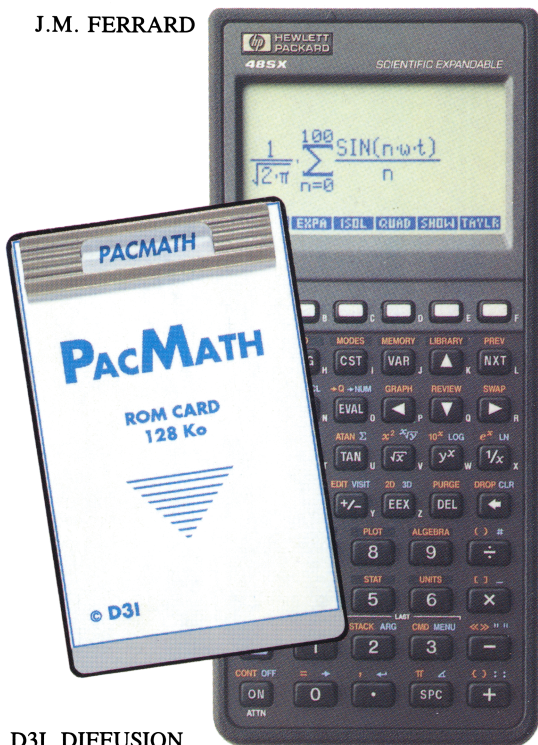


La Carte PACMATH

Pour HP48SX

J.M. FERRARD



D3I DIFFUSION

La Carte PACMATH

Pour HP48SX

J.M. FERRARD

D3I DIFFUSION

Hewlett-Packard, HP 48, HP 48 S, HP 48 SX,
sont des marques déposées

© 1992 – D3I DIFFUSION

Tous les efforts ont été faits pour que les informations, programmes et schémas présentés dans ce livre soient aussi exacts et complets que possible. Ni les auteurs, ni l'éditeur ne pourront en aucun cas être tenus pour responsables des préjudices de quelque nature que ce soit, pouvant résulter de leur utilisation, tant dans un cadre privé, que commercial ou professionnel.

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit, ou ayants cause, est illicite » (alinéa premier de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

NOTICE

Les informations contenues dans cet ouvrage peuvent être changées sans préavis.

D3I et HEWLETT-PACKARD ne donnent aucune garantie en ce qui concerne l'usage qui pourrait être fait des formules et exemples qui sont décrits dans le présent livret.

Le livret et toutes les informations qu'il contient ne peuvent être copiés en tout ou partie sans notre consentement.

HEWLETT-PACKARD est une marque déposée
de HEWLETT-PACKARD CORPORATION

© D3I - 1992

Chez le même EDITEUR et du même AUTEUR, J.-Michel FERRARD :

- **La Maîtrise de la HP 28 S**
TOME 1 : PROGRAMMATION ET EXERCICES
- **La Maîtrise de la HP 28 S**
TOME 2 : PROGRAMMATION ET APPLICATIONS
- **La Maîtrise de la HP 48**
TOME 1 : PROGRAMMATION ET EXERCICES
- **La Maîtrise de la HP 48**
TOME 2 : PROGRAMMATION ET APPLICATIONS
- **Mastering your HP 48**
PROGRAMMING AND APPLICATIONS
(Version anglaise)
- **HP 48, NOCH EFFEKTIVER NUTZEN**
ANWENDUNGSPROGRAMME FÜR FORTGESCHRITTENE
(Version allemande)
- **Les secrets de la HP 48 – TOME 1**

D3I DIFFUSION

20 rue Hermès - 31526 Ramonville St Agne - France

Tél. : (33) 62 24 66 12 - Fax : (33) 62 19 03 77

La carte PacMath

◆ Présentation

La carte **PacMath** contient une librairie de plus de 100K (105000 caractères environ), stockée sur une carte ROM de 128K.

Elle regroupe tous les programmes du tome 2 du livre "**La maîtrise de la HP48**" (D3I Diffusion), ainsi que de nombreux utilitaires extraits du tome 1. Au total, près de 400 programmes. Voilà bien la plus large collection de programmes (mathématiques pour la plupart) réunis sur un si petit support, pour être utilisés sur une calculatrice!

Le choix de la carte **PacMath** présente pour l'utilisateur bien des avantages: il est dispensé d'avoir à taper lui-même les programmes, il est sûr que ceux-ci sont conformes, et le contenu de la carte est en sécurité (pas de *Memory Clear* à craindre...).

Pour une meilleure autonomie, les programmes de la carte **PacMath** bénéficient tous d'une description dans un système d'aide spécialement mis au point à cette occasion. Pour vous donner une idée de ce que cela représente, sachez que la taille du fichier d'aide constitue environ 40% de la taille totale de la carte **PacMath**!

Dans ce système d'aide, chaque programme est brièvement décrit (son rôle, et la nature des arguments qu'il utilise et du résultat qu'il renvoie). Bien sûr, ces explications ne sauraient remplacer le texte intégral de présentation que l'on trouve dans les deux tomes de "**La maîtrise de la HP48**" (ou alors c'est une carte de 2 MégaOctets qu'il aurait fallu utiliser).

◆ Mise en route

Pour utiliser la carte **PacMath**, il vous faut l'enficher sur un des deux slots d'extension de la HP48SX (port 1 ou port 2, peu importe). Vous pouvez alors conserver indéfiniment la carte dans ce port d'extension, ou bien la retirer pour la ranger. Dans tous les cas, veillez à effectuer ces opérations **après avoir éteint la calculatrice**.

Après avoir introduit la carte dans son support, vous pouvez rallumer votre calculatrice. Vous devez alors voir apparaître un message de Copyright, ce qui est le signe que l'installation se passe correctement.

N.B: très exceptionnellement, compte tenu du volume de l'objet stocké sur la carte, il peut se produire une erreur "*Invalid Card Data*". Si cela vous arrive, vérifiez bien que la carte est correctement positionnée (poussée à fond dans son logement), et relancez l'installation par un double appui sur les touches **ON** et **C** (d'abord **ON**, puis **C**, puis relâchez les deux). Tout doit alors rentrer dans l'ordre.

◆ Vérifier l'installation

Pour contrôler que la librairie est bien installée, vous pouvez vérifier la présence d'une entrée **MATH** (c'est le nom de la librairie) dans le menu **LIBRARY**, et d'une entrée **1357** (le numéro de la librairie) dans le sous-menu **PORT1** ou **PORT2**.

◆ Attacher ou détacher la librairie

La librairie stockée sur la carte **PacMath** est "*auto-attachable*", au répertoire **HOME**, ce qui signifie que les programmes qui la composent sont immédiatement utilisables, et ce dans tous les répertoires personnels que vous pourriez créer.

Bien sûr, vous pouvez *détacher* ou *attacher* la librairie au moyen des instructions **1357 DETACH** et **1357 ATTACH**.

Quand la librairie est *détachée*, les noms des programmes qui la composent ne sont plus "connus" de la HP48 et peuvent être utilisés comme des noms quelconques.

Quand la librairie est *attachée*, tous les noms de programmes s'ajoutent aux instructions standards de la HP48 et peuvent être utilisés dans les mêmes conditions (en particulier, vous pouvez utiliser ces noms dans vos propres programmes).

La carte **PacMath** est protégée en écriture. Cela signifie que les programmes qui composent la librairie ne sont pas éditables ni modifiables.

N.B: La librairie de la carte **PacMath** s'attache automatiquement au répertoire **HOME**. Vous êtes libre de la détacher, puis de l'attacher à un sous-répertoire de **HOME** (les noms de la librairie ne sont alors connus que dans ce sous-répertoire et sa descendance).

◆ Exemple d'utilisation

Rappelons que la librairie doit être attachée (ou plutôt qu'elle ne doit pas être détachée par vous). Supposons que vous vouliez calculer le Pgcd (plus grand commun diviseur) des deux entiers 56518 et 7707. Le programme qui traite de ce problème s'appelle bien sûr **PGCD**. Vous placez les deux entiers en ligne de commande (sans valider cette ligne pour l'instant) ou sur la pile.

- ▶ Vous pouvez taper, en toutes lettres, **PGCD**, puis valider par **ENTER**. Le programme **PGCD** s'exécute aussitôt, laissant le résultat 2569 sur la pile.
- ▶ Vous pouvez passer dans le menu **LIBRARY**, pour y activer l'entrée **MATH**. Puis vous sélectionnez l'entrée **ARIT** (c'est la première). Vous actionnez alors la touche de menu **PGCD**, et le résultat est le même.
- ▶ L'entrée **ARIT** évoquée ci-dessus est un des noms de la librairie. Vous pouvez donc directement taper **ARIT** (en toutes lettres), pour vous trouver directement devant le menu débutant par **PGCD**, ce qui constitue un raccourci intéressant.

◆ Les entrées principales de la librairie

Contrairement à la structure de répertoires et de sous-répertoires que vous pouvez créer vous-même pour organiser vos données et vos programmes, les librairies sont des objets "plats".

La notion de sous-librairie n'a donc aucun sens: tous les programmes qui font partie de la librairie figurent au même niveau, dans les entrées successives d'un même menu.

Si le nombre d'entrées d'une librairie est important, le menu qui lui correspond peut être très long. La longueur du menu de la librairie **MATH** est ainsi de 66 pages de 6 entrées!

Dans de telles conditions, il peut s'avérer très long d'accéder (en "tournant les pages" du menu) à une entrée précise de la librairie (autant taper en toutes lettres le nom du programme cherché).

C'est pourquoi nous avons prévu (un peu comme les répertoires conseillé dans "**La maîtrise de la HP48**") 18 entrées principales, placées dans les 18 premières positions (les trois premières pages) du menu de la librairie.

Chacune de ces 18 entrées a pour effet d'activer un menu temporaire regroupant les différents programmes d'un sujet déterminé, facilitant ainsi l'accès à toutes les entrées de la librairie.

Voici quelles sont ces 18 entrées principales:

ARIT : arithmétique	R.C : nombres réels et complexes
POLY : polynômes	FRAC : fractions rationnelles
MATR : calcul matriciel	FORM : calcul formel
DL : développements limités	ANLY : analyse
GRPH : graphisme	GDIF : géométrie différentielle
GEOM : géométrie	TORS : torseurs
DATA : bases de données	LONG : entiers longs
PROBA : probabilités	STAT1 : statistiques simples
STAT2 : statistiques doubles	TOOLS : 70 utilitaires et un jeu

Chacune de ces 18 entrées est un nom de la librairie: on peut donc les utiliser directement depuis la ligne de commande (ou depuis un de vos programmes), pour sélectionner le menu temporaire qui lui correspond. Cela évite de passer par le menu **LIBRARY** et d'y sélectionner l'entrée **MATH**.

Le contenu de ces 18 entrées est présenté dans les pages qui vont suivre.

L'entrée **TOOLS** est un peu particulière puisqu'elle s'ouvre sur 7 entrées dont les six premières ouvrent à nouveau sur un menu temporaire (pour organiser les 70 utilitaires de l'entrée **TOOLS**).

Les six premières sections de l'entrée **TOOLS** sont:

PILE	: la pile	LIST	: les listes
CHAIN	: les chaînes	TABL	: les tableaux
VARI	: les variables	GEOP	: géométrie du plan

N.B: ces six noms ne sont pas des noms de la librairie.

Enfin la dernière entrée de **TOOLS** est le jeu **GRRR** (un petit casse tête. Il est petit mais costaud...).

♦ L'utilisation de l'aide

Chacun des menus temporaires de la librairie **MATH** se termine sur une entrée nommée **HELP!** dont le rôle, comme son nom l'indique, est de vous aider....

Le fait de sélectionner cette entrée lance la lecture d'un fichier fait de pages-écran successives. Chaque page-écran est consacrée à la description d'un programme particulier du menu temporaire en cours (dans si peu de place, les commentaires sont forcément assez brefs).

On peut ainsi facilement retrouver, si on l'a oublié, le rôle de chaque programme du répertoire en cours, ainsi que l'aspect de la pile opérationnelle avant et après l'exécution de ce programme.

On peut circuler dans cette aide d'une manière très intuitive, dans un sens ou dans l'autre (suivre les flèches). Les programmes y sont classés par ordre alphabétique. On sort de ce système d'aide par un appui sur la touche **OK**.

On peut accéder rapidement à une entrée donnée en lançant une lecture rapide (double flèche), interrompue par l'appui sur une touche quelconque.

♦ Conclusion

Il ne vous reste qu'à utiliser au mieux de vos intérêts la collection de programmes qui figure sur la carte **PacMath**.

Nous espérons, si vous êtes encore scolarisé, qu'elle sera une aide précieuse dans vos études. Elle pourra faciliter la tâche des enseignants pour inventer ou résoudre de nouveaux problèmes. Les autres y verront un moyen de reprendre goût aux mathématiques, et tous trouveront là une confirmation que la HP48 est vraiment une merveilleuse calculatrice.

L'ENTRÉE ARIT

L'entrée **ARIT** permet d'abord d'accéder aux programmes consacrés aux entiers et aux nombres rationnels:

- PGCD** : plus grand commun diviseur à deux entiers.
- PPCM** : plus petit commun multiple à deux entiers.
- SIMP** : simplification de fraction.
- FCTR** : décomposition d'un entier en produit de facteurs premiers.
- CALC** : évaluation d'expressions rationnelles.
- RACIN** : racine carrée d'un entier (ex: $495 = 3 \cdot \sqrt{55}$)
- ABCXY** : résolution de l'équation en nombres entiers $ax + by = c$.
- LPRM** : liste des diviseurs premiers d'un entier.
- LDIV** : liste de tous les diviseurs d'un entier positif.
- EULER** : indicateur d'Euler.
- MOEB** : fonction de Moebius.
- CYCLO** : calcul des polynômes cyclotomiques.

L'entrée **ARIT** contient également un certain nombre de programmes permettant de travailler sur les permutations d'un ensemble fini à n éléments (et donc sur le "groupe symétrique" S_n)

Si on note E_n l'ensemble fini $\{ 1, 2, 3, \dots, n \}$, une permutation σ de E_n sera représentée ici par la liste $\{ \sigma(1), \sigma(2), \sigma(3), \dots, \sigma(n) \}$.

Par exemple $\{ 5 \ 3 \ 7 \ 1 \ 6 \ 4 \ 2 \}$ est la permutation σ de E_7 définie par: $\sigma(1)=5$, $\sigma(2)=3$, $\sigma(3)=7$, $\sigma(4)=1$, $\sigma(5)=6$, $\sigma(6)=4$, et $\sigma(7)=2$, alors que cette permutation est notée traditionnellement:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 3 & 7 & 1 & 6 & 4 & 2 \end{bmatrix}$$

Les programmes en question sont:

- COMP** : composition de deux permutations.
- RECIP** : calcul de la permutation réciproque.
- INVRS** : nombre d'inversions d'une permutation, et liste de toutes ces inversions.
- P→CY** : décomposition d'une permutation en produits de cycles.
- CY→P** : opération inverse de **P→CY**.
- SIGN1** : signature d'une permutation (première version).
- SIGN2** : signature d'une permutation (deuxième version).
- EXP1** : puissances d'une permutation (première version).
- EXP2** : puissances d'une permutation (deuxième version).
- RNDP** : création d'une permutation aléatoire.
- TESTP** : teste si une liste représente bien une permutation.

N.B: les programmes **SIGN2** et **EXP2** sont adaptés aux permutations portant sur un grand nombre d'éléments.

L'ENTRÉE R.C

Cette entrée, consacrée aux nombres réels ou complexes, contient des programmes d'usage assez général, et dont le seul point commun est de ne pas pouvoir s'inscrire facilement sous telle ou telle entrée plus spécialisée.

Elle permet d'accéder aux programmes suivants:

- T→Q** : approximation, à la manière de l'instruction **→Q**, des éléments d'un tableau à coefficients réels ou complexes.
- TR→Q** : approximation rationnelle d'un tableau réel, avec factorisation du dénominateur commun.
- TRIG** : linéarisation des puissances de $\cos(t)$ et de $\sin(t)$, et opération inverse de la linéarisation.
- I→J** : écriture d'un nombre complexe sous la forme $a + bj$.
- PROD** : produits partiels d'un produit infini.
- RCNC** : racines n -ièmes d'un nombre réel ou complexe.
- LIPCO** : linéarisation d'un polynôme en $\cos(x)$.
- LIPSI** : linéarisation d'un polynôme en $\sin(x)$.
- LIπCS** : linéarisation de $\cos(x) \cdot \sin(x)$.
- CS↑** : expression d'une somme de $\cos(k \cdot x)$ en fonction de puissances de $\cos(x)$.
- LIPSH** : linéarisation d'un polynôme en $\sinh(x)$.
- FRCN** : calcul des fractions continues et des réduites d'un nombre réel donné.
- ITER** : permet d'effectuer des calculs récurrents (récurrence de pas quelconque)
- INTZ** : intégration le long d'un chemin du plan complexe.

L'ENTRÉE POLY

L'entrée **POLY** conduit aux programmes qui s'appliquent aux polynômes à une indéterminée (à coefficients réels ou complexes).

Les polynômes seront ici représentés par le vecteur de leurs composantes suivant les puissances décroissantes de l'indéterminée:

$P = 3X^7 - 2X^6 + X^3 - 2X$ est ainsi représenté par le vecteur $[3 \ -2 \ 0 \ 0 \ 1 \ 0 \ -2 \ 0]$.

C'est sous cette forme que devront être écrits les polynômes qui sont arguments d'un programme, et c'est sous cette forme qu'ils seront obtenus s'ils sont résultats d'un programme. Le programme **DIVPC** fait exception (il utilise le sens contraire).

L'entrée **POLY** est très complète. On y trouve:

- V→P** : transforme un polynôme (écrit sous forme de vecteur) en une notation algébrique traditionnelle.
- VALP** : valeur d'un polynôme en un point.
- ADDP** : addition de deux polynômes.
- PRODP** : produit de deux polynômes.
- DIV** : division euclidienne de deux polynômes.
- EXPO** : élévation d'un polynôme à une puissance entière.
- DIVPC** : division, suivant les puissances croissantes, de deux polynômes.
- RENV** : inverse l'ordre des coefficients d'un vecteur.
- PGDP** : pgcd de deux polynômes.
- PPMP** : ppcm de deux polynômes.
- PMAT** : calcul de polynômes de matrices.
- FSYM** : fonctions symétriques élémentaires de n nombres.
- Z→P** : trouver un polynôme à partir de ses zéros.
- TCHEB** : polynômes de Tchébitchev de première ou de seconde espèce.
- ABCUV** : recherche d'une solution de l'équation $AU + BV = C$, où A,B,C sont trois polynômes donnés, et où U,V sont deux polynômes inconnus.
- DEVPP** : développement d'un produit de puissances de polynômes.
- PPEX** : primitive, sous une forme symbolique, d'une expression du type $P(x)\text{Exp}(ax)$, où P est un polynôme.
- PPCS** : primitive, sous une forme symbolique, d'une expression du type $P(x)\text{Cos}(ax) + Q(x)\text{Sin}(ax)$, où P et Q sont deux polynômes.
- DEG2** : racines réelles ou complexes d'un polynôme de degré 2.
- DEG3** : racines réelles ou complexes d'un polynôme de degré 3.
- DEG4** : racines réelles ou complexes d'un polynôme de degré 4.
- BRST** : racines d'un polynôme de degré quelconque (méthode de Bairstow).
- DERIV** : dérivée d'un polynôme.
- PRIM** : primitive (s'annulant en zéro) d'un polynôme.
- INTP** : intégrale d'un polynôme sur un segment $[a,b]$.
- COMP** : composition de deux polynômes.
- TRNS** : translation d'un polynôme.

L'ENTRÉE **FRAC**

L'entrée **FRAC** conduit à un certain nombre de programmes s'appliquant aux fractions rationnelles, c'est-à-dire aux fonctions s'écrivant comme le quotient $R=A/B$ de deux fonctions polynômes A et B.

Voici quels sont ces programmes:

- ADDF** : addition de deux fractions rationnelles.
- PRODF** : produit de deux fractions rationnelles.
- SMPF** : simplification d'une fraction rationnelle.
- VALF** : valeur d'une fraction rationnelle en un point.
- V→F** : écriture algébrique d'une fraction rationnelle.
- F→Q** : conversion en nombres rationnels des coefficients d'une fraction rationnelle.
- EXPOF** : puissances d'une fraction rationnelle.
- DERVF** : dérivée d'une fraction rationnelle.

et surtout:

- DECPF** : décomposition d'une fraction rationnelle en éléments simples.

Dans la plupart des cas ci-dessus, une fraction rationnelle est représentée sur la pile par la superposition de deux vecteurs représentant l'un le numérateur, l'autre le dénominateur.

Par exemple, la fraction rationnelle:

$$R = \frac{X^3 - 2X^2 + 5X - 1}{11X^2 - 15X + 21}$$

est représentée sur la pile par:

2:	[1 -2 5 -1]
1:	[11 -15 21]

L'ENTRÉE MATR

L'entrée **MATR** contient les programmes suivants, consacrés aux calculs portant sur les matrices ou les systèmes linéaires.

- CRTAB** : crée un tableau dont le terme général répond à une formule donnée.
- CALCC** : effectue toutes sortes de calculs sur les colonnes d'une matrice.
- CALCL** : effectue toutes sortes de calculs sur les lignes d'une matrice.
- PIVOT** : décrit la méthode du pivot de Gauss appliquée à une matrice.
- INVN** : donne l'inverse d'une matrice carrée A à coeffs entiers, en donnant son déterminant d et la matrice (à coeffs entiers) $B = d * A^{-1}$.
- INVPR** : inverse une matrice carrée en haute précision.
- DIVPR** : divise un tableau, par une matrice carrée, en haute précision.
- PUISS** : calcule les puissances quelconques d'une matrice carrée.
- POLM** : calcule le polynôme minimal d'une matrice carrée.
- EQRL** : donne les éventuelles relations linéaires existant entre n vecteurs, ou les équations de l'espace vectoriel engendré par ces n vecteurs.
- ADDID** : borde à droite une matrice A (à n lignes) par la matrice identité d'ordre n (utile dans les programmes **POLM**, **PIVOT** et **EQRL**).
- SYST** : permet de donner l'expression symbolique de la solution générale d'un système de n équations à p inconnues. Le cas où le système possède une infinité de solutions est ici particulièrement intéressant.
- RANG** : calcule le rang d'une matrice quelconque (méthode du pivot de Gauss).
- VECTP** : donne la ou les équations du sous-espace propre d'une matrice carrée pour une valeur propre donnée.
- DEFL** : permet d'approcher les valeurs propres et vecteurs propres dans le cas de matrices de dimension supérieure à 4.
- VP234** : donne les valeurs propres d'une matrice carrée d'ordre 2,3, ou 4.
- PCAR** : donne le polynôme caractéristique d'une matrice carrée.
- TR** : calcule la trace d'une matrice carrée.
- CB** : effectue le changement de matrice d'une application linéaire (connaissant la matrice initiale et la matrice de changement de base).
- EHC** : échange deux colonnes.
- PUTC** : place une colonne dans une matrice.
- GETC** : extrait une colonne quelconque d'une matrice.
- ECHL** : échange deux lignes.
- PUTL** : place une ligne dans une matrice.
- GETL** : extrait une ligne quelconque d'une matrice.
- ALU** : décompose une matrice carrée en le produit d'une matrice triangulaire inférieure à diagonale unité et d'une matrice triangulaire supérieure.

L'ENTRÉE **ANLY**

Sous cette entrée, on trouvera un certain nombre de programmes traitant de problèmes classiques.

Dans d'autres entrées, de nombreux programmes utilisent des techniques d'analyse. On ne trouvera ici que ceux qui ne trouvent pas leur place de façon évidente dans une entrée plus spécialisée.

Plus précisément, l'entrée **ANLY** contient:

- CFIX** : Recherche de points fixes dans $y=f(x)$.
- EXTRE** : Recherche d'extremums pour la fonction $Y=F(X)$.
- INFLX** : Recherche de points d'inflexion pour la courbe $Y=F(X)$.
- TGTE** : Tangente, en un point donné, à une courbe $Y=F(X)$.
- SXY** : Résolution approchée, par la méthode de Newton, d'un système de deux équations à deux inconnues $F(X,Y)=0$, $G(X,Y)=0$.
- SXYZ** : Résolution approchée, par la méthode de Newton, d'un système de 3 équations à 3 inconnues $F(X,Y,Z)=0$, $G(X,Y,Z)=0$, $H(X,Y,Z)=0$.
- PRIM** : Recherche de la primitive symbolique d'une fonction $F(X)$, s'annulant en un point donné.
- INT23** : Calcul approché d'intégrales doubles ou triples.
- ROOTC** : Résolution d'équations d'une variable complexe.
- FOUR** : Calcul des sommes partielles de la série de Fourier d'une fonction périodique donnée.
- RK4** : Résolution approchée, par la méthode de Runge-Kutta d'ordre 4, de l'équation différentielle $Y'=F(X,Y)$.
- APMC** : Approximations au sens des moindres carrés discrets.
- LAGR** : Calcul du polynôme interpolateur de Lagrange.
- TRACE** : En liaison avec les programmes **APMC** et **LAGR**, tracé de la famille de points utilisée, puis de la courbe obtenue.
- SPPEX** : Solution particulière de $ay''+by'+cy=P(x)\exp(\alpha x)$.
- SPPC** : Solution particulière de $ay''+by'+cy=P(x)\cos(\alpha x)$.
- SPPS** : Solution particulière de $ay''+by'+cy=P(x)\sin(\alpha x)$.
- SGH1** : Solution générale de $a(x)\cdot y'+b(x)\cdot y=0$.
- SGC1** : Solution générale de $a(x)\cdot y'+b(x)\cdot y=c(x)$, sous réserve que la primitive soit connue de la HP48, et à partir de la liste $\{ a(X), b(X), c(X) \}$ au niveau 1 de la pile.
- SGH2** : Solution générale de $a\cdot y''+b\cdot y'+c\cdot y=0$.
- SPCI1** : Solution particulière de $a(x)\cdot y'+b(x)\cdot y=c(x)$ avec conditions initiales.
- SPCI2** : Solution particulière de $a\cdot y''+b\cdot y'+c\cdot y=d(x)$ avec conditions initiales.

L'ENTRÉE DL

L'entrée **DL** contient les programmes permettant de créer et de composer des développements limités, à tout ordre, d'une manière très rapide et précise. L'idée essentielle ici est qu'un développement limité comme $f(x)=a+bx+cx^2+\dots+dx^n+o(x^n)$ peut être représenté par le vecteur [a b c d].

Les programmes de l'entrée **DL** prennent des développements limités comme arguments et ils en renvoient comme résultats, en respectant ce format (sauf **VISU**, qui fournit une expression algébrique).

En tant qu'argument, un développement limité devra donc être donné sous la forme d'un vecteur. Les programmes de l'entrée **DL** interpréteront cette longueur comme donnant l'ordre exact du développement limité.

Par exemple, si vous mettez [1 3 -4] sur la pile, cela signifie que vous y mettez le développement limité $1+3*X-4*X^2+o(X^2)$. Si vous voulez calculer le développement limité de $\sin(1+3*X-4*X^2)$ à l'ordre 6, il vous faudra mettre [1 3 -4 0 0 0] sur la pile (qui représente le développement limité $1+3*X-4*X^2+o(X^6)$).

Réciproquement, lorsque un programme de l'entrée **DL** donne un développement limité comme résultat, il le calcule avec l'ordre le plus grand possible (compte tenu des données, bien sûr), et tous les coefficients obtenus sont exacts (aux erreurs d'arrondi près, qui sont très souvent négligeables).

Tous les D.L. qui seront considérés ici sont des D.L. à l'origine (c'est en tout cas obligatoire pour le D.L. de g lorsqu'il s'agira de déterminer le D.L. de gof).

L'entrée **DL** contient les programmes suivants:

- ΣDL** : somme de deux développements limités.
- πDL** : produit de deux développements limités.
- QDL** : quotient de deux développements limités.
- CPDL** : composé de deux développements limités.
- INVDL** : inverse d'un développement limité.
- DLN** : puissances entières d'un développement limité.
- VISU** : passe de la forme "vecteur" d'un développement limité à sa forme algébrique traditionnelle.
- X→XN** : remplacement de X par X^n dans un développement limité.
- X→AX** : remplacement de X par $a*X$ dans un développement limité.
- DERDL** : dérivation d'un développement limité.
- INTDL** : intégration d'un développement limité.
- DIM** : ordre et valuation d'un développement limité donné.

.../...

Pour de nombreuses fonctions usuelles, il est possible de calculer le développement limité (à tout ordre), et le composé, par une telle fonction, d'un développement limité donné. Voici les programmes correspondants:

LG	: DL de $\text{LN}(X)$ et logarithme népérien d'un DL.
EX	: DL de $\text{EXP}(X)$ et exponentielle (base e) d'un DL.
AX	: DL de a^X et exponentielle (base $a > 0$) d'un DL
XA	: DL de X^a et puissances d'un DL.
SN	: DL de $\text{SIN}(X)$ et sinus d'un DL.
CS	: DL de $\text{COS}(X)$ et cosinus d'un DL.
TG	: DL de $\text{TAN}(X)$ et tangente d'un DL.
SH	: DL de $\text{SINH}(X)$ et sinus hyperbolique d'un DL.
CH	: DL de $\text{COSH}(X)$ et cosinus hyperbolique d'un DL.
TH	: DL de $\text{TH}(X)$ et tangente hyperbolique d'un DL.
ASNS	: DL de $\text{ASIN}(X)$ et arc sinus d'un DL.
ACS	: DL de $\text{ACOS}(X)$ et arc cosinus d'un DL.
ATG	: DL de $\text{ATAN}(X)$ et arc tangente d'un DL.
ASH	: DL de $\text{ASINH}(X)$ et arc sinus hyperbolique d'un DL.
ATH	: DL de $\text{ATANH}(X)$ et arc tangente hyperbolique d'un DL.

L'ENTRÉE GEOM

L'entrée **GEOM** contient quelques programmes de géométrie affine ou euclidienne, qui se répartissent en deux groupes:

Dans un premier temps, il s'agit de trouver l'équation de tel ou tel ensemble de points (droite, plan, cercle, sphère) quand on en connaît les éléments essentiels, ou de calculer des distances ou des écarts angulaires.

Les programmes correspondants sont:

- DRTE** : équation d'une droite dont on connaît:
 - ▶ deux points, ou
 - ▶ un point et un vecteur directeur.
- PLAN** : équation d'un plan dont on connaît:
 - ▶ trois points, ou
 - ▶ deux points et un vecteur directeur, ou
 - ▶ un point et deux vecteurs directeurs.
- CERC** : équation d'un cercle ou d'une sphère dont on connaît:
 - ▶ le centre et le rayon, ou
 - ▶ deux points diamétralement opposés.
- DIST** : calcul de la distance entre:
 - ▶ un point et une droite (en dimension 2 ou 3), ou
 - ▶ un point et un plan, ou
 - ▶ deux droites (dans l'espace).
- ANGL** : calcul de l'écart angulaire entre:
 - ▶ deux vecteurs, ou
 - ▶ deux droites, ou
 - ▶ deux plans.

Le deuxième groupe de programmes de l'entrée **GEOM** constitue une "bibliothèque" permettant de travailler sur des applications affines ou linéaires, en dimension 2 ou 3.

Une application affine f , en dimension 2, et si l'espace ambiant est rapporté à un repère Oxy , est définie par un système du type (où x', y' représentent les coordonnées de l'image M' du point M de coordonnées x, y):

$$\begin{cases} x' = a_1x + b_1y + c_1 \\ y' = a_2x + b_2y + c_2 \end{cases}$$

La matrice de l'application linéaire associée est:

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}$$

et le vecteur $[c_1 \ c_2]$ représente les coordonnées de l'image, par l'application f , de l'origine du repère.

Une telle application affine f est représentée dans l'entrée **GEOM** par la matrice:

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix}$$

En choisissant $c_1=c_2=0$, on peut ainsi traiter les applications affines qui s'annulent à l'origine, ou, ce qui revient au même, les applications linéaires

De même les applications affines en dimension 3 sont représentées ici par une matrice réelle à 3 lignes et quatre colonnes. Ainsi la matrice:

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

représente l'application f définie par:

$$\begin{cases} x' = x + 3y - 4z + 1 \\ y' = -2x + 3z + 2 \\ z' = 5x + y - 3z + 4 \end{cases}$$

On utilise donc l'identification matricielle d'une application affine, permise par le choix d'un repère de l'espace. Ce repère reste implicite, et ne doit être supposé orthonormé que s'il s'agit de traiter d'applications affines où la notion d'orthogonalité intervient (projections orthogonales, déplacements et antidéplacements).

Les programmes traitant ici des applications affines sont:

- PFIX** : points fixes d'une application affine.
- COMP** : composée de deux applications affines, ou image d'un point par une application affine.
- PROJ** : projection sur une droite ou sur un plan parallèlement à une direction vectorielle donnée.
- PROJO** : projection orthogonale sur un plan ou une droite.
- SYM** : symétrie par rapport à une droite ou par rapport à un plan parallèlement à une direction vectorielle donnée.
- SYMO** : symétrie orthogonale par rapport à un plan ou une droite.
- ROTAT** : rotation d'angle donné autour d'un point ou d'une droite.
- INVAF** : inversion d'une application affine bijective.
- IDAFF** : identifier une application affine. Le programme **IDAFF** reconnaît les projections, les symétries, et toutes les isométries (déplacements et antidéplacements).

L' ENTRÉE TORS

L'entrée **TORS** ouvre une série d'utilitaires s'appliquant aux torseurs. Rappelons qu'un torseur **T** est un champ de vecteurs en dimension 3 (à tout point de l'espace on associe un vecteur) satisfaisant à la propriété suivante: Il existe un point **A** et un vecteur **R** tels que, pour tout point **M**: $T(M) = T(A) + R \wedge AM$ (1)

On a alors pour tous points **M, N**: $T(M) = T(N) + R \wedge NM$.

Rappelons les définitions et propriétés principales, pour un torseur **T**:

- Pour tout point **M**, le vecteur **T(M)** est le **moment** du torseur **T** au point **M**.
- Pour tous points **M** et **N**: $T(M).MN = T(N).MN$. On nomme cette propriété **équiprojectivité**. Elle caractérise les torseurs parmi les champs de vecteurs.
- Le vecteur **R** est unique. On l'appelle **résultante** du torseur.
- La valeur du produit scalaire $T(M).R$ ne dépend pas de **M**. On l'appelle **invariant scalaire** du torseur **T**.
- Soient 2 torseurs **T** et **T'** de résultantes **R** et **R'**, la quantité $T(M).R' + T'(M).R$ ne dépend pas de **M**. C'est le **comoment** des torseurs **T** et **T'**.
- Les **couples** sont les torseurs constants, ou encore les torseurs de résultante nulle.
- On appelle **glisseur** un couple dont le moment est nul en au moins un point **A**. Si on note **R** la résultante, alors pour tout point **M**: $T(M) = R \wedge AM$.
- Soit **T** un torseur non constant ($\Leftrightarrow$ ce n'est pas un couple $\Leftrightarrow$ sa résultante **R** n'est pas nulle). L'ensemble des points **M** tels que $T(M)$ et **R** soient proportionnels est une droite, de vecteur directeur **R**, appelée **axe central** du torseur **T**.
- Le moment du torseur en tout point de son axe central est constant. On l'appelle **invariant vectoriel** de **T**.

La définition (1) montre qu'un torseur **T** est parfaitement déterminé par la donnée d'un point **A**, du moment **T(A)**, et de la résultante **R**. On dira dans cette situation que **A** est le **point de base** du torseur. Dans l'entrée **GEOM** le torseur **T** sera défini par la liste { **A** **T(A)** **R** }, où **A**, **T(A)**, et **R** sont des vecteurs réels à trois coordonnées, ayant les significations évoquées ci-dessus.

Par exemple { [1 0 -1] [3 1 5] [-4 2 -3] } désigne le torseur dont le moment au point **A(1,0,-1)** est le vecteur **T(A)=(3,1,5)**, et dont la résultante est le vecteur **R=(-4,2,-3)**.

N.B: avec ces notations, **A** peut être omis s'il est l'origine [0,0,0]; de même si le vecteur **R** est nul, il peut être omis (de même que **A**, qui n'a alors aucune utilité).

Voici maintenant les programmes de l'entrée **TORS**:

- ΣTRS** : somme de deux torseurs.
- NEWP** : changement de point de base pour un torseur.
- MOMT** : moment d'un torseur en un point.
- INVS** : invariant scalaire d'un torseur.
- INVV** : invariant vectoriel d'un torseur.
- AXE** : axe central d'un torseur.
- MTAX** : moment d'un torseur par rapport à un axe.
- SF→T** : réduction d'un système de forces à un torseur unique.
- πTSC** : produit d'un torseur par un scalaire.
- COMT** : comoment de deux torseurs.

L'ENTRÉE GDIF

L'entrée **GDIF** contient un certain nombre de programmes de géométrie différentielle. Il s'agit de la résolution de problèmes géométriques où interviennent des calculs de dérivées et de primitives.

L'entrée **GDIF** permet d'accéder aux programmes suivants:

- RCTP** : rectification (calcul de la longueur) d'une courbe plane (donnée par une équation $Y=F(X)$, ou en polaires, ou par $X=X(T)$, $Y=Y(T)$).
- RCTG** : rectification (calcul de la longueur) d'une courbe gauche (en dimension 3), donnée sous forme paramétrée.
- CURV** : calcul d'une intégrale curviligne, dans le plan ou dans l'espace, le long d'un arc paramétré.
- AIRE** : aire d'un domaine plan limité par une courbe fermée, cette dernière étant connue par une représentation sous forme paramétrique (polaires ou cartésiennes).
- CRBRE** : calcul du rayon de courbure et du centre de courbure en un point quelconque d'une courbe plane.
- DIVRG** : calcul de la divergence d'un champ de vecteurs.
- ROTA** : calcul du rotationnel d'un champ de vecteurs.
- GRADI** : gradient d'une fonction de plusieurs variables.
- DELTA** : laplacien d'une fonction de plusieurs variables.
- DIFF** : calcul de la différentielle d'une fonction de plusieurs variables.

L'ENTRÉE GRPH

La HP48 possède un écran graphique de 131*64 points. Un grand nombre d'instructions se rapportent à la gestion de cet écran; on les trouve regroupées dans les menus **PLOT**, ou **PGR DSPL**, ou dans l'environnement **GRAPH**.

Dans le domaine du graphisme, et par rapport à la HP28S, la HP48 marque un progrès considérable. Néanmoins les quelques programmes qui suivent peuvent s'avérer utiles:

- TRACE** : module de tracé utilisé par **PAR**, **POL** et **POLP**.
- PAR** : tracé d'une courbe paramétrée ($X=X(T)$, $Y=Y(T)$).
- POL** : tracé d'une courbe en polaires ($R_o = R_o(\Theta)$).
- POLP** : tracé d'un courbe en polaires paramétrées ($R_o = R_o(T)$, $\Theta = \Theta(T)$).
- ENV** : tracé de l'enveloppe d'une famille de droites.
- MTCL** : utilisation de la méthode de Monte-Carlo pour visualiser des courbes implicites $F(X,Y)=0$.
- FAMT** : tracé d'une famille de courbes dépendant de T .
- ANIM** : production d'images-écran successives.

Les programmes **PAR**, **POL**, **POLP**, **ENV** ont pour intérêt de tracer des courbes tout en mémorisant les points utilisés dans la matrice **ΣDAT**, ce que ne permet pas l'instruction **DRAW** et ce qui est utile si vous devez tracer ces courbes sur feuille.

Le programme **MTCL** est peut-être le moins utile ici. Il est en effet possible de visualiser des courbes $F(X,Y)=0$ en traçant l'expression ' $F(X,Y)>0$ ' (ou ' $F(X,Y)<0$ ') avec l'instruction **DRAW** (et après avoir défini **TRUTH** comme type de tracé).

Néanmoins **MTCL** peut s'avérer plus rapide, car il permet de tester une partie seulement des points de l'écran (et non pas la totalité comme avec le type de tracé **TRUTH**).

L'ENTRÉE LONG

L'entrée **LONG** contient un ensemble de programmes permettant de manipuler des 'entiers longs'. Sous ce terme, on entend ici des entiers positifs ou nuls dont le nombre de chiffres peut être supérieur aux douze chiffres significatifs que le calculateur peut garder en mémoire et afficher.

Ici, un entier long **N** est représenté par un vecteur dont les composantes forment la décomposition de **N** en base 10^5 . Ces composantes doivent donc être des entiers compris entre 0 et 99999.

Ainsi: $N = 165088696783290882115695$ s'écrit [1650 88696 78329 8821 15695].
Inversement [87 132 6 78999 1] représente $N = 8700132000067899900001$.
Plus simplement 1 s'écrit [1], et [1 0] est égal à 10^5 .

L'entrée **LONG** contient les programmes permettant d'effectuer les opérations courantes sur les entiers longs:

ADDL : addition de deux entiers longs.
PRODL : produit de deux entiers longs.
EXPOL : puissances entières d'un entier long.
DIVL : division de 2 entiers longs (calcul du quotient et du reste entiers).
PGDL : pgcd de deux entiers longs.
PPML : ppcm de deux entiers longs.
FACTL : obtention, sous forme d'un entier long, de la factorielle d'un entier naturel.

On trouve également dans l'entrée **LONG**, les routines qui permettent de passer de telle à telle forme de l'écriture d'un entier long:

L→CH : écriture d'un entier long sous la forme d'une chaîne de caractères.
CH→L : traduction d'une chaîne de caractères sous forme d'un entier long.
R→L : passage de la forme "réel" à la forme entier long.
L→R : passage de la forme entier long à la forme "réel".

L'ENTRÉE PROBA

L'entrée **PROBA** contient un certain nombre de programmes permettant de connaître, sans avoir recours à des tables numériques (et donc sans avoir à y faire d'interpolations) les lois de probabilités et les fonctions de répartition des lois usuelles (discrètes ou continues).

Puisque les dénombrements interviennent fréquemment dans les calculs de probabilités, l'entrée **PROBA** contient quelques petits programmes s'appliquant aux dénombrements.

On trouvera donc sous l'entrée **PROBA**:

- AJUST** : Ajustement par une loi normale $N(m, \sigma)$.
- NRM** : fonction de répartition de la loi normale.
- NCR** : fonction de répartition de la loi normale centrée réduite.
- NRM→** : Inverse de la fonction de répartition de la loi normale.
- NCR→** : Inverse de la fonction de répartition de la loi normale centrée réduite.
- ANP** : nombre d'arrangements de p éléments parmi n .
- CNP** : nombre de combinaisons sans répétitions, de p éléments pris parmi n .
- GANP** : nombre de combinaisons avec répétitions, de p éléments pris parmi n .
- BINO** : liste des coefficients du binôme.
- BNP** : loi binômiale.
- FBNP** : fonction de répartition de la loi binômiale.
- HNAP** : loi hypergéométrique.
- FHNAP** : fonction de répartition de la loi hypergéométrique.
- POIS** : loi de Poisson.
- FPOIS** : fonction de répartition de la loi de Poisson.
- GEO** : loi géométrique.
- FGEO** : fonction de répartition de la loi géométrique.
- PRP** : loi de Pascal.
- FPRP** : fonction de répartition de la loi de Pascal.
- JRP** : loi binômiale négative.
- FJRP** : fonction de répartition de la loi binômiale négative.
- FEXP** : fonction de répartition de la loi exponentielle.

L'ENTRÉE STAT1

L'entrée **STAT1** contient un ensemble de programmes permettant d'effectuer l'étude d'une statistique simple, discrète ou donnée par classes, et connue par les effectifs (ou fréquences) correspondant à chaque valeur ou à chaque classe du caractère X qu'il s'agit d'étudier.

Dans le cas d'une statistique discrète (X_i, N_i) , où les X_i sont les valeurs du caractère X , et les N_i sont les effectifs (ou fréquences), on place les couples (X_i, N_i) dans la matrice **SDAT** (la première colonne pour les X_i , la seconde pour les N_i).

Dans le cas d'une statistique donnée par classes $([A_i, A_{i+1}[, N_i)$, on place dans une variable nommée **DATA** une liste formée elle-même de deux sous-listes: la première pour les extrémités de classes, la deuxième pour les effectifs (ou fréquences).

Par exemple la statistique:

x_i	$[0, 2[$	$[2, 6[$	$[6, 8[$	$[8, 14[$
N_i	22	25	15	10

est représentée dans la variable **DATA** par la liste $\{\{ 0 \ 2 \ 6 \ 8 \ 14 \} \{ 22 \ 25 \ 15 \ 10 \} \}$

Une variable **SDT1** est utilisée par certains programmes de l'entrée **STAT1** pour sauvegarder **SDAT** ou pour présenter les calculs intermédiaires amenant à un résultat donné (le calcul du coefficient de Gini par exemple).

Les programmes de l'entrée **STAT1** sont:

- CUM** : création d'un tableau de cumuls.
- CUM→** : retour d'un tableau de cumuls au tableau initial.
- G→D** : transformation d'une statistique par classes en stat. discrète.
- C.COL** : calcul d'une colonne dépendante des 2 colonnes de **SDAT**.
- MODΣ** : modification d'une ou des deux colonnes de **SDAT**.
- COL.N** : affiche une colonne quelconque de **SDT1** (pour **GINI & MDLE**).
- MK** : calcul du moment d'ordre k .
- MCK** : calcul du moment centré d'ordre k .
- MCRK** : calcul du moment centré réduit d'ordre k .
- MGEO** : calcul de la moyenne géométrique.
- MHAR** : calcul de la moyenne harmonique.
- ECAR** : calcul de l'écart arithmétique moyen.
- MDLE** : calcul de la médiale (tableau de calcul dans **SDT1**).
- QTLE** : calcul des quantiles.
- HIST** : tracé de l'histogramme.
- PFC** : tracé du polygone des fréquences cumulées.
- GINI** : calcul de l'indice de Gini (tableau de calcul dans **SDT1**).
- LRTZ** : tracé de la courbe de Lorentz (ou de concentration).
- YULE** : coefficients de Yule, Kelley, et Pearson.

L'ENTRÉE STAT2

L'entrée **STAT2** contient un certain nombre de programmes permettant d'étudier les statistiques doubles discrètes (X,Y) , chaque valeur (X_i,Y_j) étant connue par son effectif ou par sa fréquence. On notera N_{ij} cet effectif (ou cette fréquence).

Les différentes valeurs de X et Y (première et deuxième statistiques marginales du couple (X,Y)) doivent être placées, sous forme d'un vecteur, dans une variable nommée respectivement '**X**' et '**Y**'. Le tableau des effectifs N_{ij} doit être placé dans une variable '**SEFF**', sous forme de matrice, avec les conventions suivantes: le coefficient N_{ij} doit être placé en i -ème ligne, j -ème colonne.

Remarque:

Il est courant que l'on ait à étudier une statistique double (X,Y) consistant en un ensemble de n points (X_i,Y_i) tous d'effectif 1. Un certain nombre de programmes de l'entrée **STAT2** s'appliquent bien à ce cas simple. Néanmoins, les programmes utilisant '**SEFF**' sont utilisables à condition de placer dans '**SEFF**' la matrice identité d'ordre n .

Les programmes de l'entrée **STAT2** sont les suivants:

- MULT** : produit terme à terme des coefficients de 2 vecteurs (utile pour rechercher une variance ou une covariance)
- TOTAL** : somme des coefficients d'un vecteur ou d'une matrice.
- MODT** : modification, selon une certaine formule, des termes d'un tableau (matrice ou vecteur). **MODT** est utile dans la recherche d'ajustements au moyen de fonctions puissances ou exponentielles.
- EFFX** : effectifs de la première statistique marginale.
- EFFY** : effectifs de la deuxième statistique marginale.
- MX** : moyenne arithmétique de la première statistique marginale.
- MY** : moyenne arithmétique de la deuxième statistique marginale.
- VX** : variance de la première statistique marginale.
- VY** : variance de la deuxième statistique marginale.
- CV** : calcul de la covariance du couple (X,Y) .
- COR** : coefficient de corrélation linéaire du couple (X,Y) .
- DY→X** : équation de la droite de régression de Y en X .
- DX→Y** : équation de la droite de régression de X en Y .
- KX↑A** : ajustement par une fonction puissance $Y=k*X^a$.
- KA↑X** : ajustement par une fonction exponentielle $Y=k*a^X$.
- NXY** : création de la matrice des $N_{ij}*X_i*Y_j$, très utile pour le calcul de la covariance.

Les programmes **TOTAL**, **MULT**, **MODT**, sont destinés à permettre à l'utilisateur de mener les calculs "pas à pas".

Ils peuvent s'avérer utiles dans d'autres entrées que **STAT2**.

L'ENTRÉE Data

La capacité mémoire importante de la HP48 permet d'envisager la gestion de bases de données. Bien sûr, il ne sera pas possible de rentrer un nombre grandiose d'enregistrements. Mais, pour une calculatrice de poche, le résultat obtenu est très impressionnant.

Une base de données est un ensemble d'enregistrements (ayant en général certains rapports entre eux ...) sur lequel on peut effectuer l'une des opérations suivantes:

- ▶ ajout d'un enregistrement.
- ▶ modification d'un enregistrement.
- ▶ suppression d'un enregistrement.
- ▶ lecture de l'ensemble des enregistrements.
- ▶ tri de l'ensemble des enregistrements.
- ▶ sélection d'enregistrements.

Dans la mémoire de la HP48, une base de données se présentera sous la forme d'une liste ayant la forme suivante: { C_1 C_2 C_3 ... C_n }, où n est un entier représentant le nombre d'éléments de la liste et où $C_1, C_2, \dots, C_n$ sont les différents enregistrements.

Un enregistrement est une chaîne de caractères. Pour être totalement visible à l'écran, cette chaîne doit à priori être formée d'au plus 7 lignes, chaque ligne ayant au plus 22 caractères (On peut insérer des sauts de lignes par **NEWLINE**).

Néanmoins on verra qu'il est possible par l'option **VISIT** du programme **LIRE** d'utiliser des chaînes plus larges.

Une base de données sera placée dans une variable, et l'accès à la base de données se fera en utilisant ce nom de variable.

Les programmes de l'entrée **Data** sont les suivants:

- LIRE** : lecture de la base de données, avec possibilité de modifier ou d'effacer des enregistrements.
- AJOUT** : ajouter un enregistrement.
- FIND** : recherche du premier enregistrement contenant une sous-chaîne donnée (et lecture de la base de données à partir de celui-ci).
- SELEC** : sélection des enregistrements contenant une sous-chaîne donnée, puis lecture de la base de données créée avec les enregistrements obtenus.
- TRIBD** : tri de la base de données.

L'ENTRÉE FORM

L'entrée **FORM** contient de nombreux programmes permettant d'effectuer des calculs dits "formels", c'est-à-dire sur des objets qui sont des nombres, des noms ou des expressions. Les calculs effectués ici sont en général de nature polynômiale ou matricielle. Il a fallu avoir recours aux listes pour construire et traiter des matrices ou des polynômes formels.

Dans les lignes qui suivent, on utilisera la terminologie suivante:

- ▶ **Objet formel**: un nombre (réel ou complexe), un nom, ou une expression algébrique.
- ▶ **Vecteur formel**: une liste non vide d'objets formels.
- ▶ **Matrice formelle**: une liste non vide dont les éléments sont des vecteurs formels de même longueur (chacune de ces listes est une ligne de la matrice, et le nombre d'éléments de chacune de ces listes est le nombre de colonnes de la matrice).
- ▶ **Tableau formel**: une matrice formelle ou un vecteur formel.
- ▶ **Polynôme formel**: une liste non vide d'objets formels représentant les coefficients successifs d'un polynôme ordonné suivant ses puissances décroissantes.

La liste { '1+A' 'A*B' '5+√3' B } pourra ainsi désigner:

- ▶ le vecteur (1+A, AB, 5+√3, B) ou
- ▶ le polynôme $(1+A)X^3 + (AB)X^2 + (5+\sqrt{3})X + B$.

Le problème majeur que l'on peut rencontrer quand on effectue des calculs formels sur la HP48 est d'avoir à traiter des expressions qui deviennent vite volumineuses.

Pour prendre un exemple, l'opérateur + (addition), appliqué aux arguments 'A-1' et '1-A' donne le résultat 'A-1+(1-A)' (et non pas 0). Si ce résultat est réutilisé à l'intérieur du programme en cours d'exécution, on conçoit les complications qui vont en résulter.

Le choix qui a été effectué ici est de ne pas intervenir en cours de programme pour tenter de simplifier les résultats intermédiaires, mais plutôt de se munir d'outils permettant de traiter efficacement le résultat final du calcul (à l'aide par exemple des programmes **EDITF**, **ALLF**, **DVLP1**, **DVLP2** ou **FCTP**).

Il aurait été possible d'inclure dans certains programmes des séquences simplificatrices (le couple d'instructions **EXPAN COLCT** par exemple), mais cela aurait eu plus de défauts que d'avantages. Parmi ces défauts, citons:

Certaines expressions sont simplifiables de telle manière et pas de telle autre. Toute tentative de simplification "en aveugle" risque ainsi de ne pas être efficace.

Les instructions **COLCT**, **EXPAN**, disséminées en certains endroits d'un programme:

- ▶ en ralentissent l'exécution.
- ▶ en augmentent la taille.

.../...

Certains algorithmes qui s'appliquent bien à des tableaux numériques ne peuvent être repris ici. C'est le cas par exemple de la méthode du pivot pour calculer le déterminant ou l'inverse d'une matrice. Appliquée à un tableau contenant des expressions, la méthode du pivot conduit rapidement à des coefficients beaucoup trop volumineux et difficilement simplifiables.

Voici donc les programmes de l'entrée **FORM**:

- GENF** : générer un tableau formel.
- EDITF** : visiter et éditer un tableau formel.
- DVLP1** : développer une expression polynômiale d'une variable.
- DVLP2** : comme **DVLP1** mais avec plusieurs variables.
- ALLF** : appliquer une même procédure à tous les éléments d'un tableau formel.
- INTFR** : intégration symbolique de fractions rationnelles.
- FQ→C** : décomposition d'une forme quadratique en carrés.
- VALPF** : évaluation d'un polynôme formel en un point.
- DG2F** : résolution symbolique de $aX^2 + bX + c = 0$.
- ΣPF** : somme de deux polynômes formels.
- ∏PF** : produit de deux polynômes formels.
- DIVPF** : division euclidienne de deux polynômes formels.
- DVPCF** : comme **DIVPF** mais suivant les puissances croissantes.
- DERPF** : dérivation d'un polynôme formel.
- PRIPF** : primitivation d'un polynôme formel.
- INTPF** : intégration d'un polynôme formel sur un segment.
- FCTP** : factoriser une expression polynômiale.
- ΣMF** : somme de deux matrices formelles.
- ∏MF** : produit d'une matrice formelle par un tableau formel.
- DETF** : déterminant d'une matrice carrée formelle.
- INVf** : inverse d'une matrice carrée formelle.
- SYSTF** : résolution du système $AX=B$, où A est une matrice carrée formelle et où B est un tableau formel.
- PART** : partitions d'un entier n en somme de k entiers.
- Σ↑N** : développement d'une puissance n -ème d'une somme.
- ASSOC** : associer entre eux les éléments d'un tableau formel au moyen d'un même opérateur binaire.
- RPN** : transformer une expression en le programme RPN correspondant.
- TRI** : tri des éléments d'une liste selon un critère spécifié.
- CONF** : créer un tableau formel constant.
- GETF** : extraire un objet formel d'un tableau formel.
- PUTF** : placer un objet formel dans un tableau formel.
- SIZEF** : taille d'un tableau formel.

Trois programmes permettent de calculer des transformées de Laplace:

- LAPL** : transformée de Laplace d'une expression.
- LPPAB** : intégration symbolique (paramètre p) de $P(t)\text{Exp}(-pt)$.
- LPLFP** : transformée de Laplace d'une fonction périodique.

L'ENTRÉE TOOLS

Comme son nom l'indique, l'entrée **TOOLS** conduit à toute une série d'utilitaires, extraits du tome 1 de "La maîtrise de la HP48". Ces 70 programmes sont organisés en 6 sections, pour une meilleure accessibilité. Les noms de ces six sections: **PILE**, **LIST**, **CHAIN**, **TABL**, **VARI**, **GEOP**, ne sont pas des noms de la librairie (vous ne pouvez donc pas accéder directement à la section **PILE**, par exemple, en tapant ce nom en ligne de commande).

L'entrée **TOOLS** se termine sur le jeu **GRRR...**

- ♦ La section **PILE** regroupe les utilitaires suivants:

ECHST : échange deux objets, à des niveaux quelconques de la pile.
KEEP : conserve sur la pile les objets situés à un niveau inférieur ou égal à un niveau donné.
REVST : inverse l'ordre des objets sur la pile, jusqu'à un niveau donné.
ROLLB : effectue un **ROLL** sur une suite d'objets situés entre 2 niveaux donnés.
ROLDB : effectue un **ROLLD** vers un niveau donné *m* de la pile, sur les objets situés du niveau 1 à un niveau *n* donné.
PICKB : effectue un **PICK** sur une suite d'objets situés entre 2 niveaux donnés.

- ♦ La section **LIST** contient des programmes s'appliquant aux listes:

ECHLI : échange deux objets de position donnée dans une liste.
DELI : retire d'une liste les objets *y* figurant entre deux positions données.
INSLI : insère un objet quelconque à une position donnée dans une liste.
RECLI : renverse l'ordre des objets dans une liste.
OCCLI : calcule le nombre de fois où un objet apparaît dans une liste.
REDLI : élimine les occurrences multiples dans une liste.
NUMD : donne le nombre d'objets différents dans une liste.
DBLE : donne la liste des objets présents au moins deux fois dans la liste initiale.
UNIQ : donne la liste des objets qui ne sont présents qu'une seule fois dans la liste initiale.
INTLI : effectue l'intersection ensembliste de deux listes.
REULI : effectue la réunion ensembliste de deux listes.
DIFLI : effectue la différence ensembliste de deux listes.
XORLI : effectue la différence symétrique ensembliste de deux listes.
IN? : teste si une liste est incluse dans une autre, d'un point de vue ensembliste.
IDLI? : teste l'égalité ensembliste de deux listes.

N.B: quand on considère une liste d'un point de vue ensembliste, l'ordre dans lequel y apparaissent les objets est sans importance. Deux objets égaux apparaissant en deux endroits différents sont considérés comme distincts.

.../...

- ◆ La section **CHAIN** ouvre sur des utilitaires s'appliquant aux chaînes de caractères:

DELCH : détruit les caractères d'une chaîne, entre deux positions données.
ECHCH : échange deux caractères, dans une chaîne.
INSCH : insère une chaîne dans une autre, à une position donnée.
REPCH : remplace, dans une chaîne ch3, toutes les occurrences d'une chaîne ch2 par une chaîne chl.
OCCCH : calcule le nombre d'occurrences d'une chaîne dans une autre.
CNST : crée une chaîne formée de n exemplaires d'une chaîne donnée.
INVCH : inverse l'ordre des caractères d'une chaîne.
MAJU : force le passage en majuscules, dans une chaîne de caractères.
MINU : force le passage en minuscules, dans une chaîne de caractères.
MJMI : bascule entre les minuscules et les majuscules, dans une chaîne de caractères.
DISPD : affiche un objet, à partir d'une ligne donnée, justifié à droite.
DISPC : affiche un objet, à partir d'une ligne donnée, centré.

- ◆ La section **TABL** permet de disposer d'un grand nombre d'utilitaires s'appliquant aux tableaux:

NL→M : fabrique une matrice à partir de n vecteurs de même taille.
M→NL : libère sur la pile les différentes lignes d'une matrice.
ADDT : concatène horizontalement deux tableaux.
SUPT : concatène verticalement deux tableaux.
VMC : bascule entre vecteur et matrice colonne.
VML : bascule entre vecteur et matrice ligne.
SEPTL : sépare la première ligne d'un tableau du reste de ce tableau.
INSC : insère une colonne nulle dans une matrice.
INSL : insère une ligne nulle dans une matrice.
SUPC : supprime une colonne d'un tableau.
SUPL : supprime une ligne d'un tableau.
DIMT : donne le nombre de lignes L et le nombre de colonnes C d'un tableau (dans le cas d'un vecteur, donne 1 pour L, C étant égal au nombre de coefficients)
PUTM : place une matrice à l'intérieur d'une autre, à partir d'une position donnée.
GETM : extrait une sous-matrice d'une matrice.
MINE : calcule les mineurs d'une matrice carrée.
COMA : calcule la comatrice d'une matrice carrée.
TR? : teste si un tableau est réel, ou complexe avec partie imaginaire nulle.
TR→C : transforme un tableau réel en tableau complexe.

.../...

- ◆ La section **VARI** ouvre sur six programmes s'appliquant aux variables (et en particulier aux répertoires):

RNMV : renomme une variable (en conservant sa localisation dans le répertoire).
DEPV : déplace une variable à l'intérieur d'un même répertoire.
DEPR : déplace une variable d'un répertoire à un autre.
ALLR : exécute un programme donné dans toute l'arborescence, à partir du répertoire en cours, et vers le bas.
TRE→ : affiche l'arborescence, à partir du niveau en cours.
→TRE : crée une arborescence de répertoires, à partir du niveau en cours.

- ◆ La section **GEOP** fournit des utilitaires portant sur la géométrie du plan.

DRTAB : équation de la droite passant par deux points donnés.
PERP : équation de la perpendiculaire à une droite donnée, passant par un point donné.
MDTR : équation de la médiatrice d'un segment.
BISS : équation de la bissectrice de deux demi-droites.
INTD : intersection de deux droites données par leurs équations.
ANGV : angle de deux vecteurs du plan.
CVGL : conversion d'un angle en sa mesure principale.
PROJD : projection d'un point sur une droite.
SYMD : symétrique d'un point par rapport à une droite.
AIRT : aire intérieure d'un triangle.
PERI : périmètre d'un triangle.
ALIG? : teste si trois points sont alignés.
COCY? : teste si quatre points sont cocycliques.

- ◆ Le jeu **GRRR**

Voilà un petit casse-tête qui vous donnera du fil à retordre. Il s'agit de noircir complètement l'écran, initialement vide. Cet écran est divisé en 9 cases. chacune des touches, de 1 à 9, inverse l'état de une ou plusieurs de ces cases (deux appuis sur une même touche s'annulent). La touche 0 permet de repartir avec un écran vide.

Une autre touche interrompt le jeu.

A chaque partie la correspondance entre les touches et les cases est renouvelée au hasard, pour que toutes les parties soient différentes. Pour vous aider les cases sont "signées" par les numéros de touche qui leur correspondent.

Il y a toujours au moins une solution en 4 touches (et parfois moins, mais jamais en une seule touche!). Essayez de trouver sans aucune aide écrite (tout dans la tête...).

Achevé d'imprimer par



31240 L'UNION (Toulouse)

Tél. (16) 61.74.27.67

Dépôt légal : Novembre 1992

La carte **PACMATH**

RECETTE: prendre tous les programmes du tome 2 de **"LA MAITRISE DE LA HP48"** (3ème édition); y ajouter 70 utilitaires issus du tome 1; inclure un **SYSTEME D'AIDE** perfectionné permettant de se remémorer rapidement la signification et l'utilisation de chacun des programmes. Comme une cerise sur le gâteau, ajouter un petit jeu.

Cette recette, **D3I DIFFUSION** et **J.M. FERRARD** l'on suivie pas à pas pour vous proposer **PACMATH**, une carte ROM de **128K**.

Pour un prix modique (inférieur à celui d'une carte RAM), vous augmentez considérablement les performances de votre calculatrice. Vous entrez en possession de la plus grande bibliothèque de programmes mathématiques jamais réunie sur un tel support (presque 400 programmes)!

Ce que la carte **PACMATH** vous apporte: vous gagnez énormément de temps, vous savez que les programmes sont conformes et qu'ils sont en sécurité. Avec le système d'aide, vous êtes autonome.

La carte **PACMATH**,

Un atout-maître dans vos études!