur numérique de π .



Note

En mode de résultat numérique, n'évaluez pas un variable dont la valeur inclut son propre nom, telle qu'une variable nommée X contenant l'expression « X+Y ».

L'évaluation d'une telle variable crée une boucle sans fin. Pour sortir d'une telle boucle, vous devez exécuter un arrêt du système (\square ON \square ▲), ce qui efface en outre la pile.

De la même façon, ne créez pas de variables qui se font réciproquement référence (définition circulaire). L'évaluation d'une telle variable crée également une boucle sans fin.

Modes

Le choix d'un mode permet de modifier les résultats de nombreuses opérations. Certains modes, tels que l'unité d'angle (Degrés ou Radians), peuvent être sélectionné par la pression sur une touche de menu. Le libellé du mode contient un petit carré lorsque le mode est actif. Le libellé du mode Radians, par exemple, apparaît sous la forme **RAD■** lorsque ce mode est actif.

La plupart des modes, tels que le mode Beeper (actif ou inactif), peuvent être sélectionné en activant ou désactivant un indicateur utilisateur à l'aide des commandes **SF** (*set flag*, armer indicateur) et **CF** (*clear flag*, désarmer indicateur). L'indicateur 51, par exemple, contrôle le mode Beeper. Vous pouvez donc désactiver l'avertisseur sonore en exécutant **51 SF**.

Le présent chapitre décrit la façon dont les modes affectent le fonctionnement du calculateur et liste les libellés, les indicateurs et les témoins associés. Pour chaque mode, le manuel indique en premier l'état par défaut (celui auquel le calculateur revient après une perte de mémoire).

Modes généraux

Ces modes affectent les calculs et l'avertisseur sonore.

Mode d'angle

Ce mode détermine si les nombres réels des valeurs d'angle sont exprimés en degrés ou en radians. Ce mode affecte les arguments des fonctions trigonométriques et les résultats des fonctions trigonométriques inverses.

Mode degré (DEG , Indicateur 60 désarmé). Les nombres réels représentent les mesures d'angle en degrés.

Mode radians (RAD , Indicateur 60 armé, (2π)). Les nombres réels représentent les mesures d'angles en radians.

Mode Beeper

Ce mode contrôle le fonctionnement de l'avertisseur sonore lorsqu'une erreur a lieu ou lorsque vous exécutez BEEP.

Avertisseur actif (Indicateur 51 désarmé). Le calculateur utilise l'avertisseur.

Avertisseur inactif (Indicateur 51 armé). Le calculateur est silencieux.

Valeur principale

ISOL et QUAD nécessitent généralement des signes arbitraires (+1 ou -1) et des entiers (0, 1, 2, ...) pour représenter toutes les solutions possibles. Le mode de valeur principale détermine si les signes arbitraire et les entiers sont inclus dans les solutions de ISOL et QUAD.

Valeur principale inactive (Indicateur 34 désarmé). Les solutions de ISOL et QUAD utilisent les variables s_1, s_2, \dots , comme signes arbitraires et n_1, n_2, \dots , comme entier arbitraires.

Valeur principale active (Indicateur 34 armé). ISOL et QUAD utilisent +1 comme signe arbitraire et 0 comme entier arbitraire.

Mode des constantes

Ce mode affecte l'évaluation des constantes symboliques e , i , MINR , MAXR et π . En mode de résultat numérique (indicateur 36 désarmé), l'évaluation d'une constante symbolique donne la valeur numérique de la constante, quel que soit le mode des constantes.

Constantes symboliques (Indicateur 35 armé). L'évaluation d'une constante symbolique donne la valeur symbolique.

Constantes numériques (Indicateur 35 désarmé). L'évaluation d'une constante symbolique donne sa valeur numérique.

Mode de résultat

Le mode de résultat en cours affecte le résultat de l'évaluation d'une fonction lorsque son argument est symbolique.

Résultats symboliques (Indicateur 36 armé). Les fonctions d'arguments symboliques donnent des résultats symboliques.

Résultats numériques (Indicateur 36 désarmé). Les fonctions donnent toujours des résultats numériques, même lorsque les arguments sont symboliques et que le mode de constantes symboliques est actif. Pour cela, les fonctions évaluent de façon répétitive les arguments symboliques pour en déterminer les valeurs numériques.

Modes de saisie et d'affichage

Ces modes affectent la façon dont les objets sont entrés et sont affichés.

Mode de saisie


Le mode de saisie en cours affecte le résultat de la pression sur une touche de commande, de fonction ou de menu User. Le mode de saisie change automatiquement lorsque vous appuyez sur \square , \blacksquare ou sur α ; vous pouvez aussi le changer manuellement en appuyant sur α . L'apparence du curseur identifie le mode de saisie en cours. Pour plus de détails, consultez le chapitre 18, « La ligne de commande ».

Saisie immédiate (boîte vide). La ligne de commande est exécutée lorsque vous appuyez sur une touche de commande, de fonction ou sur une touche du menu User.

Saisie algébrique (boîte hachurée). La ligne de commande est exécutée lorsque vous appuyez sur une touche de commande.

Saisie alpha (boîte pleine, α). La ligne de commande n'est exécutée que lorsque vous appuyez sur $\boxed{\text{ENTER}}$.


Mode de remplacement et insertion

La touche  du menu du cuseur alterne entre les modes remplacement et insertion. L'apparence du curseur identifie ce mode.

Mode de remplacement (boîte). Les nouveaux caractères remplacent les caractères existants.

Mode d'insertion (flèche). Les nouveaux caractères sont insérés entre les caractères existants.

Majuscules et minuscules


La touche  alterne entre les modes majuscule et minuscule.


Mode majuscule. La pression sur une touche alphabétique écrit une lettre majuscule dans la ligne de commande.

Mode minuscule. La pression sur une touche alphabétique écrit une lettre minuscule dans la ligne de commande.

Affichage du niveau 1

De nombreux objets ne peuvent pas être affichés sur une seule ligne d'affichage. Vous pouvez choisir d'utiliser plusieurs lignes pour afficher le contenu du niveau 1 de la pile, si nécessaire, ou d'utiliser une seule ligne quelle que soit la taille de l'objet. Ce choix affecte la sortie imprimée en mode Auto.

Plusieurs lignes ( , Indicateur 45 armé). Les objets du niveau 1 sont affichés sur plusieurs lignes si nécessaire.

Une seule ligne ( , Indicateur 45 désarmé). Les objets du niveau 1 sont affichés sur une seule ligne.

Mode de symbole décimal

Vous pouvez utiliser soit la virgule soit le point comme séparateur décimal. Le symbole qui n'est pas utilisé pour séparer les parties entière et fractionnaire peut alors servir à séparer les objets sur la ligne de commande (l'espace peut toujours être utilisé comme séparateur d'objets).

Point comme séparateur décimal (`RDX.` , Indicateur 48 désarmé). Le point est le séparateur décimal et la virgule peut être utilisée pour séparer les objets.

Virgule comme séparateur décimal (`RDX,` , Indicateur 48 armé). La virgule est le séparateur décimal et le point peut être utilisé comme séparateur d'objets.

Format numérique

Ces modes déterminent le nombre de décimales affichées pour les nombres réels. Les commandes `FIX`, `SCI` et `ENG` nécessitent un argument n réel. Le format numérique en cours affecte en outre la commande `RND` (*round*, arrondi).

Format STD (`STD.`). Les nombres réels ne sont affichés avec un symbole décimal et un exposant que lorsque cela est nécessaire.

Format FIX (`FIX.`). Les nombres réels sont affichés avec n décimales. L'exposant n'est affiché que s'il est nécessaire.

Format SCI (`SCI.`). Les nombres réels sont affichés sous forme d'une mantisse, inférieure à 10 et contenant n décimales, et un exposant.

Format ENG (`ENG.`). Les nombres réels sont affichés sous forme d'une mantisse ($n + 1$ chiffres) et un exposant multiple de 3.

Base pour l'affichage des nombres binaires

Vous pouvez choisir la base utilisée pour la saisie et l'affichage des entiers binaires. Le choix de base n'affecte pas la structure interne des entiers binaires, qui sont toujours traités comme des séquences de bits.

Base DEC (`DEC`). Les entiers binaires entrés sans marqueurs de base sont interprétés comme étant en base 10. Tous les entiers binaires sont affichés en base 10 avec le marqueur de base « b ».

Base HEX (`HEX`). Les entiers binaires entrés sans marqueurs de base sont interprétés comme étant en base 16. Tous les entiers binaires sont affichés en base 16 avec le marqueur de base « h ».

Base OCT (`OCT`). Les entiers binaires entrés sans marqueurs de base sont interprétés comme étant en base 8. Tous les entiers binaires sont affichés en base 8 avec le marqueur de base « o ».

Base BIN (`BIN`). Les entiers binaires entrés sans marqueurs de base sont interprétés comme étant en base 2. Tous les entiers binaires sont affichés en base 2 avec le marqueur de base « b ».

Taille des mots binaires

La taille de mot en cours peut varier entre 1 et 64 bits. Cette taille contrôle la façon dont les entiers binaires sont affichés ; les entiers binaires sont tronqués à la taille de mot en cours lors de leur utilisation comme argument ou lorsqu'ils apparaissent comme résultat d'une opération. Pour définir la taille de mot à n , exécuter n `STWS` (*store wordsize*, stocker la taille de mot).


Modes de récupération après erreur

Les modes de récupération déterminent si le calculateur effectue des copies de la ligne de commande, de la pile et des arguments des commandes. Ces copies peuvent vous aider à corriger les erreurs.


Mode CMD



Ce mode détermine si le calculateur sauvegarde la ligne de commande lorsque vous appuyez sur `ENTER` (ou sur une touche qui exécute ENTER).


CMD actif (`CMD`). Les lignes de commandes sont sauvegardées et peuvent être récupérées avec `COMMAND`.

CMD inactif (). Les lignes de commande ne sont pas sauvegardées.

Mode UNDO


Ce mode détermine si le calculateur sauvegarde la pile lorsque vous appuyez sur  (ou sur une touche qui exécute ENTER).


UNDO actif (). La pile est sauvegardée et peut être récupérée avec .

UNDO inactif (). La pile n'est pas sauvegardée.

Mode LAST

Ce mode détermine si le calculateur sauvegarde les arguments des commandes exécutées.

LAST actif (Indicateur 31 armé, ). Le calculateur sauvegarde les arguments de la dernière commande. Ces arguments peuvent être récupérés par LAST.

LAST inactif (Indicateur 31 désarmé, ). Les arguments ne sont pas sauvegardés. Si une erreur a lieu, les arguments de la dernière commande ne sont pas remis dans la pile.

Exceptions mathématiques

Certaines erreurs apparaissant au cours de calculs sur des nombres réels sont appelés des *exceptions mathématiques*. Une exception peut agir comme une erreur ordinaire et arrêter l'exécution, ou bien peut fournir un résultat par défaut et permettre au calcul de continuer.

Résultat infini

Cette exception a lieu lorsqu'un calcul donne un résultat infini. Exemples : évaluation de ' $\text{LN}(0)$ ', ' $\text{TAN}(90)$ ' (en mode Degrés) ou ' $\text{X}/0$ '.

Résultat infini erreur (Indicateur 59 armé). Les résultats infinis donnent des erreurs.

Impression auto

Vous pouvez automatiquement imprimer tous les calculs effectués.

Impression auto inactive (TRAC , Indicateur 32 désarmé). Il n'y a pas d'impression automatique.

Impression auto active (TRAC , Indicateur 32 armé). Chaque exécution de la ligne de commande provoque l'impression de son contenu, de l'opération ayant provoqué l'exécution et du résultat présent au niveau 1.

Mode de retour chariot automatique

La plupart du temps, vous voulez envoyer des données à l'imprimante et imprimer ces données avec une seule commande. Lors de l'impression des graphiques par contre, vous devez accumuler des données dans l'imprimante sans les imprimer immédiatement.

Retour chariot auto (Indicateur 33 désarmé). Les commandes d'impression envoient automatiquement des retours-chariot à la fin de chaque transmission pour imprimer les données.

Accumule les données (Indicateur 33 armé). Les commandes d'impression envoient les données sans retour-chariot de façon à accumuler les données dans le tampon de l'imprimante.

Vitesse d'impression

Le calculateur ne peut pas détecter si l'imprimante est prête à accepter d'autres données. De ce fait, il doit calculer la vitesse à laquelle il peut envoyer les données de façon sûre. Ce mode indique au calculateur si l'imprimante est alimentée sur batteries (vitesse d'impression normale) ou sur secteur (vitesse d'impression élevée).

Vitesse d'impression normale (Indicateur 52 désarmé). Le calculateur envoie les données à l'imprimante à une vitesse convenant au fonctionnement sur batterie.

Vitesse d'impression élevée (Indicateur 52 armé). Le calculateur envoie les données à l'imprimante à une vitesse convenant au fonctionnement sur secteur.

Interligne

Ce mode détermine si le calculateur imprime le texte en interligne simple ou double.

Interligne simple (Indicateur 47 désarmé). Il n'y a pas de ligne blanche entre les informations imprimées.

Interligne double (Indicateur 47 armé). Les informations sont imprimées une ligne sur deux.

Opérations du système

Le présent chapitre décrit les combinaisons spéciales de touches qui interrompent le fonctionnement normal du HP-28S. Ces opérations incluent l'impression de l'affichage, l'ajustement du contraste d'affichage, l'arrêt des programmes, la réinitialisation de la mémoire et l'exécution des diagnostics.

Toutes les opérations système commencent par la pression sur la touche **[ON]**. Vous pouvez annuler toute opération système en appuyant sur **[DEL]** avant de relâcher **[ON]**.

Le tableau ci-dessous indique les séquences de touche des opérations système avec une brève description de la fonction.

Opérations système

Nom	Séquence
Impression de l'affichage	[ON] [L]
Contrôle du contrast	[ON] + ou [ON] -
Attention	[ON]
Arrêt du système	[ON] ▲
Réinitialisation de la mémoire	[ON] [INS] ▶
Test répétitif	[ON] ◀
Test du clavier	[ON] [NEXT]
Annulation d'une opération système	[ON] [DEL]

Impression de l'affichage

Pour imprimer l'affichage en cours :

1. Appuyez sur **[ON]** et maintenez la touche enfoncée.
2. Appuyez sur **[L]** (touche au-dessus de laquelle est imprimé « PRINT »).
3. Relâchez **[ON]**.

Contrôle du contraste

Pour modifier le contraste de l'affichage :

1. Appuyez sur **[ON]** et maintenez la touche enfoncée.
2. Appuyez sur **[+]** pour augmenter le contraste ou sur **[-]** pour le diminuer. Tant que vous maintenez la touche **[ON]** enfoncée, vous pouvez appuyer répétitivement ou de façon continue sur **[+]** ou **[-]** pour déterminer le contraste qui vous convient le mieux.
3. Relâchez **[ON]**.

Opérations d'effacement

Il y a trois opérations d'effacement, indiquées ci-dessous en ordre de sévérité.

Attention

La pression sur **[ON]** exécute « Attention » et vous renvoie à l'affichage normal de la pile. Dans certains cas, il peut être nécessaire d'appuyer deux fois sur **[ON]**. « Attention » a les effets suivants :

- Efface la ligne de commande.
- Annule toute commande ou procédure en exécution.
- Termine les opérations spéciales telles que FORM, PLOT et les catalogues.
- Relance le fonctionnement normal du clavier.

Arrêt du système

Pour arrêter un programme qui ne répond pas à **[ON]**, exécutez un arrêt du système comme indiqué ci-dessous :

1. Appuyez sur **[ON]** et maintenez la touche enfoncée.
2. Appuyez sur **[▲]**.
3. Relâchez **[ON]**.

L'arrêt système a les effets suivants :

- Tous les effets de Attention.
- Efface tous les programmes interrompus et les variables locales.
- Efface la pile.
- Efface les informations sauvegardées pour récupération (CMD, UNDO, LAST).
- Efface le menu personnalisé.
- Sélectionne HOME comme répertoire en cours.
- Active le menu du curseur.

Réinitialisation de la mémoire

Pour réinitialiser toute la mémoire :

1. Appuyez sur **[ON]** et maintenez la touche enfoncée.
2. Appuyez sur **[INS]** et maintenez la touche enfoncée.
3. Appuyez sur **[▶]**.
4. Relâchez **[INS]** et **[▶]**.
5. Relâchez **[ON]**.

La réinitialisation de la mémoire a les effets suivants :

- Tous les effets de Attention et de Arrêt système.
- Elimine toutes les variables utilisateur et tous les répertoires.
- Réinitialise tous les indicateurs utilisateur.
- Déclenche l'avertisseur et affiche Memory Lost au niveau 1.

Opérations de test

Il y a deux opérations permettant de tester le calculateur. La première teste les circuits électroniques de façon répétitive et s'exécute sans intervention de l'utilisateur. La deuxième teste le clavier et demande que vous appuyiez sur toutes les touches dans un ordre prescrit. Les deux tests exécutent un Arrêt système.

Test répétitif

Pour exécuter le test répétitif :

1. Lancez le test.
 - a. Appuyez sur et maintenez la touche enfoncée.
 - b. Appuyez sur .
 - c. Relâchez .
2. Le calculateur affiche successivement des lignes horizontales et verticales, un affichage vide, un motif quelconque et brièvement le résultat du test, puis il recommence le test.
 - Le message OK-288 indique que le test a réussi.
 - Un message tel que 1 FAIL indique que le test a détecté une erreur. Le nombre affiché indique l'erreur.

Si vous interrompez le test en appuyant sur une touche, le test donne un message d'erreur car il n'attendait pas de pression de touche. *Ce message n'indique pas qu'il y a une erreur.*
3. Terminez le test en exécutant un Arrêt système.
 - a. Appuyez sur et maintenez la touche enfoncée.
 - b. Appuyez sur .
 - c. Relâchez .

Test du clavier

Pour exécuter le test du clavier :

1. Lancez le test.
 - a. Appuyez sur **[ON]** et maintenez la touche enfoncée.
 - b. Appuyez sur **[NEXT]**.
 - c. Relâchez **[ON]**.
2. Le calculateur affiche **KEYBOARD TEST**.
 - a. Testez la première ligne du clavier de gauche en appuyant sur **[A]** **[B]** **[C]** **[D]** **[E]** **[F]**.
 - b. Testez les lignes suivantes du clavier de gauche de la même façon.
 - c. Testez la première ligne du clavier de gauche en appuyant sur **[INS]** **[DEL]** **[▲]** **[▼]** **[◀]** **[▶]**.
 - d. Testez les lignes suivantes du clavier de gauche de la même façon. (Appuyez sur la touche **[ON]** dans l'ordre prescrit—il n'interrompra pas le test.)
3. Si vous avez appuyé sur les touches dans l'ordre correct et si le clavier fonctionne correctement, le calculateur affiche **OK-288**. Un message tel que **1 FAIL** indique que vous n'avez pas suivi l'ordre indiqué ou que le calculateur a détecté une erreur. Le nombre indique la nature de l'erreur.
4. Appuyez sur **[ON]**.

3ème partie

Programmation

Page 222	26 : Structures de programmes
234	27 : Programmes interactifs
240	28 : Exemples de programmes

Structures de programmes

De nombreux programmes sont équivalents à une série de calculs au clavier à exécution immédiate. Les objets vont dans la pile et les commandes sont exécutées pour donner le résultat désiré. Ces programmes sont simplement un enregistrement des objets et commandes, écrit dans l'ordre de l'exécution au clavier. Néanmoins, les programmes peuvent effectuer des manipulations impossibles au clavier.

Dans la première partie du manuel, par exemple, vous avez écrit des programmes qui créent des variables locales. La commande spéciale \rightarrow , suivie de un ou plusieurs noms et suivie d'une procédure, est appelée une *structure de variable locale*. Vous ne pouvez pas exécuter la commande \rightarrow à partir du clavier ; elle doit apparaître dans la même programme que les noms et la procédure qui constituent la structure du programme.

Le présent chapitre présente tout d'abord la structure de variable locale. Il décrit ensuite d'autres structures de programme qui permettent d'effectuer des tests et de modifier l'exécution en fonction du résultat. Toutes les commandes pour ces structures de programme se trouvent sans le menu PROGRAM BRANCH. Consultez attentivement le premier exemple de la section « Structures conditionnelles » en page 223, qui présente les concepts utilisés dans le reste de ce chapitre.

Structure de variable locale

Au cours de la première partie, vous avez écrit plusieurs fonctions d'utilisateur, qui représentent l'application la plus importante de la structure de variable locale. Les fonctions utilisateur doivent satisfaire deux conditions :

- Elles doivent spécifier leurs arguments de façon explicite.
- Elles doivent donner un résultat unique.

La fonction utilisateur COT (chaptitre 5), par exemple, a été écrite :

```
« → x 'INV(TAN(x))' »
```

Dans ce cas, la structure de variable locale stocke un argument dans la variable locale x (la première condition est donc satisfaite) et évalue l'expression 'INV(TAN(x))' (la deuxième condition est ainsi satisfaite). La fonction utilisateur $O \rightarrow G$ (chapitre 14) contient un programme au lieu d'une expression mais donne néanmoins un résultat unique, satisfaisant ainsi la deuxième condition.

Ces conditions ne s'appliquent qu'aux fonctions utilisateur. De façon plus générale, les variables locales servent de substitut aux manipulations de la pile. L'exemple suivante permet de calculer la somme et la différence de deux nombres. Du fait que ce calcul donne deux résultats, il ne peut pas être une fonction utilisateur.

```
« → x y « x y + x y - » »
```

Les programmes du chapitre 28 donnent d'autres exemples d'utilisation de structures de variable locale au lieu de fonctions utilisateur, pour éviter les manipulations de pile.

Structures conditionnelles

Les structures conditionnelles permettent à un programme de tester une condition et de prendre une décision en fonction du résultat du test. La présente section donne un exemple de structure conditionnelle et utilise cet exemple pour présenter les structures de programme de façon générale et pour décrire d'autres types de structures conditionnelles.

Supposons que vous écriviez un programme utilisant la variable x et que vous vouliez calculer $(\sin x)/x$. Un problème se pose car le quotient n'est pas défini lorsque $x = 0$. L'exemple suivant donne $(\sin x)/x$ si $x \neq 0$, ou 1 si $x = 0$.

```
IF x 0 ≠ THEN x SIN x / ELSE 1 END
```

Voici le fonctionnement de cette structure lorsque vous exécutez le programme :

1. La commande IF marque simplement le début de la structure. Elle peut se trouver n'importe où avant la commande THEN.
2. X est évalué.
3. Le nombre 0 est placé dans la pile.
4. La commande \neq prend la valeur de X et le nombre 0 comme arguments.
 - Si les arguments ne sont pas égaux, \neq donne 1.
 - Si les arguments sont égaux \neq donne 0.
5. La commande THEN utilise 1 ou 0 comme argument.
 - Si l'argument vaut 1, THEN évalue le programme jusqu'à la commande ELSE (c'est-à-dire $\times \text{ SIN } \times \nearrow$).
 - Si l'argument vaut 0, THEN évalue le programme entre les commandes ELSE et END (c'est-à-dire 1).
6. L'exécution du programme continue après la commande END.

Voici quelques informations sur les structures de programme.

Commandes pour les structures de programme. Les commandes IF, THEN, ELSE et END sont des exemples de *commandes de structure de programme*. Leur utilisation doit respecter les combinaisons décrites dans ce chapitre.

Fonctions et commandes de test. La fonction \neq est une *fonction de test*. Pour deux arguments quelconques, \neq donne 1 ou 0, selon que le test est vrai ou faux. Les autres fonctions de test sont $<$, \leq , \geq , $>$ et $==$. (Souvenez-vous que $=$ sert pour identifier les équations, pas pour tester les égalités.) Pour deux arguments symboliques, les fonctions de test donnent des résultats symboliques.

Certaines commandes de test donnent toujours 1 ou 0. La commande de test SAME, par exemple, est similaire à $==$, mais elle teste simplement si les deux objets sont identiques. D'autres commandes de test sont disponibles pour manipuler les indicateurs (voir ci-après). La section « TESTS DE PROGRAMME » du Manuel de référence décrit les fonctions et les commandes de test plus en détail.

Indicateurs. Les nombres 1 et 0 donnés par les commandes de test sont appelés *indicateurs de pile*. Il indiquent si un test est vrai ou faux, 1 est appelé *l'indicateur vrai* et 0 est appelé *l'indicateur faux*.

Le terme « indicateur » fait en outre référence aux *indicateurs utilisateur* intégrés. Ces derniers sont numérotés 1 à 64 ; les indicateurs 31 à 64 ont des significations particulière pour le calculateur alors que les indicateurs 1 à 30 peuvent représenter toute condition vraie/fausse que vous choisissiez. Vous pouvez stocker un indicateur de pile dans un indicateur utilisateur car ils représentent tous deux le résultat vrai ou faux d'un test. Par exemple, la séquence

```
IF A B < THEN 12 SF ELSE 12 CF END
```

arme l'indicateur 12 si $A < B$, ou le désarme si $A \geq B$. Vous pouvez ensuite tester si l'indicateur 12 est armé en utilisant la séquence

```
IF 12 FS? THEN ...
```

qui donne la même valeur que le test $A B <$ initial. L'avantage de cette technique est que la valeur du test est conservée, même si les valeurs A et B changent. Les commandes permettant de modifier et de tester les indicateurs utilisateur sont présentées dans la section « TEST DE PROGRAMME » du Manuel de référence. Le terme « indicateur » fera référence à ce type d'indicateur dans la suite du manuel.

Clauses. L'ensemble des objets et des commandes situées entre deux commandes de structure de programme s'appelle une *clause*. Chaque clause est traitée comme une entité unique et est étiquetée par son rôle logique ou par la commande qui la précède. Dans le premier exemple :

- La clause entre IF et THEN ($\times \emptyset \neq$) est appelée un *test* (ou *clause IF*).
- La clause entre THEN et ELSE ($\times \text{ SIN } \times \nearrow$) est appelée *clause vraie* ou *clause THEN*.
- La clause entre ELSE et END (1) est appelée *clause fausse* ou *clause ELSE*.

Les clauses de l'exemple représentent des calculs numériques simples, mais vous pouvez inclure des séquences quelconques d'objets et de commandes. Une clause est en effet une sorte de sous-programme. Si vous écrivez un programme séparé contenant la clause et si vous stockez ce programme dans une variable, vous pouvez utiliser le nom de la variable comme clause. Dans un tel cas, un structure simple comme la suivante

```
IF A THEN B ELSE C END
```

peut représenter un processus de décision complexe avec deux résultats complexes possibles, selon les contenus de A, B et C.

IF ... THEN ... ELSE ... END

Conformément à la terminologie préalablement définie, l'évaluation de cette structure conditionnelle peut être décrite comme suit : la clause IF est évaluée et donne un indicateur. Si l'indicateur est vrai, la clause THEN est évaluée, sinon la clause ELSE est évaluée.

Cette structure est illustrée dans la section « FIB2 (Suite de Fibonacci, boucle) » en page 248.

IFTE (fonction If-Then-Else-End)

Le premier exemple de ce chapitre peut être écrit en syntaxe algébrique en utilisant la fonction IFTE :

```
' IFTE(X≠0,SIN(X)/X,1) '
```

Cette forme est très pratique pour les calculs symboliques. Si vous exécutez la version de structure de programme lorsque X est indéfini, cette expression algébrique constitue le résultat. Les arguments de IFTE doivent être représentables en syntaxe algébrique. Si vous voulez utiliser des commandes RPN dans l'expression conditionnelle, vous devez utiliser la structure de programme.

La fonction IFTE est utilisée dans la section « FIB1 (Suite de Fibonacci, version récursive) » en page 247.

IF ... THEN ... END

Si vous n'avez pas besoin de la clause ELSE, vous pouvez l'omettre de la structure. L'exemple suivant vérifie si l'objet au niveau 1 est plus grand que celui au niveau 2 et les échange si nécessaire.

```
IF DUP2 < THEN SWAP END
```

Remarquez l'utilisation de DUP2 pour copier les objets. Les copies sont ensuite détruites par la comparaison $<$. La section « SORT (tri d'une liste) » en page 270 fournit un autre exemple de IF ... THEN ... END.

IFT (commande If-Then-End)

Vous auriez pu écrire l'exemple précédent en utilisant la commande IFT au lieu de la structure de programme :

```
DUP2 < « SWAP » IFT
```

La séquence DUP2 < laisse un indicateur dans la pile, le programme « SWAP » se place dans la pile et la commande IFT utilise l'indicateur et le programme comme arguments. Si l'indicateur est vrai, IFT évalue le programme ; si l'indicateur est faux, IFT élimine le programme. Le résultat est identique à celui de la structure de programme.

Détection d'erreur

Dans certains cas, vous pouvez prédire qu'une erreur est possible pendant l'exécution d'un programme. Les erreurs arrêtent normalement l'exécution des programmes ; mais vous pouvez utiliser *détection* l'erreur en plaçant la commande qui la provoque dans une structure particulière.

Le problème $(\sin x)/x$, par exemple donne une erreur « Infinite Result » lorsque $x = 0$. Néanmoins, vous pouvez définir $(\sin 0)/0 = 1$ en utilisant :

```
IFERR X SIN X / THEN DROP2 1 END
```

Ceci signifie : évaluer la clause IFERR (X SIN X /). Si une erreur a lieu, évaluer la clause THEN (DROP2 1).

Cet exemple contient la commande DROP2 pour éliminer les deux zéros qui provoquent l'erreur. Remarquez que ceci suppose que LAST est actif. Si LAST est désactivé, les zéros ne sont pas présents et la commande DROP2 ne peut pas être utilisée. Vous devez donc vérifier l'état de LAST lorsque vous utilisez les détections d'erreur.

Un autre exemple de IFERR ... THEN ... END se trouve dans « BDISP (affichage binaire) » en page 259. Vous pouvez en outre inclure une clause ELSE à évaluer s'il n'y a pas d'erreur :

```
IFERR ... THEN ... ELSE ... END
```

Structures de boucles finies

Les structures de boucles contiennent une *clause de boucle* qui est évaluée de façon répétitive. Dans une *structure de boucle finie*, le programme spécifie en avance le nombre d'évaluations de la boucle. Vous pouvez aussi construire une structure de programme, appelée *structure de boucle indéfinie*, qui utilise un test pour déterminer si la boucle doit être évaluée à nouveau. Cette section décrit les structures de boucle finies ; les structures de boucles indéfinies sont décrites en page 231.

START ... NEXT

L'exemple suivant émet quatre signaux sonores.

```
1 4 START 440 ,1 BEEP NEXT
```

Cette structure fonctionne de la façon suivante :

1. La commande START prend les valeurs 1 et 4 dans la pile et crée un *compteur*. Le compteur sera utilisé pour savoir combien de fois la boucle doit être exécutée. La valeur 1 spécifie la *valeur initiale* du compteur et la valeur 4 spécifie la *valeur finale*.

2. La clause de boucle `440 , 1 BEEP` est exécutée.
3. La commande `NEXT` ajoute 1 au compteur.
4. La valeur en cours du compteur est comparée à la valeur finale.
 - Si la valeur en cours du compteur est inférieure à la valeur finale, les étapes (2), (3) et (4) sont répétées.
 - Si la valeur en cours du compteur dépasse la valeur finale, la structure de boucle finie est terminée et l'exécution du programme continue après la commande `NEXT`.

Dans cet exemple, les étapes (2), (3) et (4) sont répétées quatre fois. Le compteur de boucle est incrémenté tout d'abord de 1 à 2, puis à 3, à 4 et enfin à 5. Lorsque la valeur du compteur vaut 5, elle est supérieure à 4 et le calculateur met alors fin à la structure de boucle finie. Remarquez que l'étape (1) est effectuée avant l'exécution des tests, de telle façon que la clause de boucle soit toujours effectuée au moins une fois. La section « FIB2 (suite de Fibonacci) » en page 248 donne un autre exemple de `START ... NEXT`.

FOR compteur... NEXT

Dans de nombreux cas, il est pratique d'utiliser la valeur en cours du compteur comme variable dans la clause de boucle. Pour cela, il suffit de remplacer `START` par `FOR nom`. Le compteur devient alors une variable locale portant le nom spécifié. Comme précédemment, ce manuel présente les noms locaux en minuscules de façon à les différencier des noms globaux. L'exemple suivant place dans la pile les carrés des entiers de un à cinq.

```
1 5 FOR x x SQ NEXT
```

La séquence `FOR x` n'est exécutée qu'une seule fois. La séquence `x SQ` est la clause de boucle, qui est exécutée de façon répétitive.

Les exemples précédents ont tous utilisés une valeur initiale de compteur égale à 1 ; néanmoins, vous pouvez utiliser toute valeur entière. Le compteur étant utilisé comme une variable, définissez les valeurs initiales et finales du compteur égales aux valeurs des variables initiale et finale désirées. L'exemple suivant place les carrés des entiers de trois à neuf.

```
3 9 FOR x x SQ NEXT
```

Un autre exemple apparaît dans « BDISP (affichage binaire) » en page 259.

... *incrément* STEP

La commande NEXT incrémente toujours le compteur de 1. Pour spécifier un incrément différent, remplacez NEXT par *n* STEP, où *n* est l'incrément désiré. STEP est souvent utilisé après FOR *compteur*, comme indiqué dans les exemples ci-dessous, mais il peut aussi être utilisé après START. L'exemple suivant place les carrés des entiers impairs entre 1² et 5².

```
1 5 FOR x x SQ 2 STEP
```

La clause de boucle *x SQ 2* est exécutée trois fois. La commande STEP incrémente d'abord le compteur de 1 à 3, puis à 5 et enfin à 7. Lorsque le compteur vaut 7, il est supérieur à la valeur finale 5 et le calculateur termine la structure de boucle finie.

Les exemples ont jusqu'à présent tous utilisés des valeurs *croissantes* du compteur. Vous pouvez utiliser des valeurs *décroissantes* en spécifiant un incrément négatif. L'exemple suivant place dans la pile les carrés des entiers impaires entre 5² et 1².

```
5 1 FOR x x SQ -2 STEP
```

La séquence *-2 STEP* décrémente le compteur de 5 à 3, puis à 1 et enfin à -1. Lorsque la valeur du compteur vaut -1, elle est inférieure à 1 et le calculateur termine la structure de boucle fine.

Le programme « SORT (tri d'une liste) » en page 270 utilise *-1 STEP* pour décrémente le compteur d'une unité. Dans ce cas, la commande STEP modifie la valeur d'une unité, comme le ferait NEXT, mais la valeur décroît au lieu d'augmenter.

Structures de boucles indéfinies

Si vous ne pouvez pas spécifier en avance le nombre de répétition d'une boucle, vous pouvez écrire une *structure de boucle indéfinie* qui contient à la fois une clause de boucle et une clause de test. Les clauses sont exécutées alternativement et le résultat du test détermine si l'exécution doit continuer.

Cette section décrit deux types de structures de boucle indéfinie. La première, `DO... UNTIL... END`, exécute la clause de boucle avant le test. La clause de boucle est par conséquent toujours exécutée au moins une fois. Le deuxième type, `WHILE... REPEAT... END`, exécute le test en premier. Dans certains cas, la clause de boucle n'est donc jamais exécutée.

DO...UNTIL...END

L'exemple suivant évalue répétitivement un objet jusqu'à ce que l'évaluation ne modifie plus l'objet. L'évaluation devant avoir lieu au moins une fois avant qu'un test puisse être effectué, l'exemple utilise la structure `DO... UNTIL... END`.

```
DO DUP EVAL UNTIL DUP ROT SAME END
```

Cette structure fonctionne de la façon suivante :

1. La clause de boucle `DUP EVAL` est exécutée et laisse l'objet et le résultat de l'évaluation dans la pile.
2. La clause de test `DUP ROT SAME` est exécutée et laisse dans la pile le résultat de l'évaluation et un indicateur. L'indicateur permet de savoir si l'objet et le résultat sont identiques.
3. L'indicateur est pris dans la pile et sa valeur détermine si la structure de boucle doit être répétée.
 - Si l'indicateur est faux, les étapes (1), (2) et (3) sont répétées.
 - Si l'indicateur est vrai, la structure de boucle est terminée.

Supposons que vous vouliez évaluer complètement ' $A+B$ ', lorsque A contient ' $P+Q$ ' et P contient 2. La première évaluation de la clause de boucle donne ' $A+B$ ' et ' $P+Q+B$ '. Ces expressions sont différentes, le calculateur évalue par conséquent la boucle une autre fois et donne ' $P+Q+B$ ' et ' $2+Q+B$ '. Ces expressions sont encore différentes ; la clause de boucle est donc évaluée une troisième fois et donne ' $2+Q+B$ ' et ' $2+Q+B$ '. Ces expressions sont maintenant identiques, et la calculateur termine la structure de boucle.

L'effet de cet exemple est similaire à celui de $\rightarrow\text{NUM}$, sauf que $\rightarrow\text{NUM}$ donne une erreur si un nom est indéfini. La section « MULTI (exécution multiple) » en page 253 présente une version plus performante.

WHILE ... REPEAT ... END

L'exemple suivant prend un nombre quelconque de vecteurs dans la pile et les ajoute à la matrice statistique en cours. Du fait que l'exemple doit vérifier si l'objet présent au niveau 1 est un vecteur *avant* de l'ajouter, l'exemple utilise la structure WHILE ... REPEAT ... END.

```
WHILE DUP TYPE 3 == REPEAT Σ+ END
```

1. La clause de test `DUP TYPE 3 ==` est évaluée et laisse un indicateur dans la pile. L'indicateur permet de déterminer si l'objet au niveau 2 est un vecteur réel.
2. L'indicateur est pris dans la pile. Sa valeur détermine si la clause de boucle est exécutée.
 - Si l'indicateur est vrai, la clause de boucle `Σ+` est exécutée, elles ajoutent le vecteur à la matrice statistique en cours et les étapes (1) et (2) sont répétées.
 - Si l'indicateur est faux, la structure de boucle se termine.

Remarquez que WHILE ... REPEAT ... END se termine lorsque l'indicateur est faux, mais que DO ... UNTIL ... END se termine lorsque l'indicateur est vrai. Vous pouvez inverser la valeur vraie ou fausse d'une clause de test en ajoutant NOT comme dernière commande : WHILE ... NOT REPEAT ou UNTIL ... NOT END.

La structure WHILE ... REPEAT ... END est aussi illustrée dans la section « PAD (ajout d'espaces de remplissage) » en page 257.

Structures de programmes emboîtées

Une structure de programme étant similaire à un sous-programme, une clause peut elle-même contenir une structure de programme. La structure située dans la clause est appelée une structure *interne* et celle contenant la clause est appelée structure *externe*. Le programme « SORT (tri d'une liste) » en page 270 démontre les boucles finies emboîtées.

Le calculateur ne limite pas le nombre de niveau d'emboîtements. Dans certains cas, il peut être plus facile de stocker les structures internes dans des programmes et d'utiliser leurs noms comme clause dans les structures externes.

Programmes interactifs

Certains programmes nécessitent des informations, de la part de l'utilisateur, en cours d'exécution. Le programme peut demander à l'utilisateur soit de fournir des valeurs, soit d'effectuer un choix.

Le présent chapitre présente la façon dont un programme peut demander une valeur ou un choix, en utilisant les commandes suivantes du menu PROGRAM CONTROL.

Commande	Description
HALT	Arrête l'exécution du programme.
s WAIT	Arrête l'exécution du programme pendant s secondes.
KEY	Renvoie une chaîne de touche si vous appuyez sur une touche.
f s BEEP	Emet un signal sonore de fréquence f pendant s secondes.
CLLCD	Efface l'affichage.
n DISP	Affiche un objet dans la ligne n de l'affichage.
CLMF	Rappelle l'affichage normal lorsque l'exécution reprend.

Demande de saisie

La séquence suivante crée un menu de saisie personnalisé pour les variables A, B et C, émet un signal sonore pour avertir l'utilisateur et s'arrête pour attendre une donnée.

```
... ( STO A B C ) MENU 440 , 1 BEEP HALT ...
```

Le menu affiché contient les libellés `A`, `B`, and `C`, qui ressemblent aux libellés du menu de l'algorithme de résolution d'équations. Après avoir entré la valeur dans la pile, l'utilisateur peut simplement appuyer sur l'une de ces touches pour stocker la valeur dans la variable correspondante. Après l'entrée des valeurs, l'utilisateur doit appuyer sur `CONT` pour relancer l'exécution du programme.

Demande de choix

Pour les tâches complexes, il est souvent mieux d'écrire un série de petits programmes, exécutant chacun une tâche limitée.

Supposons qu'une tâche soit terminée et que l'utilisateur doivent choisir parmi les tâches SAUT, BOND et DETOUR. La séquence suivante crée un menu utilisateur personnalisé pour les programmes SAUT, BOND et DETOUR, émet un signal sonore pour avertir l'utilisateur et arrête l'exécution.

```
... ( SAUT BOND DETOUR ) MENU 440 ,1 BEEP »
```

Le menu affiché contient les libellés `SAUT`, `BOND` et `DETO`, qui ressemblent à ceux du menu User. Lorsque l'utilisateur appuie sur l'une des touche de menu, le calculateur exécute la tâche suivante. Cette tâche peut se terminer par une séquence similaire, offrant à l'utilisateur un nouveau jeu d'options.

Exemple plus complexe

L'exemple ci-dessous affiche un message, attend que l'utilisateur appuie sur une touche et vérifie que cette touche est définie (c'est-à-dire qu'elle représente un des choix). Si la touche est définie, l'action correspondante est effectuée ; si la touche n'est pas définie, le calculateur affiche un message d'erreur et le processus recommence.

Une structure `DO ... UNTIL ... END` « externe » se répète jusqu'à ce que l'utilisateur appuie sur une touche définie. La clause `DO` externe contient une structure `DO ... UNTIL ... END` « interne » qui se répète jusqu'à ce que l'utilisateur appuie sur une touche.

La clause UNTIL externe contient une condition qui affiche un message d'erreur si la touche n'est pas définie. Dans la liste ci-dessous, les indentations identifient la structure externe, les clauses, les structures interne et leurs clauses.

Séquence	Commentaires
{ "Ananas" "Banane" "Cerise" }	Cette liste contient les résultats possibles. Elle reste dans la pile jusqu'à ce que la structure DO...UNTIL...END suivante donne 1, 2 ou 3, indiquant le choix de l'utilisateur.
DO	Commence la clause de la boucle externe. Cette clause affiche les messages d'option, qui présente à l'utilisateur les choix et en demande une réponse.
CLLCD	Efface l'affichage.
"Appuyez sur "	Le message d'option pour la ligne 1.
1 DISP	Affiche le message.
" [A] pour Ananas"	Le message d'option pour la ligne 2.
2 DISP	Affiche le message.
" [B] pour Banane"	Le message d'option pour la ligne 3.
3 DISP	Affiche le message.
" [C] pour Cerise"	Le message d'option pour la ligne 4.
4 DISP	Affiche le message.
DO UNTIL KEY END	Cette boucle interne non définie se répète jusqu'à ce que l'utilisateur appuie sur une touche. La commande KEY donne 0 si vous n'avez pas appuyé sur une touche, ou donne une chaîne (représentant la touche) et 1 si vous avez appuyé sur une touche. A la fin de la boucle, la chaîne reste dans la pile.
UNTIL	Commence la clause de test externe. Cette clause vérifie si la touche est une touche définie.

("A" "B" "C")	Cette liste contient les touches définies. Chaque touche définie correspond à une issue possible.
SWAP POS	Fait correspondre la chaîne de touche à la liste de touches définies. POS (<i>position</i>) donne 1 si la elle est "A", 2 si elle est "B", 3 si elle est "C" ou 0 s'il n'y a pas de correspondance.
IF DUP	Copie la position pour l'utiliser comme indicateur. Si la position est 1, 2 ou 3, exécute la clause THEN. Si la position est 0, exécute la clause ELSE.
THEN 1	La touche est une touche définie, l'indicateur vrai est placé dans la pile.
ELSE	La touche n'est pas définie, affiche un message d'erreur et émet un signal sonore.
CLLCD "TOUCHE ERRONEE"	Affiche un message d'erreur.
1 DISP	
440 ,1 BEEP	Emet un signal sonore.
1 WAIT	Attend une seconde.
END	Termine la structure IF ... THEN ... ELSE ... END. Si la touche est définie, la position et l'indicateur vrai se trouve dans la pile. Si elle n'est pas définie, seule la position (qui est en outre un indicateur faux) se trouve sur la pile.
END	Termine la boucle non définie externe. Si la touche est définie, la boucle se termine en laissant la position dans la pile. Si la touche n'est pas définie, la clause de boucle se répète.
GET	Détermine le résultat en fonction d'une liste d'issues possibles et d'une position.

EVAL

Evalue le résultat. Dans cet exemple, EVAL n'a aucun effet car le résultat est une chaîne. Dans une application réelle, le résultat peut être un programme (peut-être stocké dans une variable), et EVAL serait dans ce cas nécessaire.

CLMF

Donne l'affichage normal de la pile.

Lors de l'exécution de cette séquence, l'utilisateur voit les messages d'option.

```
Appuyez sur
[A] pour Ananas
[B] pour Banane
[C] pour Cerise
```

Si l'utilisateur appuie sur une touche autre que [A], [B] ou [C], le calculateur émet un signal sonore et affiche le message d'erreur pendant une seconde.

```
Bad key
```

Les messages d'options réapparaissent ensuite. Lorsque l'utilisateur appuie sur [A], [B] ou [C], le calculateur place la chaîne "Ananas", "Banane" ou "Cerise" au niveau 1.

Vous pouvez rendre cette séquence plus significative en modifiant la liste des issues possibles, les messages d'option et la liste des touches définies. De façon plus générale, l'utilisation de variables locales et le stockage de la séquence dans une structure de variable locale permet de créer le programme suivant.

```

« → touches m1 m2 m3 m4
« DO CLLCD
  m1 1 DISP
  m2 2 DISP
  m3 3 DISP
  m4 4 DISP
  DO UNTIL KEY END
UNTIL touches SWAP POS
  IF DUP
  THEN 1
  ELSE CLLCD "TOUCHE ERRONEE" 1 DISP
    440 ,1 BEEP 1 WAIT
  END
END
»
GET EVAL CLMF
»

```

Si vous stockez ce programme dans une variable nommée TOUCHE?, vous pouvez effectuer l'exemple précédent en exécutant

```

{ "Ananas" "Banane" "Cerise" }
{ "A" "B" "C" }
"Appuyez sur "
" [A] for Ananas"
" [B] for Banane"
" [C] for Cerise"
TOUCHE?

```

Exemples de programme

Ce chapitre contient 20 programmes pour votre HP-28S. Ces programmes démontrent les techniques de programmation. Chaque programme est accompagné des informations suivantes.

- **Diagramme de pile.** Un diagramme de pile est un tableau à deux colonnes contenant les « Arguments » et les « Résultats ». La colonne Arguments indique ce qui doit se trouver dans la pile avant l'exécution du programme ; la colonne Résultats montre ce qui se trouve dans la pile après l'exécution du programme.

Le diagramme de pile n'indique pas tout ce qui se passe ; un programme qui modifie la mémoire utilisateur ou affiche des objets peut n'avoir aucun effet sur le pile.

- **Techniques.** Lorsque vous aurez bien étudié une technique dans ce chapitre, vous pourrez l'utiliser dans vos propres programmes.
- **Programmes nécessaires.** Certains programmes en appellent d'autres comme des sous-routines. Vous pouvez entrer les programmes nécessaires et le programme principal dans un ordre quelconque, mais vous devez les entrer tous avant d'exécuter le programme principal.
- **Programme et commentaires.** Ce chapitre formate le listage de programme pour illustrer une structure de programme et le processus. Il n'est pas nécessaire de suivre le format du listage lorsque vous entrez un programme. Néanmoins, veuillez à entrez des espaces lorsqu'ils apparaissent dans le listage ou entre les objets apparaissant sur des lignes différentes.

Vous pouvez entrer un programme caractère par caractère, ou vous pouvez utiliser les menus pour entrer directement des commandes. Le résultat est le même tant que le programme correspond au listage.

Lorsque vous entrez le programme, vous pouvez omettre toutes les parenthèse de droite et tous les délimiteurs de droite *qui apparaissent à la fin du programme*. Lorsque vous appuyez sur **ENTER**, le calculateur ajoute des parenthèses et délimiteurs pour vous.

- Exemple. Les exemples supposent que le format d'affichage STD est actif. Pour choisir le format d'affichage STD, appuyez sur STD **ENTER** ou utilisez le menu MODE.

La technique la plus importante présentée dans ce chapitre est la *programmation structurée* : chaque programme global est divisé en tâches simples correspondant chacune à un programme. Les programmes suivants sont utilisés par d'autres programmes de ce chapitre.

- SBOITE est utilisé dans RBOITE.
- MULTI est utilisé dans EXCO.
- COMPLETE et PRESERVE sont utilisés dans BAFFICH.
- Σ LIT est utilisé dans $\Sigma X2$, $\Sigma Y2$ et ΣXY .
- TRI et LMED sont utilisés dans MEDIANE.

Fonctions de boîtes

Cette section contient deux programmes :

- SBOITE calcule la surface totale d'une boîte.
- RBOITE utilise SBOITE pour calculer le rapport entre la surface et le volume d'une boîte.

SBOITE (Surface d'une boîte)

Ce programme calcule la surface totale des six faces d'une boîte en fonction de la longueur, largeur et hauteur.

Arguments	Résultats
3 : hauteur	3 :
2 : largeur	2 :
1 : longueur	1 : surface

Techniques :

- Structure de variable locale. Les variables locales vous permettent d'affecter des noms aux arguments sans créer de conflit avec les variables globales. De même que les variables globales, les variables locales permettent d'utiliser plusieurs fois un même argument sans avoir à se souvenir de sa position dans la pile. Contrairement aux variables globales, les variables locales disparaissent lorsque la structure de programme qui les a créé se termine.

Une structure de variable locale comporte trois parties.

1. Commande « \rightarrow ». Lorsque vous entrez cette commande, veillez à les faire précéder et suivre d'espaces. (Comme toute commande, \rightarrow s'écrit avec des caractères normaux et n'est reconnue que si elle est séparée des autres objets par des espaces. Ne confondez pas cette commande d'un seul caractère avec les délimiteurs tels que # ou *.)
2. Un ou plusieurs noms.
3. Une procédure (expression, équation ou programme) comprenant les noms. Cette procédure est appelée procédure de *définition*.

Lors de l'évaluation d'une structure de variable locale, le calculateur crée une variable locale pour chaque nom. Les valeurs des variables locales sont prises dans la pile. La procédure de définition est ensuite évaluée, après substitution des valeurs aux variables locales. Une comparaison de la version de SBOITE ci-dessous et de celle présentée en page 244, illustre la puissance des variables locales.

- Fonction utilisateur. Ce type de programme fonctionne en syntaxe RPN ou algébrique. Une fonction utilisateur est un programme qui consiste uniquement d'une structure de variable locale et qui donne un résultat unique.

Programme

Commentaires

* \rightarrow h l a l o	Commence le programme. Crée les variables locales nécessaires pour la hauteur, la longueur et la largeur. Conformément aux conventions établies au début du manuel, nous utiliserons des minuscules pour les noms de variables locales. Les valeurs sont prises dans la pile (syntaxe RPN) ou dans les arguments de la fonction utilisateur (syntaxe algébrique).
------------------------------	--

Programme

'2*(h*1a+h*1o+1a*1o)'

*

ENTER

SBOITE STO

Commentaires

L'expression de définition pour la surface. L'évaluation de la fonction utilisateur provoque l'évaluation de cette expression, et renvoie la surface dans la pile. Termine le programme.

Place le programme dans la pile. Stocke le programme sous le nom SBOITE.

Exemple. Un des avantages des fonctions utilisateur est qu'elles fonctionnent aussi bien en syntaxe RPN qu'algébrique. Calculez la surface d'une boîte de 12 cm de haut, 16 cm de large et 24 cm de long. Effectuez le calcul d'abord en syntaxe RPN puis en syntaxe algébrique.

Pour la version RPN, entrez d'abord la hauteur et la largeur.

USER

12 ENTER

16 ENTER

3:					
2:					12
1:					16
SBOIT					

Entrez ensuite la longueur et exécutez SBOITE.

24 SBOIT

3:					
2:					
1:					1728
SBOIT					

La surface est 1 728 cm carré.

Essayez maintenant la version algébrique.

SBOIT (12,16,24 EVAL

3:					
2:					1728
1:					1728
SBOIT					

La surface est à nouveau 1 728.

SBOITE sans variables locales

Le programme suivant utilise uniquement les opérations sur la pile pour calculer la surface d'une boîte. Comparez ce programme à SBOITE.

Arguments	Résultats
3 : hauteur	3 :
2 : largeur	2 :
1 : longueur	1 : surface

Programme

```
⌘
DUP2 *
ROT
4 PICK
*
+
ROT ROT

*
+
2 *
⌘
```

Commentaires

Commence le programme.
Calcule *lalo*.
Place *la* au niveau 1.
Copie *h* au niveau 1.
Calcule *lalo*.
Calcule *lalo* + *lah*.
Place *h* et *lo* dans les niveaux 2 et 1.
Calcule *hlo*.
Calcule *lalo* + *lah* + *hlo*.
Calcule 2(*lalo* + *lah* + *hlo*).
Termine le programme.

Cette version de SBOITE n'étant pas une fonction utilisateur, elle ne peut pas être utilisée en syntaxe algébrique.

RBOITE (rapport de la surface et du volume d'une boîte)

Ce programme calcule le rapport entre la surface et le volume d'une boîte en fonction de sa hauteur, sa largeur et sa longueur.

Arguments	Résultats
3 : hauteur	3 :
2 : largeur	2 :
1 : longueur	1 : surface/volume

Techniques :

- Fonctions utilisateur emboîtées. RBOITE est une fonction utilisateur dont l'expression de définition utilise SBOITE. RBOITE peut lui-même être utilisé pour définir d'autres fonctions utilisateur.

Souvenez-vous que SBOITE a été défini avec les variables locales *h*, *la* et *lo*, et remarquez que, ci-dessous, SBOITE utilise *x*, *y* et *z* comme arguments dans la définition de RBOITE. Il n'est pas nécessaire que les variables locales des deux définitions se correspondent, car les deux jeux de variables sont indépendants. Il est cependant essentiel que les variables locales soient homogènes dans une même définition.

Programme

```
«  
  → x y z  
  
  'SBOITE(x,y,z)  
  
  /(x*y*z)'  
  
»
```

ENTER

▣ RBOITE ▣ STO

Commentaires

Commence le programme.

Crée les variables locales nécessaires pour la hauteur, la largeur et la longueur. Ce programme utilise *x*, *y* et *z*, au lieu de *h*, *la* et *lo*.

Commence l'expression de définition avec la fonction utilisateur SBOITE.

Divise la surface par le volume de la boîte.

Termine le programme.

Place le programme dans la pile. Stocke le programme sous le nom RBOITE.

Exemple. Calculez le rapport entre la surface et le volume d'une boîte de 9 cm de haut, 18 cm de large et 21 cm de long. Effectuez le calcul tout d'abord en notation RPN puis en notation algébrique.

Pour la notation RPN, entrez d'abord la hauteur et la largeur.

USER
9 ENTER
18 ENTER

3:				
2:				9
1:				18
RBOIT	RBOIT			

Entrez ensuite la longueur et exécutez RBOITE.

21 RBOIT

3:				
2:				
1:				,428571428571
RBOIT	RBOIT			

Le rapport est 0,428571428571.

Essayez maintenant la version algébrique.

▢ RBOIT (9.18.21 EVAL

3:				
2:				,428571428571
1:				,428571428571
RBOIT	RBOIT			

Le rapport est à nouveau 0,428571428571.

Suite de Fibonacci

Etant donné un entier n , calculez le n ème nombre de Fibonacci F_n , où

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}$$

Cette section contient deux programmes, illustrant chacun une approche différente du problème.

- FIB1 est une fonction utilisateur définie de façon *réursive* ; son expression de définition contient son propre nom. FIB1 est une fonction courte et explicite, qui peut être utilisée dans les objets algébriques.
- FIB2 est un programme contenant une boucle définie. Ce programme ne peut pas être utilisé dans les objets algébriques, est plus long et plus complexe que FIB1, mais est aussi plus rapide.

FIB1 (Nombres de Fibonacci, par récursion)

Arguments	Résultats
1 : n	1 : F_n

Techniques :

- IFTE (fonction If-Then-Else). L'expression de définition de FIB1 contient une fonction conditionnelle IFTE qui peut être utilisée en syntaxe RPN ou algébrique.)
- Récursion. L'expression de définition de FIB1 est écrite en fonction de FIB1, de la même façon que F_n est défini en fonction de $F_n - 1$ et $F_n - 2$.

Programme

```

*
→ n
'

  IFTE<n≤1,
  n,
  FIB1<n-1>+FIB1<n-2>
'
*
```

ENTER

▢ FIB1 ▢ STO

Commentaires

Commence le programme.

Définit une variable locale.

Commence l'expression de définition.

Si $n \leq 1$,

Alors $F_n = n$;

Sinon $F_n = F_{n-1} + F_{n-2}$.

Termine l'expression de définition.

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom FIB1.

Exemple. Calculez F_6 en utilisant la syntaxe RPN et F_{10} en utilisant la syntaxe algébrique.

Calculez tout d'abord F_6 en RPN.

USER

6 FIB1

3:	
2:	
1:	8
FIB1 R0017 R0017	

Calculez ensuite F_{10} en algébrique.

▢ FIB1 (10 EVAL

3:	
2:	8
1:	55
FIB1 R0017 R0017	

FIB2 (Nombres de Fibonacci, par boucle)

Arguments	Résultats
1 : n	1 : F_n

Techniques :

- IF ... THEN ... ELSE ... END. FIB2 utilise une structure de programme comme condition. (FIB1 utilise IFTE.)
- START ... NEXT (boucle définie). Pour calculer F_n , FIB2 commence avec F_0 et F_1 , puis répète une boucle pour calculer les F_i successifs.

Programme

```

«
→ n
«
IF n 1 ≤
THEN n
ELSE
  0 1
  2 n
  START
  DUP

  ROT

  +

  NEXT
  SWAP DROP
END
»
»
[ENTER]
[FIB2] [STO]

```

Commentaires

Commence le programme.
 Crée une variable locale.
 Commence le programme de définition.
 Si $n \leq 1$,
 Alors $F_n = n$;
 Commence la clause ELSE.
 Place F_0 et F_1 dans la pile.
 De 2 à n ,
 Effectue la boucle suivante :
 Effectue une copie du dernier F (initialement F_1).
 Place le F précédent (initialement F_0) au niveau 1.
 Calcule le F suivant (initialement F_2).
 Répète la boucle.
 Elimine F_{n-1} .
 Termine la clause ELSE.
 Termine le programme de définition.
 Termine le programme.
 Place le programme dans la pile.
 Stocke le programme sous le nom FIB2.

Exemple. Calculez F_6 et F_{10} . Remarquez que FIB2 est plus rapide que FIB1.

Calculez F_6 .

USER

6 FIB2

3:						
2:						8
1:						
FIB2 FIB1 REOIT SEOIT						

Calculez F_{10} .

10 FIB2

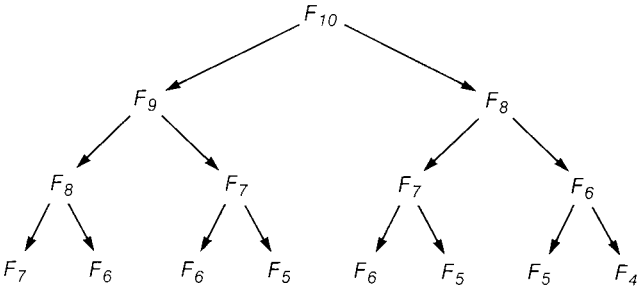
3:						
2:						8
1:						55
FIB2 FIB1 REOIT SEOIT						

Comparaison de FIB1 et FIB2

FIB1 calcule plusieurs fois les valeurs intermédiaires de F_i , alors que FIB2 ne les calcule qu’une seule fois. FIB2 est donc plus rapide.

La différence de vitesse augmente avec la valeur de n car la durée du calcul de FIB1 croît de façon exponentielle en fonction de n , alors que la durée du calcul de FIB2 croît de façon linéaire en fonction de n .

Le diagramme ci-dessous illustre les premières étapes du calcul de F_{10} avec FIB1. Remarquez le nombre de calculs intermédiaires : 1 dans la première ligne, 2 dans la deuxième, 4 dans la troisième et 8 dans la quatrième.



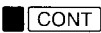
Exécution pas à pas

Il est plus simple de comprendre le fonctionnement d'un programme si vous l'exécutez pas à pas, pour étudier son effet sur la pile. Ceci peut vous aider à corriger les erreurs de programmation et à comprendre les programmes écrits par d'autres personnes.

Cette section vous montre comment exécuter FIB2 pas à pas. Vous pourrez ensuite appliquer ces techniques à tout programme. Les règles générales sont les suivantes :

1. Utilisez VISIT pour insérer la commande HALT dans le programme. Placez HALT là où vous voulez commencer l'exécution pas à pas. (Vous allez voir comment la position de HALT dans FIB2 affecte l'exécution.)
2. Exécutez le programme. Lorsque la commande HALT est exécutée, le programme s'arrête (indiqué par le témoin d'arrêt).
3. Sélectionnez le menu PROGRAM CONTROL.
4. Appuyez une fois sur **SST** pour afficher puis exécuter l'étape suivante du programme.

Vous pouvez maintenant :

- Maintenir la touche **SST** enfoncée pour afficher et exécuter les étapes suivantes.
 - Appuyez sur  **CONT** pour reprendre l'exécution normale.
 - Appuyez sur **KILL** pour terminer l'exécution du programme.
5. Lorsque vous voulez reprendre l'exécution normale du programme, utilisez VISIT pour enlever la commande HALT du programme.

Pour le premier exemple, insérez HALT comme première commande de FIB2.

Effacez la pile et sélectionnez le menu USER.

CLEAR
USER

```
3:  
2:  
1:  
FIB2 FIB1 ABORT SEQUIT
```

Utilisez VISIT pour renvoyer FIB2 dans la ligne de commande.

FIB2 **VISIT**

```
→ n  
IF n 1 ≤  
THEN n
```

Insérez la commande HALT.

INS **CTRL** HALT

```
HALT ↔ n  
IF n 1 ≤  
SST HALT ABORT KILL WAIT KEY
```

Stockez la version modifiée de FIB2.

ENTER

```
3:  
2:  
1:  
SST HALT ABORT KILL WAIT KEY
```

Calculez F_1 . La seule action est l'apparition du témoin \odot .

USER
1 FIB2

```
3:  
2:  
1: 1  
FIB2 FIB1 ABORT SEQUIT
```

Sélectionnez le menu PROGRAM CONTROL et exécutez SST (*single-step*, pas à pas). Remarquez que la première étape est affichée dans la ligne du haut de l'affichage avant d'être exécutée.

CTRL
SST

```
→ n  
2:  
1: 1  
SST HALT ABORT KILL WAIT KEY
```

Remarquez que $\rightarrow n$ constitue une étape ; le terme « étape » fait référence à une unité logique plutôt qu'à un objet du programme.

Consultez les règles générales au début de cette section. Vous avez effectué les quatre premières étapes et vous pouvez maintenant choisir l'une des trois alternatives pour l'étape 5.

Pour cet exemple, appuyez de façon répétitive sur **SSRT** jusqu'à ce que le témoin **●** disparaisse, indiquant que FIB2 est terminé. (Ces étapes ne sont pas illustrées dans le manuel.) Le calcul de F_1 n'exécute que la clause THEN de FIB2.

Pour le deuxième exemple, exécutez 3 FIB2 et effectuez le calcul de F_3 pas à pas. Cette séquence exécute la clause ELSE, qui contient la boucle START ... NEXT. Vous allez voir que pour $n = 3$, la boucle START ... NEXT est exécutée deux fois.

Pour le troisième exemple, supposons que vous vouliez exécuter la boucle START ... NEXT pas à pas, pour voir la pile avant chaque itération de la boucle, mais sans exécuter pas à pas toutes les étapes de FIB2 ou de la boucle elle-même. Pour cela, placez la commande HALT dans la boucle. Le programme FIB2 est alors exécuté normalement jusqu'à ce qu'il atteigne la boucle. Vous pouvez utiliser **CONT** pour exécuter la boucle itération par itération.

Utilisez VISIT pour renvoyer FIB2 dans la ligne de commande.

USER

FIB2 **VISIT**

```

■ HALT → n
  ✖
    IF n 1 ≤
    THEN n

```

Utilisez les touches du menu du curseur pour supprimer HALT. Insérez ensuite HALT comme indiqué (après la commande START).

```

    IF n 1 ≤
    THEN n
    ELSE 0 1 2 n
        START HALT DUP R...

```

Stockez la version modifiée de FIB2.

ENTER

```

3:
2:
1:
FIB2 FIB1 BOMB BOMB

```

Commencez le calcul de F_3 . FIB2 s'arrête avant d'exécuter la boucle.

3 **FIB2**

```

3:
2:
1:
FIB2 FIB1 BOMB BOMB

```

Continuez l'exécution de la boucle. FIB2 s'arrête avant la deuxième exécution de la boucle.

 CONT

3:	
2:	1
1:	1
FIB2	FIB1
REOIT	REOIT

Continuez l'exécution de la boucle. Cette itération est la dernière, par conséquent, FIB2 s'exécute jusqu'à la fin.

 CONT

3:	
2:	
1:	2
FIB2	FIB1
REOIT	REOIT

Lorsque vous avez fini d'expérimenter l'exécution pas à pas avec FIB2, n'oubliez pas d'utiliser VISIT pour supprimer la commande HALT.

Développement et mise en facteur complets

Cette section contient deux programmes :

- MULTI répète un programme jusqu'à ce qu'il n'ait plus d'effet.
- EXCO utilise MULTI pour développer et mettre en facteur de façon complète.

MULTI (exécution multiple)

Etant donné un objet et un programme qui agit sur cet objet, MULTI exécute répétitivement le programme avec l'objet comme argument, jusqu'à ce que l'objet soit inchangé.

Arguments	Résultats
2 : <i>objet</i>	2 :
1 : « <i>programme</i> »	1 : <i>objet résultat</i>

Techniques :

- DO ... UNTIL ... END (boucle non définie). La clause DO contient les étapes à répéter ; la clause UNTIL contient le test qui détermine si les deux clauses doivent être répétées (si faux) ou si le programme doit être terminé (si vrai).
- Programmes comme arguments. Bien que les programmes soient généralement nommés et exécutés ensuite en appelant leurs noms, ils peuvent aussi être placés dans la pile et utilisés comme arguments d'autres programmes.
- Evaluation des variables locales. L'argument du programme à exécuter de façon répétitive est stocké dans une variable locale. Le stockage d'un objet dans une variable locale est une technique pratique lorsque vous ne connaissez pas le nombre de copies dont vous aurez besoin.

MULTI illustre une des différences entre les variables locales et globales : si une variable globale contient un nom ou un programme, le contenu de la variable est évalué lorsque le nom est évalué. Par contre, s'il s'agit d'une variable locale, le contenu est simplement rappelé dans tous les cas. Par conséquent, MULTI utilise le nom local pour placer l'argument programme dans la pile et exécute ensuite une commande EVAL explicite pour évaluer le programme.

Programme	Commentaires
⌘	Commence le programme.
→ P	Crée une variable locale <i>p</i> contenant l'argument programme.
⌘	Commence le programme de définition.
DO	Commence la clause DO.
DUP	Effectue une copie de l'objet.
P EVAL	Exécute le programme avec l'objet pour donner une nouvelle version de ce dernier. (La commande EVAL est nécessaire pour exécuter le programme car les variables locales renvoient simplement leurs contenus dans la pile sans évaluation.)
UNTIL	Commence la clause UNTIL.
DUP	Effectue une copie de la nouvelle version de l'objet.
ROT	Place l'ancienne version au niveau 1.
SAME	Teste si l'ancienne version et la nouvelle sont identiques.

Programme

END

»

»

ENTER

1 MULTI STO

Commentaires

Termine la clause UNTIL.

Termine le programme de définition.

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom MULTI.

EXCO (développement et regroupement complets)

Etant donné un objet algébrique, cette commande exécute EXPAN de façon répétitive jusqu'à ce que l'expression algébrique soit inchangé, puis exécute COLCT de façon répétitive jusqu'à ce que l'expression algébrique soit inchangée. Dans certains cas, le résultat sera un nombre.

Arguments	Résultats
1 : 'expression algébrique '	1 : 'expression algébrique '
1 : 'expression algébrique '	1 : z

Techniques :

- Programmation structurée. EXCO appelle deux fois le programme MULTI. Même si vous n'utilisez pas le programme MULTI dans d'autres programmes, la rédaction de MULTI comme programme séparé est justifiée par le gain d'efficacité apporté.

Programmes nécessaires :

- MULTI (page 253) exécute de façon répétitive les programmes fournis comme arguments par EXCO.

Programme

```

«
  « EXPAN »
  MULTI

  « COLCT »
  MULTI
»

```

[ENTER]
 ['] EXCO [STO]

Commentaires

Commence le programme.
 Place EXPAN dans la pile.
 Exécute EXPAN jusqu'à ce que
 l'objet algébrique soit inchangé.
 Place COLCT dans la pile.
 Exécute COLCT jusqu'à ce que
 l'objet algébrique soit inchangé.
 Termine le programme.

Place le programme dans la pile.
 Stocke le programme sous le nom
 EXCO.

Exemple. Développe et regroupe complètement l'expression

$$3x(4y + z) + (8x - 5z)^2.$$

Entrez l'expression.

[USER]
 ['] 3 [x] X [x]
 [(] 4 [x] Y [+] Z [)] [+]
 [(] 8 [x] X [-] 5 [x] Z [)] [^] 2
 [ENTER]

```

2:
1: '3*X*(4*Y+Z)+(8*X-5*
  Z)^2'
EXCO MULT FIB2 FIB1 RBOIT SBOIT

```

Développe et regroupe complètement.

[EXCO]

```

2:
1: '12*X*Y-77*X*Z+64*X^
  2+25*Z^2'
EXCO MULT FIB2 FIB1 RBOIT SBOIT

```

Les expressions contenant de nombreux produits de sommes ou des
 puissances dont le développement nécessite de nombreuses itérations
 de EXPAN, augmentent la durée d'exécution de EXCO.

Affichage d'un entier binaire

Cette section contient trois programmes :

- **COMPLET** est un utilitaire qui convertit un objet en une chaîne justifiée à droite.
- **PRESERVE** est un utilitaire pour les programmes modifiant les modes du calculateur (unité d'angle, base d'affichage binaire, etc.)
- **BAFFICHE** affiche un entier binaire en bases HEX, DEC, OCT et BIN. Ce programme appelle **COMPLET** pour afficher les nombres justifiés à droite et appelle **PRESERVE** pour conserver la base d'affichage des nombres binaires.

COMPLET (ajoute des espaces devant un nombre)

Convertit un objet en chaîne et, si l'objet contient moins de 23 caractères, ajoute des espaces devant l'objet de façon qu'il soit aligné à droite.

Lorsque **DISP** affiche une chaîne courte, elle apparaît *justifiée à gauche* : le premier caractère de la chaîne se trouve à l'extrémité gauche de l'affichage. La position du dernier caractère est déterminée par la longueur de la chaîne.

COMPLET ajoute des espaces au début d'une chaîne courte de façon que le dernier caractère de la chaîne se trouve à l'extrémité droite de l'affichage. Elle apparaît alors *justifiée à droite*.

COMPLET n'a aucun effet sur les chaînes de plus de 22 caractères.

Arguments	Résultats
1 : <i>objet</i>	1 : " <i>objet</i> "

Techniques :

- **WHILE ... REPEAT ... END** (boucle non définie). La clause **WHILE** contient un test qui détermine s'il faut exécuter à nouveau la clause **REPEAT** et recommencer le test (test vrai), ou s'il faut ignorer la clause **REPEAT** et terminer le programme (test faux).

- Opérations de chaîne. COMPLET démontre comment convertir un objet sous forme de chaîne, compter le nombre de caractères et concaténer deux chaînes.

Programme

```

⌘
→STR

WHILE
  DUP SIZE 23 <

REPEAT
  " " SWAP +

END
⌘

```

ENTER

PAD STO

Commentaires

Commence le programme.
 Convertit l'objet en une chaîne (les chaînes ne sont pas modifiées).
 Commence la clause WHILE.
 Teste si la chaîne contient moins de 23 caractères.
 Commence la clause REPEAT.
 Ajoute un espace devant la chaîne.
 Termine la clause REPEAT.
 Termine le programme.
 Place le programme dans la pile.
 Stocke le programme sous le nom COMPLET.

PRESERVE (stocke et rappelle l'état précédent)

Etant donné un programme dans la pile, stocke l'état en cours, exécute le programme et rappelle l'état précédent.

Arguments	Résultats
1 : ⌘ <i>programme</i> ⌘	1 : (<i>résultat du programme</i>)

Techniques :

- RCLF et STOF. PRESERVE utilise RCLF (*recall flags*, rappelle indicateur) pour enregistrer l'état en cours du calculateur dans un entier binaire et STOF (*store flags*, stocke indicateur) pour rappeler l'état.

- Structure de variable locale. PRESERVE crée une variable locale de façon à retirer brièvement l'objet de la pile ; son programme de définition évalue simplement l'argument programme présent dans la pile.

Programme

```

«
RCLF

→ f

«
EVAL
f STO

»

»

[ENTER]
['] PRESERVE [STO]

```

Commentaires

Commence le programme.

Rappelle un entier binaire de 64 bits représentant les états des 64 indicateurs utilisateurs.

Stocke l'entier binaire dans une variable locale.

Commence le programme de définition.

Exécute l'argument programme.

Réactive l'état précédent des 64 indicateurs utilisateur.

Termine le programme de définition.

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom PRESERVE.

BAFFICHE (affichage binaire)

Affiche un nombre dans les bases HEX, DEC, OCT et BIN

Arguments	Résultats
1 : # <i>n</i>	1 : # <i>n</i>
1 : <i>n</i>	1 : <i>n</i>

Techniques :

- IFERR ... THEN ... END (détection d'erreur). BAFFICHE contient la commande $R \rightarrow B$ (réel en binaire) de façon à pouvoir traiter les nombres réels. Néanmoins, cette commande provoque une erreur si l'argument est *déjà* un entier binaire.

De façon à pouvoir poursuivre l'exécution si une telle erreur se présente, la commande $R \rightarrow B$ est placée dans une clause IFERR. La clause THEN ne contient pas de commande car aucune action n'est nécessaire en cas d'erreur.

- Activation de LAST. Au cas où une erreur se présente, LAST doit être activé de façon à remettre l'argument dans la pile. BAFFICHE arme l'indicateur 31 de façon à activer par programme la récupération d'erreur par LAST.
- Boucle FOR ... NEXT (boucle définie avec compteur). BAFFICHE exécute une boucle de 1 à 4 et affiche chaque fois le nombre n dans une base différente sur une ligne différente.

Le compteur de boucle (nommé j dans ce programme) est une variable locale. Il est créé par la structure de programme FOR ... NEXT (plutôt que par une commande \rightarrow) et il est automatiquement incrémenté par NEXT.

- sous-programmes. BAFFICHE illustre trois utilisations des sous-programmes.
 1. BAFFICHE contient un sous-programme principal et un appel à PRESERVE. Le sous-programme principal est placé dans la pile pour être évalué par PRESERVE.
 2. Lorsque BAFFICHE crée une variable locale pour n , le programme de définition est un sous-programme.
 3. Quatre sous-programmes « personnalisent » l'action de la boucle. Chaque sous-programme contient une commande modifiant la base d'affichage des nombres binaires et chaque itération exécute un de ces sous-programmes.

Programmes nécessaires :

- COMPLET (page 257) ajoute des espaces devant une chaîne de façon qu'elle soit justifiée à droite.
- PRESERVE (page 258) stocke l'état en cours, exécute le sous-programme principal et rappelle l'état.

Programme

```
«
«
  DUP
  31 SF

  IFERR
    R→B
  THEN
  END

  → n
  «

  CLLCD
  « BIN »
  « OCT »
  « DEC »
  « HEX »
  1 4

  FOR J
    EVAL

    n →STR

  COMPLET
  J DISP

  NEXT
  »

  »

  PRESERVE

  »

  ENTER
  BAFFICHE STO
```

Commentaires

Commence le programme.

Commence le sous-programme principal.

Effectue une copie de n .

Arme l'indicateur 31 pour activer LAST.

Commence la détection d'erreur.

Convertit n en un entier binaire.

Si une erreur se présente, ne fait rien (pas de commande dans la clause THEN).

Crée une variable locale n .

Commence le programme de définition.

Efface l'affichage.

Sous-programme pour BIN.

Sous-programme pour OCT.

Sous-programme pour DEC.

Sous-programme pour HEX.

Première et dernière valeurs du compteur.

Commence la boucle avec j .

Evalue un des sous-programme de base (en commençant par celui de HEX).

Crée une chaîne affichant n dans la base en cours.

Justifie la chaîne à droite.

Affiche la chaîne dans la j ème ligne.

Incrémente j et répète la boucle.

Termine le programme de définition.

Termine le sous-programme principal.

Stocke l'état en cours, exécute le programme principal et rappelle l'état.

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom BAFFICHE.

Exemple. Passez en base DEC, affichez # 100 dans toutes les bases et vérifiez que BAFFICHE vous a bien renvoyé en base DEC.

Effacez la pile et sélectionnez le menu BINARY.

☐ CLEAR
☐ BINARY

```
3:
2:
1:
DEC= HEX OCT BIN STKS RCWS
```

Vérifiez que la base en cours est DEC et entrez # 100.

DEC
100 ENTER

```
3:
2:
1: # 100d
DEC= HEX OCT BIN STKS RCWS
```

Exécutez BAFFICHE. (Ne changez pas de menu, car vous devrez utiliser le menu BINARY à l'étape suivante).

BAFFICHE ENTER

```
# 64h
# 100d
# 144o
# 1100100b
```

Retournez à l'affichage normal de la pile et vérifiez la base en cours.

ON

```
3:
2:
1: # 100d
DEC= HEX OCT BIN STKS RCWS
```

Bien que le sousprogramme principal ait laissé le calculateur en base BIN, PRESERVE a réactivé la base DEC.

Vérifiez que BAFFICHE fonctionne aussi pour les nombres réels en entrant 144.

USER
144 BAFFI

```
# 90h
# 144d
# 220o
# 10010000b
```

Sommations

Lors des calculs statistiques sur des couples d'observations, il est souvent utile de calculer la somme des carrés (Σx^2 et Σy^2) et la somme des produits (Σxy) des deux variables.

La présente section contient cinq programmes :

- **SOMME** crée une variable ΣCOV contenant la matrice de covariance pour la matrice statistique ΣDAT en cours.
- ΣLIT extrait un nombre à la position spécifiée de ΣCOV .
- ΣX2 utilise ΣLIT pour extraire Σx^2 de ΣCOV .
- ΣY2 utilise ΣLIT pour extraire Σy^2 de ΣCOV .
- ΣXY utilise ΣLIT pour extraire Σxy de ΣCOV .

Si ΣDAT contient n colonnes, ΣCOV est une matrice $n \times n$. Les programmes ΣX2 , ΣY2 et ΣXY font référence à ΣPAR (paramètres statistiques) pour déterminer quelles colonnes contiennent les données x (appelée C_1) et les données y (appelée C_2).

Techniques :

- Opérations sur les matrices. Ces programmes illustrent la transposition de matrice, la multiplication de deux matrices et l'extraction d'un élément d'une matrice.
- Programmes utilisable dans des objets algébriques. ΣX2 , ΣY2 et ΣXY étant conformes à la syntaxe algébrique (pas d'arguments dans la pile et un résultat unique placé dans la pile), vous pouvez utiliser leurs noms comme des variables ordinaires dans une expression ou une équation.
- Convention ΣPAR . Plusieurs commandes de statistiques sur deux échantillons utilisent une variable nommée ΣPAR pour spécifier une paire de colonnes dans ΣDAT . ΣPAR contient une liste avec quatre nombres ; les deux premiers spécifient les colonnes (les deux autres sont la pente et l'intercept de la régression linéaire.)

SOMME vérifie que ΣPAR en exécutant 0 **PREDV DROP** ; la commande **PREDV** (*predicted value*, valeur prévue) crée ΣPAR avec des valeurs par défaut si ΣPAR n'existe pas déjà et **DROP** élimine la valeur prévue calculée pour 0.

ΣX2 , ΣY2 et ΣXY utilisent les valeurs stockées dans ΣPAR pour déterminer l'élément à extraire de ΣCOV .

SOMME (matrice de sommation)

Crée une variable ΣCOV contenant la matrice de covariance de la matrice de statistique ΣDAT .

Exemple : si ΣDAT est la matrice $n \times 2$

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix},$$

alors ΣCOV contiendra la matrice de covariance

$$\begin{bmatrix} \Sigma x^2 & \Sigma xy \\ \Sigma xy & \Sigma y^2 \end{bmatrix}.$$

Arguments	Résultats
1 :	1 :

Programme

```
«
  RCLΣ
  DUP
  TRN
  SWAP *
  'ΣCOV' STO
  Ø PREDV DROP
  »
```

ENTER

↑ SUMS STO

Commentaires

Commence le programme.
 Rapelle le contenu de la matrice de statistiques $n \times m$ ΣDAT .
 Effectue une copie.
 Transpose la matrice. Le résultat est une matrice $m \times n$.
 Multiplie les matrices pour donner la matrice de covariance $m \times m$. Sans l'échange de matrices, le produit serait une matrice $n \times n$.
 Stocke la matrice de covariance dans la variable ΣCOV .
 Vérifie que ΣPAR existe.
 Termine le programme.

Place le programme dans la pile.
 Stocke le programme sous le nom SOMME .

ΣLIT (extrait un élément de ΣCOV)

Etant donné p et q , indiquant chacun la première ou la dernière *position* dans ΣPAR, ΣLIT extrait l'élément rs de ΣCOV, où r et s sont les premier et deuxième *éléments* correspondants dans ΣPAR.

ΣLIT est appelé par ΣX2, ΣY2 et ΣXY avec les arguments suivants.

- Pour or ΣX2, $p = 1$ et $q = 1$.
- Pour ΣY2, $p = 2$ et $q = 2$.
- Pour ΣXY, $p = 1$ et $q = 2$.

Arguments	Résultats
2 : 1 ou 2	2 :
1 : 1 ou 2	1 : élément rs de ΣCOV

Programme

```

«
  ΣCOV
  ΣPAR
  DUP
  5 ROLL
  GET
  SWAP
  4 ROLL
  GET
  2 →LIST
  GET
  »

```

ENTER

ΣGET STO

Commentaires

Commence le programme.
 Place la matrice de covariance dans la pile.
 Place la liste des paramètres statistiques dans la pile.
 Effectue une copie.
 Place p au niveau 1.
 Appelle r , le p ème élément de ΣPAR.
 Place ΣPAR au niveau 1.
 Place q au niveau 1.
 Appelle s , le q ème élément de ΣPAR.
 Place $\{ r, s \}$ dans la pile.
 Rappelle l'élément rs de ΣCOV.
 Termine le programme.
 Place le programme dans la pile.
 Stocke le programme sous le nom ΣLIT.

ΣX^2 (somme des carrés de x)

Calcule Σx^2 , où les x sont les éléments de C_1 (la colonne spécifiée par le premier paramètre de ΣPAR).

Arguments	Résultats
1 :	1 : Σx^2

Programme

«

1 1

ΣLIT

»

ΣX^2

Commentaires

Commence le programme.

Spécifie deux fois C_1 .

Extrait Σx^2 .

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom ΣX^2 .

ΣY^2 (Somme des carrés de y)

Calcule Σy^2 , où y sont les éléments de C_2 (la colonne spécifiée par le deuxième paramètre de ΣPAR).

Arguments	Résultats
1 :	1 : Σy^2

Programme

«

2 2

ΣLIT

»

ΣY^2

Commentaires

Commence le programme.

Spécifie deux fois C_2 .

Extrait Σy^2 .

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom ΣY^2 .

ΣXY (Somme des produits de x et y)

Calcule Σxy , où x et y sont les éléments correspondants de C_1 et C_2 (les colonnes spécifiées par les premier et deuxième paramètres de ΣPAR).

Arguments	Résultats
1 :	1 : Σxy

Programme

```
«
  1 2
  Σ LIT
»
```

```
[ENTER]
['] ΣXY [STO]
```

Commentaires

Commence le programme.
Spécifie C_1 et C_2 .
Extrait Σxy .
Termine le programme.
Place la programme dans la pile.
Stocke le programme sous le nom ΣXY .

Exemple. Calculez ΣX^2 , ΣY^2 et ΣXY pour les données statistiques suivantes :

18	12
4	7
3	2
11	1
31	48
20	17

Les étapes générales sont les suivantes.

1. Entrez les données statistiques.
2. Exécutez SOMME pour créer la matrice de covariance ΣCOV .
3. Exécutez ΣX^2 , ΣY^2 et ΣXY .
4. Si ΣDAT contient plus de deux colonnes (c'est-à-dire si chaque point comporte plus de deux variables) :
 - a. Exécutez $\text{COL}\Sigma$ pour spécifier de nouvelles valeurs pour C_1 et C_2 . Les valeurs sont stockées dans ΣPAR .
 - b. Exécutez ΣX^2 , ΣY^2 et ΣXY .

Essayez maintenant l'exemple ci-dessus.

Effacez la pile, sélectionnez le menu STAT et effacez ΣDAT .

■ CLEAR
■ STAT
 CLΣ

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

Entrez les données et vérifiez que vous entrez les six observations.

I 18.12 **Σ+**
I 4.7 **Σ+**
I 3.2 **Σ+**
I 11.1 **Σ+**
I 31.48 **Σ+**
I 20.17 **Σ+**
■ NZ

```
3:
2:
1:
Σ+ Σ- NZ CLΣ STOΣ RCLΣ 6
```

Éliminez le nombre d'observations.

■ DROP

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

Créez la matrice de covariance ΣCOV .

USER
SUMS

3:						
2:						
1:						
	ΣPAR	ΣCOV	ΣDAT	ΣXY	ΣY2	ΣX2

Calculez Σx^2 .

ΣX2

3:						
2:						
1:						1831
	ΣPAR	ΣCOV	ΣDAT	ΣXY	ΣY2	ΣX2

Calculez Σy^2 .

ΣY2

3:						
2:						1831
1:						2791
	ΣPAR	ΣCOV	ΣDAT	ΣXY	ΣY2	ΣX2

Calculez Σxy .

ΣXY

3:						1831
2:						2791
1:						2089
	ΣPAR	ΣCOV	ΣDAT	ΣXY	ΣY2	ΣX2

Si la matrice de statistiques comporte plus de deux colonnes, vous pouvez spécifier de nouvelles valeurs pour C_1 et C_2 . En exemple, spécifiez $C_1 = 1$ et $C_2 = 2$ (les valeurs en cours).

La commande $\text{COL}\Sigma$ est présente dans le menu STAT , mais il est plus facile de rester dans le menu USER et d'écrire son nom.

1
2 $\text{COL}\Sigma$

3:						1831
2:						2791
1:						2089
	ΣPAR	ΣCOV	ΣDAT	ΣXY	ΣY2	ΣX2

Vous pourriez maintenant exécuter ΣX2 , ΣY2 et ΣXY pour la nouvelle paire de colonnes C_1 et C_2 .

N'oubliez pas d'exécuter à nouveau SOMME lorsque vous ajoutez ou supprimez des données de la matrice statistiques ΣDAT .

Médiane de données statistiques

Cette section contient trois programmes :

- TRI trie les éléments d'une liste.
- LMED calcule la médiane d'une liste triée.
- MEDIAN utilise TRI et MED pour calculer la médiane des données statistiques en cours.

TRI (trie une liste)

Trie une liste en ordre croissant.

Arguments	Résultats
1 : { liste }	1 : { liste triée }

Techniques :

- Tri par échange de paires de clés. En commençant par le premier et le deuxième nombre de la liste, TRI compare des nombres adjacents et place le plus grand vers la fin de la liste. Ce processus est effectué une fois pour placer le plus grand nombre à la fin de la liste, puis pour placer le nombre immédiatement précédent en ordre dans la position précédente dans la liste, etc.
- Boucles définies emboîtées. La boucle externe contrôle la position d'arrêt lors de chaque exécution du processus ; la boucle interne s'exécute de 1 à la position d'arrêt.
- Structures emboîtées de variable locale. TRI contient deux structures de variable locales, la seconde structure se trouvant dans le programme de définition de la première. Cet emboîtement est effectué car il est plus facile de créer la première variable locale dès que sa valeur est calculée, de façon à pouvoir retirer sa valeur de la pile, plutôt que de calculer les deux valeurs et de créer les deux variables locales en même temps.

- FOR ... STEP et FOR ... NEXT (boucles définies). TRI utilise deux compteurs : — 1 STEP et NEXT.

Programme

```

«
  DUP SIZE 1 - 1
  FOR j

    1 j
    FOR k

      k GETI → n1

    «
      GETI → n2

    «
      DROP
      IF n1 n2 >
      THEN
        k n2 PUTI
        n1 PUT

      END

    »
  »
NEXT
-1 STEP
»

[ENTER]
[ ] TRI [STO]

```

Commentaires

Commence le programme.

De la pénultième position à la première.

Commence la boucle externe avec le compteur j .

De la première position à la j ème position,

Commence la boucle interne avec le compteur k .

Appelle le k ème nombre de la liste et le stocke dans une variable locale n_1 .

Commence le programme de définition externe.

Appelle le nombre suivant dans la liste et le stocke dans une variable locale n_2 .

Commence le programme de définition interne.

Elimine le compteur.

Si les nombres sont dans le mauvaise ordre, Alors, exécute :

Place le deuxième nombre en k ème position.

Place le k ème nombre dans la position suivante.

Fin de la clause THEN.

Termine le programme de définition interne.

Termine le programme de définition externe.

Incrémente j et répète la boucle interne.

Décrémente j et répète la boucle externe.

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom TRI.

Exemple.

Trie la liste { 8, 3, 1, 2, 5 }.

USER

{ 8.3.1.2.5 TRI

3:	
2:	
1:	{ 1 2 3 5 8 }
SORT SPAR SCOV SDAT SNV SVE	

LMED (médiane d'une liste)

Etant donné une liste triée, calculez la médiane. Si la liste contient un nombre impair d'éléments, la médiane est la valeur de l'élément central. Si la liste contient un nombre pair d'éléments, la médiane est la moyenne des deux valeurs centrales.

Arguments	Résultats
1 : { liste triée }	1 : médiane de la liste triée

Techniques :

- FLOOR et CEIL. Pour un entier, FLOOR et CEIL donnent cet entier ; pour un non entier, FLOOR et CEIL donnent les deux entiers successifs qui entourent cette valeur.

Programme

```
⌘
  DUP SIZE
  1 + 2 /

→ P

⌘

  DUP
  P FLOOR GET

  SWAP
  P CEIL GET
```

Commentaires

Commence le programme.
Taille de la liste.
Position du centre de la liste (valeur fractionnaire pour les listes ayant un nombre pair d'éléments).
Stocke la position du centre dans une variable locale *p*.
Commence le programme de définition.
Effectue une copie de la liste.
Appelle le nombre à la position du centre ou immédiatement en dessous.
Place la liste au niveau 1.
Appelle le nombre à la position du centre ou immédiatement au-dessus.

Programme

```
+ 2 /  
  
»  
  
»  
  
[ENTER]  
['] LMED [STO]
```

Commentaires

Moyenne des deux nombres les plus proches du centre.
Termine le programme de définition.
Termine le programme.
Place le programme dans la pile.
Stocke le programme sous le nom LMED.

Exemple.

Calculez la médiane de la liste triée avec SORT.

```
[USER]  
LMED
```



LMED est appelé par MEDIANE.

MEDIANE (médiane des données statistiques)

Donne un vecteur représentant les médianes des colonnes de données statistiques.

Arguments	Résultats
1 :	1 : [x_1 x_2 ... x_m]

Techniques :

- Eléments de tableaux, listes et pile. MEDIANE extrait une colonne de données de ΣDAT sous forme d'un vecteur. Pour convertir le vecteur en une liste, MEDIANE place les éléments du vecteur dans la pile et les combine en une liste. La médiane est calculée par SORT et LMED à partir de cette liste.
La médiane de la même colonne est calculée en premier et celle de la première colonne en dernier. Chaque médiane calculée est placée dans la pile au-dessus de la médiane précédente.

Lorsque toutes les médianes sont calculées et correctement positionnées dans la pile, elles sont combinées en un vecteur.

- FOR ... NEXT (boucle définie avec compteur). MEDIANE utilise une boucle pour calculer la médiane de chaque colonne. Les médianes étant calculées en ordre inverse (dernière colonne en premier), le compteur permet d'en renverser l'ordre.

Programmes nécessaires :

- SORT (page 220) arrange une liste en ordre croissant.
- LMED (page 272) calcule la médiane d'une liste triée.

Programme	Commentaires
«	Commence le programme.
RCLΣ	Place une copie de la matrice statistique ΣDAT en cours dans la pile pour la sauvegarder.
DUP SIZE	Place la liste { n m } dans la pile, où n est le nombre de lignes dans ΣDAT et m est le nombre de colonne.
LIST→ DROP	Place n et m dans la pile. Elimine la valeur de la taille de la liste.
→ n m	Crée des variables locales pour n et m .
«	Commence le programme de définition.
'ΣDAT' TRN	Transpose ΣDAT. n est maintenant le nombre de colonnes de ΣDAT et m le nombre de ligne.
1 m	Première et dernière lignes.
FOR J	Pour chaque ligne, faites :
Σ-	Extrait la dernière ligne de ΣDAT. Au début, c'est la même ligne, qui correspond à la même colonne du tableau ΣDAT original.
ARRY→ DROP	Place les éléments de ligne dans la pile. Elimine la liste d'index { n }, car n est déjà stockée dans une variable locale.
n →LIST	Crée une liste de n éléments.
SORT	Tri la liste.
LMED	Calcule la médiane de la liste.

Programme

```
  J ROLL D  
  
NEXT  
m 1 →LIST  
→ARRY  
  
»  
  
SWAP  
  
STOΣ  
  
»  
  
[ENTER]  
['] MEDIANE [STO]
```

Commentaires

Place la médiane dans le niveau approprié de la pile.
Incrémente j et répète la boucle.
Crée la liste $\{ m \}$.
Combine toutes les médianes dans un vecteur de m éléments.
Termine le programme de définition.
Place le ΣDAT original dans le niveau 1.
Redonne à ΣDAT sa valeur précédente.
Termine le programme.

Place le programme dans la pile.
Stocke le programme sous le nom MEDIANE.

Exemple. Calculez la médiane des données de la page 268. (Cet exemple suppose que vous avez déjà introduit les données.) Il y a deux colonnes de données, MEDIANE donne donc un vecteur à deux éléments.

Calculez la médiane.

```
[USER]  
MEDI
```



```
3:  
2:  
1: [ 14.5 9.5 ]  
ΣDAT MEDI LMEM SORT ΣPAR ΣCON
```

Les médianes sont 14,5 pour la première colonne et 9,5 pour la seconde.

Changement de répertoires

Cette section contient deux programmes :

- PREC vous donne un menu des répertoires parents.
- SUIV vous donne un menu des sous-répertoire.

PREC (répertoires parents)

Crée un menu contenant les noms du répertoire parent, du parent de ce dernier, et ainsi de suite jusqu'au répertoire HOME.

Arguments	Résultats
1 :	1 :

Techniques :

- Liste des répertoires parents. PREC utilise PATH pour donner les noms du répertoire parents et de tous les précédents.
- Sous-ensemble d'une liste. PREC utilise SUB pour retirer le nom du répertoire en cours dans la liste PATH.
- Menu personnalisé. PREC utilise MENU pour créer un menu personnalisé des répertoires parents à partir de la liste PATH.

Programme

«

PATH

1

OVER SIZE 1 -

SUB

MENU

»

PREC

Commentaires

Commence le programme.

Place la liste du chemin dans la pile.

Place 1 dans la pile.

Place la taille - 1 dans la liste.

Crée un sous-ensemble de la liste PATH incluant tous les noms sauf le dernier (répertoire en cours).

Crée un menu des répertoires parents.

Termine le programme.

Place le programme dans la pile.

Stocke le programme sous le nom PREC.

Exemple. A partir du répertoire HOME, créez une hiérarchie de sous-répertoires D1, D2 et D3 ; puis utilisez PREC pour aller de D3 à D1.

Effacez la pile et allez au répertoire HOME.

☐ CLEAR
☐ MEMORY HOME

```
3:  
2:  
1:  
MEM MENU ORDER PATH HOME CDIR
```

Créez un sous-répertoire D1 et allez dans ce sous-répertoire.

☐ D1 CDIR
D1 ☐ ENTER

```
3:  
2:  
1:  
MEM MENU ORDER PATH HOME CDIR
```

Répétez le processus pour les sous-répertoires D2 et D3.

☐ D2 CDIR
D2 ☐ ENTER
☐ D3 CDIR
D3 ☐ ENTER

```
3:  
2:  
1:  
MEM MENU ORDER PATH HOME CDIR
```

Affichez le menu des répertoires parents.

UP ☐ ENTER

```
3:  
2:  
1:  
HOME D1 D2
```

Allez au répertoire D1.

D1 :

```
3:  
2:  
1:  
HOME D1 D2
```

SUIV (aller à un sous-répertoire)

SUIV crée un menu contenant les noms de tous les sous-répertoires du répertoire en cours.

Arguments	Résultats
1 :	1 :

Techniques :

- Liste de variables. SUIV utilise VARS pour donner une liste des variables et sous-répertoires du répertoire en cours.
- Détection d'erreur. Pour vérifier si un nom de la liste VARS est un sous-répertoire, SUIV utilise le nom comme argument de RCL ; du fait que les répertoires ne peuvent pas être rappelés dans la pile, une erreur apparaît si le nom est celui d'un répertoire et ce nom est ajouté à la liste des sous-répertoires.

Programme

```
«  
VARS  
→ v  
«  
{ }  
1 v SIZE  
FOR j  
  v j GET  
  IFERR RCL DROP  
  
  THEN +  
  
END  
  
NEXT  
MENU  
  
»  
»  
[ENTER]  
[ ] SUIV [STO]
```

Commentaires

Commence le programme.
Place dans la pile une liste des noms de toutes les variables et sous-répertoires.
Stocke la liste VARS dans une variable locale *v*.
Commence le programme de définition.
Place la liste des noms de sous-répertoires dans la pile (vide au début).
Place 1 et la taille de *v* dans la pile.
Pour chaque nom dans *v*, fait :
Appel le nom.
Essaie de rappeler le contenu de la variable ; si réussi, élimine le contenu.
Si RCL provoque une erreur, le nom est celui d'un sous-répertoire, le programme ajoute donc ce nom à la liste des noms de sous-répertoire.
Fin de la clause THEN et de la structure de programme.
Répétition pour le nom suivant dans *v*.
Crée un menu personnalisé des noms de répertoire.
Termine le programme de définition.
Termine le programme.
Place le programme dans la pile.
Stocke le programme sous le nom SUIV.

Exemple. Dans l'exemple précédent (page 277), vous avez créé un hiérarchie de sous-répertoires D1, D2 et D3 et vous vous êtes arrêté dans le répertoire D1. Dans l'exemple suivant, vous allez vous déplacer à D2 puis à D3.

Affichez le menu des sous-répertoires.

SUIV

3:									
2:									
1:									
	D2								

Allez dans D2.

D2

3:									
2:									
1:									
	D2								

Affichez le menu des sous-répertoires.

SUIV

3:									
2:									
1:									
	D3								

Allez dans D3.

D3

3:									
2:									
1:									
	D3								

Annexes & index

Page 282	A: Assistance, Batteries et Service
296	B : Les menus et leurs jeux de libellés
317	Index des touches
323	Index des sujets

A

Assistance, Batteries et Service

Cette annexe contient des informations qui vous aideront si votre calculateur vous occasionne des problèmes. Si vous ne trouvez pas de rubrique dans l'index ou la Table des matières (page 5) qui s'applique à votre cas, parcourez la liste des « Réponses aux questions fréquemment posées », ci-dessous. Dans le cas où rien de tout cela ne s'applique à votre problème, contactez le Service de support technique calculateurs, en utilisant l'adresse et les numéros de téléphone qui figurent en page intérieure de couverture, à l'arrière du manuel.

Si vous avez besoin de remplacer les piles, consultez la page 286. Si le calculateur ne fonctionne pas bien, lisez « Déterminer si votre calculateur a besoin d'être réaroé » en page 289. Si c'est le cas, consultez « Garantie » en page 207 et « Maintenance » en page 293.

Réponses à quelques questions fréquemment posées

Q: *Le calculateur ne s'allume pas lorsque j'appuie sur [ON]. Que se passe-t-il ?*

A: Il s'agit peut-être d'un problème anodin, susceptible d'être éliminé immédiatement, ou bien le calculateur a besoin d'être réparé. Voyez « Déterminer si votre calculateur a besoin d'être réparé », en page 289.

Q: *Comment puis-je être sûr du bon fonctionnement de mon calculateur ?*

A: Exécutez le test automatique décrit à la page 290.




Q: *Comment effacer complètement la mémoire du calculateur ?*

A: Appuyez simultanément sur [ON] [INS] [▶] et maintenez-les enfoncées, puis relâchez-les, comme indiqué sous « Effacer la mémoire » en page 20.

Q: *Que signifient trois points (...) à l'extrémité droite de la ligne d'affichage ?*

A: Ces trois points s'appellent une *ellipse* et ils indiquent que l'objet est trop long pour tenir sur une seule ligne.

Q: *Comment afficher la totalité d'un objet ?*


A: Utilisez  [EDIT] ou  [VISIT] pour renvoyer l'objet en ligne de commande, comme indiqué sous « Corriger un objet » en page 173. Vous pouvez alors utiliser les touches de commande du curseur pour afficher n'importe quelle partie de l'objet. Pour annuler une correction, appuyez sur  [ON].

Q: *Qu'entend-on exactement par « objet » ?*


A: C'est un terme général qui s'applique à presque tous les éléments avec lesquels vous travaillez. Les nombres, les expressions, les tableaux, les programmes, etc. sont tous différents types d'objets. Lisez « Caractéristiques principales et concepts de base », en page 25, vous y trouverez une brève description des types d'objets, ou bien voyez le chapitre 16, « Objets », pour une définition plus détaillée.

Q: *MLe calculateur émet une tonalité et affiche*

Bad Argument Type. Que se passe-t-il ?

A: Les objets de la pile ne sont pas d'un type utilisable par la commande que vous essayez d'utiliser. Par exemple, exécuter  [STO] sans qu'un nom soit présent au niveau 1 produit cet erreur. Utilisez CATALOG pour vérifier quels sont les arguments corrects pour la commande en question, comme indiqué dans « Le catalogue des commandes », en page 86.


Q: *Le calculateur émet un bip et affiche Too Few Arguments. Qu'est-ce qui ne va pas ?*

A: La pile ne contient pas assez d'objets pour la commande que vous essayez d'exécuter. Par exemple, exécuter  [+] avec seulement un nombre dans la pile produit cette erreur. Utilisez CATALOG pour vérifier quels sont les arguments corrects de cette commande, comme indiqué dans « Le catalogue des commandes » en page 31.




Q: *Le calculateur émet un bip, et il affiche un message d'erreur différent de ceux cités ci-dessus. Comment puis-je savoir ce qui ne va pas ?*

A: Consultez l'annexe A, « Messages », dans le Manuel de référence.

Q: *Comment supprimer la tonalité ?*

A: Frappez 51 SF  [ENTER]. Ceci arme l'indicateur 51, qui désactive la tonalité.

Q: *Comment imprimer une copie de l'affichage ?*

A: Appuyez sur  [ON] et maintenez la pression, appuyez simultanément sur  [L], relâchez ensuite  [ON].

Q: Les touches de [A] à [R] ne fonctionnent plus. Qu'est-ce qui se passe ?

A: Vous avez accidentellement appuyé sur « Menu Lock », ce qui fait que toutes les fiches de [A] à [R] sélectionnent des menus, à moins d'appuyer d'abord sur [■]. Pour désactiver ce mode de verrouillage « Menu Lock », appuyez sur [■] [MENUS].

Q: Je ne retrouve pas certaines des variables que j'ai utilisées il y a un moment. Que leur est-il arrivé ?

A: Vous avez peut-être utilisé ces variables dans un autre répertoire. Si vous ne vous souvenez pas du répertoire que vous utilisiez, il vous faudra ouvrir tous les répertoires.

Q: Comment puis-je vérifier la quantité de mémoire restante ?

A: Exécutez MEM [ENTER] et la machine affichera le nombre d'octets disponibles en mémoire.

Q: Pourquoi le curseur a-t-il changé de forme ?

A: Le curseur indique le mode de saisie courant. Les modes de saisie sont le mode exécution (curseur « vide »), le mode alphabétique (curseur partiellement rempli) ou le mode alpha (curseur plein). La forme du curseur identifie le mode de remplacement (curseur carré) ou l'insertion (flèche). Voir « Indication des modes par le curseur », en page 172.

Q: J'ai frappé un nom/appuyé sur une touche du menu USER, mais le nom n'est pas apparu sur l'affichage. Pourquoi ?


A: Vous avez frappé le nom sans guillemets, ce qui réfère au contenu d'une variable. Pour placer un mot sur la pile, appuyez sur la touche ['] d'abord (lisez le paragraphe concernant les noms avec ou sans guillemets en page 57).



Q: Lorsque je calcule la racine carrée de -27, pourquoi le résultat n'est-il pas -3 ?

A: Tout nombre possède trois racines carrées, dont deux sont des nombres complexes. Le HP-28S vous donne l'une d'entre elles, nommée la valeur principale. Pour les arguments positifs réels, la valeur principale est la racine réelle ; pour les arguments négatifs réels la valeur principale est l'une des racines complexes. Pour calculer la racine réelle bème d'un nombre a , utilisez le programme suivant :

« → a b 'SIGN(a)*ABS(a)^INV(b)' »

Appuyez sur ['] RREEL [STO] pour stocker le programme dans une variable RREEL (racine réelle). Vous pouvez alors trouver la racine réelle de -27 en frappant 27 [CHS] [ENTER] 3 [ENTER] RREEL [ENTER].

Q: Le fonctionnement du calculateur s'est ralenti et le témoin  clignote sur l'affichage. Que se passe-t-il ?



A: Le calculateur est en mode d'impression automatique (TRACE). Appuyez sur  **PRINT**  **TRAC** pour désactiver ce mode d'impression.


Q: L'imprimante imprime quelques lignes très rapidement, puis ralentit fortement. Quelle en est la raison ?

A: Le calculateur transmet rapidement une certaine quantité de données vers l'imprimante, puis ralentit pour être sûr que l'imprimante puisse le suivre.

Q: Comment peut-on accélérer l'impression ?

A: Si votre imprimante est branchée à un adaptateur, le calculateur peut envoyer des données à un rythme plus soutenu. Pour accélérer l'impression, frappez 52 SF

 **ENTER**. Cette séquence arme l'indicateur 52, qui contrôle la vitesse d'impression. Si l'imprimante n'est pas branchée à un adaptateur, frappez 52 CF  **ENTER** pour effacer l'indicateur 52 et revenir à la vitesse d'impression normale.

Q: L'imprimante « oublie » un caractère ou imprime le caractère . Que se passe-t-il ?



A: La distance ou l'angle entre imprimante et calculateur est peut-être trop grand(e), ou bien il y a un obstacle gênant la transmission. Consultez le manuel d'utilisation de l'imprimante pour les détails.

Q: Quelle est la différence entre STO et STORE ?

A: La commande STO assigne des valeurs spécifiques à une variable. Le menu STORE contient des commandes qui effectuent des opérations arithmétiques directes, utilisant la valeur d'une variable comme argument et affectant le résultat de l'opération à cette même variable.

Q: A la place du résultat symbolique que j'attendais, j'obtiens un résultat numérique. Pourquoi ?

A: Il y a des valeurs affectées à une ou plusieurs variables. Éliminez le contenu des variables (page 52) et essayez à nouveau.

Q: Lorsque j'appuie sur  **DRAW**, l'affichage s'efface, le témoin  clignote puis s'arrête, mais je ne vois aucun des points de la courbe sur l'affichage. Pourquoi ?

A: Les valeurs calculées sont en dehors du domaine du tracé actuel. Voyez « Modifier l'échelle du tracé » en page 91.

Q: J'ai évalué une variable (ou une expression) et le calculateur ne répond plus. Même l'appui sur **ON** ne produit pas d'effets. Que s'est-il passé ?

A: Vous avez créé une variable en l'utilisant elle-même dans sa définition, créant ainsi une « boucle sans fin ». Pour l'interrompre, effectuez un arrêt système :

1. Appuyez sur **ON**, et maintenez-la.
2. Appuyez sur **▲**.
3. Relâchez **ON**.

Puis redéfinissez la variable pour éliminer la définition circulaire.


Si vous ne trouvez pas ici la réponse à votre question, contactez le Support technique, les numéros de téléphone et les adresses se trouvent à l'intérieur du dos de couverture de ce manuel.

Batteries

The HP-28S est alimenté par trois piles alcalines. Trois nouvelles piles assurent environ six mois à un an d'utilisation ; leur longévité dépend directement de la fréquence avec laquelle vous utilisez votre calculateur.

N'utilisez que des piles alcalines neuves de type N. N'utilisez pas de piles rechargeables.

Indicateur de faible charge

Lorsque le témoin de faible charge apparaît, () , le HP-28S peut continuer à fonctionner au moins 10 heures. Si le calculateur est éteint lorsque le témoin apparaît, la mémoire permanente sera préservée pendant environ un mois.

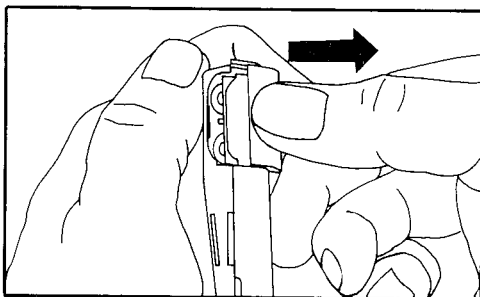
Installation des piles

Si vous venez d'acquérir votre HP-28S et que vous êtes en train d'installer les trois piles pour la première fois, vous pouvez prendre tout le temps nécessaire pour cette procédure.

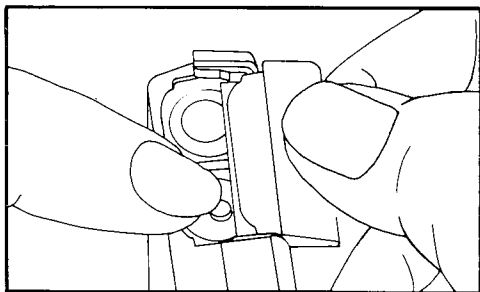
Cependant, si vous remplacez les piles, souvenez-vous qu'il y a un délai maximum pour effectuer ce remplacement si vous désirez conserver intact le contenu de la mémoire. Une fois que le compartiment des piles est ouvert, il vous faut remplacer les piles et refermer le compartiment en moins d'une minute pour éviter la perte de la mémoire continue. Déballiez donc et préparez les trois piles avant d'ouvrir ce compartiment. Assurez-vous également que votre calculateur est bien hors tension avant d'opérer.

Pour installer les piles :

1. Préparez trois piles neuves de type N.
2. Ouvrez le calculateur ; assurez-vous qu'il est éteint. *Ne touchez pas à la touche **ON** pendant ce processus, vous effaceriez le contenu de la mémoire permanente.*
3. Tenez le calculateur avec le compartiment des piles tourné vers le haut. Faites glisser le couvercle du compartiment vers vous (vers le haut).

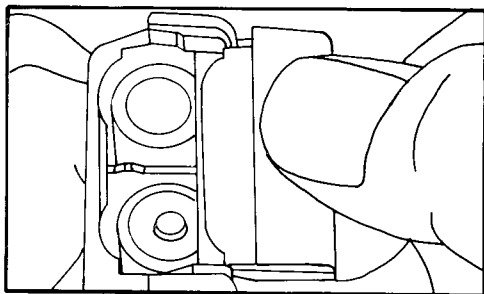


4. Penchez le calculateur pour extraire les anciennes piles.
5. Insérez les nouvelles piles. Placez-les comme indiqué sur le diagramme moulé sur le dos du calculateur. Respectez les polarités (+ et -).
6. Engagez les piles dans le compartiment et enfoncez-les à l'aide de la partie en plastique de la porte jusqu'à ce que la plaque métallique de contact soit alignée sur les guides dans le boîtier du calculateur.



7. Faites glisser la plaque métallique de contact dans les glissières. Si nécessaire, appuyez sur les piles pour les maintenir dans le compartiment pour que le couvercle puisse glisser par-dessus. Faites glisser le couvercle dans ses glissières jusqu'à ce qu'il se mette en place avec un déclic.

La loi du 15 juillet 1975 relative à l'élimination des déchets et à la récupération des matériaux régit la manipulation des piles au mercure, qui constituent un danger potentiel pour l'environnement. *Ce calculateur n'utilise pas de piles au mercure*, mais sachez cependant que nombre de commerçants disposent d'un service de récupération des piles usagées.



Entretien du calculateur

Utilisez un chiffon propre et doux légèrement humide pour nettoyer l'écran. Evitez de mouiller le calculateur et ne lubrifiez pas la charnière.

Environnement





Pour maintenir le bon fonctionnement du calculateur, les limites de température et d'humidité suivantes doivent être respectées :

- Température de fonctionnement : 0 ° à 45 °C.
- Température de stockage : -20 ° à 65 °C.
- Humidité (stockage et fonctionnement) : 90 % à 40 °C.

Déterminer si votre calculateur a besoin d'être réparé

Suivez cette procédure pour éliminer le plus de causes de panne possible. S'il s'avère que votre calculateur a réellement besoin d'être réparé, lisez « Garantie » en page 291 et « Si votre calculateur doit être réparé » en page 293.

Si rien ne se produit lorsque vous appuyez sur  :

1. Vérifiez le contraste de l'affichage.
 - a. Appuyez sur  et maintenez la pression.
 - b. Appuyez plusieurs fois sur .
 - c. Relâchez la touche .
 - d. Si l'écran reste vide, appuyez sur  et répétez les points a, b et c.

2. Changez les piles selon la procédure décrite à la page 286.
3. Si le fonctionnement de votre calculateur n'est pas rétabli après les étapes 1 et 2, il doit être réparé. Voyez les instructions concernant la garantie (page 291) et l'envoi du calculateur en réparation (page 293).

Si l'écran est lisible, mais que rien ne se passe lorsque vous appuyez sur les touches :

1. Exécutez un « arrêt système ».
 - a. Appuyez sur **ON** et maintenez votre pression.
 - b. Appuyez sur **▲**.
 - c. Relâchez la touche **ON**.
2. Si le calculateur ne répond toujours pas, exécutez une réinitialisation de la mémoire.
 - a. Appuyez sur **ON** et maintenez la pression.
 - b. Appuyez sur **INS** et sur **▶** et maintenez la pression.
 - c. Relâchez **INS** et **▶**.
 - d. Relâchez **ON** en dernier lieu.
3. Si les étapes 1 et 2 ne rétablissent pas le fonctionnement du calculateur, il a besoin d'être réparé. Lisez les paragraphes concernant la garantie (en page 291) et la réparation (en page 293).

Le test automatique

Le calculateur fonctionne, mais vous estimez qu'il ne fonctionne pas normalement :

1. Si vous possédez une imprimante, mettez-la sous tension. Pendant son test, le calculateur imprime des caractères qui peuvent faciliter le diagnostic.
2. Lancez le test automatique.
 - a. Appuyez sur **ON** et maintenez votre pression.
 - b. Appuyez sur **◀**.
 - c. Relâchez **ON**.

Ce test à répétition est automatique. S'il ne s'enclenche pas, c'est que vous avez appuyé sur **ON** **▼** par erreur. Cette frappe lance un autre test, utilisé en usine, qui nécessite des frappes au clavier. Quittez ce test en exécutant une réinitialisation, décrite au point 4 ci-dessous, puis relancez le test automatique à répétition.

3. Le test affiche des lignes horizontales et verticales, un écran vide, un motif composé de caractères aléatoires et affiche le résultat du test.
 - Le message **OK-28S** indique que le calculateur a réussi le test.
 - Un message **1 FAIL** indique l'échec. Le nombre précise la nature de la panne. Si vous envoyez votre calculateur en réparation, joignez-y ce nombre et la sortie imprimante, si cela est possible.

Si vous interrompez la répétition du test par la frappe d'une touche, le test produit un message d'erreur parce qu'il n'attendait aucune frappe sur le clavier. *Un tel message n'indique aucunement que le calculateur est en panne.*

4. Interrompez la succession de tests en exécutant un arrêt système.
 - a. Appuyez sur **ON** et maintenez la pression.
 - b. Appuyez sur **▲**.
 - c. Relâchez votre pression sur **ON**.
5. Si le message vous donne un message d'erreur et *que vous n'avez pas causé l'échec du teste en appuyant sur une touche*, il nécessite réparation. Consultez les paragraphes sur la garantie, ci-dessous, et sur les réparations, en page 293.

Garantie

Le HP-28S (sauf pour ce qui concerne les piles ou d'éventuels dommages causés par les piles) est garanti par Hewlett-Packard contre tout vice de matière et de fabrication pour une durée d'un an à partir de la date de livraison, la facture d'achat faisant foi. Hewlett-Packard s'engage à réparer ou, éventuellement, à remplacer les pièces

qui se révéleraient défectueuses pendant la période de garantie. Cette garantie couvre les pièces et la main d'œuvre*. Elle disparaît en cas d'utilisation en dehors des spécifications ou de modification ou maintenance par un centre non reconnu par Hewlett-Packard.

Seuls les essais effectués à partir des programmes Hewlett-Packard seront considérés comme faisant foi lors des litiges concernant le fonctionnement du matériel. Aucune autre garantie explicite ou implicite n'est accordée. La responsabilité de Hewlett-Packard ne peut être engagée dans le cas d'une application particulière. La société ne peut être tenue responsable des dommages indirects.

Modifications

Les appareils vous sont livrés selon les spécifications en vigueur au moment de la fabrication. Hewlett-Packard n'est pas tenu de modifier les appareils déjà vendus.

Coût de la maintenance

Les réparations sont effectuées pour un prix forfaitaire incluant pièces et main d'œuvre. Ce forfait est sujet à la T.V.A. en France ou à des taxes similaires dans d'autres pays. Ces taxes apparaissent en détail sur les factures.

Les calculateurs endommagés par accident ou utilisation hors des spécifications ne sont pas couverts par le coût forfaitaire. Le prix de la réparation est alors fonction des pièces changées et du temps passé.

Garantie des réparations effectuées après la période de garantie

Tout appareil réparé par Hewlett-Packard est garanti, pièces et main d'œuvre, pendant 90 jours à compter de la date de réparation.

* Valable pour la France uniquement.

Lorsque l'acheteur est non-professionnel, ou consommateur au sens de la loi 78-23 du 10 janvier 1978, les obligations de HP, définies ci-dessus, ne sont pas exclusives de la garantie légale en matière de vices cachés (article 1641 et suivants du Code Civil).

Informations

Si vous désirez des informations plus précises concernant les termes de cette garantie, veuillez contacter :

- **En Europe :** Contactez votre distributeur ou le bureau commercial HP le plus proche ou le Siège Social de HP Europe. *N'envoyez pas votre calculateur pour réparation avant d'avoir contacté un bureau Hewlett-Packard ;*

Hewlett-Packard S.A.
150, Route du Nant-d'Avril
P.O. Box CH 1217 Meyrin 2
Genève, Suisse
Téléphone : (022) 82 81 11

- **Aux Etats-Unis :**

Hewlett-Packard Co.
1000, N.E. Circle Boulevard
Corvallis, OR 97330
(503) 757-2000

- **Dans les autres pays :**

Hewlett-Packard International
3495 Deer Creek Road
Palo Alto, CA 94304
U.S.A.
Téléphone : (415) 857-1501

Instructions d'expédition

Si vous devez envoyer votre calculateur pour réparation, conformez-vous aux indications suivantes :

- Joignez au calculateur la carte de maintenance portant la description de la panne.
- Si l'appareil est sous garantie, joignez une copie de la facture ou une preuve de la date d'achat.

- Expédiez le calculateur et les différents documents dans la boîte d'origine, ou, éventuellement, dans un autre emballage de protection pour éviter toute détérioration en cours de transport qui ne serait pas couverte par la garantie. Nous vous conseillons d'assurer le colis.
- Que le calculateur soit sous garantie ou non, les frais d'expédition, et éventuellement de douane, sont à votre charge. Le retour est effectué port payé.

Maintenance

Les appareils sont généralement réparés et ré-expédiés dans un délai de cinq jours ouvrables à dater de leur réception au centre de réparation approprié. Il s'agit d'un délai moyen pouvant varier selon l'époque de l'année et la charge de travail du service après-vente.

Réparations

Si votre calculateur doit être réparé, adressez-vous à un distributeur officiel de Hewlett-Packard, qui le fera parvenir à un centre de maintenance Hewlett-Packard.

Tous les centres de maintenance Hewlett-Packard ne sont pas équipés pour assurer la maintenance des calculateurs. Cependant, s'il y a un distributeur officiel Hewlett-Packard dans le pays où vous vous trouvez, vous pouvez être sûr que HP dispose d'un centre de maintenance dans ce pays.

S'il n'y a pas de distributeur officiel Hewlett-Packard dans le pays où vous vous trouvez, vous pouvez contacter le bureau commercial Hewlett-Packard le plus proche pour plus d'informations.

Sécurité - conformité aux normes

Le HP-28S a été testé selon les normes en vigueur aux Etats-Unis et dans le reste du monde. Ces tests passent en revue la sécurité mécanique et électrique du calculateur, les possibilités d'interférences radio, des études ergonomiques et acoustiques ainsi que l'innocuité des matériaux employés. Là où les règlements nationaux l'exigent, des approbations par les différents organismes concernés ont été obtenues et elles figurent sur l'étiquette du produit.

Interférences radio : France

Le HP-28S a été testé avec son imprimante Hewlett-Packard et a été trouvé conforme à toutes les normes françaises concernant les interférences radio pour les limites de classe B.

Si vous utilisez du matériel non fabriqué ou non recommandé par Hewlett-Packard, la configuration doit être en accord avec ces normes.

Utilisation en aviation (U.S.A.)

Le HP-28S et l'imprimante HP 82240A satisfont aux spécifications du RTCA (Radio Technical Commission for Aeronautics), Docket 160B, Section 21. La plupart des compagnies aériennes permettent l'utilisation de calculateurs en vol sur la base de cette qualification. Une simple vérification auprès d'un membre de l'équipage vous indiquera la position de la compagnie aérienne sur l'usage de calculateurs en vol.

B

Les menus et leurs jeux de libellés

Cette annexe rassemble toutes les commandes de chacun des menus du HP-28S. Les menus sont présentés en ordre alphabétique, d'ALGEBRA à TRIG. Pour trouver des informations détaillées sur un menu, consultez le *Dictionnaire*, qui se trouve dans le *Manuel de référence*. Le dictionnaire décrit tous les menus. Pour des renseignements plus précis sur une des commandes, voyez l'index des opérations qui se trouve à l'arrière du *Manuel de référence*. L'index des opérations dresse la liste de toutes les commandes et vous renvoie à une page où vous trouverez la description de la commande.

Cette annexe ne comprend pas les menus des opérations interactives qu'offrent CATALOG, FORM, l'algorithme de résolution d'équations et UNITS.

- CATALOG est décrite au chapitre 22 et exemplifiée à la page 31.
- FORM est décrite en page 112. Pour les détails, consultez le *Manuel de référence*.
- L'algorithme de résolution d'équations est décrit au chapitre 8, « Résolution d'équations ». Pour plus de détails, voyez « SOLVE », dans le *Manuel de référence*.
- UNITS est décrite en page 141.

Pour chacun des menus de cette annexe, les commandes sont groupées en rangées qui apparaissent sur l'affichage à un moment donné. En appuyant sur **[NEXT]**, vous passez à la ligne suivante, et l'appui sur **[PREV]** vous ramène à la ligne précédente.

La colonne intitulée « Commande » est le nom tel qu'il apparaît sur l'affichage. Sous le titre « Description » se trouve une brève description de la commande ou son nom complet. Vous trouverez à la page qui est indiquée ensuite, un exemple, une description ou une mention de cette commande dans ce manuel. Pour les commandes sans référence, consultez l'index des opérations dans le *Manuel de référence*.

ALGEBRA

	Commande	Description	Page
Ligne 1	COLCT	Regrouper des termes	111
	EXPAN	Développer des produits	111
	SIZE	Taille	
	FORM	Former une expression algébrique	112
	OBSUB	Substitution d'objet	
	EXSUB	Substitution d'expression	
NEXT			
Ligne 2	TAYLR	Série de Taylor	
	ISOL	Isoler une variable	112
	QUAD	Résolution d'équation du seconde degré	
	SHOW	Montre une variable	
	OBGET	Extraure un objet d'une expression	
	EXGET	Extraire une sous-expression	

ARRAY

	Commande	Description	Page
Ligne 1	→ARRY	Pile vers tableau	275
	ARRY→	Tableau vers pile	274
	PUT	Placer un élément dans un tableau ou une liste	
	GET	Appeller un élément	
	PUTI	Placer un élément et incrémenter l'index	
	GETI	Appeller un élément et incrémenter l'index	
NEXT			
Ligne 2	SIZE	Taille	274
	RDM	Redimensionner un tableau	
	TRN	Transposer	264
	CON	Tableau constant	
	IDN	Matrice identité	
	RSD	Reste	
NEXT			
Ligne 3	CROSS	Produit vectoriel	126
	DOT	Produit scalaire	126
	DET	Déterminant	128
	ABS	Valeur absolue	
	RNRM	Norme de ligne	
	CNRM	Norme de colonne	
NEXT			
Ligne 4	R→C	Réel en complexe	
	C→R	Complexe en réel	
	RE	Partie réelle	
	IM	Partir imaginaire	
	CONJ	Conjugué	
	NEG	Changer de signe	

BINARY

	Commande	Description	Page
Ligne 1	DEC	Mode décimal	140
	HEX	Mode hexadécimal	139
	OCT	Mode octal	140
	BIN	Mode binaire	140
	STWS	Stocker la taille de mot	139
	RCWS	Rapeller la taille de mot	
NEXT			
Ligne 2	RL	Permutation circulaire d'un bit vers la gauche	
	RR	Permutation circulaire d'un bit vers la droite	
	RLB	Permutation circulaire d'un octet vers la gauche	
	RRB	Permutation circulaire d'un octet vers la droite	
	R→B	Réel en binaire	261
	B→R	Binaire en réel	
NEXT			
Ligne 3	SL	Déplacer d'un bit vers la gauche	
	SR	Déplacer d'un bit vers la droite	
	SLB	Déplacer d'un octet vers la gauche	
	SRB	Déplacer d'un octet vers la droite	
	ASR	Déplacement arithmétique d'un bit vers la droite	
NEXT			
Ligne 4	AND	ET logique	
	OR	OU logique	
	XOR	OU EXCLUSIF logique	
	NOT	NON logique	

COMPLEX

	Commande	Description	Page
Ligne 1	R→C	Réel en complexe	83
	C→R	Complexe en réel	83
	RE	Partie réelle	83
	IM	Partie imaginaire	84
	CONJ	Conjugé	84
	SIGN	Signe	84
NEXT			
Ligne 2	R→P	Rectangulaire en polaire	86
	P→R	Polaire en rectangulaire	85
	ABS	Valeur absolue	85
	NEG	Changer de signe	85
	ARG	Argument	85

LIST

	Commande	Description	Page
Ligne 1	→LIST	Pile vers liste	181
	LIST→	Liste vers pile	181
	PUT	Placer un élément dans un tableau ou une liste	271
	GET	Appeler un élément	237
	PUTI	Placer un élément et incrémenter l'index	271
	GETI	Appeler un élément et incrémenter l'index	271
Ligne 2	NEXT		
	POS	Position	237
	SUB	Extraire une partie d'une liste ou d'une chaîne	276
	SIZE	Taille	271

LOGS

	Commande	Description	Page
Ligne 1	LOG	Logarithme en base 10	78
	ALOG	Antilogarithme en base 10	78
	LN	Logarithme népérien	78
	EXP	Exponentielle	78
	LNPI	Logarithme népérien de $1 + x$	78
	EXPM	Exponentielle moins 1	78
NEXT			
Ligne 2	SINH	Sinus hyperbolique	78
	ASINH	Sinus hyperbolique inverse	78
	COSH	Cosinus hyperbolique	78
	ACOSH	Cosinus hyperbolique inverse	78
	TANH	Tangente hyperbolique	78
	ATANH	Tangente hyperbolique inverse	78

MEMORY

	Commande	Description	Page
Ligne 1	MEM	Mémoire disponible	188
	MENU	Créer un menu personnalisé	195
	ORDER	Trier les variables	184
	PATH	Chemin en cours	67
	HOME	Sélectionner le répertoire HOME	71
	CRDIR	Créer un répertoire	66
NEXT			
Ligne 2	VARS	Variables du répertoire en cours	184
	CLUSR	Effacer le répertoire en cours	184
	DIR		
	DIR		
	DIR		

MODE

	Commande	Description	Page
Ligne 1	STD	Format d'affichage numérique standard	38
	FIX	Format d'affichage numérique fixe	38
	SCI	Format d'affichage numérique scientifique	38
	ENG	Format d'affichage numérique ingénieur	38
	DEC	Angles en degrés	74
	RAD	Angles en radians	74
NEXT			
Ligne 2	CMD	Valider ou invalider COMMAND	210
	UNDO	Valider ou invalider UNDO	211
	LAST	Valider ou invalider LAST	211
	ML	Valider ou invalider le mode d'affichage multi-ligne	208
	RDX,	Valider ou invalider RDX,	37
	PRMD	Modes d'impression et d'affichage	

PLOT

	Commande	Description	Page
Ligne 1	STEQ	Stocker une équation	90
	RCEQ	Rappeler une équation	
	PMIN	Limite inférieure de traçage	95
	PMAX	Limite supérieure de traçage	95
	INDEP	Variable indépendante	
	DRAW	Tracer	90
NEXT			
Ligne 2	PPAR	Rappeler les paramètres de traçage	90
	RES	Résolution	
	AXES	Axes	
	CENTR	Centre	94
	*W	Multiplier la largeur	
	*H	Multiplier la hauteur	93
NEXT			
Ligne 3	STOΣ	Stocker sigma	
	RCLΣ	Rappeler sigma	
	COLΣ	Choisir colonnes de la matrice statistiques	
	SCLΣ	Mettre à l'échelle les paramètres de traçage en fonction de sigma	
	DRWΣ	Tracer sigma	
	 		
NEXT			
Ligne 4	CLLCD	Effacer l'affichage	
	DGTIZ	Numériser	
	PIXEL	Allumer un pixel	
	DRAX	Tracer les axes	
	CLMF	Effacer l'indicateur de message	
	PRLCD	Imprime l'affichage	

PRINT

	Commande	Description	Page
Ligne 1	PR1	Imprimer le contenu du niveau 1	151
	PRST	Imprimer le contenu de la pile	152
	PRVAR	Imprimer une variable	152
	PRLCD	Imprimer l'affichage	149
	CR	Retour chariot	
	TRAC	Valider ou invalider le mode Auto	150
<div>NEXT</div> Ligne 2	PRSTC	Imprimer la pile (format compact)	
	PRUSR	Imprimer les variables utilisateur	
	PRMD	Modes d'impression	
	PRM1		
	PRM2		
	PRM3		

PROGRAM BRANCH

	Commande	Description	Page
Ligne 1	IF	Commencer la clause IF	226
	IFERR	Commencer la clause IF ERROR	227
	THEN	Commencer la clause THEN	226
	ELSE	Commencer la clause ELSE	226
	END	Terminer la structure de programme	226
NEXT			
Ligne 2	START	Commencer une boucle définie	228
	FOR	Commencer une boucle définie	229
	NEXT	Terminer une boucle définie	228
	STEP	Terminer une boucle définie	230
	IFT	Commande If-Then	227
	IFTE	Fonction If-Then-Else	226
NEXT			
Ligne 3	DO	Élément d'une boucle indéfinie	231
	UNTI	Élément d'une boucle indéfinie	231
	END	Terminer une structure de programme	231
	WHIL	Élément d'une boucle indéfinie	232
	REPEA	Élément d'une boucle indéfinie	232
	END	Terminer une structure de programme	232

PROGRAM CONTROL

	Commande	Description	Page
Ligne 1	SST	Exécution pas-à-pas	250
	HALT	Suspendre un programme	234
	ABORT	Abandonner un programme	
	KILL	Abandonner les programmes suspendus	250
	WAIT	Faire une pause dans un programme	234
	KEY	Afficher une chaîne représentant la plus ancienne touche en mémoire tampon	234
NEXT			
Ligne 2	BEEP	Tonalité	234
	CLLCD	Effacer l'affichage	234
	DISP	Afficher un objet	234
	CLMF	Effacer l'indicateur de message	234
	ERRN	Afficher le dernier numéro d'erreur	
	ERRM	Message d'erreur	

PROGRAM TEST

	Commande	Description	Page
Ligne 1	SF	Armer un indicateur binaire	205
	CF	Désarmer un indicateur binaire	205
	FS?	Indicateur binaire armé ?	225
	FC?	Indicateur binaire désarmé ?	
	FS?C	Indicateur armé ? Désarmer	
	FC?C	Indicateur désarmé ? Désarmer	
NEXT			
Ligne 2	AND	And	
	OR	Or	
	XOR	OU EXCLUSIF logique	
	NOT	NON logique	232
	SAME	Identité	231
	==	Egalité	222
NEXT			
Ligne 3	STOF	Stocker les indicateurs binaires	156
	RCLF	Rappeler les indicateurs binaires	156
	TYPE	Donner le type d'un objet	232

REAL

	Commande	Description	Page
Ligne 1	NEG	Changer de signe	78
	FACT	Factorielle (gamma)	78
	RAND	Nombre aléatoire	78
	RDZ	Racine des nombres aléatoires	78
	MAXR	Constante symbolique réelle maximale	79
	MINR	Constante symbolique réelle minimale	79
NEXT			
Ligne 2	ABS	Valeur absolue	
	SIGN	Signe	
	MANT	Mantisse	
	XPON	Exposant	
NEXT			
Ligne 3	IP	Partie entière	
	FP	Partie fractionnaire	
	FLOOR	Entier le plus petit suivant	272
	CEIL	Entier le plus grand suivant	272
	RND	Arrondi	
NEXT			
Ligne 4	MAX	Maximum	
	MIN	Minimum	
	MOD	Modulo	
	%T	Pourcentage du total	

SOLVE

	Commande	Description	Page
Ligne 1	STEQ	Stocker une équation	64
	RCEQ	Rappeler une équation	
	SOLVR	Menu des variables de l'algorithme de résolution d'équations	102
	ISOL	Isoler une variable	110
	QUAD	Solution d'un polynôme du second degré	108
	SHOW	Montrer une variable	
NEXT			
Ligne 2	ROOT	Trouver une racine numérique	

STACK

	Commande	Description	Page
Ligne 1	DUP	Dupliquer l'objet du niveau 1	178
	OVER	Dupliquer l'objet du niveau 2	178
	DUP2	Dupliquer deux objets	178
	DROP2	Eliminer deux objets	179
	ROT	Déplacer l'objet du niveau 3 au niveau 1	178
	LIST→	Liste vers pile	181
NEXT			
Ligne 2	ROLLD	Faire descendre les objets d'un niveau	178
	PICK	Dupliquer le même objet au niveau 1	178
	DUPN	Dupliquer n objets	178
	DROPN	Eliminer n objets	179
	DEPTH	Compter les objets de la pile	181
	→LIST	Pile vers liste	181

STAT

	Commande	Description	Page
Ligne 1	Σ+	Ajouter une observation	132
	Σ-	Soustraire une observation	133
	NΣ	Nombre d'observations	134
	CLΣ	Effacer les observations	132
	STOΣ	Stocker la matrice statistique	275
	RCLΣ	Rappeler la matrice statistique	264
NEXT			
Ligne 2	TOT	Total	
	MEAN	Moyenne	134
	SDEV	Ecart type	135
	VAR	Variance	135
	MAXΣ	Valeur maximale	
	MINΣ	Valeur minimale	
NEXT			
Ligne 3	COLΣ	Choisir des colonnes dans la matrice statistique	136
	CORR	Corrélation	136
	COV	Covariance	136
	LR	Régression linéaire	137
	PREDV	Valeur prévue	137
	Σ		
NEXT			
Ligne 4	UTPC	Loi du Chi carré	
	UTPF	Loi de F (Snedecor)	
	UTPN	Loi normale	
	UTPT	Loi de T (Student)	
	COMB	Combinaisons	
	PERM	Permutations	

STORE

	Commande	Description	Page
Ligne 1	STO+	Stocker une somme arithmétique	
	STO-	Stocker une différence arithmétique	
	STO*	Stocker un produit arithmétique	
	STO/	Stocker un quotient arithmétique	
	SNEG	Changer le signe d'une variable	
	SINV	Stocker l'inverse	
NEXT			
Ligne 2	SCONJ	Stocker le conjugué	

STRING




	Commande	Description	Page
Ligne 1	→STR	Object en chaîne	258
	STR→	Chaîne en objet	175
	CHR	Créer une chaîne de un caractère	156
	NUM	Code de caractère	156
	→LCD	Chaîne vers affichage	157
	LCD→	Affichage vers chaîne	157
NEXT			
Ligne 2	POS	Position	
	SUB	Sous-ensemble	
	SIZE	Taille	258
	DISP	Afficher	156

TRIG

	Commande	Description	Page
Ligne 1	SIN	Sinus	74
	ASIN	Arc sinus	74
	COS	Cosinus	74
	ACOS	Arc cosinus	74
	TAN	Tangente	74
	ATAN	Arc tangente	74
NEXT			
Ligne 2	P→R	Polaire en rectangulaire	76
	R→P	Rectangulaire en polaire	76
	R→C	Réel en complexe	76
	C→R	Complexe en réel	76
	ARG	Argument	76
NEXT			
Ligne 3	→HMS	Heures décimales en heures-minutes-secondes	76
	HMS→	Heures-minutes-seconde en heures décimales	76
	HMS+	Ajouter des heures-minutes-secondes	76
	HMS-	Soustraire des heures-minutes-secondes	76
	D→R	Degrés en radians	77
	R→D	Radians en degrés	77



















Index des touches

Vous trouverez ici la description des touches du clavier de votre calculateur. En premier lieu vous trouverez un index alphabétique des touches du clavier de gauche de votre calculateur ; il est suivi de l'index alphabétique des touches du clavier de droite. Il y a aussi un index spécial des touches du menu du curseur (les libellés imprimés en blanc au-dessus de la rangée supérieure de touches, sur le clavier de droite).

Cet index inclut les touches préfixées (comme  **ARRAY** et  **OFF**). Il ne traite pas des touches de caractères, c'est-à-dire des touches de **A** à **Z** et de **0** à **9**, qui inscrivent simplement un caractère en ligne de commande. Les autres touches de caractères comprennent **[**, les opérateurs tels que **=** et les constantes symboliques telles que  **π** . Ces caractères ont une signification particulière pour le calculateur, mais ces touches sont simplement des touches de caractères. Si vous ne trouvez pas une touche dans cet index, c'est qu'il s'agit d'une touche de caractère.

Pour chaque touche, il y a une courte description de son action et une référence vous renvoie à une page. Si une touche n'est pas mentionnée du tout dans ce manuel, ou si vous désirez plus d'informations sur une touche quelconque, cherchez dans l'index des opérations qui se trouve au dos du *Manuel de référence*.

Clavier de gauche



Touche	Description	Page
 ALGBRA	Sélectionne le menu ALGEBRA.	110
 ARRAY	Sélectionne le menu ARRAY.	124
 BINARY	Sélectionne le menu BINARY.	138
 BRANCH	Sélectionne le menu PROGRAM BRANCH.	222
 CATALOG	Commence l'affichage du catalogue.	196
 COMPLX	Sélectionne le menu COMPLEX.	83
 CONTRL	Sélectionne le menu PROGRAM CONTROL.	234
LC	Alterne entre mode minuscules et mode maj.	168
 LIST	Sélectionne le menu LIST.	102
 MENUS	Alterne entre verrouillage du menu et mode normal.	192
 MEMORY	Sélectionne le menu MEMORY.	182
 PRINT	Sélectionne le menu PRINT.	149
 REAL	Sélectionne le menu REAL.	78
 STACK	Sélectionne le menu STACK.	176
 STAT	Sélectionne le menu STAT.	131
 STORE	Sélectionne le menu STORE.	191
 STRING	Sélectionne le menu STRING.	156
 TEST	Sélectionne le menu PROGRAM TEST.	225
 UNITS	Sélectionne le catalogue UNITS.	141
α	Alterne entre modes de saisie.	171













Clavier de droite

Touche	Description	Page
ATTN (ON)	Met fin à l'exécution du programme ; efface la ligne de commande ; sort des catalogues, de FORM, des affichages de tracés.	216
CHS	Change le signe d'un nombre en ligne de commande ou exécute NEG.	168
CLEAR	Efface la pile.	179
COMMAND	Déplace une saisie de la <i>pile</i> de commande en <i>ligne</i> de commande.	174
CONT	Continue un programme interrompu.	235
CONVERT	Effectue une conversion d'unités.	143
CUSTOM	Sélectionne le dernier menu personnalisé ayant été affiché.	192
d/dx	Dérivée.	117
DROP	Elimine un objet de la pile.	179
EDIT	Copie l'objet du niveau 1 en ligne de commande pour correction.	173
EEX	Entre l'exposant en ligne de commande.	168
ENTER	Analyse et évalue la ligne de commande.	173
EVAL	Evalue un objet.	118
LAST	Renvoie les derniers arguments.	179
LOGS	Sélectionne le menu LOGS.	77
MODE	Sélectionne le menu MODE.	36
NEXT	Affiche la ligne suivante de libellés de menu.	192
ON (ATTN)	Met le calculateur sous tension ; met fin à l'exécution d'un programme ; efface la ligne de commande ; sort des catalogues, de FORM, des tracés de courbes.	216
OFF	Eteint le calculateur.	20
PLOT	Sélectionne le menu PLOT.	89
PREV	Affiche la ligne précédente de libellés de menu.	192

Touche	Description	Page
PURGE	Elimine une ou plusieurs variables.	183
RCL	Rappelle le contenu d'une variable, sans l'évaluer.	183
ROLL	Déplace l'objet de niveau $n+1$ au niveau 1.	178
SOLV	Sélectionne le menu SOLVE.	99
STO	Stocke un objet dans une variable.	183
SWAP	Alterne les objets des niveaux 1 et 2.	178
TRIG	Sélectionne le menu TRIG.	74
UNDO	Remplace le contenu de la pile.	180
USER	Sélectionne le menu USER.	49
VIEW↑	Déplace la fenêtre d'affichage d'une ligne vers le haut.	177
VIEW↓	Déplace la fenêtre d'affichage d'une ligne vers le bas.	177
VISIT	Copie un objet en ligne de commande pour correction.	173
x^2	Met un nombre ou une matrice au carré.	40
$1/x$	Inverse (réciproque).	40
+	Ajoute deux objets.	41
-	Soustrait deux objets.	41
×	Multiplie deux objets.	41
÷	Divise deux objets.	42
%	Pour-cent.	43
%CH	Pourcentage de changement.	43
^	Elève un nombre à une puissance.	42
\sqrt{x}	Extrait la racine carrée.	40
\int	Intégrale définie ou indéfinie.	120
	Touche préfixe, « Shift ».	29
	Sélectionne le menu du curseur ou rétablit le dernier menu.	168
	Espace arrière.	168
→NUM	Force un résultat numérique.	75

Le menu du curseur

Les libellés de ce menu sont imprimés en blanc au-dessus de la rangée supérieure de touches du clavier de droite. Le menu est en action lorsque la ligne de commande est affichée et qu'aucun libellé de menu n'est visible. Pour sélectionner le menu du curseur lorsque des libellés *sont* affichés, appuyez sur la touche . Pour revenir au menu précédent, appuyez sur  une seconde fois.

Touche	Description	Page
	Alterne entre mode de remplacement et mode d'insertion.	167
	Supprime tous les caractères placés à la gauche du curseur.	168
	Supprime le caractère se trouvant sous le curseur.	167
	Supprime le caractère sous le curseur et tous ceux qui se trouvent à sa droite.	168
	Déplace le curseur vers le haut.	167
	Déplace le curseur vers l'extrémité supérieure de l'affichage.	168
	Déplace le curseur vers le bas.	167
	Déplace le curseur vers l'extrémité inférieure de l'affichage.	168
	Déplace le curseur vers la gauche.	167
	Déplace le curseur vers le début de la ligne.	168
	Déplace le curseur vers la droite.	167
	Déplace le curseur vers l'extrémité de droite de la ligne.	168

Index des sujets

Les nombres de pages imprimées en **grasses** indiquent les références primaires ; celles imprimées en caractères normaux indiquent les références secondaires.

A

Affichage, impression, 149, 216
Algorithme de résolution
 d'équations, 63-64, **98-109**
Angles d'un tracé, 94
Annonceurs, 27, **29**
Annulation, opérations système, 215
Arguments
 définis, 25
 ordre des-, 41, 43
 utilisation, 197
Arguments précédents, 179-180
Arrêt système, 217
Assombrir l'affichage, 21
Attention, 216

B

Base de nombre entier, 209-210
Base des entiers binaires, 139
Batteries, 286-288
BDISP, programme, 259-262
Boucles finies, **228-230**, 248, 260, 274
 emboîtées, 270-271
Boucles indéfinies, **231-232**, 254, 257
BOXR, programme, 245-246
BOXS, programme, 241-244
Boîtier, ouverture et fermeture, 18

C

Calculateurs algébriques, 302-305
Calculs en chaîne, 45
Calculs financiers, 103-106
Calculs sur des flux constants,
 103-106
Caractère nouvelle ligne, 169
Carré, 40
Catalogue
 de commandes, 196-197
 d'unités, 141-143
Catalogue des unités, 26, 29
Chaînes, 156-157, 258
Chaînes graphiques, 157
Chaînes d'unités, 144-145
Changement
 de répertoire, 275-279
 de variable, 51
Changement de signe, 39
Chemin en cours, 60, 184-185
Classes d'objets, 199
Clause, 225-226
Clavier, 26-27, 328-330
Combinaison de termes, 115
Commandes, 164-165
 catalogue des-, 26, 29, 31-33
Compteur, **228-230**, 260, 271, 274
Constantes symboliques, 163
Constantes, mode, 206-207

- Contraste de l'affichage, 21
- Contraste, affichage, 21, 216
- Conversion de températures, 143–144
- Conversion en pieds cubiques, 146
- Coordonnées polaires, 84–88
- Copie d'objets dans la pile, 178
- Correction d'erreurs, 47
- Correction des erreurs de programmation, 250
- Corrélation, 136
- COT, programme, 80–81
- Cotangente, 80–81
- Couvercle de la batterie, emplacement, 19
- Covariance, 136
- Covariance, matrice, 263
- Création
 - d'un répertoire, 183
 - d'une variable, 49, 54
- Curseur, indiquant les modes, 172
- Curseur, menu, 30, 69, 166–168

D

- Degrés-minutes-secondes, 76
- Diagnostics, 218–219
- Diagramme de la pile, définition, 240
- Différentiation, 117–120
- Données, 199
- Données dépendantes, 136
- Données indépendantes, 136
- Décimal, séparateur, 36, 209
- Décimales, 37
- Délimiteurs, 26, 28, 169
- Dépassement inférieur de capacité, 212
- Dépassements de capacité, 212
- Déplacement d'objets dans la pile, 178
- Détection d'erreur, 227–228, 259, 278
- Déterminant, 128
- Développement d'une expression algébrique, 111, 256

E

- Ecart-type, 135
- Echelle d'un tracé, 91
- Eclaircir l'affichage, 21
- Effacer
 - des données statistiques, 132
 - la pile, 44
 - le contenu entier de la mémoire, 20
- Éléments de tableaux, 272
- Élimination
 - d'un répertoire, 187
 - d'une variable, 52
- Élimination d'objets de la pile, 179
- ENTER, 24, 173
- Entiers binaires, 156
- Entrer un exposant, 39
- Entretien et soins, 289
- Equation en cours, 90
- Equations linéaires, système d'—, 130
- Equations, 162–163
 - du second degré, 107
 - évaluation, 203
 - racines d'une—, 107
 - tracage, 97
- Espace arrière, 27, 30
- Estimations pour l'algorithme de résolution d'équations, 99, 102
- Etat en cours, 258
- Évaluation d'une expression avec l'algorithme de résolution d'équations, 65
- Évaluation d'une variable, 50, 56
- Évaluation différée, 198
- Évaluation, 198–199
- Exceptions, mathématiques, 211–212
- EXCO, programme, 255–256
- Exécution pas-à-pas, 250–253
- Exposant, 38
- Expressions et équations du second degré, 107

Expressions, 34, 161–162
 évaluation, 202–203
 évaluation utilisant l'algorithme de
 résolution d'équations, 65
 provenant de calculs de la pile,
 59–60
 zéro d'une—, 92, **98–100**, 107
Extinction automatique, 20
Extrême d'un graphe ou d'un tracé,
 96

F

FIB1, programme, 247
FIB2, programme, 248–253
Fibonacci, nombres de—, 246–249
Fonction analytique, 164–165
Fonction, 164–165
 diadique, 41
 évaluation, 203–204
 monadique, 41
Fonctionnement, 164–165
Fonctions diadiques, 40
Fonctions et commandes de test, 224
Fonctions exponentielles, 77–78
Fonctions hyperboliques, 77–78
Fonctions logarithmiques, 77–78
Fonctions trigonométriques, 73–77
Fonctions utilisateur, 79–81, 161,
 202, 242
 emboîtées, 245
Force, unités de—, 146

G

Gallons, conversion, 146
Gamma, fonction, 78
Garantie, 291–293
Globales, variables, 80, 182–183
Grammes, conversion, 147–148
Guillemets, noms entre—, 57
G→O, programme, 148

H

Heures-minutes-secondes, 76
HOME, répertoire, 60, 71
HP RPN, calculateurs, 296–301
HP Solve. Voir algorithme de
 résolution d'équations

I

Impression auto, 150, 213
Impression d'un relevé continu, 150
Impression d'un tracé, 91
Inch, conversion, 144
Incrément pour le compteur, 230
Indicateurs d'utilisateur, 205, 225,
 258
Indicateurs de pile, 225
Indicateurs utilisateur, 225
Interligne, 214
Intégration numérique, 122–123
Intégration symbolique, 121
Intégration, 120–123
Inverse, 40
Inverse, 40
Inversion d'une matrice, 128
Isolation d'une variable, 109–116

K, L

KEY?, programme, 239
Libellés de menu, 27, 31
Ligne de commande, 22, 166
 récupération, 174
Listes, 158, 276
 éléments d'une—, 272
LMED, programme, 272–273
Logique de la pile, 25

M

Manque de mémoire, 188–190
Mantisse, 38
Marqueur de base (libellés), 139
Matrice de covariance, 263
Matrice statistique en cours, 132
Matrice, définie, 124
Maximum d'une expression, 100
MEDIAN, programme, 273–275
Médiane définie, 272
Mémoire disponible, 188
Mémoire permanente, 20
Mémoire utilisateur, 48
Mémoire, manque de—, 188–190
Mémoire, réinitialisation, 20, 217
Menu d'utilisateur personnalisé, 235
Menu de saisie personnalisé, 234–235
Menu du curseur, 30, 69, 166–168
Menu Lock, 169
Menus personnalisés, 192, 195, 276, 277
Message, impression, 151
Milles par heure, conversion, 145
Millimètres, conversion, 144
Minimum d'une expression, 100
Mise en facteurs, 114–115
Mise hors tension automatique, 20
Mode « plusieurs lignes », 208
Mode « retour chariot » automatique, 213
Mode Angle, 73, 205–206
Mode avertisseur, 206
Mode constantes, 206–207
Mode d'affichage des nombres, 37, 209
Mode d'insertion, 70
Mode de résultat numérique, 203–204
Mode de résultat symbolique, 203

Mode de résultats, 207
Mode de saisie algébrique, 34, 51, 170–172
Mode de saisie alpha(bétique), 55, 170–172
Mode de saisie immédiate, 170–172
Mode Degrés, 73
Mode minuscules (LC), 26, 28
Modes de récupération après erreur, 210–211
Modes de saisie, 51, 169–172, 207
Modes par défaut, 205
Modes, 205–214
 indication par le curseur, 172
Modifications, 69, 173
 de données statistiques, 133
Monadiques, fonctions, 40
Moyenne, 134
MULTI, programme, 253–255

N

Négation, 40, 79
Niveau 1, impression, 151
Niveaux de la pile, 27, 31, 176
Nombres aléatoires, 78
Nombres complexes, 82, 155
Nombres négatifs, 39
Nombres réels, 155
Noms globaux, 159
 évaluation, 200–201
Noms locaux, 159
 évaluation, 200
Noms réservés, 159–160
Noms sans guillemets, 57
Noms, 159–160
 Noms entre guillemets ou non, 57
Noms, évaluation, 199–201
Nouvelle ligne, caractère, 169
Numérisation, 93, 99

O

Objets, 154
Objets, évaluation, 199
Objets, types, 26–29
Objets algébriques, 161–163
évaluation, 202–203
Observations, 132
Onces, conversion, 147–148
Opérations sur les matrices, 263
Optimisation des performances,
190–191
Ordre des arguments, 41, 43
Organisation de la mémoire,
190–191
Ouverture du boîtier, 18
O→G, programme, 147

P

PAD, programme, 257–258
Paramètres de traçage, 89
Paramètres statistiques, 136, 263
Période, 169
Permutation de termes, 113–114
Pi, 74–75
Pieds par seconde, conversion, 145
Pieds, unités, 146
Pile, **176–181**
Pile, 22, 272
effacement, 44
impression, 152
Port d'imprimante, emplacement, 19
Pourcentages, 43
Préservation de l'état en cours, 258
PRESERVE, programme, 258–259
Prêt, calculs, 103–106
Procédure, évaluation d'objets, 199,
201–204
Produit scalaire, 126
Produit vectoriel, 126
Programmation structurée, 202, 241,
255

Programmes, 160–161
dans les expressions algébriques,
263
en tant qu'arguments, 254
évaluation, 201–202
Proposition, 162
PSUM, programme, 86–88
Puissances, 42

R

Racines, 42
Racine carrée, 40
Racines d'une équation, 107
Radians, mode, 73
Rappel d'une variable, 50, 56
Rappel de la pile, 180
Rassembler une expression
algébrique, 111, 256
Recommencer un test, 218
Récursion, 246–247
Refermer le boîtier, 18
Règle de dérivation, 118–119
Régression linéaire, 137
Remise à zéro de la mémoire, 20
RENAME, programme, 54–55
Renommer une variable, 52
Répertoire en cours, 60, 184
Répertoire parent, 60, 183, 275
Répertoires, **183–187**
Représentation graphique d'une
expression, **89–97**
Résultat infini, 211–212
Retour en arrière, 47
RPN, 25
avantages des—, 62, 66, 71, 183
changement, 275–279
création, 60

S

Séparateur décimal, 36, 209
Séparateur décimal, choix, 36
Séparateurs, 169
Séries de Taylor, 120
Service après-vente, 293–295
Shift, touche, 27, 29
SORT, programme, 270–272
Sous-programmes, 260
Sous-répertoire, 60, 183, 275
Stockage de paramètres de tracé, 96
Structures conditionnelles, 223–228
Structures de boucles, 228
Structures de programme emboîtées, 233
 boucles définies, 270–271
 fonctions d'utilisateur, 245
 structures de variables locales, 270
Structures de programme, 161
 évaluation, 201
SUMS, programme, 263–264
 Σ GET, programme, 265
 Σ X2, programme, 266
 Σ XY, programme, 267
 Σ Y2, programme, 266
Système d'équations linéaires, 130

T

Tableaux
 définition, 124
 en syntaxe algébrique, 157
 utilisation minimum de la
 mémoire, 191
Taille de mot, entiers binaires, 210
Taille de mots, 138–139
Test d'égalité, 224
Test du clavier, 219
Tests automatiques, 218–219

Touches de menu, 27, 31
Translation d'un tracé, 93
Transposition, 264
Tri par échanges de paires de clés, 270
Types d'objets, 26–29

U

Unité préfixée, 144
Utilisation des commandes, 197

V

Valeur principale, 206
Valeurs prévues, 137
Variable formelle, 200
Variable numérique, 49
Variables globales, 80, 182–183
Variables locales, 80, 86, 147, 179, 222–223, 242, 259
 emboîtées, 270
 évaluation, 254
Variables, 48
 création, 49, 54
 élimination, 52
 impression, 152
 isolation, 109–116
Variance, 135
Vecteurs, définition, 124
Virgule, 169
Vitesse d'impression, 213

Z

Zéro d'une expression, 92, 98–100, 107, 162

Comment contacter Hewlett-Packard

Renseignements sur l'utilisation du calculateur. Si vous avez des questions relatives au fonctionnement du calculateur et que vous ne pouvez trouver de réponse dans ce manuel (après avoir consulté **Réponses à des questions fréquemment posées**, l'index et la table des matières) ou dans le manuel de référence, consultez votre **distributeur Hewlett-Packard** ou bien adressez-vous directement à :

Pour la France :

Appelez le numéro d'assistance
téléphonique :
(1) 43 37 63 50

Pour la Belgique :

Hewlett-Packard Belgique S.A./N.V.
100, boulevard de la Woluwe
B-1200 Bruxelles
Tél. : (02) 762-32.00

Pour la Suisse francophone :

Hewlett-Packard (Suisse) S.A.
7, rue du Bois-du-Lan
CH-1217 Meyrin 1-Genève
Tél. : (22) 83.11.11

Pour le Canada francophone :

Hewlett-Packard (Canada)
17500 Route Transcanadienne
Voie de Service Sud
Kirkland, Québec H9J 2M5
Tél. : (514) 697-42.32

L'annexe A indique comment déterminer si le calculateur nécessite réellement une réparation. Elle indique également comment procéder pour faire réparer votre calculateur.

Table des matières

Page 15 Comment utiliser ce manuel

17 1ère partie : principes de base

Pour commencer • Arithmétique • Utiliser des variables
Répéter un calcul • Fonctions opérant sur les nombres réels

Fonctions opérant sur les nombres complexes • Représentation graphique d'expressions • Résolution d'équations
Solutions symboliques • Calcul différentiel et intégral • Vecteurs et matrices

Statistiques • Arithmétique binaire • Conversion d'unités
Impression

153 2ème partie : résumé des caractéristiques du calculateur

Objets • Opérations, commandes et fonctions

La ligne de commande • La pile opérationnelle • Mémoire • Menus

Catalogue des commandes • Evaluation • Modes
Opérations au niveau du système

221 3ème partie : programmation

Structures de programmes • Programmes interactifs
Exemples de programmation

281 Annexes et index

Garantie et service après-vente

Les menus et leurs jeux de libellés • Index des touches •
Index des sujets



**HEWLETT
PACKARD**

**Reorder Number
00028-90080**

00028-90081 Français, French
Printed in West Germany 2/88
Imprimé en R.F.A.

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please to not make copies of this scan or
make it available on file sharing services.